



MEKANİZMA TASARIMINDA KİNEMATİK SENTEZ İLE BÜTÜNLEŞİK YAZILIM GELİŞTİRİLMESİ

DOKTORA TEZİ

ÖMER OĞUZ TEMEL

**MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MAKİNE MÜHENDİSLİĞİ
ANABİLİM DALI**

**MERSİN
TEMMUZ - 2024**

MEKANİZMA TASARIMINDA KİNEMATİK SENTEZ İLE BÜTÜNLEŞİK YAZILIM GELİŞTİRİLMESİ

DOKTORA TEZİ

ÖMER OĞUZ TEMEL
ORCID ID: 0009-0007-6282-2025

MERSİN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAKİNE MÜHENDİSLİĞİ
ANABİLİM DALI

DANIŞMAN
DOÇ. DR. HÜSEYİN MUTLU
ORCID ID: 0000-0002-4770-2873

MERSİN
TEMMUZ - 2024

ÖZET

MEKANİZMA TASARIMINDA KİNEMATİK SENTEZ İLE BÜTÜNLEŞİK YAZILIM GELİŞTİRİLMESİ

Makine tasarımı alanında en önemli adımlardan birini, bir makine ve alette çoğu zaman birden fazla sayıda bulunan mekanizmaların tasarımı oluşturmaktadır. Bu çalışma kapsamında imalat, otomotiv, mekanik ölçüm cihazları vb. çok geniş bir alanda uygulama alanına sahip çeşitli mekanizma tiplerinin tasarımı için matematiksel yöntemler açıklanmış ve bu yöntemler modül adı verilen birer bilgisayar programı haline getirilerek sonuçların kullanıcıya metin tabanlı, animasyon ve çeşitli CAD programlarında açılabilen dosya tiplerinde verilmesi sağlanmıştır.

Kam mekanizması modülü kapsamında, kamın 1 turda 4 hareket değişimiyle kısıtlandığı ve sikloid hareket eğrisi kullanılarak iç ve/veya dıştan temaslı kam tasarımı yapan program geliştirilmiştir. Modül; mekanizma hareketini, yer değiştirme ve basınç açısı grafiklerini animasyon şeklinde sunmakla birlikte kam profili ise xlsx, dxf ve step dosya türlerinde kaydedilebilmektedir.

Fonksiyon sentezi modülü kapsamında geliştirilen program ile dört çubuk mekanizmasının 5 parametrelili kapalı çözümü, Kesin Nokta, Altbölge ve Galerkin yöntemleri ile hesaplanmış, mevcut tüm çözümlere ulaşılmıştır. Elde edilen sonuçlar elenerek metin tabanlı dosya ve animasyon şeklinde kullanıcıya sunulmuştur.

Dört çubuk mekanizmasının bir tipi olan kol sarkaç mekanizması modülü kapsamında geliştirilen program ile kol-sarkaç mekanizması tasarımında etkili parametrelerin 8 farklı durumda kullanılması ile tasarım gerçekleştirilmiş, sonuçlar elenerek animasyon, metin tabanlı ve CAD programlarında açılabilir dosya olarak kullanıcıya sunulmuştur.

Dört çubuk mekanizmasının yörünge sentezi modülü kapsamında geliştirilen program ile 9 noktada istenen yörüngeyi sağlayan mekanizma tasarımının öncelikle 5 parametrelili kapalı çözümünün tüm çözümleri elde edilmiştir. Ardından mekanizmanın dokuz boyutsal parametresine bağlı biyel eğrisinin cebirsel ifadesine ulaşılmıştır. Birinci aşamada elde edilen sonuçların başlangıç değeri olarak kullanılması ile dokuz parametrelili optimizasyon yapılarak istenen yörünge ile gerçek yörünge arasındaki hatanın azaltılması sağlanmıştır.

Özgün bir döngüsel kompresör yapısının geometrik modellenmesi modülü kapsamında ise üç rotora sahip bir kompresör yapısının geometrik tasarımı için gerekli parametrik eğri ailesi ve zarf eğrilerinin elde edilmesi için bir matematiksel model geliştirilmiş ve kompresör tasarımında sonraki hesaplamalı akışkanlar dinamiği, verim, performans vb. çalışmalarında yapılacak geometrik iyileştirmeler için kullanılacaktır.

Açıklanan çözüm yöntemleri ve geliştirilen bilgisayar programlarının etkinliği çeşitli sayısal uygulamalar yapılarak örneklenmiştir.

Anahtar Kelimeler: Kinematik Sentez, Fonksiyon Sentezi, Kol-Sarkaç Mekanizması, Bağlama Açısı, Yörünge Sentezi, Kapalı Çözüm, Ters Kinematik, Zarf Eğrisi, Döngüsel Kompresör, Bilgisayar Destekli Tasarım

Danışman: Doç. Dr. Hüseyin MUTLU, Mersin Üniversitesi, Makine Mühendisliği Anabilim Dalı, Mersin.

ABSTRACT

INTEGRATED SOFTWARE DEVELOPMENT WITH KINEMATIC SYNTHESIS IN MECHANISM DESIGN

One of the most important steps in the field of machine design is the design of mechanisms, which are often found in multiple numbers in a machine or tool. Within the scope of this study, mathematical methods for the design of various types of mechanisms, which have applications in a wide range of fields such as manufacturing, automotive, mechanical measurement devices, etc., have been explained. These methods have been transformed into computer programs called modules, allowing results to be presented to the user in text-based formats, animations, and file types that can be opened in various CAD programs.

In the cam mechanism module, a program has been developed to design a cam with internal and/or external contact using a cycloid motion curve in which the cam is constrained by four motion changes per revolution. The module presents the mechanism's motion, displacement, and pressure angle graphs as animations, and saves the cam profile in xlsx, dxf, and step file formats.

The program developed within the function synthesis module provides the 5-parameter closed solution of the four-bar mechanism, calculated using the precision point, subdomain, and Galerkin methods, reaching all existing solutions. The obtained results were sifted and presented to the user in the form of text-based files and animations.

Within the module for a type of four-bar mechanism known as the crank-rocker mechanism, a program was developed that designs the crank-rocker mechanism using the effective parameters in 8 different cases. The results are filtered and presented to the user as animations, text-based files, and files that can be opened in CAD programs.

With the program developed within the path synthesis module of the four-bar mechanism, initially, all solutions of the 5-parameter closed solution of the mechanism design that provides the desired path at nine points were obtained. Then, the algebraic expression of the coupler curve dependent on the nine dimensional parameters of the mechanism was derived. By using the results obtained in the first stage as initial values, a nine-parameter optimization was performed to reduce the error between the desired trajectory and the actual trajectory.

In the module for the geometric modeling of a novel rotary type compressor structure, a mathematical model has been developed to obtain the parametric curve family and envelope curves required for the geometric design of a compressor structure with three rotors and will be used for geometric improvements to be made in future computational fluid dynamics, efficiency, performance, etc. studies in compressor design.

The effectiveness of the solution methods described and the computer programs developed are demonstrated through various numerical applications.

Keywords: Kinematic Synthesis, Function Synthesis, Crank-Rocker Mechanism, Coupling Angle, Path Synthesis, Closed Solution, Inverse Kinematics, Envelope Curve, Cyclic Compressor, Computer Aided Design

Advisor: Assoc. Prof. Huseyin MUTLU, Department of Mechanical Engineering, Mersin University, Mersin.

TEŞEKKÜR

Tez çalışmam boyunca bilgi birikimiyle bana her konuda yardımcı olan, tecrübeleriyle bana yol gösteren ve karşılaştığım sorunlarda desteğini esirgemeyen tez danışmanım Doç. Dr. Hüseyin MUTLU'ya,

Aldığım kararları her zaman destekleyen, beni cesaretlendiren ve moral desteğini benden esirgemeyen başta annem ve babam olmak üzere, bu süreçte her konuda yanımda olan kardeşim ve eşime,

Çalışmamın tüm aşamalarında gerek bilimsel gerek manevi anlamda yol gösteren, desteğini esirgemeyen ve hayatımın her alanında bana rehberlik eden çok değerli hocam Doç. Dr. Salih AKSAY'a,

Çalışmalarım boyunca her zaman benimle fikir alışverişinde bulunan ve bana destek olan değerli arkadaşlarım Dr. Süleyman Çınar ÇAĞAN ve Dr. Hasan İBİLOĞLU'na sonsuz teşekkür ederim.



İÇİNDEKİLER

	Sayfa
İÇ KAPAK	i
ONAY	ii
ETİK BEYAN	iii
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
TABLOLAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
1.1. Kam Mekanizmaları	3
1.2. 4 Çubuk Mekanizmaları	7
2. KAYNAK ARAŞTIRMALARI	11
3. MATERYAL ve YÖNTEM	15
3.1. Kam Mekanizması Modülü Teorisi	15
3.2. Kam Mekanizması Modülü Program Arayüzü	19
3.3. Fonksiyon Sentezi Modülü Teorisi	22
3.4. Fonksiyon Sentezi Modülü Program Arayüzü	27
3.5. Kol Sarkaç Mekanizması Modülü Teorisi	30
3.5.1. Durum 1: α , ZO ve β kullanılarak diğer parametrelerin bulunması	31
3.5.2. Durum 2: α , ZO ve Φ kullanılarak diğer parametrelerin bulunması	31
3.5.3. Durum 3: α , ZO ve u/p kullanılarak diğer parametrelerin bulunması	32
3.5.4. Durum 4: α , β ve u/p kullanılarak diğer parametrelerin bulunması	33
3.5.5. Durum 5: α , β ve Φ kullanılarak diğer parametrelerin bulunması	33
3.5.6. Durum 6: α , Φ ve u/p kullanılarak diğer parametrelerin bulunması	33
3.5.7. Durum 7: α , ZO ve u/r kullanılarak diğer parametrelerin bulunması	33
3.5.8. Durum 8: Bağlama açısına göre diğer parametrelerin bulunması	34
3.6. Kol Sarkaç Mekanizması Modülü Program Arayüzü	38
3.7. Yörünge Sentezi Modülü Teorisi	43
3.7.1. Yörünge Sentezi Ön Tasarım Süreci	43
3.7.2. Yörünge Sentezi Optimizasyon Süreci	48
3.8. Yörünge Sentezi Modülü Program Arayüzü	51
3.9. Özgün Bir Döngüsel Kompresör Yapısının Geometrik Modellenmesi Modülü Teorisi	55
3.9.1. Kompresörün Ön Geometrisinin Oluşturulması	56
3.9.2. Kompresörün Çıkış Geometrisinin Oluşturulması	57
3.9.3. Kompresör Geometri Hesabı Program Arayüzü	64
3.9.4. Kompresörün Çalışma Prensibi	67
4. BULGULAR ve TARTIŞMA	71
4.1. Kam Mekanizması Modülü Örnek Uygulamaları	71
4.1.1. Dairesel Dokuma Tezgâhı Kam Profili Tasarımı Örnek Uygulaması	71
4.1.2. Beklemeli Pim Basma Mekanizması Kam Profili Tasarımı Örnek Uygulaması	73
4.2. Fonksiyon Sentezi Modülü Örnek Uygulamaları	76
4.2.1. $y=e^x$ Fonksiyonu Örnek Uygulaması	76
4.2.2. $y=x^2$ Fonksiyonu Örnek Uygulaması	77
4.2.3. $y=\sin(x)$ Fonksiyonu Örnek Uygulaması	79
4.3. Kol Sarkaç Modülü Örnek Uygulamaları	81
4.3.1. Durum 1 (α , ZO ve β kullanılarak) Örnek Uygulaması	81
4.3.2. Durum 2 (α , ZO ve Φ kullanılarak) Örnek Uygulaması	82
4.3.3. Durum 3 (α , ZO ve u/p kullanılarak) Örnek Uygulaması	84
4.3.4. Durum 4 (α , β ve u/p kullanılarak) Örnek Uygulaması	85

İÇİNDEKİLER

	Sayfa
4.3.5. Durum 5 (α , β ve Φ kullanılarak) Örnek Uygulaması	86
4.3.6. Durum 6 (α , Φ ve u/p kullanılarak) Örnek Uygulaması	87
4.3.7. Durum 7 (α , ZO ve u/r kullanılarak) Örnek Uygulaması	89
4.3.8. Durum 8 (Bağlama açısı kullanılarak) Örnek Uygulaması	90
4.4. Yörünge Sentezi Modülü Örnek Uygulaması	91
4.5. Kompresör Modülü Örnek Uygulamaları	94
5. SONUÇLAR VE ÖNERİLER	101
KAYNAKLAR	105
EKLER	109
EK 1. Fonksiyon Sentezi Modülü Denklem (3.3 9) Katsayıları	109
EK 2. Fonksiyon Sentezi Modülü Denklem (3.3 10) Katsayıları	113
EK 3. Yörünge Sentezi Modülü Denklem (3.7 10) Katsayıları	118
EK 4. Yörünge Sentezi Modülü Denklem (3.7 12) Katsayıları	122
EK 5. Yörünge Sentezi Modülü Denklem (3.7 22) Katsayıları	168
EK 6. Yörünge Sentezi Modülü Denklem (3.7 23) Katsayıları	172
ÖZGEÇMİŞ	177

TABLOLAR DİZİNİ

	Sayfa
Tablo 3.1. Kam Tasarım Programı Hesaplama Algoritması	21
Tablo 3.2. Fonksiyon Sentezi Programı Hesaplama Algoritması	29
Tablo 3.3. Kol Sarkaç Mekanizması Programı Hesaplama Algoritması	40
Tablo 3.4. Yörünge Sentezi Programı Hesaplama Algoritması	53
Tablo 3.5. Kompresör Çıkış Geometrisi Hesaplama Algoritması	64
Tablo 4.1. Fonksiyon Sentezi Modülü $y=e^x$ Fonksiyonu Örneği Çözüm Sonuçlarının Karşılaştırılması	76
Tablo 4.2. Fonksiyon Sentezi Modülü $y=x^2$ Fonksiyonu Örneği Çözüm Sonuçlarının Karşılaştırılması	78
Tablo 4.3. Fonksiyon Sentezi Modülü $y=\sin(x)$ Fonksiyonu Örneği Çözüm Sonuçlarının Karşılaştırılması	79
Tablo 4.4. Durum 1 Örnek Uygulaması Tasarım Sonuçları	81
Tablo 4.5. Durum 2 Örnek Uygulaması Tasarım Sonuçları	82
Tablo 4.6. Durum 3 Örnek Uygulaması Tasarım Sonuçları	84
Tablo 4.7. Durum 4 Örnek Uygulaması Tasarım Sonuçları	85
Tablo 4.8. Durum 5 Örnek Uygulaması Tasarım Sonuçları	86
Tablo 4.9. Durum 6 Örnek Uygulaması Tasarım Sonuçları	88
Tablo 4.10. Durum 7 Örnek Uygulaması Tasarım Sonuçları	89
Tablo 4.11. Durum 8 Örnek Uygulaması Tasarım Sonuçları	90
Tablo 4.12. Kompresör referans tasarıma göre r yarıçapındaki değişimin kompresör geometrisi ve kapasitesine etkilerinin sayısal sonuçları	95
Tablo 4.13. Kompresör referans tasarıma göre r_2 yarıçapındaki değişimin kompresör geometrisi ve kapasitesine etkilerinin sayısal sonuçları	96
Tablo 4.14. Kompresör referans tasarıma göre r_3 yarıçapındaki değişimin kompresör geometrisi ve kapasitesine etkilerinin sayısal sonuçları	98

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Kam mekanizması genel yapısı (Megep, 2011)	3
Şekil 1.2. Kam şekline göre sınıflandırma (Soylemez, 2007)	4
Şekil 1.3. İzleyici hareketine (a) ve şekline (b) göre sınıflandırma (Norton, 2009)	4
Şekil 1.4. Şekil kapalı kam mekanizması (Norton, 2009)	5
Şekil 1.5. Kam Mekanizması Tipik Hareket Eğrisi	5
Şekil 1.6. Kam Tasarımında Sikloid Hareket Eğrisi (Rothbart, 2004)	6
Şekil 1.7. 4 Çubuk Mekanizması Yapısı	7
Şekil 1.8. Grasshof Teoremine Göre 4 Çubuk Mekanizması Hareketleri (Norton, 2020)	8
Şekil 3.1. Kam Mekanizması Tasarım Parametreleri	15
Şekil 3.2. 4 hareketli kam mekanizması eğrisi	17
Şekil 3.3. Kam Tasarım Programı Kullanıcı Arayüzü	19
Şekil 3.4. Kam Tasarım Programı Animasyon ve Grafik Sunumu	20
Şekil 3.5. Xlsx dosya türünde kaydedilen kam tasarım sonuçları	20
Şekil 3.6. Dxf (a) ve Step (b) Dosya Türünde Kaydedilen Kam Tasarım Sonuçları	21
Şekil 3.7. Kam Tasarım Programı Akış Diyagramı	22
Şekil 3.8. Fonksiyon Sentezi Tasarım Parametreleri	23
Şekil 3.9. Fonksiyon sentezi sonuç dosya görüntüsü	28
Şekil 3.10. Fonksiyon sentezi programı arayüzü	28
Şekil 3.11. Fonksiyon Sentezi Programı Akış Diyagramı	29
Şekil 3.12. Kol Sarkaç Mekanizması Parametreleri	30
Şekil 3.13. Kol sarkaç mekanizmasında minimum ve maksimum bağlama açısı	34
Şekil 3.14. Kol Sarkaç Mekanizması sentezi sonuç dosya görüntüsü	38
Şekil 3.15. Kol Sarkaç Mekanizması Modülü Programı Arayüzü	39
Şekil 3.16. Dxf Dosya Türünde Kaydedilen Kol-Sarkaç Tasarım Sonuçları	39
Şekil 3.17. Kol Sarkaç Mekanizması Programı Akış Diyagramı	42
Şekil 3.18. Yörünge Sentezi Tasarım Parametreleri	43
Şekil 3.19. Yörünge Sentezi Ön Tasarım Sonuçları	51
Şekil 3.20. Yörünge Sentezi Modülü Programı Arayüzü	52
Şekil 3.21. Dxf Dosya Türünde Kaydedilen Yörünge Sentezi Tasarım Sonuçları	52
Şekil 3.22. Yörünge Sentezi Programı Akış Diyagramı	54
Şekil 3.23. Pozitif Deplasmanlı Kompresör Modeli	55
Şekil 3.24. Kompresör Rotor Kesitlerinin Gövde Üzerindeki Konumları	56
Şekil 3.25. Kompresör Rotor Kesitlerinin Tasarım Parametreleri ve Kinematik Düzenleme	57
Şekil 3.26. Ters Dönüşüm Mekanizması ve Kinematik Boyutları	58
Şekil 3.27. Ters Dönüşüm Mekanizmasında Yörünge Eğri Ailesi	59
Şekil 3.28. a' ve d' noktasının yerini belirleyen bağımsız değişkenler	60
Şekil 3.29. Ters Dönüşüm Mekanizmasının Yuvarlanma Hareketi	62
Şekil 3.30. B silindirik kesitinin hesaplanan büyüklükleri	63
Şekil 3.31. Kompresör modelinin geometrik düzenlemesi	63
Şekil 3.32. Kompresör kapasitesini belirleyen büyük (a) ve küçük (b) hacimsel bölgeyi oluşturan rotor konumları	64
Şekil 3.33. Kompresör Geometri Hesabı Programı Akış Diyagramı	65
Şekil 3.34. Etkileşimli Kullanıcı Arayüzünün Ana ve Simülasyon Penceresi	66
Şekil 3.35. Parametre Giriş Menüsü	66
Şekil 3.36. Etkileşimli Kullanıcı Arayüzünün Ana ve Çıkış Geometri Penceresi	67
Şekil 3.37. Giriş ve Çıkış Geometrisinin Dxf Formatı (a) ve Step Formatı (b)	67
Şekil 3.38. Örnek 1'deki kompresör yapısının 1 çevrimdeki I. Evre (a), II. Evre (b), III. Evre (c), IV. Evre (d) ve V. Evre (e) kompresyon evreleri.	68
Şekil 4.1. Dairesel Dokuma Tezgahı	71
Şekil 4.2. Mekik Ağzı Açma Sistemi	71
Şekil 4.3. Örnek 1 İçin İzleyicinin Kam Açısız Dönmesine Bağlı Yer Değiştirme Eğrisi	72
Şekil 4.4. Örnek 1'e Ait İç Kam(a) ve Dış Kam (b) Basınç Açıkları	73

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 4.5. Örnek 1'e Ait Programdan Elde Edilen Dxf (a) ve Step (b) Dosya Görüntüleri	73
Şekil 4.6. Beklemeli Pim Basma Mekanizması	74
Şekil 4.7. Örnek 2 İçin İzleyicinin Kam Açısıl Dönmesine Bağlı Yer Değiştirme Eğrisi	74
Şekil 4.8. Örnek 2'ye ait Dış Kam Basınç Açıları	75
Şekil 4.9. Örnek 2'ye Ait Programdan Elde Edilen Dxf (a) ve Step (b) Dosya Görüntüleri	75
Şekil 4.10. Fonksiyon Sentezi Modülü $y=e^x$ Fonksiyonu Örneğine Ait Karşılaştırmalı Yapısal Hata Dağılımı	76
Şekil 4.11. Fonksiyon Sentezi Modülü $y=e^x$ Fonksiyonu Örneğine Ait Tasarım Sonucunun Çalışma Aralığının Başlangıç (a) ve Bitiş (b) Konumları	77
Şekil 4.12. Fonksiyon Sentezi Modülü $y=e^x$ Örneğine Ait Tasarım Sonucunun CAD Programı ile Yapılan Doğrulaması	77
Şekil 4.13. Fonksiyon Sentezi Modülü $y=x^2$ Fonksiyonu Örneğine Ait Karşılaştırmalı Yapısal Hata Dağılımı	78
Şekil 4.14. Fonksiyon Sentezi Modülü $y=x^2$ Fonksiyonu Örneğine Ait Tasarım Sonucunun Çalışma Aralığının Başlangıç (a) ve Bitiş (b) Konumları	78
Şekil 4.15. Fonksiyon Sentezi Modülü $y=x^2$ Örneğine Ait Tasarım Sonucunun CAD Programı ile Yapılan Doğrulaması	79
Şekil 4.16. Fonksiyon Sentezi Modülü $y=\sin(x)$ Fonksiyonu Örneğine Ait Karşılaştırmalı Yapısal Hata Dağılımı	80
Şekil 4.17. Fonksiyon Sentezi Modülü $y=\sin(x)$ Fonksiyonu Örneğine Ait Tasarım Sonucunun Çalışma Aralığının Başlangıç (a) ve Bitiş (b) Konumları	80
Şekil 4.18. Fonksiyon Sentezi Modülü $y=\sin(x)$ Örneğine Ait Tasarım Sonucunun CAD Programı ile Yapılan Doğrulaması	81
Şekil 4.19. Kol Sarkaç Modülü Durum 1'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	81
Şekil 4.20. Kol Sarkaç Modülü Durum 1'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	82
Şekil 4.21. Kol Sarkaç Modülü Durum 2'ye Ait Tasarım Sonuçlarından Birinci Çözümün Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	83
Şekil 4.22. Kol Sarkaç Modülü Durum 2'ye Ait Tasarım Sonuçlarından İkinci Çözümün Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	83
Şekil 4.23. Kol Sarkaç Modülü Durum 2'ye Ait Tasarım Sonuçlarından Birinci (a) ve İkinci (b) Çözümün İlk ve Son Sınır Konumları	83
Şekil 4.24. Kol Sarkaç Modülü Durum 3'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	84
Şekil 4.25. Kol Sarkaç Modülü Durum 3'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	85
Şekil 4.26. Kol Sarkaç Modülü Durum 4'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	85
Şekil 4.27. Kol Sarkaç Modülü Durum 4'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	86
Şekil 4.28. Kol Sarkaç Modülü Durum 5'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	87
Şekil 4.29. Kol Sarkaç Modülü Durum 5'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	87
Şekil 4.30. Kol Sarkaç Modülü Durum 6'ya Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	88
Şekil 4.31. Kol Sarkaç Modülü Durum 6'ya Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	88
Şekil 4.32. Kol Sarkaç Modülü Durum 7'ye Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	89

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 4.33. Kol Sarkaç Modülü Durum 7'ye Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	90
Şekil 4.34. Kol Sarkaç Modülü Durum 8'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları	91
Şekil 4.35. Kol Sarkaç Modülü Durum 8'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları	91
Şekil 4.36. Yörünge Sentezi Modülü Örneği Optimizasyon Öncesi Sonuçları	92
Şekil 4.37. Yörünge Sentezi Modülü Örneği Son Tasarım Sonuçları	93
Şekil 4.38. Yörünge Sentezi Modülü Örneğine Ait Amaçlanan Elips Yörünge ile Ön Tasarım ve Son Tasarım Aşamalarındaki Yörüngelerin Karşılaştırmalı Görüntüsü	94
Şekil 4.39. Kompresör referans tasarımı	94
Şekil 4.40. Kompresör referans tasarıma göre r yarıçapındaki değişimin kompresör geometrisi ve kapasitedeki etkileri	95
Şekil 4.41. Kompresör kapasitesinin belirleyen V_b , V_s ve V_t hacimlerinin r yarıçapına göre değişim grafiği	96
Şekil 4.42. Kompresör referans tasarıma göre r_2 yarıçapındaki değişimin kompresör geometrisi ve kapasitedeki etkileri	97
Şekil 4.43. Kompresör kapasitesinin belirleyen V_b , V_s ve V_t hacimlerinin r_2 yarıçapına göre değişim grafiği	98
Şekil 4.44. Kompresör referans tasarıma göre r_3 yarıçapındaki değişimin kompresör geometrisi ve kapasitedeki etkileri	99
Şekil 4.45. Kompresör kapasitesinin belirleyen V_b , V_s ve V_t hacimlerinin r_3 yarıçapına göre değişim grafiği	100

SİMGELER VE KISALTMALAR

Kısaltma/Simgesi	Tanım
R_i	Kam iç profil eğrisine çizilebilecek merkezi kam ile aynı olan en küçük daire yarıçapı
R_d	Kam dış profil eğrisine çizilebilecek merkezi kam ile aynı olan en küçük daire yarıçapı (R_i+t)
R_2	Kam iç ve dış profil eğrilerine temas eden izleyici yarı çapı
t	Kam iç ve dış profilleri arası mesafe
$r_{\theta i}$	İç izleyici merkezi ile kam merkezi arası mesafe
$r_{\theta d}$	Dış izleyici merkezi ile kam merkezi arası mesafe
s	İzleyicilerin hareket doğrultularının normali
t_i	İç izleyicinin temas anında kam iç profil eğrisine teğeti
t_d	Dış izleyicinin temas anında kam dış profil eğrisine teğeti
n_i	t_i teğetinin normali
n_d	t_d teğetinin normali
Φ_d	Dış izleyici basınç açısı
Φ_i	İç izleyici basınç açısı
x_1	Giriş uzvu yani krank kolu boyu
x_2	Biyel uzvu boyu
x_3	Çıkış uzvu boyu
x_4	krank-biyel ortak mafsalları ile x_5 arası dik mesafe
x_5	İstenen yörüngeyi izleyen P biyel noktasının biyel uzvuna dik mesafesi
x_6	Gövde uzvu boyu, çıkış uzvu gövde mafsalları giriş uzvu gövde mafsalları göre yatay mesafesi
x_7	Giriş uzvu gövde mafsalları sabit OXY referans sistemine göre yatay mesafesi
x_8	Giriş uzvu gövde mafsalları sabit OXY referans sistemine göre düşey mesafesi
x_9	Çıkış uzvu gövde mafsalları giriş uzvu gövde mafsalları göre düşey mesafesi
d_{45}	İstenen yörüngeyi izleyen P biyel noktasının krank-biyel ortak mafsalları olan mesafesi
u/p	Çıkış uzvu boyunun gövde uzvu boyuna oranı
r/p	Biyel uzvu boyunun gövde uzvu boyuna oranı
q/p	Giriş uzvu boyunun gövde uzvu boyuna oranı
u/r	Çıkış uzvu boyunun biyel uzvu boyuna oranı
ψ	Giriş uzvu dönüş açısı
α	Çıkış uzvu dönüş açısı
δ	Biyel uzvunun yatay eksenle yaptığı açı
η	Kol-Sarkaç mekanizması iki sınır konumu arası giriş uzvu açısı
β	Kol-Sarkaç mekanizması başlangıç sınır konumunda arası giriş uzvu ile yatay eksen arasındaki açı
Φ	Kol-Sarkaç mekanizması başlangıç sınır konumunda arası çıkış uzvu ile yatay eksen arasındaki açı
μ_{max}	Kol-Sarkaç mekanizması maksimum bağlama açısı
μ_{min}	Kol-Sarkaç mekanizması minimum bağlama açısı
ZO	Zaman oranı
r	birinci rotorun yarıçapı
r_2	ikinci rotorun yarıçapı
r_3	üçüncü rotorun yarıçapı

1. GİRİŞ

Makineler, tanım olarak enerji ve gücü mekanik enerjiye veya işe çeviren yapılardır. Isıl dönüşümler dışında kalan, yani statik, kinetik ve kinematik prensiplere göre çalışan mekanik makineler mekanizma tekniği branşının temel çalışma alanını oluşturmaktadır. Makine tanımına uygun olarak bir amaca yönelik kuvvet ve enerji dönüşümü yaparken bünyesinde bir veya daha fazla sayıda genel amaçla tasarlanmış ve farklı makinelerde de bulunabilen, çeşitli hareket ve hareket dönüşümü yapan mekanizmaları bulundurabilir (Akcalı ve Mutlu, 2014). Harekete neden olan kuvvetler ve bu kuvvetlerin nedenlerinden bağımsız olarak katı (rijit) cisimlerin geometrisine bağlı hareketin oluşumunu inceleyen kinematik biliminin alanı olan yapılar mekanizmalardır. Mekanizmaları oluşturan her bir eleman uzuv olarak tanımlanırken, uzuvların birbirine bağlanması için kullanılan bölümüne kinematik eleman, bağlantı noktasına ise kinematik çift veya mafsal adı verilmektedir. İhtiyaca ve mekanizmanın tipine uygun olarak uzuv, kinematik eleman, kinematik çift ve mafsallar çok çeşitli biçimlerde olabilmektedir.

Mekanizmalar günlük hayatımızda birçok uygulamada karşımıza çıkmaktadır. Sürekli dönme hareketi ile git-gel biçiminde doğrusal hareket arasında dönüşüm yapan krank biyel mekanizması, içten yanmalı motorlarda yakıtın yanması ile ortaya çıkan kuvvetin pistonu ötelemesi sonucu krank milinde dönme hareketi elde edilmesini sağlamaktadır. Aynı mekanizma kompresör ve pompa uygulamalarında ise elektrik motorundan alınan sürekli dönme hareketini piston tarafında öteleme hareketine çevirerek akışkanı sıkıştırma veya öteleme amacıyla kullanılmaktadır.

Bir kaynaktan alınan sürekli dönme hareketinin, üzerinde bulunan slot sayısı ile orantılı biçimde beklemeli dönme hareketi olarak alındığı geneva mekanizması ise çoğunlukla saatlerde, otomatik makinalarda zamanlayıcı olarak kullanılmaktadır.

İstenen amaç harekete yönelik imal edilen kam profilleri ve izleyiciden oluşan kam mekanizmaları yaygın kullanım alanına sahip bir diğer mekanizma tipidir. Kam mekanizmaları çok çeşitli hareket tasarımlarına imkân sağlamaları, kolay tasarlanabilir ve kolay değiştirilebilir olmaları nedeniyle en çok tercih edilen mekanizma tiplerinden biridir. Hareketin doğruluğu için malzemenin düşük toleransta işlenmesi ve tasarım doğruluğu yüksek maliyet oluşturmakta, bunula birlikte zamanla oluşan aşınmalar ve yenileme ihtiyacı nedeniyle yüksek hızda çalışan makinelerde kullanımı sınırlı tutulmaktadır (Soylemez, 2007).

Birçok alanda kullanılan bir diğer mekanizma tipi ise kol mekanizmalarıdır. İnşaat makinelerinden (ekskavatör, dozer vb.) robotik taşıma kollarına, ayarlı penseden mekanik manometreye kadar çok çeşitli makine ve alet içerisinde kol mekanizmaları bulunmaktadır. İhtiyaca göre uzuv sayısı değişken olmakta ve sık kullanılan tiplere dört çubuk mekanizması, kol-sarkaç mekanizması, çift sarkaç mekanizması gibi özel adlandırılmalar yapılmaktadır.

Yukarıda bazı tipleri ve kullanım alanlarına örnek verilenler de dâhil tüm mekanizmaların tasarımından uygulamaya kadar ilerleyen süreçte çalışmalar iki ana grupta toplanmaktadır. Tipi ve boyutları bilinen bir mekanizmanın hareket özelliklerinin belirlenmesi işlemi kinematik analiz olmakta

ve bu aşamada mekanizmanın seçilen bir noktasında sırasıyla yer değiştirme yani konum analizi, hız analizi ve ivme analizi yapılmakta, bir referans noktasına göre kinematik değerleri bulunmaktadır. Kinematik analizin tersi durumda, yani istenen kinematik değerlere uygun mekanizma tasarlanması süreci ise kinematik sentez olarak adlandırılmaktadır. Sentez çalışmaları öncelikle aynı işlevi gören fakat çalışma alanı, üretim şekli, işletme şartları, maliyet vb. durumlar göz önüne alınarak mekanizmanın tip sentezi ile başlar. Tipi belirlenen mekanizmaya ait uzuv sayısı ve tipleri, mafsalları ve tipleri ve uzuvların konumlarını belirlemek için gerekli parametre sayısı olarak tanımlanan serbestlik derecesi ile oluşturulabilecek kinematik zincirler ve mekanizmaların belirlenmesi olan sayı sentezi ile devam edilir.

Amaca uygun kinematik zinciri oluşturulan mekanizmanın istenen kinematik değerlere uygun biçimde uzuv boyları, birbirine göre açıların ve mafsalları uzaklıklarının belirlenmesi işlemine ise boyutsal sentez adı verilir. Araba sileceklerinde kullanılan sürekli bir dönüş hareketini istenen sıklıkta salınım hareketine çeviren kol-sarkaç mekanizmasının boyutlandırılması uygulamalardan bir örnektir. Boyutsal sentez kapsamında, hareketin giriş ve çıkış uzuvları arasındaki aktarma fonksiyonu adı verilen analitik bağıntının amaca uygun belirli bir fonksiyon olması için mekanizma boyutlarının belirlenmesine ise fonksiyon sentezi adı verilir. Kam mekanizmalarında kam profilinin dönüşüne bağlı olarak izleyicinin hareketi arasındaki ilişki veya manometreler içerisinde bulunan bourdon tüpünün basınç etkisiyle hareketi sonucu çıkış uzvunda, yani ibrede basınç değerini okuyabilmemiz fonksiyon sentezinin birer uygulamasıdır. Boyutsal sentezin bir diğer uygulaması da mekanizma uzuvlarının istenen konumlarda hareket etmesine yönelik hareket ve yörünge sentezi çalışmalarıdır. Hareket sentezinde hareketin alındığı uzvun birden fazla noktasının istenen konumlardan geçmesi, yörünge sentezinde ise hareket uzvunun izleyeceği yörüngenin tam veya kısmi olarak belirlenmesi çalışmaları yapılır. Sürekli dönüş hareketinin belirli zaman aralıklarında öteleme hareketine dönüştürüldüğü manyetik film makineleri ve paketleme makineleri, üstü açılan bir arabanın bagaja kendiliğinden yerleşen tavanının yaptığı hareketin hesabı yörünge sentezi kapsamında örnek verilebilir (Akcalı, 1994); (Hartenberg ve Denavit, 1964).

Mekanizma tekniği alanında yapılan çalışmalar tüm uzuv, mafsalları ve mekanizmanın hareketinin düzlemsel, küresel ve uzaysal olması durumuna göre olarak 3 ana grupta toplanır. Güncel çalışmaların büyük bir kısmını oluşturan düzlemsel mekanizmalarda uzuv ve mafsallar bir referans düzlem üstünde veya referans düzleme paralel düzlemler üstünde hareket eder. Bu hareket doğrusal veya eğrisel halde öteleme, dönme ya da bunların bileşimi şeklinde olmaktadır.

Kinematik sentez çalışmalarında fiziksel problem öncelikle matematiksel bir probleme dönüştürülür ve bulunan sonuçlar kısmi ve tam çözüm olarak iki farklı tipte olabilir. Tam çözümde elde edilen boyutlarla mekanizmanın hareket aralığının tamamında matematiksel olarak kesinlik sağlanırken, kısmi çözümde uygun görülen hata oranı kadar farkla çözüme ulaşılır.

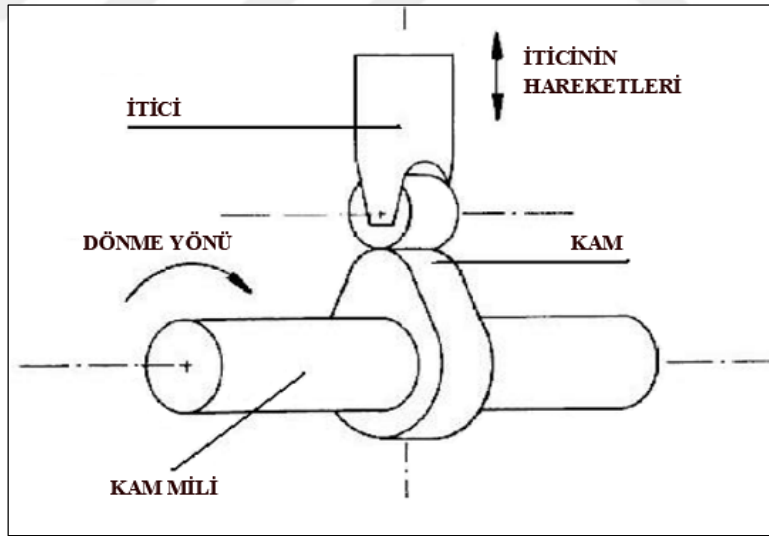
Mekanizmalarda sentez problemlerinin çözümünde, çizim hassasiyeti ve tekrarlama zorluklarını da barındıran grafik yöntemler ile gelişen bilgisayar teknolojisiyle beraber daha yaygın olarak analitik

yöntemler kullanılmaktadır. Mekanizmanın sentez problemine konu olan istenen yörünge veya fonksiyonun, gerçekte üretilen yörünge ve fonksiyonla olan farkı yapısal hata olarak adlandırılır ve bu hatanın mekanizmanın belirli bir aralığında kabul edilebilir seviyede olması amaçlanır. Başlangıç çözüm değerine bağlı ardışık işlem gerektiren (iteratif) metodlarda çözüme ulaşma garantisi olmamakla birlikte, yakınsama halinde tek çözüm elde edilir. Kesin nokta (Hartenberg ve Denavit, 1964), alt bölge (Akçalı., I.D. ve Lindholm., 1979) ve Galerkin (Akcali ve Dittrich, 1989) yöntemleri gibi kapalı çözüm metodlarında ise daha az işlem yüküne karşılık, çözüm bir denkleme ait tüm kökler olarak bulunmaktadır (Mutlu, 1994).

Daha önce belirtildiği gibi bir makinenin hareket dönüşümlerini yapması için bir veya daha fazla sayıda mekanizmanın bir araya getirilmesi makine tasarımının temel adımlarından biridir. Öncelikle istenen hareket dönüşümünü sağlayan mekanizmanın tasarlanması sonrasında uygun imalat yöntemleri, dayanım sınırları vb. mühendislik hesaplamalarına geçilmektedir.

1.1. Kam Mekanizmaları

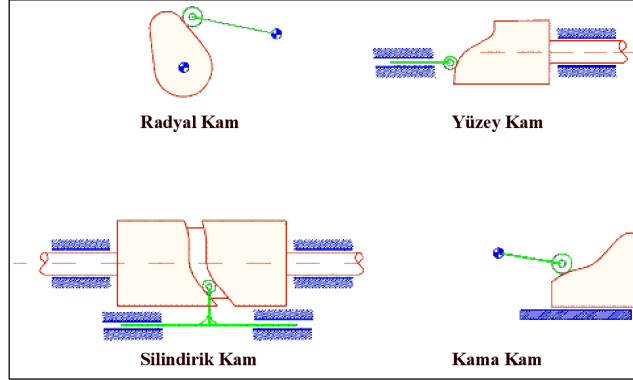
Kam mekanizmaları, genel yapısı Şekil 1.1’de gösterilen, yüzeyi ihtiyaca uygun biçimlendirilmiş ve sürekli dönme hareketi yapan kam ve kam ile sürekli noktasal veya çizgisel temas halinde olan itici’den oluşan, bir serbestlik derecesine sahip mekanizmalardır.



Şekil 1.1. Kam mekanizması genel yapısı (Megep, 2011)

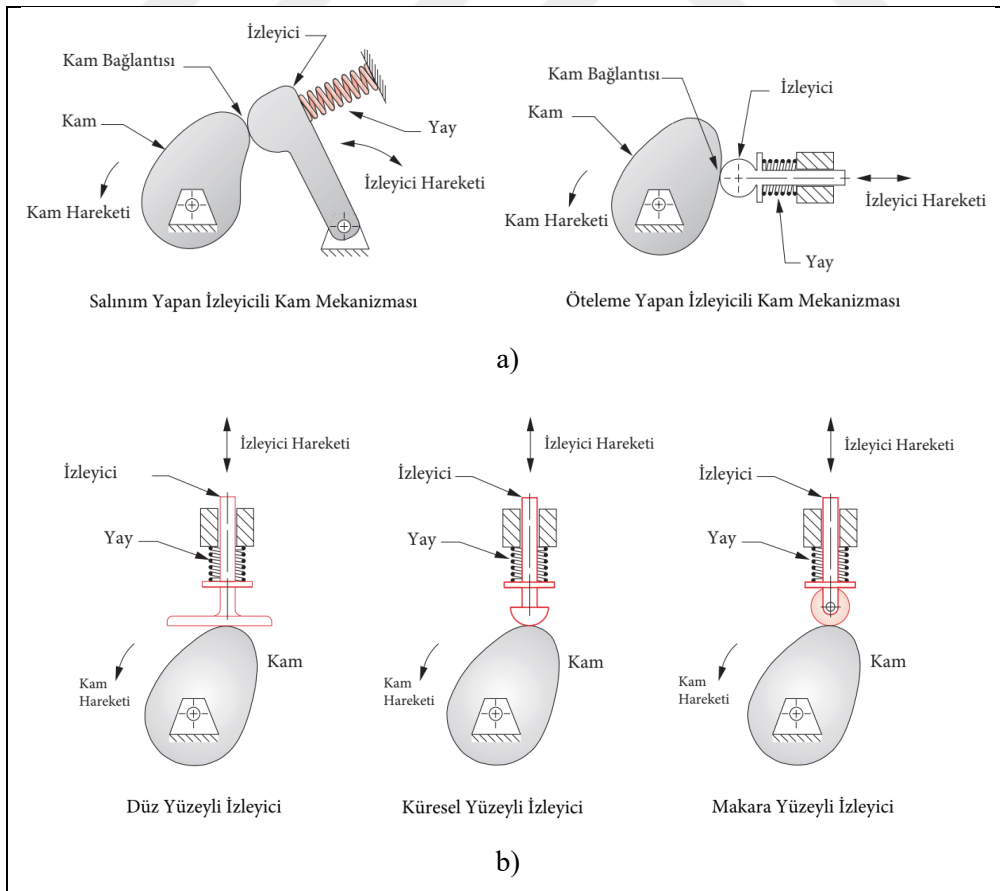
Kam mekanizmaları nispeten kolay tasarım, hareket çeşitliliği ve ihtiyaca uygun değişim kolaylığı nedeniyle çok geniş bir alanda kullanılmakta ve izleyici hareketinin istenen miktarda ve durumda sınırlandırılması ile bir mekanik programlama elemanı olarak da tanımlanmaktadır. (Rothbart, 2004)

Kam mekanizmalarının sınıflandırılmasında çeşitli yaklaşımlar bulunmaktadır. Kamın şekline göre sınıflandırmada Şekil 1.2’de gösterildiği gibi radyal, yüzey, silindirik ve kama tipinde olması esas alınır.



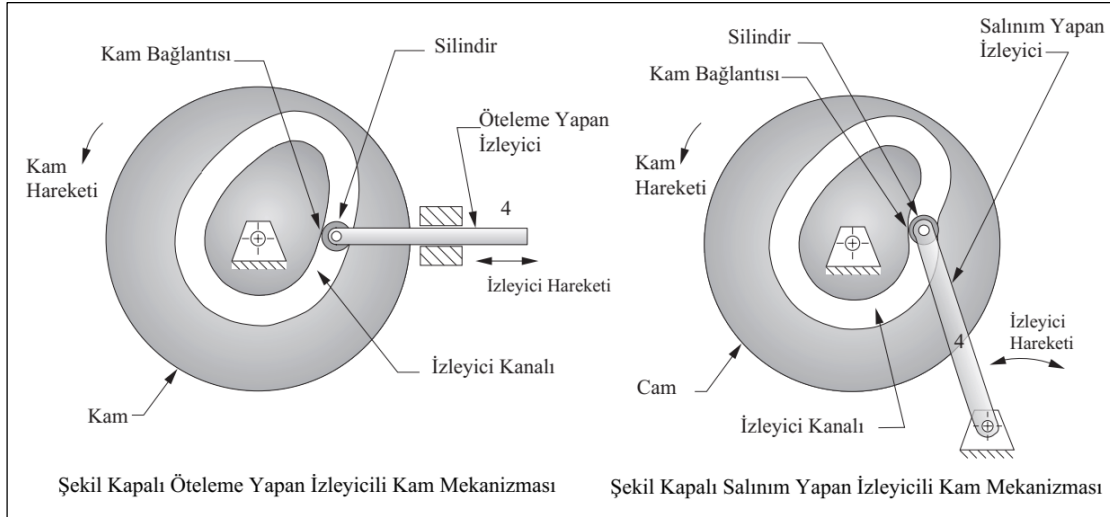
Şekil 1.2. Kam şekline göre sınıflandırma (Soyleman, 2007)

İzleyici hareketine ve şekline göre sınıflandırmalarda ise, Şekil 1.3a’da gösterilen izleyicinin salınım ya da öteleme hareketi yapması, Şekil 1.3b’de gösterilen biçimde izleyicinin kam ile temas eden yüzeyi dikkate alınır. Bu yüzey düz, küresel veya silindirik, makaralı (yuvarlanmalı) biçimde olabilir.



Şekil 1.3. İzleyici hareketine (a) ve şekline (b) göre sınıflandırma (Norton, 2009)

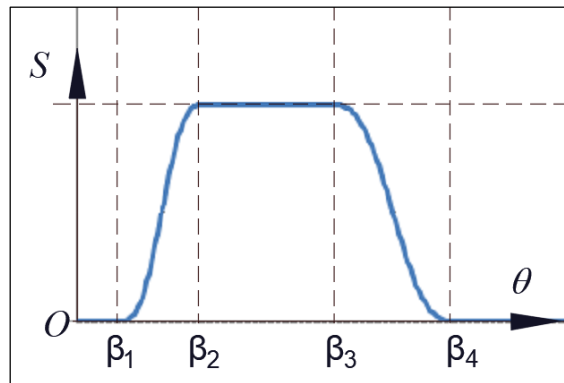
Şekil 1.3’ de gösterilen mekanizmalar kam mekanizmalarının bir diğer sınıflandırma şekli olan temas durumunun, kuvvet kapalı mekanizma olma durumuna örnek olup, şekil kapalı mekanizma olması durumu Şekil 1.4’te gösterilmiştir.



Şekil 1.4. Şekil kapalı kam mekanizması (Norton, 2009)

Kam mekanizmalarının tarifi yapılırken kam yapısı, izleyici yapısı ve hareketi, temas şekli gibi sınıflandırma türlerinin birlikte kullanılarak en kapsamlı biçimde ifade edilmek istenir.

Kam mekanizması tasarımı izleyicinin amaca uygun hareketi yapması için kam yüzey profilinin belirlenmesidir. Bu aşamada izleyicinin istenen konumda, hızda ve ivmede olması amaçlanmaktadır. Tasarım aşamasında çalışma koşulları göz önüne alınarak sistem dinamiğinde atalet kuvvetlerinin ihmal veya dahil edileceğine karar verilerek ilerlenmelidir.



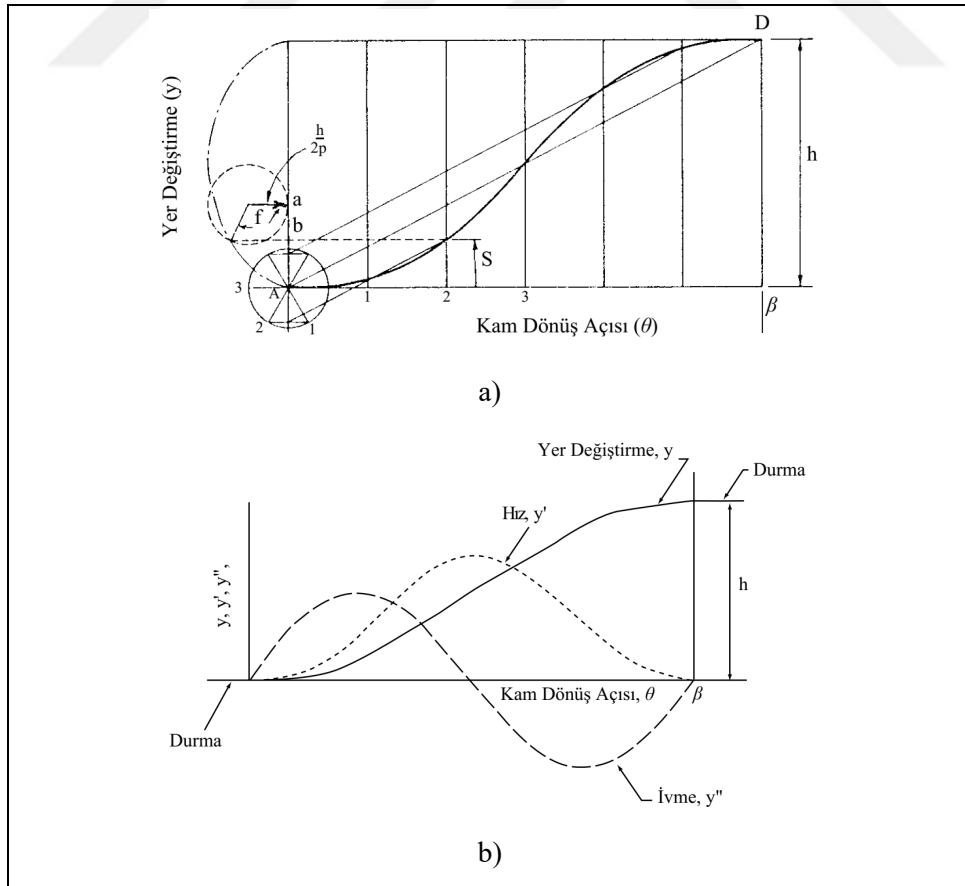
Şekil 1.5. Kam Mekanizması Tipik Hareket Eğrisi

Bir kam mekanizmasına ait tipik hareket eğrisi Şekil 1.5’teki biçimde, θ ile gösterilen kam dönüşünün bir tam turunda izleyicinin S kadar hareketi şeklinde gösterilir. Şekil 1.5’te verilen hareket

eğrisinde izleyici $0-\beta_1$ ve $\beta_2-\beta_3$ aralıklarında bekleme konumunda, $\beta_1-\beta_2$ ve $\beta_3-2\pi$ aralıklarında ise hızı ve ivmesi değişmektedir. İzleyicinin S miktarda yer değişimine karşılık, ilerleme hızı ve ivmesi β aralıklarının arttırılıp azaltılması ile ayarlanır. Şekil 1.5'te gösterilen hareket eğrisinde 2 adet bekleme ve 2 adet hareket durumu bulunmaktadır. Tasarım aşamasında, 1 turda kaç adet bekleme ve hareket yapılacağı belirlenir.

Kam mekanizmalarında izleyici hareketini belirleyen eğrileri elde etmek için kullanılan çeşitli yöntemler vardır. Tasarımında yaygın olarak ve tez kapsamında hazırlanan bilgisayar programında da kullanılan hareket türü sikloiddir. Bir doğru üzerinde kaymadan yuvarlanma hareketi yapan bir daire üzerindeki herhangi bir noktanın izlediği yol olarak tanımlanan sikloid eğrisinin (Şekil 1.6a) kam mekanizmaları tasarımında kullanımı aşağıdaki adımlarla uygulanır:

- Kam dönüş açısı (θ) eksenini ve merkezi A-D doğrusu üzerinde, çevresi izleyici öteleme miktarı (h) kadar olan çember eşit miktarda aralıklara bölünür.
- Çember üzerindeki bir noktanın yatay izdüşümünün dikey eksenini kestiği noktadan A-D doğrusuna paralel bir doğru çizilir.
- Açık ekseninde numaralandırma olarak çember üzerindeki denkleme noktadan dikey bir doğru çizilir
- Dikey doğru ile A-D doğrusuna paralel çizilen doğrunun kesişimi hareket eğrisi üzerindeki bir nokta olarak belirlenir.

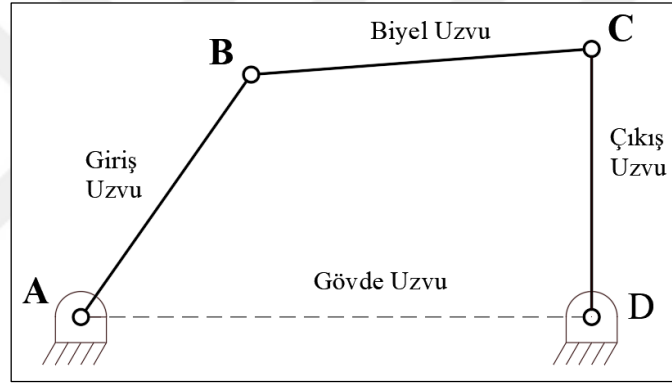


Şekil 1.6. Kam Tasarımında Sikloid Hareket Eğrisi (Rothbart, 2004)

Sikloid eğrisi ile kam tasarımı Şekil 1.6b’de görülen ivmenin sıfır olarak başlayıp bitmesi ile süreksizlik sorununun olmaması, mekanik olarak daha az gerilim ve yorulma yaratması nedenleriyle tercih edilmektedir.

1.2. 4 Çubuk Mekanizmaları

4 çubuk mekanizmaları geniş bir uygulama alanı olan düzlemsel bir mekanizma türü olup, kinematik yapısı Şekil 1.7’de görüldüğü biçimde 4 uzuvdan ve 4 döner mafsaldan (A, B, C, D) oluşan tek serbestlik dereceli bir mekanizmadır. Uygulamada çok çeşitli biçimlerde bulunabilen ve birleştirilmeleriyle yeni mekanizmaların elde edilebildiği bir mekanizma türüdür. Gövde uzvu sabit olup hareket etmeyen uzuv olduğundan, uygulamalarda sadece 3 eleman hareketli olarak görünür. Biyel uzvu ise gövde ile teması olmayan uzuvdur.



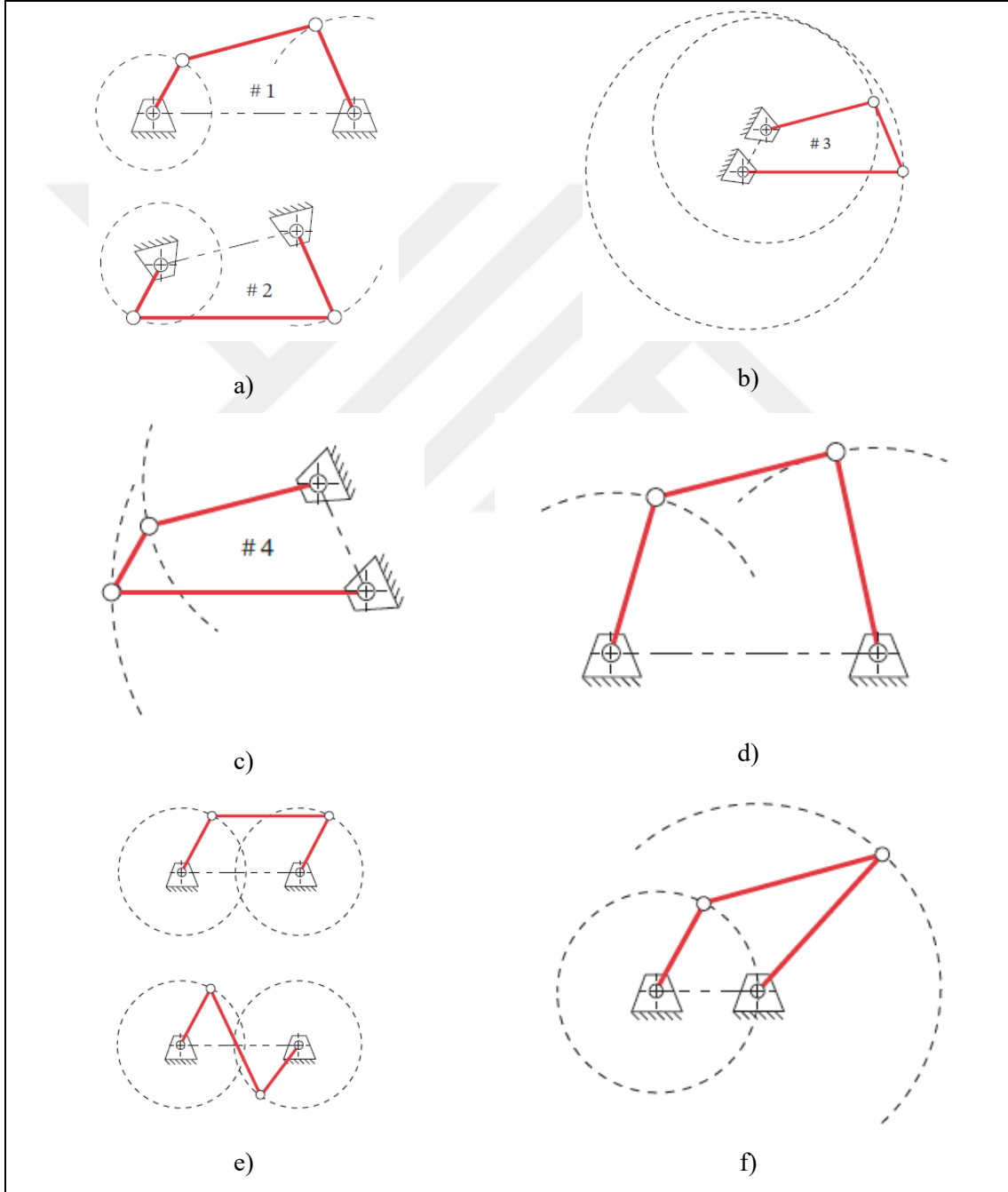
Şekil 1.7. 4 Çubuk Mekanizması Yapısı

Mekanizmada uzuvlar yaptıkları hareket tiplerine göre özel adlandırılmaktadırlar. Uzuvlar tam bir tur atmaları durumunda kol veya krank, belirli bir aralıkta salınım yapmaları durumunda ise sarkaç olarak ifade edilirler. Kolların boyundan gelen kısıtlamalara bağlı olarak kol-sarkaç, çift sarkaç, çift kol şeklinde ifade edilirler.

Hareket kısıtlamaları Grashof teoremine göre en kısa uzuv S, en uzun uzuv L, diğer iki uzuv da P ve Q olduğu düşünüldüğünde şu şekilde sınıflandırılmaktadır.

- $S + L < P + Q$ ve en kısa uzuv gövde uzvuna bağlı ise bu en kısa uzvun tam dönme hareketi yaptığı Şekil 1.8a’da görülen iki farklı kol-sarkaç
- $S + L < P + Q$ ve en kısa uzuv gövde uzvu ise Şekil 1.8b’de görülen çift krank
- $S + L < P + Q$ ve en kısa uzuv gövde uzvu uzvunun karşısında ise Şekil 1.8c’de görülen çift sarkaç
- $S + L > P + Q$ durumunda sabit uzva komşu uzuvların salınım hareketi yaptığı Şekil 1.8d’de görülen 4 farklı çift sarkaç (sadece biri gösterilmiştir.)

- e) $S + L = P + Q$ durumunda mekanizma, hareketi öngörülemez belirsiz bir noktaya sahip olmak üzere yine a, b, c maddelerindeki mekanizmalar meydana gelmektedir. Ek olarak eşit uzunluktaki kolların karşılıklı olması durumunda iki farklı bağlantı tipi olan paralelogram (Şekil 1.8e) mekanizması oluşmaktadır. Eşit uzunlukta kolların yan yana olması durumunda ise, uzun uzuvlardan birinin gövde olması durumunda kol sarkaç, kısa uzuvlardan birinin gövde olması durumunda ise diğer kısa kolun 2 turuna karşılık uzun kolun tek tur attığı ve Galloway adıyla anılan 4 çubuk mekanizması oluşmaktadır (Şekil 1.8f).



Şekil 1.8. Grasshof Teoremine Göre 4 Çubuk Mekanizması Hareketleri (Norton, 2020)

4 çubuk mekanizmalarında giriş uzvu ile çıkış uzvunun açısal dönme miktarları arasında kurulan ilişkiye fonksiyon sentezi adı verilmektedir. Bu durumda ihtiyaca yönelik tasarımda, çıkış uzvunun yapacağı dönme miktarı giriş uzvu dönme miktarının bir fonksiyonu olacak şekilde ilişkilendirilir.

Mekanizmanın hareketi sırasında gövde uzvu ile teması olmayan biyel uzvunun üzerindeki herhangi bir biyel noktasının izleyeceği eğri biyel eğrisi olarak adlandırılmaktadır. Biyel üzerindeki belirlenen bir noktanın mekanizmanın belirli bir hareket aralığında kabul edilebilir yapısal hata sınırlarında istenen bir doğru, çember veya eğri çizmesi için yapılan çalışmalar ise yörünge sentezi kapsamındadır.

Bu çalışmada düzlemsel kam mekanizması, 4 çubuk mekanizmaları ve döngüsel bir kompresör yapısının kinematik sentezine yönelik yapılan özgün çözüm yöntemleri kullanan ve sonuçların değerlendirilmesine yönelik bilgisayar programı geliştirilmiştir. Her mekanizma ve sentez tipine yönelik modül adı verilen ayrı bir program hazırlanmıştır. Programın temel amacı matematiksel olarak bulunan sonuçlardan uygun olmayanların kolaylıkla elenmesi, uygun olanların görsel olarak kullanıcıya hemen sunulması, sonuçların çeşitli dosya tiplerinde kaydedilerek farklı bilgisayar platformlarında kullanılmak üzere depolanması işlerini otomatik hale getirerek tasarım ve kontrol sürecini hızlandırmaktır.

Geliştirilen yazılım günümüzde kişisel bilgisayarlarda yaygın olarak kullanılan işletim sistemi olan Microsoft Windows ortamında çalışmak üzere popülaritesi giderek artan ve mühendislik alanında çok sayıda hazır kütüphaneye sahip Python programlama dili ile geliştirilmiştir. Geliştirilen programın; kolay ve etkileşimli veri girişi, elde edilen tasarım sonuçlarının grafik ortamda kolay ve hızlı olarak değerlendirilmeye elverişli olması, tasarım sonuçlarının sanal ortamda simülasyonlarının yapılarak uygun tasarıma hızlı ve kolay ulaşılır olması, sonuçların daha detaylı değerlendirilmesi ve imalata uygun hale getirilmesi amacıyla yaygın olarak kullanılan bazı universal yazılımlara uygun çıktılar üretebilmesi gibi özellikleri, endüstri, tarım ve günlük yaşamda görülen makinaların pek çoğunun tasarımında kinematik sentez yöntemlerine gereksinim olması nedenleriyle ulusal ve uluslararası ölçekte pazarlanabilir bir ürün olması amaçlanmaktadır.



2. KAYNAK ARAŞTIRMALARI

Mekanizmalar alanında endüstrideki yeni ihtiyaçların karşılanması, mevcut mekanizmaların gelişen teknoloji ve imkanlarla daha iyi tasarlanması ve imal edilmesi, çalışma sonuçlarının hızlı biçimde değerlendirilmesi amacıyla pek çok akademik ve ticari çalışmalar bulunmaktadır. Bu bölümün konusu, ulaşılabilirliği kolaylaşan ve artan bilgisayar teknolojisi sayesinde mekanizmaların kinematik sentez yöntemlerinde yapılan güncel araştırmalardır.

Flores (2013), yuvarlanmalı izleyici tipine sahip düzlemsel kam mekanizmalarının boyut optimizasyonu üzerine hesaplamalı bir yaklaşım sunmaktadır. Çalışmada, amaç fonksiyonu kamın temel yarıçapı, izleyici yarıçapı ve izleyici öteleme miktarının yanı sıra basınç açısı ve eğrilik yarıçapı kısıtlamaları da hesaplamalara dahil edilmiştir. Çalışmada, yaklaşımın hız ve ivme gibi diğer büyüklüklerin hesaplanması noktasında geliştirilebilir olduğu belirtilmiştir.

Pan vd. (2022) ise, kam temel yarıçapı, izleyici yarıçapı ve izleyici öteleme miktarına ek olarak izleyici kol uzunluğunun da dahil edildiği kam tasarım parametrelerinin optimizasyonu için bir hesaplama yaklaşımı sunmuştur. Matlab yazılımında optimize edilebilir nonlinear hesaplama yöntemi, motor eksantrik mili üzerinde bir örnek olarak uygulanarak etkinliği gösterilmiştir.

Zhang ve Shin, (2004), kam profilinin oluşturulmasında, basınç açısı ve eğrilik yarıçapı hesabında göreceli hareket yöntemi adında izleyicinin hız ve ivme değerlerini kullanan bir yaklaşım ortaya koymuşlardır. Sadece yuvarlanmalı değil, değişik izleyici tipleri için de uygulanabilir olan bu yaklaşım, kam şekli tasarımı ve kinematik simülasyon için verimli ve hızlı kullanım amacıyla bir bilgisayar programı ile otomatize edilmiş ve örneklendirilmiştir.

Petropoulou vd. (2009), kam mekanizmalarının tasarım ve üretim süreçlerini kolaylaştırmak amacıyla kam profilindeki üç ardışık noktanın vektör farklarını kullanarak iterasyonlu biçimde izleyicinin salınım, hız, ivme ve sarsım (jerk) değerlerinin hesaplanmasını içeren bilgisayar destekli sayısal bir tasarım yöntemi sunmuşlardır. Sayısal örneklerle, yöntemin kabul edilebilir sınırlar içerisinde olduğu belirtilmiştir.

Trajcevski ve Geramitchioski (2013), Delphi dili kullanarak geliştirdikleri CAMS programı ile düz yüzeyli ve yuvarlanmalı makara tipli izleyicilere sahip kam mekanizmaları için kinematik sentez ve analiz hesaplamaları yapan profil, kinematik, dinamik, animasyon şeklinde çeşitli alt programlardan oluşan bir program geliştirmişlerdir. Yapılan hesaplamalardan sonra program, giriş ve çıkış bileşenlerinin tamamen grafiksel bir görselleştirmesini ve animasyonunu gösterebilmektedir.

Mundo vd. (2006), mekanik preslerin kinematik davranışını iyileştirmek amacıyla presin koçu için sabit giriş hızıyla çalışan ve optimal bir hareket oluşturabilen bir fonksiyon üreten mekanizma tasarlamak üzere çalışmışlardır. Bu amaçla, öncelikle istenen harekete yönelik tip sentezi yapılarak çok serbestlik dereceli bir mekanizma seçilmiş ve sonrasında sistemin hareketi bir veya daha fazla kam kullanılarak tek girişli olarak kısıtlanmıştır. Sentez sürecini boyut ve kinematik kriterlere göre optimize

etmek için genetik bir algoritma kullanılmış ve önerilen metodoloji, derin çekme ve hassas kesme işlemleri için mekanik preslerin kinematik optimizasyonuna uygulanmıştır.

Shala vd. (2013), izleyicinin bilinen ivme diyagramını temel alan kam profili oluşturmak üzere analitik bir yöntem sunmuşlar ve kam profilindeki üretim hatalarının mekanizmanın kinematik ve dinamik karakteristiklerini çok etkilediğini MathCad ve Working Model yazılımları ile simule ederek göstermişlerdir.

Zhu ve Lan (2017), yaptıkları kam tasarımında matematik hesaplarını kolaylaştırmak adına yapıyı iki elemanlı kinematik uzuv zincirine dönüştürmüş, kam profilinin basınç açısını ve eğrilik yarıçapını dahil ederek mekanizmaya açısal hızı ters kinematik yöntemle uygulamış ve kam profili elde etmişlerdir.

Simionescu ve Ortega (2020), Working Model 2D yazılımını kullanarak çeşitli izleyici tiplerine sahip kam mekanizmaları için hareketi tersinden vererek kam profili oluşturmuş, takipçinin hareketi, temel çember yarıçapı, takipçi geometrisi ve kamın alt kesimi arasındaki ilişkileri göstermişlerdir. Sundukları yöntem ile düz yüzlü veya silindirik olmayan takipçilere sahip kam mekanizmalarının diğerlerine nazaran daha kolay sentezlendiği sonucuna varmışlardır.

Simionescu ve Neagoe (2020), izleyici hareketini bir dosyadan okuyarak ve Matlab yazılımı kullanarak kam sentezi yapmak için, kamın sabit tutularak izleyicinin ters yönde hareket ettiği bir yaklaşım uygulamışlar ve sonuç profilinin Autocad dosya tipinde polyline olarak verilmesini sağlamışlardır.

Zhu vd. (2023), yüksek hızlarda çalışan kam mekanizmalarının tasarımı için yapıyı öncelikle 5 çubuk mekanizması olarak matematik model oluşturmuş, mafsallar arası mesafe ve hızın dahil edildiği bir çözüm ortaya koymuşlardır. Sonrasında ise, kam basınç açısı ve izleyici ivme değerlerini en aza indirerek en uygun yapısal parametreleri ve çıkış hızı eğrisini bulmak üzere optimizasyon yapılmışlardır.

Yapılan akademik çalışmaların yanında kam tasarımı ve analizlerinin yapıldığı, çeşitli platformlarda çalışabilen ticari birçok yazılım da bulunmaktadır. Bunlardan bazıları kendi başına bir ürünken, bir kısmı çeşitli CAD programlarında kullanılmak üzere geliştirilmiş eklentiler şeklindedir.

Li vd. (2016), fonksiyon sentezi için özgün bir analitik yöntem olarak yaptıkları çalışmada çıkış uzvu dönme-açısı fonksiyonunu, giriş uzvu dönme açısının Fourier serisi bileşenlerine göre formüle ederek ve bir vektörel döngü denklemi oluşturarak tasarım parametreleri ve Fourier katsayıları arasında yeni tasarım denklemleri ile ilişki kurmuş ve problemi kübik bir polinom denkleminin çözümüne indirgemişlerdir. Mevcut optimizasyon tekniklerinin aksine, yalnızca tek bir çözüm sağlamak yerine, önerilen yöntem tarafından sağlanan üç çözüm, tasarımcıya daha fazla seçenek sunarak yöntemin doğruluğu tatmin edici olmadığı durumlarda daha fazla optimizasyon için başlangıç çözümleri olarak da kullanılabilir olacağı belirtilmiştir.

Shariati ve Norouzi (2011), dört çubuk mekanizmalarında istenen bir matematik fonksiyonunun üretimi için üretilen fonksiyon ile istenen fonksiyon arasındaki hata karelerinin en az olması temelinde bir çalışma yapmışlardır. Amaç fonksiyonunun ve kısıtlamanın doğrusal olmaması nedeniyle amaç

fonksiyonunu minimize etmek ve en iyi mekanizma için SQP (sequential quadratic programming) yöntemi kullanılmıştır. Sentez için 5 farklı kesin nokta kullanılarak bir dizi nonlineer denklem elde edilmiş ve boyutsal parametreler yerine bu denklemler kullanılmıştır.

Wang vd. (2021), fonksiyon sentezi ve hareket sentezini birlikte yaptıkları numerik çalışmada yöntem, bağımlı ve bağımsız kombinasyon olarak iki bölümden oluşmaktadır. Bağımsız kombinasyonda fonksiyon ve hareket sentezleri ayrı ayrı yapılmakta, bağımlıda ise sentezlerin ise en az bir adet ortak pozisyonu olmaktadır. Çalışmada bağımlı kombinasyon için beş uygulanabilir durum ve bağımsız kombinasyon için iki uygulanabilir durum analiz edilmiştir.

Stoilov Todorov (2015), istenen bir giriş-çıkış hareketini üretmek amacıyla dört çubuk mekanizmalarının sentezi için yeni bir yöntem tanımlamaktadır. Sentez yöntemi, Chebyshev'in Alternasyon teoreminin Freudenstein Denklemine uygulanmasına dayanmaktadır. Bu yaklaşım sayesinde, maksimum yapısal hata önceden tahmin edilebilmektedir. Çalışmadaki yenilik, hedef fonksiyonun kesin olarak tahmin edildiği bir denklem olarak sunulmasıdır. Bu denklemin çözümünün Freudenstein denkleminde yerine konmasıyla, Chebyshev'in genelleştirilmiş bir polinomu elde edilerek Chebyshev'in alternasyon teoremiyle minimize edilmektedir. Önerilen bu yaklaşımın avantajlarından biri, çözümün başlangıcında sentez problemini göz önünde bulundurarak mekanizmanın özelliklerinin tahmin edilebilmesidir.

Soong ve Chang (2011), orjinal 4 çubuk mekanizmasının yapısını, giriş uzvu ile biyel uzvu mafsalını silindir çifti haline getirerek değişken kol boyları için fonksiyon üreten bir tasarım yöntemi ortaya koymuştur. Tasarım için gerekli kesin noktaların sınırlaması olmadığı belirtilen yöntem, çeşitli örneklerde yapısal hataların, geleneksel üç kesin nokta tarafından üretilen hatalardan önemli ölçüde daha küçük olduğunu göstermiştir.

Shiakolas vd. (2002), ilk tahmini değerlerin iyi seçilmesine dayanan tekniklerden farklı olarak başlangıçta tanımlanan ilk sınırlardan farklı aramalar yapabilen diferansiyel gelişim, yeni sundukları ve kesin noktalarının geometrik merkezi (GCPP) olarak adlandırdıkları teknik ve başlangıç sınırlamasındaki kesin noktaların uzaklığı bilgilerini birleştiren bir yöntem sunmuşlardır. 4 çubuk mekanizmalarının yörünge sentezinde uygulanan yöntemle, öngörülen bir zamanlamayla biyel noktasının doğru sırada ve makul bir doğruluk seviyesinde bir dizi noktadan geçmesiyle biyel eğrilerinin bulunması sağlanmıştır. Ayrıca bu yöntemin sonuçlarının istenen doğruluk seviyesinde iyileştirilmesi için ilk tahmini değerler olarak da kullanılabileceği belirtilmiştir.

Buśkiewicz (2015), yörünge sentezinde tasarım parametre sayısını azaltmak üzere yaptığı çalışmada, biyel uzvunu mekanizmanın boyutlarından bağımsız olarak istenen yörüngeyi kesin olarak yapan serbest bir cisim olarak ele almış ve açısal konumunu bir sinüs fonksiyonu olarak tanımlamıştır. Metodun amacı bu biyelin kullanıldığı bir mekanizmada biyel mafsallarının yörüngesi öngörülen bir doğruluk içindeyken biyel boyutlarını bulmaktır. Yaklaşımın farkı biyel noktalarının değil, biyel uzvuna ait mafsal noktalarının yörüngelerindeki sapmanın ölçülmesidir.

Zarkandi (2021), dört çubuk ve krank biyel mekanizmalarının yörünge sentezi problemlerinin devamlılık metotlarından farklı çoklu çözümü (cognates) için Yerçekimi Arama Algoritması (GSA) ile minimize edilen, önceki sonuçları ve kongratlarını dikkate alan yeni bir hata fonksiyonu ve optimizasyon tabanlı bir algoritma sunarak sezgisel algoritmaların uygulanabilirliğini çalışmıştır. Çeşitli örneklerle uygulanan yeni yöntemin diğer yöntemlere göre daha yüksek doğruluk ve yakınsama oranına sahip olduğu belirtilmiştir.

Bureerat ve Slesongsom (2021), kısıtlama tekniği olarak giriş açısı ile harici ceza fonksiyonları kullanımından farklı olarak giriş kolu dönüşü ve Grashof kriterini birlikte kullanarak 4 çubuk mekanizmalarında yörünge sentezi üzerine yaptıkları çalışmada, öğretme-öğrenme tabanlı optimizasyon ile üst sezgisel algoritma kullanmışlardır. Uygulanan tekniğin harici ceza fonksiyonu kullanan tekniklerden çok daha iyi sonuçlar verdiği ve ilerleyen çalışmalarda uçak kanatlarının flaplerinde kullanılacağı belirtilmiştir.

Kafas ve Nahvi (2017), dört çubuk mekanizmasında biyel uzvu üzerinde bulunan iki mafsalin her zaman çembersel birer yörüngeleri olduğundan yola çıkarak yeni bir çözüm yaklaşımı sunmaktadırlar. Bu yaklaşımda, iki elemanlı, iki serbestlik dereceli olmasından dolayı serbest uzvu üzerinde sınırsız sayıda nokta ve yörünge tanımlanabilen bir dyad kullanılır. Serbest uzuv üzerinde çembere yakın bir yörüngesi olan nokta aranır ve böyle bir nokta bulunduğunda dyadın yeni mafsali noktası olarak tanımlanır. Bu amaçla bir çember yakınlık fonksiyonu ve buna bağlı bir amaç fonksiyonu tanımlanarak diferansiyel gelişim yöntemiyle optimize edilmektedir. Yöntemin dokuz tasarım değişkeni yerine dört optimizasyon değişkeni ile işlem yapması ve optimizasyon değişkeni sayısının istenen kesin noktalardan bağımsız olmasının avantaj olduğu belirtilmiştir.

Nadal vd. (2015), çalışmalarında biyel noktası tarafından üretilen ve hedef yörünge arasındaki hataları karşılaştırmak için elde edilen hatanın sabit uzvun yer değişimine ve dönmesine, mekanizmanın ölçek değişimine bağlı olmamasından dolayı dönüş fonksiyonlarının kullanımını açıklamışlardır. Bu sayede sentez için kullanılan genetik algoritmada optimizasyon için arama alanı azaltılmaktadır. Metod, amaç ve gerçek yörüngeleri karşılaştırarak bunlar arasında normalize edilmiş yay ve açı taramaları arasında bir fonksiyon oluşturur. Dört çubuk mekanizmasında uygulanan yöntemin başka mekanizma tipleri için de uygulanabileceğini belirtmişlerdir.

t_i : iç izleyicinin temas anında kam iç profil eğrisine teğeti

t_d : dış izleyicinin temas anında kam dış profil eğrisine teğeti

n_i : t_i teğetinin normali

n_d : t_d teğetinin normali

Φ_d : dış izleyici basınç açısı

Φ_i : iç izleyici basınç açısı olarak ifade edilmiştir.

Şekil 3.1’de hem iç hem dış temaslı kam yapısı birlikte gösterilmiş olup sadece iç ve sadece dış temaslı kam tasarımları da aynı parametrelerle yapılmaktadır.

Kam merkezi ve izleyici merkezleri arası mesafeler şu şekilde bulunmaktadır;

$$r_{\theta d} = R_d + R_2 + S_n(\theta), n = 1,2,3 \dots 16 \quad (3.1)$$

$$r_{\theta i} = R_i + R_2 + S_n(\theta), n = 1,2,3 \dots 16 \quad (3.2)$$

İzleyici merkezlerinin kartezyen koordinat düzlemindeki noktaları ise;

$$X'_{ci} = r_{\theta i} \cos \theta_i \quad (3.3)$$

$$Y'_{ci} = r_{\theta i} \sin \theta_i \quad (3.4)$$

$$X'_{cd} = r_{\theta d} \cos \theta_d \quad (3.5)$$

$$Y'_{cd} = r_{\theta d} \sin \theta_d \quad (3.6)$$

denklemleri ile bulunmaktadır.

Kam mekanizmasının önemli bir parametresi olan basınç açıları ise iç ve dış izleyici için;

$$\Phi_i = \tan^{-1} \left(\frac{1}{r_{\theta i}} \frac{dS_n(\theta)}{d\theta} \right) \quad (3.7)$$

$$\Phi_d = \tan^{-1} \left(\frac{1}{r_{\theta d}} \frac{dS_n(\theta)}{d\theta} \right) \quad (3.8)$$

bağıntılarıyla bulunmaktadır.

Yarıçapı R_2 birim olan izleyicilerin kam iç ve dış profil eğrilerine temas ettiği noktalar olan P_i ve P_d noktalarının bulunmasında ise aşağıdaki eşitlikler kullanılmaktadır.

$$\gamma_d = \theta - \Phi_i \quad (3.9)$$

$$\gamma_d = \theta - \Phi_d \quad (3.10)$$

$$X'_{pi} \theta = X'_{ci} \theta - R_2 \cos \gamma_d \quad (3.11)$$

$$Y'_{pi}\theta = Y'_{ci}\theta - R_2 \cos \gamma_d \quad (3.12)$$

$$X'_{pd}\theta = X'_{cd}\theta - R_2 \cos \gamma_d \quad (3.13)$$

$$Y'_{pd}\theta = Y'_{cd}\theta - R_2 \cos \gamma_d \quad (3.14)$$

İzleyici yer değiştirmesi (H), toplam yer değiştirme miktarı süresindeki kam dönme açısı (β) ve kamın toplam dönme açısı (θ) olmak üzere mekanizmanın sikloid eğrisi aşağıda belirtilen genel formül ile bulunmaktadır.

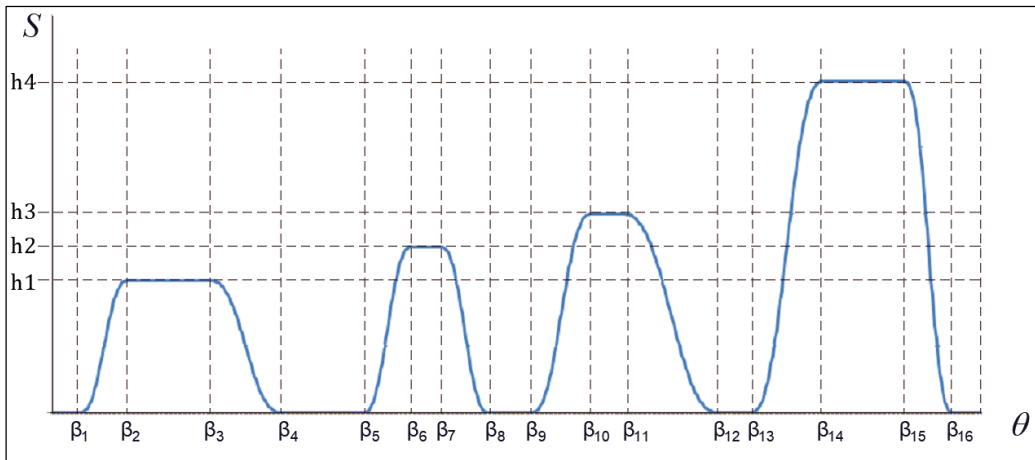
$$S(\theta) = \frac{H}{\pi} \left(\frac{\pi\theta}{\beta} - \frac{1}{2} \sin\left(\frac{2\pi\theta}{\beta}\right) \right) \quad (3.15)$$

İzleyici yer değiştirme miktarını veren (3.15) denkleminin birinci türevi ile hız (3.16), ikinci türevi ile de ivme (3.17) denklemleri elde edilmektedir. Tez kapsamında kam mekanizması kinematik olarak değerlendirilerek yalnızca profil eğrisinin hesaplanması ve izleyici ilerleme miktarı hesaplanmıştır. Hız ve ivme değerleri yalnızca katsayı olarak hesaplanmıştır

$$V = \frac{H\omega}{\beta} \left(1 - \cos\left(\frac{2\pi\theta}{\beta}\right) \right) \quad (3.16)$$

$$a = \frac{2H\omega^2\pi}{\beta^2} \sin\left(\frac{2\pi\theta}{\beta}\right) \quad (3.17)$$

Tez kapsamında yapılan bilgisayar programında kam mekanizması tasarımı bir kam dönmesine karşılık 4 farklı hareket değişimi olacak şekilde sınırlandırılmış olup bu hareketi ve parametrelerini gösteren sikloid eğrisi Şekil 3.2'deki gibidir.



Şekil 3.2. 4 hareketli kam mekanizması eğrisi

Bu eğride gösterilen θ kam dönme açılarına karşılık gelen her izleyici yer değişimi miktarları aşağıdaki gibi hesaplanmıştır;

$$S_1(\theta) = 0 \quad (3.18)$$

$$S_2(\theta) = \frac{H}{\pi} \left(\frac{\pi(\theta - \beta_{n-3})}{\beta_{n-2} - \beta_{n-3}} \right) - \frac{1}{2} \sin\left(\frac{2\pi(\theta - \beta_{n-3})}{\beta_n - \beta_{n-3}}\right), n = 1,2,3 \dots 16 \quad (3.19)$$

$$S_3(\theta) = 0 \quad (3.20)$$

$$S_4(\theta) = H - \frac{H}{\pi} \left(\frac{\pi(\theta - \beta_{n-1})}{\beta_n - \beta_{n-1}} \right) - \frac{1}{2} \sin\left(\frac{2\pi(\theta - \beta_{n-1})}{\beta_n - \beta_{n-1}}\right), n = 1,2,3 \dots 16 \quad (3.21)$$

S değerlerinin birinci türevleri alınarak elde edilen $ds/d\theta$ eşitlikleri ise hız denklemleri olup her izleyici yer değişimi için şu şekildedir;

$$\frac{ds_1(\theta)}{d\theta} = 0 \quad (3.22)$$

$$\frac{ds_2(\theta)}{d\theta} = \frac{H}{\beta_{n-2} - \beta_{n-3}} \left(1 - \cos\left(\frac{2\pi((\theta - \beta_{n-3}))}{\beta_{n-2} - \beta_{n-3}}\right) \right), n = 1,2,3 \dots 16 \quad (3.23)$$

$$\frac{ds_3(\theta)}{d\theta} = 0 \quad (3.24)$$

$$\frac{ds_4(\theta)}{d\theta} = -\frac{H}{\beta_n - \beta_{n-1}} \left(1 - \cos\left(\frac{2\pi((\theta - \beta_{n-1}))}{\beta_n - \beta_{n-1}}\right) \right), n = 1,2,3 \dots 16 \quad (3.25)$$

$ds/d\theta$ eşitliklerinin birinci türevleri alınarak elde edilen $d^2s(\theta)/d\theta^2$ eşitlikleri ise ivme denklemleri olup her izleyici yer değişimi için şu şekildedir;

$$\frac{d^2s_1(\theta)}{d\theta^2} = 0 \quad (3.26)$$

$$\frac{d^2s_2(\theta)}{d\theta^2} = \frac{H}{(\beta_{n-2} - \beta_{n-3})^2} (2\pi \sin\left(\frac{2\pi(\theta - \beta_{n-3})}{\beta_{n-2} - \beta_{n-3}}\right)), n = 1,2,3 \dots 16 \quad (3.27)$$

$$\frac{d^2s_3(\theta)}{d\theta^2} = 0 \quad (3.28)$$

$$\frac{d^2s_4(\theta)}{d\theta^2} = -\frac{H}{(\beta_{n-2} - \beta_{n-3})^2} \left(2\pi \sin\left(\frac{2\pi((\theta - \beta_{n-1}))}{\beta_n - \beta_{n-1}}\right) \right), n = 1,2,3 \dots 16 \quad (3.29)$$

3.2. Kam Mekanizması Modülü Program Arayüzü

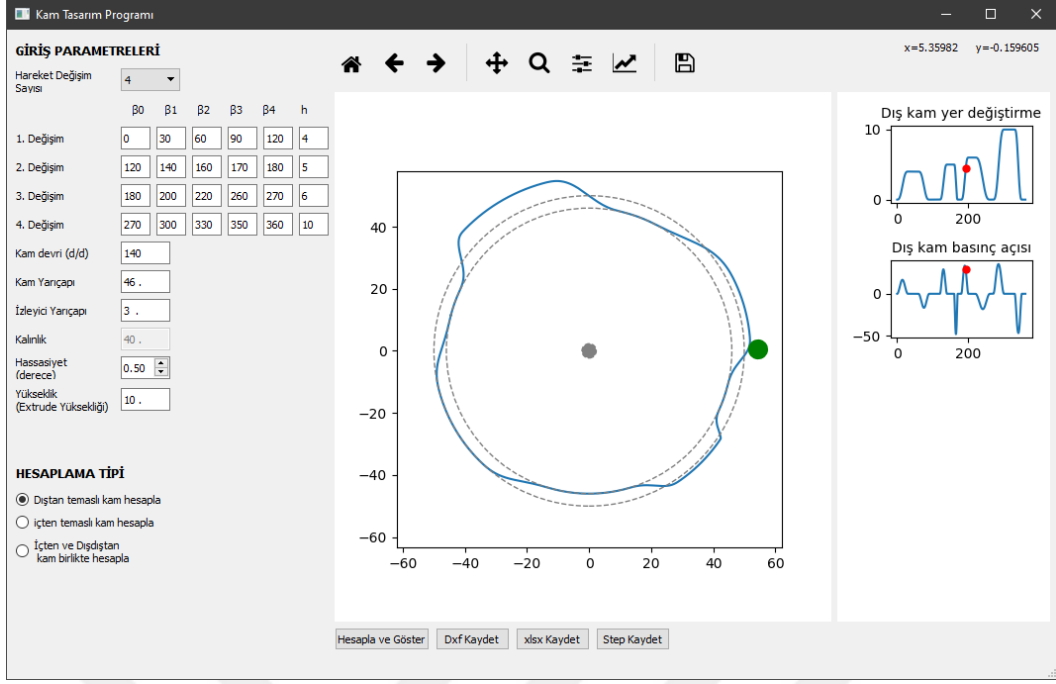
Bölüm 3’de açıklanan teori kullanılarak geliştirilen bilgisayar programı günümüzde en popüler olan programlama dillerinden birisi olan Python ile yazılmıştır. Python geniş ve güncellenen kütüphane desteği, platformlar arası uyumluluk, etkin bir kullanıcı topluluğu gibi nedenlerle tercih edilmiştir.

Kam programı öncelikle yapılan hesaplamaların kontrolü için metin tabanlı konsol uygulaması olarak yazılmış, sonrasında ise Şekil 3.3’de görülen kullanıcı arayüzü “PyQt5” kütüphanesi kullanılarak oluşturulmuştur. Kullanıcı arayüzünün sol üst köşesinde bulunan “hareket değişim sayısı” açılır menüsü ile bir kam devrinde olması istenen değişim miktarı belirlenmekte olup tez kapsamında dört ile sınırlandırılmıştır.

Yapılan seçim sonrasında, kullanıcıdan Şekil 3.2’de belirtilen θ ve β değerleriyle birlikte hangi hesaplama tipinin kullanılacağı, kam devri, kam yarıçapı, izleyici yarıçapı, kam kalınlığı (iç ve dış izleyici arası mesafe), derece cinsinden hesap hassasiyeti ve 3D formatında sonuç vermek üzere yükseklik bilgilerinin girilmesi istenmektedir.

Şekil 3.3. Kam Tasarım Programı Kullanıcı Arayüzü

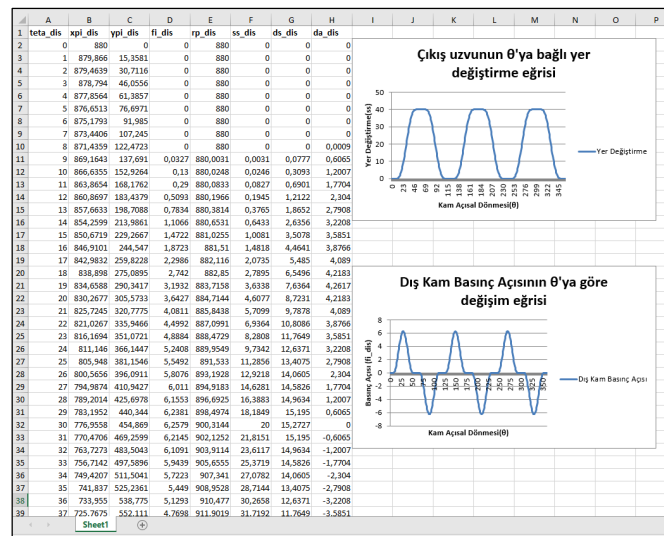
Kullanıcının “Hesapla ve Göster” butonuna tıklaması sonrası seçim tipi ve girilen parametrelere uygun hesaplanan sonuçlar hareketli görsek olarak kullanıcıya gösterilmektedir. Sağ yan bölümde ise anlık olarak izleyici yer değiştirme miktarları ve basınç açıları takip edilebilmektedir (Şekil 3.4).



Şekil 3.4. Kam Tasarım Programı Animasyon ve Grafik Sunumu

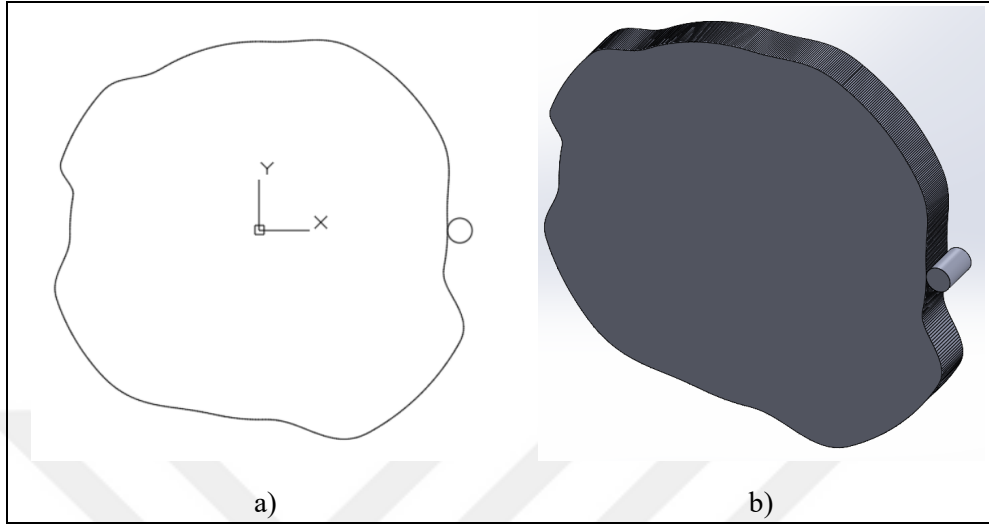
Kullanıcının kayıt öncesi önizleme yapabilmesi, istediğinde durdurma ve devam etme, yaklaşma ve uzaklaşma gibi görsel düzenlemeler ve grafikler matplotlib kütüphanesi kullanılarak yapılmıştır.

Programda kullanıcıya detaylı inceleme yapmak üzere sonuçların çeşitli dosya tiplerinde verilmesi sağlanmıştır. Şekil 3.5'te kullanıcının “xlsx kaydet” butonuna basması sonucu MS Excel dosya tipinde tüm sonuçlar belirlenen hassasiyetle tablo ve grafik olarak sunulmaktadır. Bu dosya kayıt işlemleri “xlswriter” kütüphanesi kullanılarak yapılmıştır.



Şekil 3.5. Xlsx dosya türünde kaydedilen kam tasarım sonuçları

Sonuçlar, incelemek veya başka bir CAD programında düzenlenmek üzere “ezdxf” kütüphanesi kullanılarak hazırlanan 2D dosya formatı olarak “dxf” ve FreeCAD kütüphanesi kullanılarak hazırlanan 3D dosya formatı “Step” biçimlerinde verilebilmektedir (Şekil 3.6).

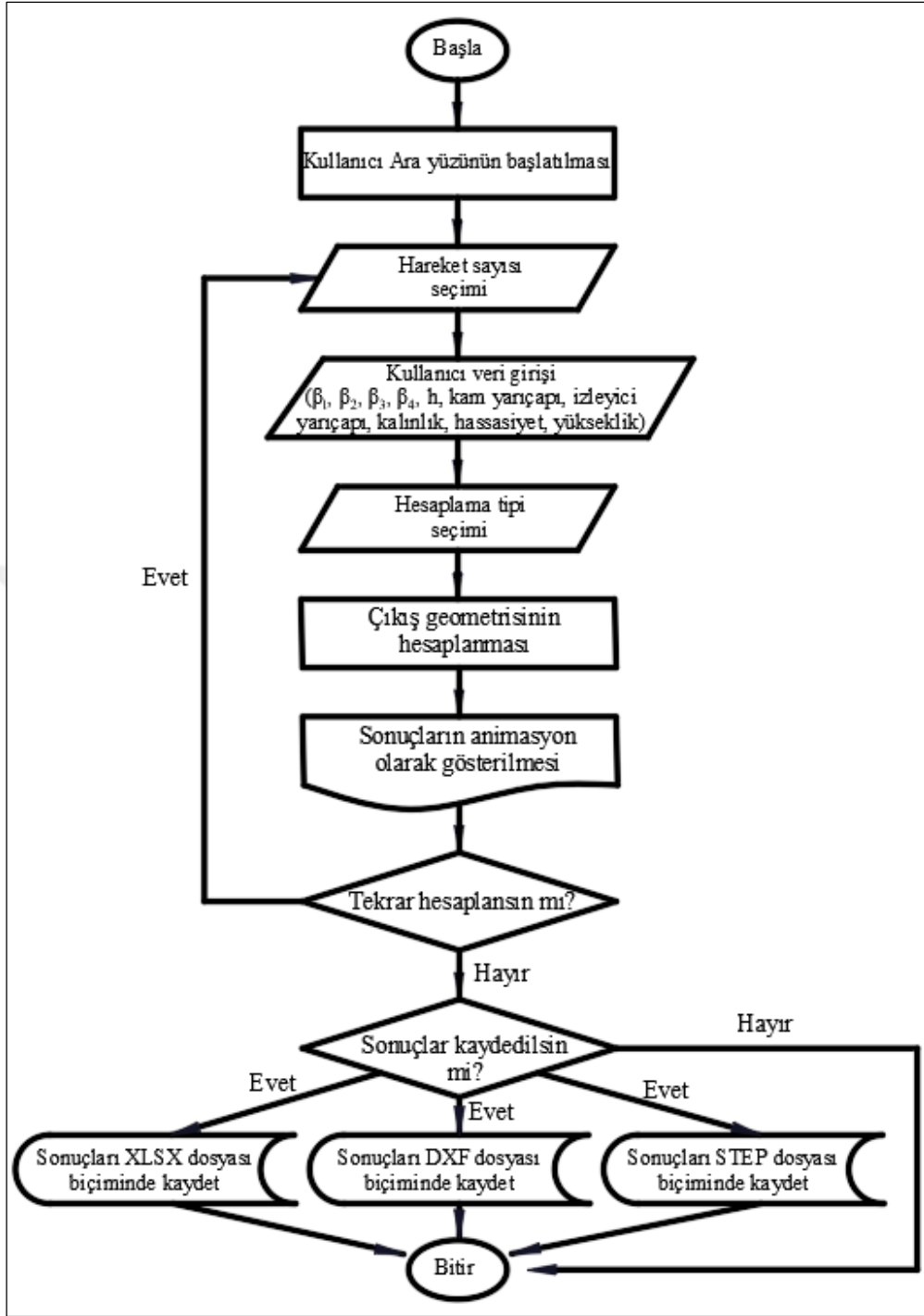


Şekil 3.6. Dxf (a) ve Step (b) Dosya Türünde Kaydedilen Kam Tasarım Sonuçları

Programın hesap adımları Tablo 3.1’de, bu adımlara dayalı oluşturulan akış diyagramı Şekil 3.7’de sunulmuştur.

Tablo 3.1. Kam Tasarım Programı Hesaplama Algoritması

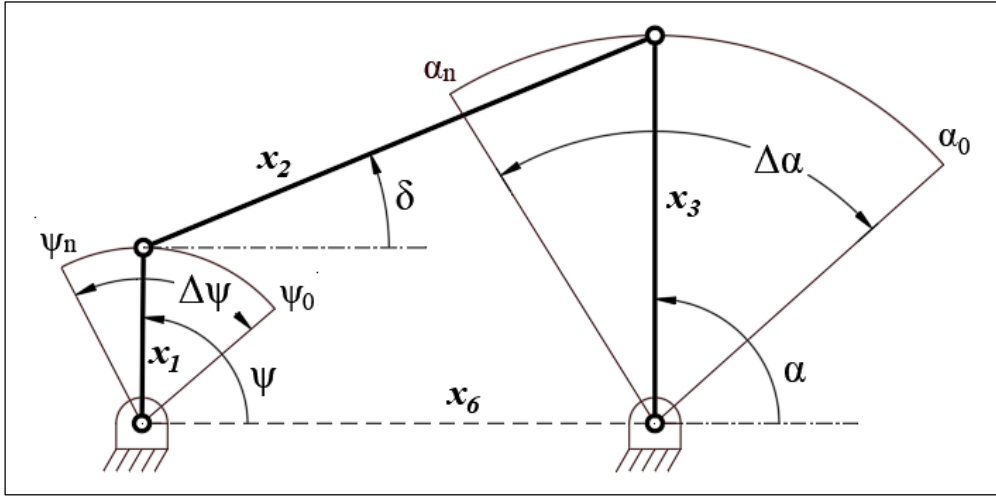
1.	Kullanıcıdan hareket sayısı, $\beta_1, \beta_2, \beta_3, \beta_4, h$, kam yarıçapı, izleyici yarıçapı, kalınlık, hassasiyet, yükseklik parametrelerinin alınması
2.	Kam merkezi ve izleyici merkezleri arası mesafelerin ($r_{\theta d}$ ve $r_{\theta i}$) denklem (3.1) ve (3.2) kullanılarak hesaplanması
3.	İç ve dış izleyici merkezlerinin koordinatlarının ($X'_{ci}, Y'_{ci}, X'_{cd}$ ve Y'_{cd}) denklem (3.3), (3.4), (3.5) ve (3.6) kullanılarak hesaplanması
4.	İç ve dış izleyicilerin, basınç açılarının (Φ_i ve Φ_d) denklem (3.7) ve (3.8) kullanılarak hesaplanması
5.	İç ve dış izleyicilerin, kam iç ve dış profil eğrilerine temas ettiği noktalar olan P_i ve P_d noktalarının denklem (3.9), (3.10), (3.11), (3.12), (3.13) ve (3.14) kullanılarak hesaplanması
6.	Sikloid hareket eğrisine göre izleyici hareket miktarının her harekette (s) denklem (3.18), (3.19), (3.20) ve (3.21) kullanılarak hesaplanması
7.	Her hareket için izleyici hız miktarının denklem (3.22), (3.23), (3.24) ve (3.25) kullanılarak hesaplanması
8.	Her hareket için izleyici ivme miktarının denklem (3.26), (3.27), (3.28) ve (3.29) kullanılarak hesaplanması



Şekil 3.7. Kam Tasarım Programı Akış Diyagramı

3.3. Fonksiyon Sentezi Modülü Teorisi

Mekanizmanın giriş uzvunun hareketine karşılık çıkış uzvunun hareketinin belirlenmiş bir fonksiyona göre olduğu mekanizmaların tasarımına fonksiyon sentezi adı verilmektedir. Bu modül Şekil 3.8’de gösterilen tasarım parametreleri ile fonksiyon üreten 4 çubuk mekanizmasını Mutlu (2011)’de açıklanan yöntemi temel alarak tasarlamaktadır.



Şekil 3.8. Fonksiyon Sentezi Tasarım Parametreleri

Bu tasarım parametrelerinde:

x_1 : giriş uzvu yani krank kolu boyu

x_2 : biyel uzvu boyu

x_3 : çıkış uzvu boyu

x_6 : gövde uzvu boyu

ψ : giriş uzvu dönüş açısı

α : çıkış uzvu dönüş açısı

δ : biyel uzvunun yatay eksenle yaptığı açı olarak belirlenmiştir.

$\psi - \alpha$ arasındaki ilişki aşağıdaki vektörel olarak QABM dörtgeni içinde aşağıdaki denklemlerle ifade edilmektedir.

$$x_1 \cos \psi + x_2 \cos \delta = x_6 + x_3 \sin \alpha \quad (3.30)$$

$$x_1 \sin \psi + x_2 \sin \delta = x_3 \sin \alpha \quad (3.31)$$

Denklem (3.30)ve (3.31)'den δ açısı yok edilerek fonksiyon değişkeni olan x , giriş uzvu dönüş açısı ψ (harekete başlama açısı ψ_0), çıkış uzvu dönüş açısı α (harekete başlama açısı α_0)'a bağlı bağımlı değişken olan y arasında (3.32) ve (3.33) bağlantıları kurularak (3.34) elde edilir.

$$Z_1 = \frac{x_3}{x_1}; Z_2 = \frac{x_3}{x_6}; Z_3 = \frac{x_1^2 - x_2^2 + x_3^2 + x_6^2}{2x_1x_6} \quad (3.32)$$

$$\psi = \psi_0 + \psi'; \psi' = R_x (x - x_0); R_x = \frac{\Delta\psi}{\Delta x} \quad (3.33)$$

$$\alpha = \alpha_0 + \alpha'; \alpha' = R_y (y - y_0); R_y = \frac{\Delta\alpha}{\Delta y}$$

$$\begin{aligned}
 a_i Z_3 + Z_1(b_i \cos \alpha_0 - c_i \sin \alpha_0) - Z_2(d_i \sin \alpha_0 - \psi_0) + e_i \cos(\alpha_0 - \psi_0) \\
 + f_i \sin \psi_0 - g_i \cos \psi_0 = 0
 \end{aligned}
 \tag{3.34}$$

$$i = 1, 2, 3, 4, 5$$

(3.34) denkleminde bulunan $\Delta\psi = \psi_n - \psi_0$ giriş uzvu çalışma aralığı, $\Delta\alpha = \alpha_n - \alpha_0$ çıkış uzvu çalışma aralığı olmakta ve $\Delta x = x_n - x_0$ bağımsız değişkenin, $\Delta y = y_n - y_0$ bağımlı değişkenin değişim miktarlarını ifade etmektedir. Bu denkleme ait kesin nokta, alt bölge ve galerkin çözüm yöntemlerinin katsayıları (3.35), (3.36), (3.37) denklem takımlarında verilmiştir.

Kesin Nokta Yöntemi Katsayıları:

$$\begin{aligned}
 a_i &= 1, \quad b_i = \cos \alpha_i', \quad c_i = \sin \alpha_i', \\
 d_i &= \sin(\alpha_i' - \psi_i'), \quad e_i = \cos(\alpha_i' - \psi_i'), \\
 f_i &= \sin \psi_i', \quad g_i = \cos \psi_i'
 \end{aligned}
 \tag{3.35}$$

$$i = 1, 2, 3, 4, 5$$

Alt Bölge Yöntemi Katsayıları:

$$\begin{aligned}
 a_i &= \int_{x_{i-1}}^{x_i} dx, \quad b_i = \int_{x_{i-1}}^{x_i} \cos \alpha' dx, \quad c_i = \int_{x_{i-1}}^{x_i} \sin \alpha' dx, \\
 d_i &= \int_{x_{i-1}}^{x_i} \sin(\alpha' - \psi') dx, \quad e_i = \int_{x_{i-1}}^{x_i} \cos(\alpha' - \psi') dx, \quad f_i = \int_{x_{i-1}}^{x_i} \sin \psi' dx, \\
 g_i &= \int_{x_{i-1}}^{x_i} \cos \psi' dx
 \end{aligned}
 \tag{3.36}$$

$$i = 1, 2, 3, 4, 5$$

Galerkin Yöntemi Katsayıları:

$$\begin{aligned}
 a_i &= \int_{x_0}^{x_n} w_i dx, \quad b_i = \int_{x_0}^{x_n} \cos \alpha' w_i dx, \quad c_i = \int_{x_0}^{x_n} \sin \alpha' w_i dx \\
 d_i &= \int_{x_0}^{x_n} \sin(\alpha' - \psi') w_i dx, \quad e_i = \int_{x_0}^{x_n} \cos(\alpha' - \psi') w_i dx, \\
 f_i &= \int_{x_0}^{x_n} \sin \psi' w_i dx, \quad g_i = \int_{x_0}^{x_n} \cos \psi' w_i dx
 \end{aligned}
 \tag{3.37}$$

$$i = 1, 2, 3, 4, 5$$

Çözüm için (3.34) numaralı denklemde bulunan Z oranları Z2, Z3, Z1 sırasıyla yok edilip $v = \tan(\psi_0/2)$ ve $t = \tan(\alpha_0/2)$ parametrelerine bağlı (3.38) numaralı denklemler elde edilir. Denklemdeki katsayılar uzun olduğundan dolayı EK-1 de bulunmaktadır.

$$\begin{aligned}
 & v^4 \left\{ (a_{pk} + b_{pk} + c_{pk} + d_{pk})t^4 + 4(f_{pk} - g_{pk})t^3 + 2(a_{pk} + b_{pk} - \right. \\
 & \left. 3c_{pk} - 3d_{pk})t^2 + 4(g_{pk} - f_{pk})t + (a_{pk} + b_{pk} + c_{pk} + d_{pk}) \right\} + \\
 & v^3 \{ -4(e_{pk} + f_{pk})t^4 + 16c_{pk}t^3 - 8(e_{pk} - 3f_{pk})t^2 - 16c_{pk}t - 4(e_{pk} + f_{pk}) \} + \\
 & v^2 \left\{ 2(a_{pk} - 3b_{pk} - 3c_{pk} + d_{pk})t^4 - 4(3f_{pk} + g_{pk})t^3 + 4(a_{pk} - 3b_{pk} + \right. \\
 & \left. 9c_{pk} - 3d_{pk})t^2 + 4(3f_{pk} + g_{pk})t + 2(a_{pk} - 3b_{pk} - 3c_{pk} + d_{pk}) \right\} + \\
 & v \{ 4(e_{pk} + f_{pk})t^4 - 16c_{pk}t^3 + 8(e_{pk} - 3f_{pk})t^2 + 16c_{pk}t + 4(e_{pk} + f_{pk}) \} + \\
 & \left\{ (a_{pk} + b_{pk} + c_{pk} + d_{pk})t^4 + 4(f_{pk} - g_{pk})t^3 + 2(a_{pk} + b_{pk} - 3c_{pk} - \right. \\
 & \left. 3d_{pk})t^2 + 4(g_{pk} - f_{pk})t + (a_{pk} + b_{pk} + c_{pk} + d_{pk}) \right\} = 0, \\
 & k = 1, 2
 \end{aligned} \tag{3.38}$$

(3.38) denkleminde v değeri yok edilerek 12. dereceli t 'ye bağlı (3.39) numaralı polinom elde edilmektedir. Denklemdaki w katsayıları uzun olduğundan dolayı EK-2 de bulunmaktadır.

$$\begin{aligned}
 & w_0 - w_1t + (w_2 - 2w_0)t^2 + (2w_1 - w_3)t^3 + (3w_0 - 2w_2 + w_4)t^4 + \\
 & (-3w_1 + 2w_3 - w_5)t^5 + (-4w_0 + 3w_2 - 2w_4 + w_6)t^6 + \\
 & (3w_1 - 2w_3 + w_5)t^7 + (3w_0 - 2w_2 + w_4)t^8 + \\
 & (-2w_1 + w_3)t^9 + (w_2 - 2w_0)t^{10} + w_1t^{11} + w_0t^{12} = 0
 \end{aligned} \tag{3.39}$$

(3.39) denkleminin çözümü sonrasında ψ_0 'ın sanal ve gerçekteğerlerden oluşan 12 adet çözüm kümesi elde edilir.

$$\psi_{0i} = 2 \tan^{-1} t_i, \quad i=1,2,3,4,5 \dots 12 \tag{3.40}$$

Bulunan ψ_{0i} değerlerinin (3.38) denkleminde yerine konarak v^4 terimi yok edildiğinde çıkış uzvuna ait α_0 'ın bulunması için (3.41) 2. dereceden polinomun kökleri kullanılır.

$$(aa_2bb_1 - aa_1bb_2) + (aa_1cc_2 - aa_2cc_1)v + (aa_1bb_2 - aa_2bb_1)v^2 = 0 \tag{3.41}$$

aa_k, bb_k, cc_k (3.42) denklem takımındaki değerleri sonucu (3.41) denkleminin kökleri ile çıkış uzvu açısı α_0 (3.43)'teki gibi olur;

$$\begin{aligned}
 aa_k &= a_{pk} + b_{pk} + (c_{pk} + d_{pk})\cos 2\psi_{0i} + (-f_{pk} + g_{pk})\sin 2\psi_{0i} \\
 bb_k &= -4(c_{pk}\sin 2\psi_{0i} + f_{pk}\cos 2\psi_{0i} + e_{pk})
 \end{aligned} \tag{3.42}$$

$$cc_k = 2(a_{pk} - 3b_{pk} + (-3c_{pk} + d_{pk})\cos 2\psi_{0i} + (3f_{pk} + g_{pk})\sin 2\psi_{0i}$$

$$k=1,2 \text{ ve } i=1,2,\dots,12$$

$$\alpha_{0i\pm} = \arctan v_{i\pm} \quad i = 1, 2, \dots, 12 \tag{3.43}$$

İstenen fonksiyona uygun hareket eden dört çubuk mekanizmasının uzuv boyları olan x_1, x_2, x_3 parametreleri hesaplanırken Z_1, Z_2, Z_3 oranları kullanılır.

$$Z_{2i\pm} = - \frac{\begin{pmatrix} \cos(\alpha_{0i\pm} - \psi_{0i}) \begin{bmatrix} a_k(-c_2f_1 + c_1f_2 - b_2g_1 + b_1g_2) + \\ a_2(c_kf_1 - c_1f_k + b_kg_1 - b_1g_k) + \\ a_1(-c_kf_2 + c_2f_k - b_kg_2 + b_2g_k) \end{bmatrix} + \\ \cos(\alpha_{0i\pm} + \psi_{0i}) \begin{bmatrix} a_k(c_2f_1 - c_1f_2 - b_2g_1 + b_1g_2) + \\ a_2(-c_kf_1 + c_1f_k + b_kg_1 - b_1g_k) + \\ a_1(c_kf_2 - c_2f_k - b_kg_2 + b_2g_k) \end{bmatrix} + \\ \sin(\alpha_{0i\pm} - \psi_{0i}) \begin{bmatrix} a_k(-b_2f_1 + b_1f_2 + c_2g_1 - c_1g_2) + \\ a_2(b_kf_1 - b_1f_k - c_kg_1 + c_1g_k) + \\ a_1(-b_kf_2 + b_2f_k + c_kg_2 - c_2g_k) \end{bmatrix} + \\ \sin(\alpha_{0i\pm} + \psi_{0i}) \begin{bmatrix} a_k(b_2f_1 - b_1f_2 + c_2g_1 - c_1g_2) + \\ a_2(-b_kf_1 + b_1f_k - c_kg_1 + c_1g_k) + \\ a_1(b_kf_2 - b_2f_k + c_kg_2 - c_2g_k) \end{bmatrix} \end{pmatrix}}{\begin{pmatrix} \cos(2\alpha_{0i\pm} - \psi_{0i}) \begin{bmatrix} a_k(-c_2d_1 + c_1d_2 - b_2e_1 + b_1e_2) + \\ a_2(c_kd_1 - c_1d_k + b_ke_1 - b_1e_k) + \\ a_1(-c_kd_2 + c_2d_k - b_ke_2 + b_2e_k) \end{bmatrix} + \\ \cos\psi_{0i} \begin{bmatrix} a_k(c_2d_1 - c_1d_2 - b_2e_1 + b_1e_2) + \\ a_2(-c_kd_1 + c_1d_k + b_ke_1 - b_1e_k) + \\ a_1(c_kd_2 - c_2d_k - b_ke_2 + b_2e_k) \end{bmatrix} + \\ \sin(2\alpha_{0i\pm} - \psi_{0i}) \begin{bmatrix} a_k(-b_2d_1 + b_1d_2 + c_2e_1 - c_1e_2) + \\ a_2(b_kd_1 - b_1d_k - c_ke_1 + c_1e_k) + \\ a_1(-b_kd_2 + b_2d_k + c_ke_2 - c_2e_k) \end{bmatrix} + \\ \sin\psi_{0i} \begin{bmatrix} a_k(b_2d_1 - b_1d_2 + c_2e_1 - c_1e_2) + \\ a_2(-b_kd_1 + b_1d_k - c_ke_1 + c_1e_k) + \\ a_1(b_kd_2 - b_2d_k + c_ke_2 - c_2e_k) \end{bmatrix} \end{pmatrix}} \quad \begin{matrix} i=1,..,12 \\ k=3/4/5 \end{matrix} \quad (3.44)$$

$$Z_{2i\pm} \left\{ \begin{matrix} \sin(\alpha_{0i\pm} - \psi_{0i})(a_kd_1 - c_1d_k) + \\ \cos(\alpha_{0i\pm} - \psi_{0i})(a_ke_1 - a_1e_k) \end{matrix} \right\} + \\ Z_{1i\pm} = - \frac{\begin{matrix} \sin\psi_{0i}(a_1f_k - a_kf_1) + \\ \cos\psi_{0i}(a_kg_1 - a_1g_k) \end{matrix}}{\begin{matrix} \cos\alpha_{0i\pm}(a_1b_k - a_kb_1) + \\ \sin\alpha_{0i\pm}(a_kc_1 - a_1c_k) \end{matrix}} \quad \begin{matrix} i = 1, \dots, 12 \\ k = 2/3/4/5 \end{matrix} \quad (3.45)$$

$$Z_{1i\pm}(b_k\cos\alpha_{0i\pm} - c_k\sin\alpha_{0i\pm}) - \\ Z_{2i\pm} \left[\begin{matrix} d_k\sin(\alpha_{0i\pm} - \psi_{0i}) + \\ e_k\cos(\alpha_{0i\pm} - \psi_{0i}) \end{matrix} \right] + \\ Z_{3i\pm} = - \frac{f_k\sin\psi_{0i} - g_k\cos\psi_{0i}}{a_k} \quad \begin{matrix} i = 1, \dots, 12 \\ k = 1/2/3/4/5 \end{matrix} \quad (3.46)$$

Hesaplanan kol boyları şu şekildedir;

$$\left. \begin{aligned} x_{3i\pm} &= Z_{2i\pm}x_6 \\ x_{1i\pm} &= \frac{x_{3i\pm}}{Z_{1i\pm}} \\ x_{2i\pm} &= \sqrt{x_{1i\pm}^2 + x_{3i\pm}^2 + x_6^2 - 2Z_{3i\pm}x_{1i\pm}x_6} \end{aligned} \right\} i = 1, 2, \dots, 12 \quad (3.47)$$

Hesaplanan kol boylarında bulunan gövde uzvu boyu olan x_6 mafsal noktaları arası mesafe olup diğer boyutların belirlenmesinde ölçek katsayısı olarak kullanıldığından, herhangi bir değer alabilmektedir.

Hesaplanan uzuv boyları, giriş uzvu ve çıkış uzvu açılarının istenen fonksiyona çalışma aralığında uygunluğu (3.33)'te hesaplanan teorik sonuçlar ve (3.48) denkleminde hesaplanan gerçek sonuçlar kullanılarak (3.49) denklemindeki yapısal hata ile tespit edilir.

$$\alpha_{ac\pm} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \quad (3.48)$$

$$e = \alpha_{th} - \alpha_{ac} \quad (3.49)$$

(3.48) denkleminde A, B, C değerleri şunlardır;

$$\begin{aligned} A &= x_1^2 - x_2^2 + x_3^2 - 2x_3x_6 + x_6^2 + (2x_1x_3 - 2x_1x_6)\cos(\psi_0 + \psi') \\ B &= -4x_1x_3\sin(\psi_0 + \psi') \\ C &= x_1^2 - x_2^2 + x_3^2 + 2x_3x_6 + x_6^2 - (2x_1x_3 + 2x_1x_6)\cos(\psi_0 + \psi') \end{aligned} \quad (3.50)$$

3.4. Fonksiyon Sentezi Modülü Program Arayüzü

Bölüm 3.3'te açıklanan teori kullanılarak geliştirilen bilgisayar programı öncelikle yine metin tabanlı konsol uygulaması olarak yazılmış ve çeşitli kod hataları kontrol edilmiştir. Sonuçlar Şekil 3.9'de görüldüğü biçimde elenmiş olarak kullanıcıya incelemesi için sunulmaktadır.

```

1 **** Precision point method ile fonksiyon dort cubuk tasarlar ****
2
3 **** Girilen tasarim degerleri ****
4 dps = 90
5 dal = 60.0000
6 x0 = 0.0
7 xn = 1.0
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

coz.no	x1	x2	x3	ps0	al0	hata
7	0.991534	1.844513	-0.555468	146.410462	-121.530615	6.00E-06
8	0.991534	1.844513	-0.555468	146.410462	58.469385	6.00E-06
19	-0.991534	1.844513	0.555468	-33.589538	-121.530615	1.88E-06
20	-0.991534	1.844513	-0.555468	-33.589538	58.469385	1.88E-06

```

***** 7. cozum hata analiz sonuclari *****
***** 1 ici Dort-Cubuk analiz sonuclari *****

```

x	psi	y	alac	alth	delta	hata
0.0000	146.4105	0.0000	238.4694	238.4694	123.6476	-0.00000000
0.1000	155.4105	0.0100	239.0694	239.0694	118.8165	0.00000000
0.2000	164.4105	0.0400	240.8456	240.8694	114.0454	0.00041477
0.3000	173.4105	0.0900	243.8215	243.8694	109.3868	0.00083571
0.4000	182.4105	0.1600	248.0177	248.0694	104.8710	0.00090239
0.5000	191.4105	0.2500	253.4385	253.4694	100.5040	0.00053856
0.6000	200.4105	0.3600	260.0694	260.0694	96.2672	0.00000000
0.7000	209.4105	0.4900	267.8854	267.8694	92.1185	-0.00027877
0.8000	218.4105	0.6400	276.8694	276.8694	87.9945	0.00000000
0.9000	227.4105	0.8100	287.0370	287.0694	83.8097	0.00056485
1.0000	236.4105	1.0000	298.4694	298.4694	79.4514	-0.00000000

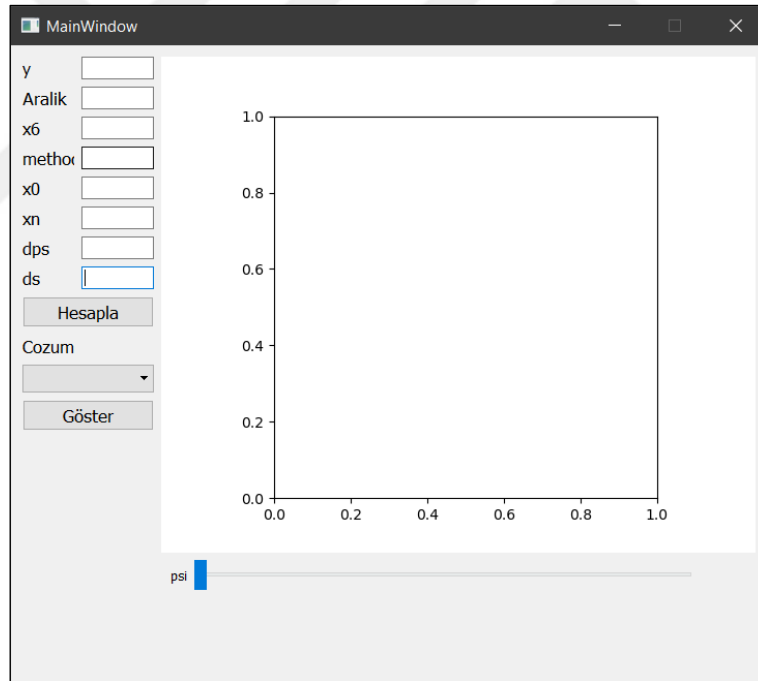
```

***** 2 ici Dort-Cubuk analiz sonuclari *****

```

Şekil 3.9. Fonksiyon sentezi sonuç dosya görüntüsü

Ardından Şekil 3.10'daki kullanıcı arayüzü daha önce bahsedilen Python kütüphaneleri kullanılarak geliştirilmiştir.



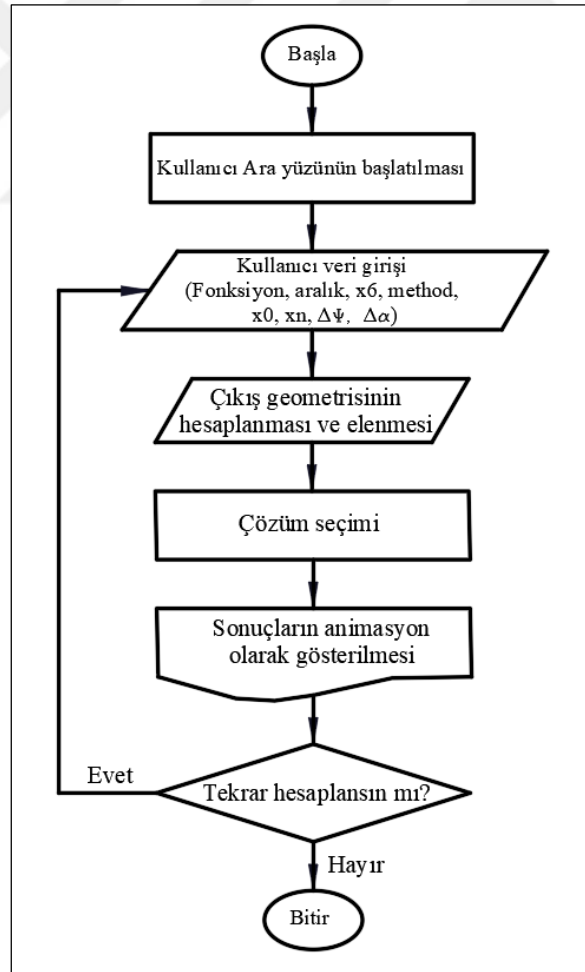
Şekil 3.10. Fonksiyon sentezi programı arayüzü

Kullanıcı arayüzünün sol üst köşesinde bulunan, teoride tarif edilen parametre girişlerinin ardından metin tabanlı çıktıda olduğu biçimde elenmiş sonuçlar kullanıcıya verilmektedir. Açılır menüden kullanıcının seçeceği uygun çözüm arayüze eklenen bir kayar çubuk vasıtasıyla istenen ψ değerinde durdurularak kontrol edilebilmektedir.

Programın hesap adımları Tablo 3.2’de, bu adımlara dayalı oluşturulan akış diyagramı Şekil 3.11’de sunulmuştur.

Tablo 3.2. Fonksiyon Sentezi Programı Hesaplama Algoritması

1.	Kullanıcıdan fonksiyon, x_6 , yöntem seçimi, x_0 , x_n , $\Delta\psi$, $\Delta\alpha$ parametrelerinin alınması
2.	(3.33) denkleminde kullanılarak R_x ve R_y değerlerinin hesaplanması
3.	Yöntem seçimine bağlı olarak (3.35), (3.36) veya (3.37) denklemlerindeki a_i , b_i , c_i , d_i , e_i , f_i , g_i katsayıların hesaplanarak (3.34) denkleminde yerine konması
4.	(3.34) denkleminde Z_2 , Z_3 , Z_1 sırasıyla yok edilerek (3.38) denkleminin elde edilmesi
5.	(3.38) denkleminde v değerleri yok edilerek (3.39) denkleminin elde edilmesi
6.	(3.39) denklemi çözümü ile ψ_{0i} çözüm kümesinin bulunması
7.	ψ_{0i} değerlerinin (3.38) denkleminde kullanılması ve v^4 terimini yok edilmesi, (3.41) denklemi ve (3.42) denklem takımı kullanılarak α_{0i} değerlerinin hesaplanması
8.	(3.44), (3.45) ve (3.46) denklemleri kullanılarak Z_{2i} , Z_{1i} , Z_{3i} oranlarının hesaplanması
9.	(3.47) denklem takımı kullanılarak x_{1i} , x_{2i} , x_{3i} uzuv boylarının hesaplanması
10.	(3.50) denkleminde hesaplanan değerlerin (3.48) denkleminde yerine konması ile gerçek fonksiyon sonuçlarının hesaplanması
11.	(3.49) denklemini kullanılarak gerçek ve teorik sonuçlar arasında arasındaki yapısal hatanın hesaplanması



Şekil 3.11. Fonksiyon Sentezi Programı Akış Diyagramı

Bu denklem takımındaki Φ ifadesi ise (3.54) şeklindedir.

$$\Phi = \tan^{-1} \left[\frac{\tan \beta [\sin(\beta + \eta) + \sin(\alpha - \beta - \eta)]}{\sin(\beta + \eta) - \tan \beta \cos(\alpha - \beta - \eta)} \right] \quad (3.54)$$

Denklemlerde kullanılan oranlar mekanizma kol uzuv boylarının gövde uzvu boyuna oranı şeklinde gösterilmiş olup bu durum, tasarım aşamasında gövde uzununun kullanılmaması fakat gövde uzununun hesaplanan boyutlar için bir ölçek faktörü olarak kullanılabileceği anlamına gelmektedir.

Kol sarkaç mekanizmasında B₁-B₂ arası gidiş zamanı ve B₂-B₁ arası dönüş zamanının birbirine oranı zaman oranı olarak adlandırılıp “ZO” şeklinde ifade edilmekte ve bu zaman oranına göre belirlenen sınır konumları arası η açısı (3.55) denklemindeki şekildedir.

$$\eta = 180 \left[\frac{ZO - 1}{ZO + 1} \right] \quad (3.55)$$

Bu tasarım parametrelerinden diğer uzuv boylarının gövde uzvuna oranları, Φ , η ve β değerlerinden ikisi ve α salınım miktarı tasarım aşamasında belirlendiğinde, bilinmeyen diğer dört parametre bulunmaktadır. Tasarım ihtiyacına göre hazırlanan çeşitli durumlar ve çözüm adımları şunlardır;

3.5.1. Durum 1: α , ZO ve β kullanılarak diğer parametrelerin bulunması

- a) ZO değeri (3.55) denkleminde kullanılarak η değeri yani sınır konumları arasındaki açı hesaplanır
- b) α ve η değerleri (3.54) denkleminde yerine konarak Φ açısı hesaplanır
- c) α , η , Φ değerleri (3.51), (3.52), (3.53) denkleminde yerine konarak kol uzuvlarının gövdeye oranları bulunur.

3.5.2. Durum 2: α , ZO ve Φ kullanılarak diğer parametrelerin bulunması

- a) ZO değeri (3.55) denkleminde kullanılarak η değeri yani sınır konumları arasındaki açı hesaplanır
- b) α , η , Φ değerleri (3.54) denkleminde yerine konur ve $f(\beta)=0$ şeklindeki bir denkleme $t_\beta = \tan\left(\frac{\beta}{2}\right)$ dönüşümü yapılarak katsayılar (3.57)’de verilen (3.56) numaralı polinom elde edilir.

$$w_0 + w_1 t_\beta + w_2 t_\beta^2 - w_1 t_\beta^3 + w_0 t_\beta^4 = 0 \quad (3.56)$$

$$w_0 = \cos(\eta - \Phi) - \cos(\eta + \Phi)$$

$$w_1 = -4[\sin(\eta - \Phi) + \sin(\alpha - \eta + \Phi)] \quad (3.57)$$

$$w_2 = 8\cos(\alpha - \eta + \Phi) - 6\cos(\eta - \Phi) - 2\cos(\eta + \Phi)$$

- c) Polinomun kökleri bulunarak $i=1,2,3,4$ olmak üzere $\beta_i=2\text{Arctan}(t_{\beta i})$ çözüm kümesi elde edilir.
- d) Hesaplanan β_i ve α , η , Φ değerleri (3.51), (3.52), (3.53) denkleminde yerine konarak kol uzunlarının gövdeye oranları bulunur.

3.5.3. Durum 3: α , ZO ve u/p kullanılarak diğer parametrelerin bulunması

- a) ZO değeri (3.55) denkleminde kullanılarak η değeri yani sınır konumları arasındaki açı hesaplanır
- b) u/p, η ve α değerleri önce (3.51) sonra (3.54) denklemlerine yerine konularak Φ açısının yok edildiği β 'ya bağlı (3.58) denklemi elde edilir.

$$2 - \left(\frac{u}{p}\right)^2 - \cos\alpha - \cos 2\beta + \cos(\alpha + 2\beta) - \cos(\alpha - 2\eta) + (u/p)^2 \cos(2\alpha - 2\eta) + \cos(\alpha - 2\beta - 2\eta) - \cos(2\beta + 2\eta) = 0 \quad (3.58)$$

- c) Bu denklemde $t_\beta = \tan\left(\frac{\beta}{2}\right)$ dönüşümü yapılarak katsayıları (3.60)'da verilen (3.59) numaralı polinom elde edilir.

$$w_0 + w_1 t_\beta + w_2 t_\beta^2 - w_1 t_\beta^3 + w_0 t_\beta^4 = 0 \quad (3.59)$$

$$w_0 = 1 - (u + (u \cos(2\alpha - 2\eta) - \cos 2\eta$$

$$w_1 = 16 \sin 0.5\alpha \sin(0.5\alpha - \eta) \sin \eta \quad (3.60)$$

$$w_2 = 2[5 - (u - 4\cos\alpha - 4\cos(\alpha - 2\eta) - (u \cos(2\alpha - 2\eta) + 3\cos 2\eta]$$

- d) Polinomun kökleri bulunarak $i=1,2,3,4$ olmak üzere $\beta_i=2\text{Arctan}(t_{\beta i})$ çözüm kümesi elde edilir.
- e) Verilen α , hesaplanan η ve β_i değerleri (3.54) denkleminde yerine konarak $i=1,2,3,4$ olmak üzere Φ_i değerleri bulunur.
- f) Verilen ve hesaplanan açı değerleri (3.52) ve (3.53) denklemlerinde yerine konarak kol uzunlarının gövdeye oranları bulunur.

3.5.4. Durum 4: α , β ve u/p kullanılarak diğer parametrelerin bulunması

- a) β ve u/p değerleri (3.51) denkleminde yerine konarak Φ açısı hesaplanır.
- b) Hesaplanan Φ açısı ve verilen β ve α açıları (3.5) denkleminde yerine konarak η açısı bulunur.
- c) η değeri ile (3.55) denkleminde ZO bulunur.
- d) Verilen ve hesaplanan açı değerleri (3.52) ve (3.53) denklemlerinde yerine konarak kol uzunlarının gövdeye oranları bulunur.

3.5.5. Durum 5: α , β ve Φ kullanılarak diğer parametrelerin bulunması

- a) β ve Φ değerleri (3.51) denkleminde yerlerine konarak u/p oranı hesaplanır.
- b) β ve Φ değerleri (3.54) denkleminde yerlerine konarak η hesaplanır.
- c) η değeri ile (3.55) denkleminde ZO bulunur.
- d) Verilen ve hesaplanan açı değerleri (3.52) ve (3.53) denklemlerinde yerine konarak kol uzunlarının gövdeye oranları bulunur.

3.5.6. Durum 6: α , Φ ve u/p kullanılarak diğer parametrelerin bulunması

- a) Φ ve u/p değerleri (3.51) denkleminde yerine konarak β açısı hesaplanır.
- b) Hesaplanan β açısı ve verilen Φ ve α açıları (3.5) denkleminde yerine konarak η açısı bulunur.
- c) η değeri ile (3.55) denkleminde ZO bulunur.
- d) Verilen ve hesaplanan açı değerleri (3.52) ve (3.53) denklemlerinde yerine konarak kol uzunlarının gövdeye oranları bulunur.

3.5.7. Durum 7: α , ZO ve u/r kullanılarak diğer parametrelerin bulunması

- a) ZO değeri (3.55) denkleminde kullanılarak η değeri yani sınır konumları arasındaki açı hesaplanır
- b) (3.51) ve (3.52) denklemlerinin birbirine oranı ile u/r oranı hesaplanır. Bu oran ve (3.54) denkleminde Φ değerinin yarım açı tanjant formülü ile yok edilmesi sonrasında β 'ya bağlı (3.61) denklemi elde edilir.

$$2 - [(u/r)^2 - 1]\cos\alpha - \cos 2\beta + \cos(\alpha + 2\beta) - \cos(\alpha - 2\eta) - \cos(2\beta + 2\eta) + \cos(\alpha - 2\beta - 2\eta) - (u/r)^2 \sin^2 0.5\alpha [\cos(\alpha + 2\beta) + 2\cos(\alpha - \eta) + 2\cos(2\beta + \eta) + \cos(\alpha - 2\beta - 2\eta)] = 0 \quad (3.61)$$

- c) Denklemin kapalı çözümü için $t_\beta = \tan(\frac{\beta}{2})$ dönüşümü yapılarak katsayıları (3.63)'te verilen (3.62) numaralı polinom elde edilir.

$$w_0 + w_1 t_\beta + w_2 t_\beta^2 - w_1 t_\beta^3 + w_0 t_\beta^4 = 0 \quad (3.62)$$

$$w_0 = -8\cos^2 0.5\eta [-(u/r)\sin(\alpha - 0.5\eta) - 2\sin 0.5\eta - (u/r)\sin 0.5\eta] [-(u/r)\sin(\alpha - 0.5\eta) + 2\sin 0.5\eta - (u/r)\sin 0.5\eta]$$

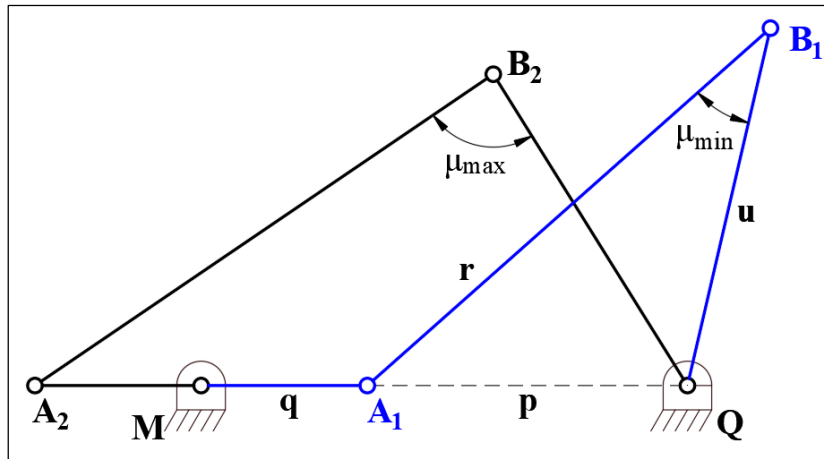
$$w_1 = 32\cos 0.5\eta \sin 0.5\alpha \sin 0.5\eta [2(u\sin 0.5\alpha + 4\sin(0.5\alpha - \eta) - (u\sin(0.5\alpha - \eta) + (u\sin(3\alpha/2 - \eta)) \quad (3.63)$$

$$w_2 = -2[20 + 7(u + 16\cos\alpha - 10(u\cos\alpha + 3(u\cos 2\alpha + 16\cos(\alpha - 2\eta) - 6(u\cos(\alpha - 2\eta) + 3(u\cos(2\alpha - 2\eta) + 10(u\cos(\alpha - \eta) - 2(u\cos(2\alpha - \eta) - 14(u\cos\eta - 12\cos 2\eta + 3(u\cos 2\eta + 6(u\cos(\alpha + \eta))]$$

- d) Polinomun t_β kökleri bulunarak $\beta_i = 2\arctan(t_{\beta_i})$, ($i=1,2,3,4$) çözüm kümesi elde edilir.
- e) α ve hesaplanan β_i ve η açı değerleri (3.54) denkleminde yerine konarak $i=1,2,3,4$ olmak üzere Φ_i değerleri bulunur.
- f) Verilen ve hesaplanan açı değerleri (3.52) ve (3.53) denklemlerinde yerine konarak kol uzuvlarının gövdeye oranları bulunur.

3.5.8. Durum 8: Bağlama açısına göre diğer parametrelerin bulunması

Kol sarkaç mekanizmalarında giriş uzvunun yatay eksenle çakışık olduğu ($\psi=0$ ve $\psi=180$) durumlarda Şekil 3.13'de gösterilen halde minimum ve maksimum değerlere ulaşan bağlama açısı, tasarım sonucu ortaya çıkan mekanizmada giriş uzvundan çıkış uzvuna aktarılabilecek kuvvetin ne kadar iş yapabileceğinin bir ölçütüdür. Optimum olarak 90° derece olması istenen fakat her an değişen bağlama açısının mümkün olduğunca 90° dereceden az sapması istenir.



Şekil 3.13. Kol sarkaç mekanizmasında minimum ve maksimum bağlama açısı

Şekil 3.13’de gösterilen minimum bağlama açısı ($\mu_{\min}$) ve maksimum bağlama açısı ($\mu_{\max}$) değerleri için kosinüs teoremi kullanılarak (3.64) ve (3.65) denklemleri elde edilir.

$$\cos \mu_{\min} = \frac{u^2 + r^2 - p^2 - q^2}{2ur} + \frac{pq}{ur} \quad (3.64)$$

$$\cos \mu_{\max} = \frac{u^2 + r^2 - p^2 - q^2}{2ur} - \frac{pq}{ur} \quad (3.65)$$

Bu denklemlerde (3.51), (3.52), (3.53) denklemleri yerine konarak sadeleştirildiğinde $\cos \mu_{\min} = C\mu_{\min}$ ve $\cos \mu_{\max} = C\mu_{\max}$ olmak üzere 3 bilinmeyenli 3 adet denklem elde edilir.

$$f_1(\beta, \eta, \varphi) = -\cos(2\beta + \eta - \varphi) - \cos(\alpha - \eta + \varphi) + \cos(\alpha - 2\beta - \eta + \varphi) + \cos(\eta + \varphi) = 0 \quad (3.66)$$

$$f_2(\beta, \eta, \varphi) = -\cos(\alpha + 3\beta/2 - \varphi) + \cos(\alpha - \beta/2 + \varphi) - \cos(\alpha + \beta/2 + \varphi) - \cos(\alpha - 5\beta/2 - \eta + \varphi) - \cos(\alpha + \beta/2 - \eta + \varphi) + \cos(\alpha - 3\beta/2 + 3\varphi) + 2\sin(3\beta/2 - 2\varphi)\sin(\alpha - \beta - \eta + \varphi) + 2\cos(3\beta/2) C\mu_{\min}[\cos(\alpha + \beta) + \cos(\alpha - \beta - \eta)] - 2\cos(\eta/2)\cos(\alpha - \beta - \eta/2 + 2\varphi) = 0 \quad (3.67)$$

$$f_3(\beta, \eta, \varphi) = 4\sin(\beta/2 - \varphi)\sin(\beta - \varphi)\sin(\alpha + \varphi) + 2\sin(\alpha - \beta - \eta + \varphi)[- \cos(3\beta/2) + 2\cos(3\beta/2 - 2\varphi)] + 4\sin(\beta/2) C\mu_{\max}[\sin(\beta - \varphi)\sin(\alpha + \varphi) - \sin\varphi \sin(\alpha - \beta - \eta + \varphi)] = 0 \quad (3.68)$$

Bu denklemlerde φ , β ve η değerlerinin sırasıyla iki tanesini yok etmek için yarım açı tanjant formülleri kullanmak amacıyla denklemler yok edilecek parametreye göre yeniden yazılırsa (3.72) olmak üzere (3.69), (3.70), (3.71)’deki şekilde olur.

$$f_1(\beta, \eta, \varphi) = aw_0 + aw_1 t_\eta - aw_0 t_\eta^2 = 0 \quad (3.69)$$

$$f_2(\beta, \eta, \varphi) = bw_0 + bw_1 t_\eta + bw_2 t_\eta^2 = 0 \quad (3.70)$$

$$f_3(\beta, \eta, \varphi) = cw_0 + cw_1 t_\eta + cw_2 t_\eta^2 = 0 \quad (3.71)$$

$$t_\eta = \tan \frac{\eta}{2} \quad (3.72)$$

Bu denklemlerdeki katsayılar aşağıda verilmiştir.

$$aw_0 = \cos(\alpha/2 - \beta + \varphi)\sin(\alpha/2)\sin\beta \quad (3.73)$$

$$aw_1 = -\cos(\alpha - \beta + \varphi)\sin\beta + \cos\beta\sin(\beta - \varphi) \quad (3.74)$$

$$bw_0 = 0.5\cos(\beta/2)[- \cos(\alpha + \beta - \varphi) - \cos(\alpha - 2\beta + \varphi) + \cos(\alpha - \beta + \varphi) + \cos(\alpha - 2\beta + 3\varphi) + C\mu_{\min}(\cos(\alpha - \beta) + \cos(\alpha + \beta) - 2\cos(\alpha - \beta + 2\varphi))] \quad (3.75)$$

$$w_1 = 2\cos(\alpha - \beta + \varphi)\sin\varphi[\cos(3\beta/2 - \varphi) - \cos(\beta/2)C\mu_{\min}] \quad (3.76)$$

$$bw_2 = 0.5\sin(\beta/2)[\sin(\alpha + \beta - \varphi) + 2\sin(\alpha + \varphi) + \sin(\alpha - 2\beta + \varphi) + \sin(\alpha - \beta + \varphi) - \sin(\alpha - 2\beta + 3\varphi) + 4\cos^2(\beta/2)\sin\alpha C\mu_{\min}] \quad (3.77)$$

$$cw_0 = 0.5\sin(\beta/2)[\cos(\alpha + \beta - \varphi) - \cos(\alpha - 2\beta + \varphi) - \cos(\alpha - \beta + \varphi) + \cos(\alpha - 2\beta + 3\varphi) + C\mu_{\max}(\cos(\alpha - \beta) + \cos(\alpha + \beta) - 2\cos(\alpha - \beta + 2\varphi))] \quad (3.78)$$

$$cw_1 = 2\cos(\alpha - \beta + \varphi)\sin\varphi[\sin(3\beta/2 - \varphi) - \cos(\beta/2)C\mu_{\max}] \quad (3.79)$$

$$cw_2 = 0.5\cos(\beta/2)[\sin(\alpha + \beta - \varphi) - 2\sin(\alpha + \varphi) - \sin(\alpha - 2\beta + \varphi) + \sin(\alpha - \beta + \varphi) + \sin(\alpha - 2\beta + 3\varphi) + 2(\cos\beta - 1)\sin\alpha C\mu_{\max}] \quad (3.80)$$

Önce (3.69) ve (3.70), sonra (3.69) ve (3.71) denklemlerinde t_η yok edilerek β ve φ 'ye bağlı iki denklem elde edilir ve β değerine göre yine yarım açı tanjant formülleri kullanarak düzenlendiğinde katsayıları (3.84)-(3.90) olan (3.81) ve (3.82) polinomları olur.

$$g_1(\beta, \varphi) = ap_0 + ap_1t_\beta + ap_2t_\beta^2 = 0 \quad (3.81)$$

$$g_2(\beta, \varphi) = bp_0 + bp_1t_\beta + bp_2t_\beta^2 = 0 \quad (3.82)$$

$$t_\beta = \tan \frac{\beta}{2} \quad (3.83)$$

$$ap_0 = 2[\cos\varphi - C\mu_{\min}][\cos\alpha + \cos(\alpha + 2\varphi) - 2 + 2C\mu_{\min}(\cos\varphi - \cos(\alpha + \varphi))] \quad (3.84)$$

$$ap_1 = 4\sin\varphi[\cos\alpha + \cos(\alpha + 2\varphi) - 2 + 2C\mu_{\min}(\cos\varphi - \cos(\alpha + \varphi))] \quad (3.85)$$

$$ap_2 = 4\sin^2\varphi[\cos(\alpha + \varphi) + C\mu_{\min}] \quad (3.86)$$

$$bp_0 = 4\sin^2\varphi[C\mu_{\max} - \cos(\alpha + \varphi)] \quad (3.87)$$

$$bp_1 = 4\sin\varphi[\cos\alpha + \cos(\alpha + 2\varphi) - 2 + 2C\mu_{\max}(-\cos\varphi + \cos(\alpha + \varphi))] \quad (3.88)$$

$$bp_2 = -2[\cos\varphi + C\mu_{\max}][\cos\alpha + \cos(\alpha + 2\varphi) - 2 + 2C\mu_{\max}(-\cos\varphi - \cos(\alpha + \varphi))] \quad (3.89)$$

(3.81) ve (3.82) polinomlarında t_β yok edilerek φ 'ye bağlı denklemde yarım açı tanjant formülleri kullanıldığında (3.90) ifadesine ulaşılır.

$$z(\varphi) = z_1(\varphi) z_2(\varphi) = 0 \quad (3.90)$$

(3.5) denkleminde bulunan $z_1(\varphi)$ ve $z_2(\varphi)$ fonksiyonları(3.91) ve (3.92) olup polinom katsayıları da altında verilmiştir.

$$z_1(\varphi) = p_0 + p_1 t_\varphi + p_2 t_\varphi^2 + p_3 t_\varphi^3 + p_4 t_\varphi^4 = 0 \quad (3.91)$$

$$z_2(\varphi) = w_0 + w_1 t_\varphi + w_2 t_\varphi^2 + w_3 t_\varphi^3 + w_4 t_\varphi^4 + w_5 t_\varphi^5 + w_6 t_\varphi^6 + w_7 t_\varphi^7 + w_8 t_\varphi^8 = 0 \quad (3.92)$$

$$t_\varphi = \tan \frac{\varphi}{2} \quad (3.93)$$

$$p_0 = (1 - \cos \alpha)(1 + C\mu_{\max})(C\mu_{\min} - 1)$$

$$p_1 = 2\sin \alpha(1 + C\mu_{\max})(C\mu_{\min} - 1)$$

$$p_2 = 4\cos^2 \frac{\alpha}{2}(C\mu_{\min} - C\mu_{\max}) \quad (3.94)$$

$$p_3 = 2\sin \alpha(C\mu_{\max} - 1)(C\mu_{\min} + 1)$$

$$p_4 = (\cos \alpha - 1)(C\mu_{\max} - 1)(C\mu_{\min} + 1)$$

$$w_0 = 8\sin^4 \frac{\alpha}{2}(1 + C\mu_{\max})^2(C\mu_{\min} - 1)^2$$

$$w_1 = 32\cos \frac{\alpha}{2}\sin^3 \frac{\alpha}{2}(1 + C\mu_{\max})^2(C\mu_{\min} - 1)^2$$

$$w_2 = 16\sin^2 \frac{\alpha}{2}(C\mu_{\min} - 1)(1 + C\mu_{\max})$$

$$(-5\cos \alpha + 2C\mu_{\min} \cos \alpha + C\mu_{\max}(-2\cos \alpha + (1 + \cos \alpha)C\mu_{\min}))$$

$$w_3 = -8\sin \alpha(C\mu_{\min} - 1)(1 + C\mu_{\max})(9 - \cos \alpha - 3C\mu_{\min}(\cos \alpha - 1) -$$

$$2\sin^2 \frac{\alpha}{2}C\mu_{\max}(3 + C\mu_{\min}))$$

$$w_4 = 2(33 + 36\cos \alpha - 5\cos 2\alpha +$$

$$4\sin^2 \frac{\alpha}{2}(4(3 + \cos \alpha)(C\mu_{\max}C\mu_{\min} -$$

$$(1 + 7\cos \alpha)C\mu_{\min}^2 +$$

$$C\mu_{\max}^2(-1 - 7\cos \alpha + (3 + 5\cos \alpha)C\mu_{\min}^2)))$$

$$w_5 = 8\sin \alpha(C\mu_{\min} + 1)(C\mu_{\max} - 1)(9 - \cos \alpha -$$

$$2\sin^2 \frac{\alpha}{2}C\mu_{\max}(C\mu_{\min} - 3) +$$

$$3(\cos \alpha - 1)C\mu_{\min})$$

$$w_6 = 16\sin^2 \frac{\alpha}{2}(C\mu_{\min} + 1)(C\mu_{\max} - 1)$$

$$(-5 - \cos \alpha - 2C\mu_{\min} \cos \alpha +$$

$$C\mu_{\max}(2\cos \alpha + (1 + \cos \alpha)C\mu_{\min}))$$

$$w_7 = -32\cos \frac{\alpha}{2}\sin^3 \frac{\alpha}{2}(C\mu_{\max} - 1)^2(C\mu_{\min} + 1)^2$$

$$w_8 = 8\sin^4 \frac{\alpha}{2}(C\mu_{\max} - 1)^2(C\mu_{\min} + 1)^2$$

(3.95)

Açıklanan bu hesaplamalara göre belirli bir α salınım miktarı, minimum ve maksimum bağlama açıları kullanılarak diğer mekanizma parametrelerinin bulunmasında izlenecek adımlar şunlardır;

- Verilen tasarım parametreleri (3.94) ve (3.95)'te yerine yazılarak (p_i , $i=0,1,2,3,4$) ve (w_i , $i=0,1,...,8$) katsayıları hesaplanır
- Bulunan katsayılar (3.91), (3.92) denklemlerinde kullanılarak 12 adet (t_{ϕ_i} , $i=1,2,...,12$) kökleri bulunur.
- Bulunan kökler (3.93) eşitliğinde kullanılarak 12 adet çıkış uzvu sınır açısı (ϕ_i , $i=1,2,...,12$) çözümüne ulaşılır.
- α salınım miktarı, minimum ve maksimum bağlama açıları ve hesaplanan (ϕ_i , $i=1,2,...,12$) değerleri önce (3.84), (3.85), (3.86) denklemlerinde yerine konarak ap_0, ap_1, ap_2 katsayıları, bu katsayılar da (3.81) veya (3.82) denkleminde yerine konarak ve (3.83) eşitliği kullanılarak toplam 24 adet $\beta_{\pm}$ çözüm kümesi elde edilir.
- Hesaplanan her (ϕ_i , $i=1,2,...,12$) değerleri ve bunların her birine karşılık ikişer adet olmak üzere 24 adet hesaplanan (β_+ , $\phi_i, i=1,2,...,12$) ve (β_- , $\phi_i, i=1,2,...,12$) değerleri (3.69), (3.70) veya (3.71) denklemlerinde yerine konarak 48 adet mekanizmanın sınır konumları arasındaki açı $\eta_{\pm}$ hesaplanır.
- Elde edilen 48 adet çözüm kümesinin gerçek değere sahip olanları (3.51), (3.52) ve (3.53) denklemlerinde yerine konarak uzuvların gövdeye olanları bulunur.
- Elde edilen uzunluk oranları (3.64), (3.65) denklemlerinde yerine konarak bağlama açıları hesaplanır ve tasarım başlangıcında verilen değerlerle karşılaştırılarak aynı olmayanlar elenir.

3.6. Kol Sarkaç Mekanizması Modülü Program Arayüzü

Bölüm 3.5'te açıklanan durumlar ve adımlar kullanılarak geliştirilen bilgisayar programı öncelikle yine metin tabanlı konsol uygulaması olarak yazılmış ve çeşitli kod hataları kontrol edilmiştir. Sonuçlar Şekil 3.14'te görüldüğü biçimde elenmiş olarak kullanıcıya incelemesi için sunulmaktadır.

```

1 **** Precision point method ile fonksiyon dört cubuk tasarlar ****
2
3 **** Girilen tasarım değerleri ****
4 dps = 90
5 dal = 60.0000
6 x0 = 0.0
7 xn = 1.0
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

coz.no	x1	x2	x3	ps0	al0	hata
7	0.991534	1.844513	0.555468	146.410462	-121.530615	6.00E-06
8	0.991534	1.844513	-0.555468	146.410462	58.469385	6.00E-06
19	-0.991534	1.844513	0.555468	-33.589538	-121.530615	1.88E-06
20	-0.991534	1.844513	-0.555468	-33.589538	58.469385	1.88E-06

```

17 ***** 7. cozum hata analiz sonuclari *****
18
19
20 ***** 1 ici Dort-Cubuk analiz sonuclari *****
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

x	psi	y	alac	alth	delta	hata
0.0000	146.4105	0.0000	238.4694	238.4694	123.6476	-0.00000000
0.1000	155.4105	0.0100	239.0694	239.0694	118.8165	0.00000000
0.2000	164.4105	0.0400	240.8456	240.8694	114.0454	0.00041477
0.3000	173.4105	0.0900	243.8215	243.8694	109.3868	0.00083571
0.4000	182.4105	0.1600	248.0177	248.9694	104.8710	0.00090239
0.5000	191.4105	0.2500	253.4385	253.4694	100.5040	0.00053856
0.6000	200.4105	0.3600	260.0694	260.0694	96.2672	0.00000000
0.7000	209.4105	0.4900	267.8854	267.8694	92.1185	-0.00027877
0.8000	218.4105	0.6400	276.8694	276.8694	87.9945	0.00000000
0.9000	227.4105	0.8100	287.0370	287.0694	83.8097	0.00056485
1.0000	236.4105	1.0000	298.4694	298.4694	79.4514	-0.00000000

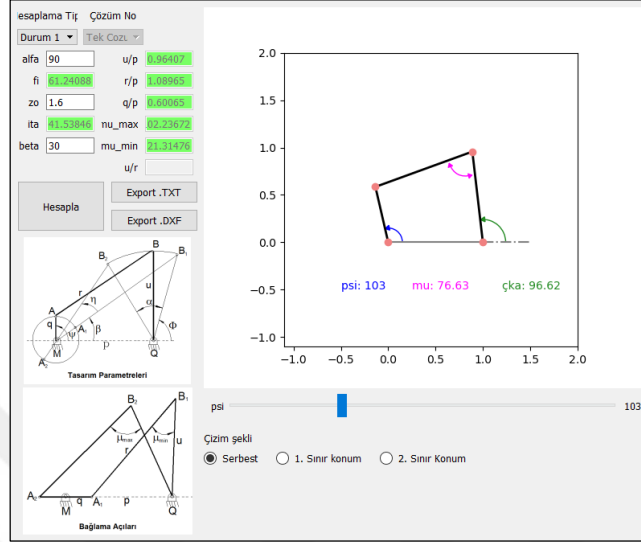
```

35 ***** 2 ici Dort-Cubuk analiz sonuclari *****
36
37

```

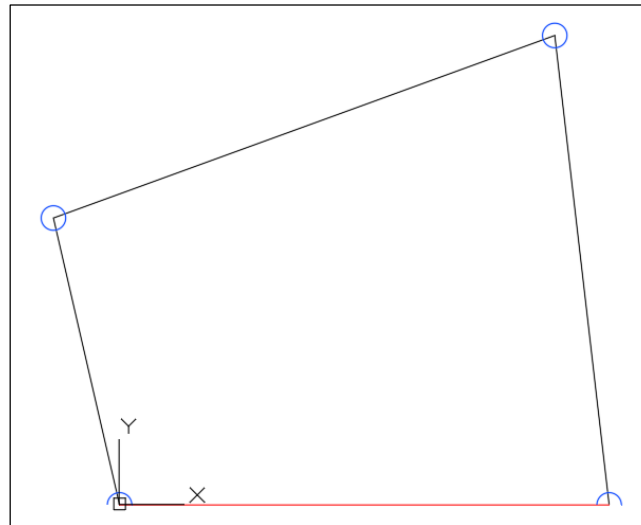
Şekil 3.14. Kol Sarkaç Mekanizması sentezi sonuç dosya görüntüsü

Ardından Şekil 3.15'teki kullanıcı arayüzü daha önce bahsedilen Python kütüphaneleri kullanılarak geliştirilmiştir.



Şekil 3.15. Kol Sarkaç Mekanizması Modülü Programı Arayüzü

Kullanıcı arayüzünün sol üst köşesinde, ihtiyaç duyulan duruma yönelik hesaplama tipi seçilmekte ve ön izleme şeklinde programa gömülü tasarım parametreleri ile parametre girişlerinin ardından metin tabanlı çıktıda olduğu biçimde elenmiş sonuçlar kullanıcıya verilmektedir. Açılır menüden kullanıcının seçeceği uygun çözüm arayüze eklenen bir kayar çubuk vasıtasıyla istenen ψ değerinde durdurularak veya sınır konumlara otomatik olarak giderek kontrol edilebilmektedir. Ayrıca sonuçlar “dxf” dosya formatında 2D olarak Şekil 3.16’te görüldüğü gibi kullanıcıya verilerek CAD programlarında kullanma imkanı sağlanmaktadır.



Şekil 3.16. Dxf Dosya Türünde Kaydedilen Kol-Sarkaç Tasarım Sonuçları

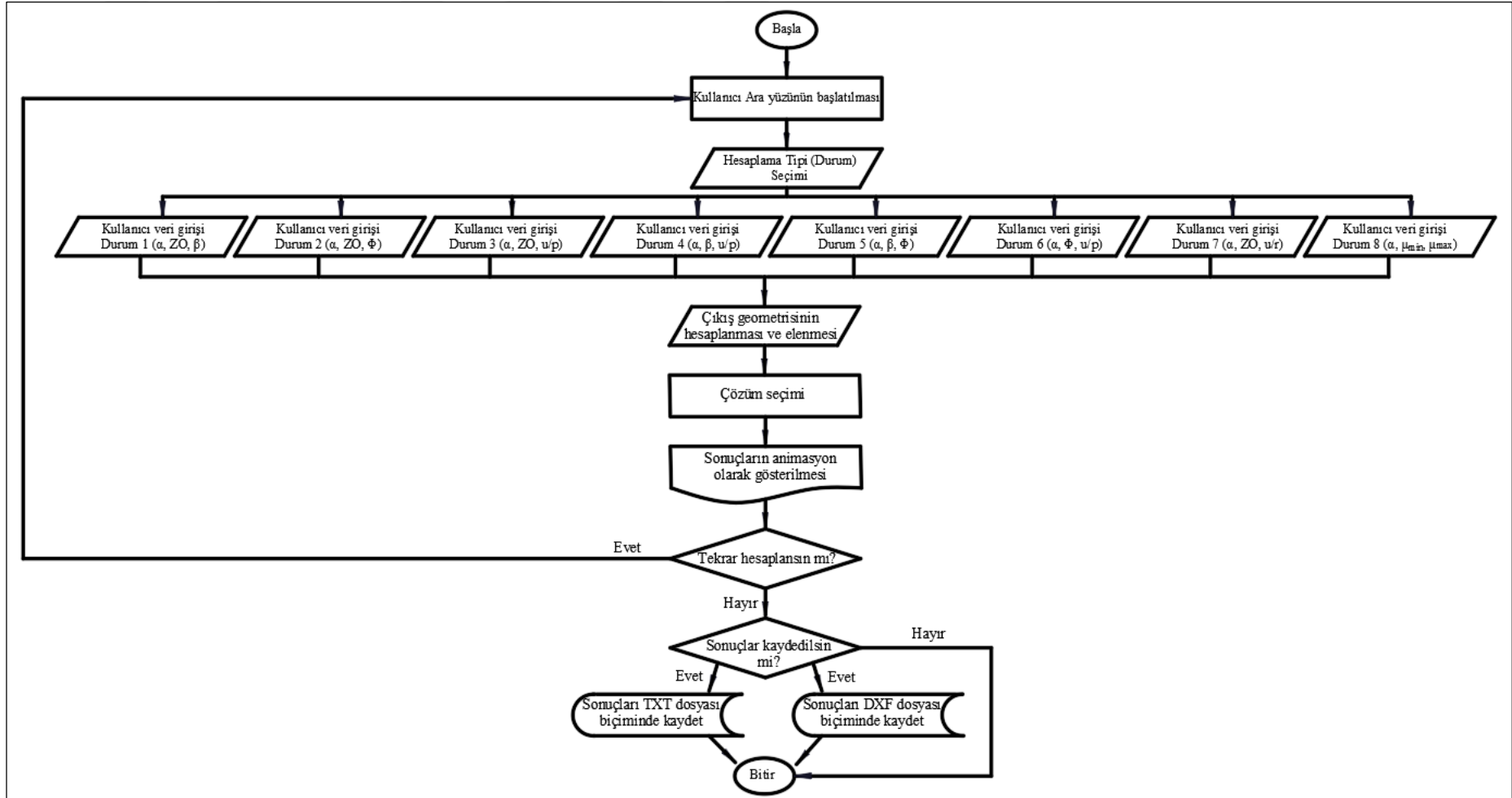
Programın hesap adımları Tablo 3.3’de, bu adımlara dayalı oluşturulan akış diyagramı Şekil 3.17’de sunulmuştur.

Tablo 3.3. Kol Sarkaç Mekanizması Programı Hesaplama Algoritması

1.	Durum 1 (α, ZO, β) ile hesaplama
1.1.	ZO değeri (3.55) denklemde kullanılarak η değeri hesaplanması
1.2.	A, η değerleri (3.54) denklemde yerine konarak Φ açısı hesaplanması
1.3.	α , η , Φ değerleri (3.51), (3.52), (3.53) denklemde yerine konarak u/p, r/p, q/p oranları hesaplanması
2.	Durum 2 (α, ZO, Φ) ile hesaplama
2.1.	ZO değeri (3.55) denklemde kullanılarak η değeri hesaplanması
2.2.	α , η , Φ değerleri (3.54) denklemde yerine konarak katsayılar (3.57)’de verilen (3.56) hesaplanması
2.3.	Polinomun (3.56) kökleri bulunarak $t_{\beta i}$ çözüm kümesi hesaplanması
2.4.	β_i , α , η , Φ değerleri (3.51), (3.52), (3.53) denklemde yerine konarak u/p, r/p, q/p oranları hesaplanması
3.	Durum 3 (α, ZO, u/p) ile hesaplama
3.1.	ZO değeri (3.55) denklemde kullanılarak η değeri hesaplanması
3.2.	u/p, η , α değerleri (3.51) sonra (3.54) denklemlerine yerine konarak (3.58) denklemi hesaplanması
3.3.	(3.58) denklemde $t_{\beta} = \tan\left(\frac{\beta}{2}\right)$ dönüşümü yapılarak katsayıları (3.59) numaralı polinomun hesaplanması
3.4.	(3.59) Polinomun kökleri bulunarak β_i çözüm kümesi hesaplanması
3.5.	α , η , β_i değerleri (3.54) denklemde yerine konarak Φ_i değerleri hesaplanması
3.6.	α , η , β_i , Φ_i değerleri (3.52) ve (3.53) denklemde yerine konarak r/p, q/p oranları hesaplanması
4.	Durum 4 (α, β, u/p) ile hesaplama
4.1.	β , u/p değerleri (3.51) denklemde yerine konarak Φ açısı hesaplanması
4.2.	Φ , β , α açıları (3.54) denklemde yerine konarak η açısı hesaplanması
4.3.	η değeri (3.55) denklemde yerine konarak ZO hesaplanması
4.4.	α , η , β , Φ değerleri (3.52) ve (3.53) denklemde yerine konarak r/p, q/p oranları hesaplanması
5.	Durum 5 (α, β, Φ) ile hesaplama
5.1.	β , Φ değerleri (3.51) denklemde yerlerine konarak u/p oranı hesaplanması
5.2.	β , Φ değerleri (3.54) denklemde yerlerine konarak η hesaplanması
5.3.	η değeri ile (3.55) denklemden ZO hesaplanması
5.4.	α , η , Φ , β değerleri (3.52) ve (3.53) denklemde yerine konarak r/p, q/p oranları hesaplanması
6.	Durum 6 (α, Φ, u/p) ile hesaplama
6.1.	Φ ve u/p değerleri (3.51) denklemde yerine konarak β hesaplanması
6.2.	β , Φ , α açıları (3.54) denklemde yerine konarak η hesaplanması.
6.3.	η değeri ile (3.55) denklemden ZO bulunur.
6.4.	α , η , Φ , β değerleri (3.52) ve (3.53) denklemde yerine konarak r/p, q/p oranları hesaplanması
7.	Durum 7 (α, ZO, u/r) ile hesaplama
7.1.	ZO değeri (3.55) denklemde yerine konarak η hesaplanması
7.2.	(3.51) ve (3.52) denklemlerinin birbirine oranı ile u/r oranı hesaplanması. Bu oran ve (3.54) denklemde Φ değerinin yarım açı tanjant formülü ile yok edilmesi ile (3.61) denkleminin hesaplanması

Tablo 3.4. (Devamı)

7.3.	(3.61) denklemin kapalı çözümü için $t_\beta = \tan(\frac{\beta}{2})$ dönüşümü (3.5) numaralı polinomun hesaplanması.
7.4.	(3.62) polinomun t_β kökleri bulunarak $\beta_i = 2\arctan(t_{\beta_i})$, ($i=1,2,3,4$) çözüm kümesi hesaplanması
7.5.	α , β_i , η açılı değerleri (3.54) denkleminde yerine konarak Φ_i değerleri hesaplanması
7.6.	α , η , Φ_i , β_i değerleri (3.52) ve (3.53) denkleminde yerine konarak r/p , q/p oranları hesaplanması
8.	Durum 8 (α, $\mu_{\min}$, $\mu_{\max}$) ile hesaplama
8.1.	α , $\mu_{\min}$, $\mu_{\max}$ değerleri (3.94) ve (3.95) denkleminde yerine konarak p_i ve w_i katsayıları hesaplanması
8.2.	p_i , w_i katsayılarının (3.91), (3.92) denklemlerinde yerine konarak t_{ϕ_i} kökleri hesaplanması
8.3.	t_{ϕ_i} köklerin (3.5) eşitliğinde yerine konarak ϕ_i çözüm kümesi hesaplanması
8.4.	α , $\mu_{\min}$, $\mu_{\max}$, ϕ_i değerleri (3.84), (3.85), (3.86) denklemlerinde yerine konarak ap_0, ap_1, ap_2 katsayıları hesaplanması
8.5.	ap_0, ap_1, ap_2 değerleri (3.81) veya (3.82) denklemlerinde yerine konarak ve (3.83) eşitliği kullanılarak $\beta_\pm$ çözüm kümesi hesaplanması
8.6.	ϕ_i , β_+ , β_- değerleri (3.69), (3.70) veya (3.71) denklemlerinde yerine konarak $\eta_\pm$ hesaplanması
8.7.	Hesaplanan 48 adet çözüm kümesinin gerçek değere sahip olanları (3.51), (3.52) ve (3.53) denklemlerinde yerine konarak u/p , r/p , q/p oranları hesaplanması
8.8.	u/p , r/p , q/p oranlarının (3.64), (3.65) denklemlerinde yerine konarak $\mu_{\min}$ ve $\mu_{\max}$ hesaplanarak tasarım başlangıcında verilen değerlerle karşılaştırılıp aynı olmayanlar elenmesi



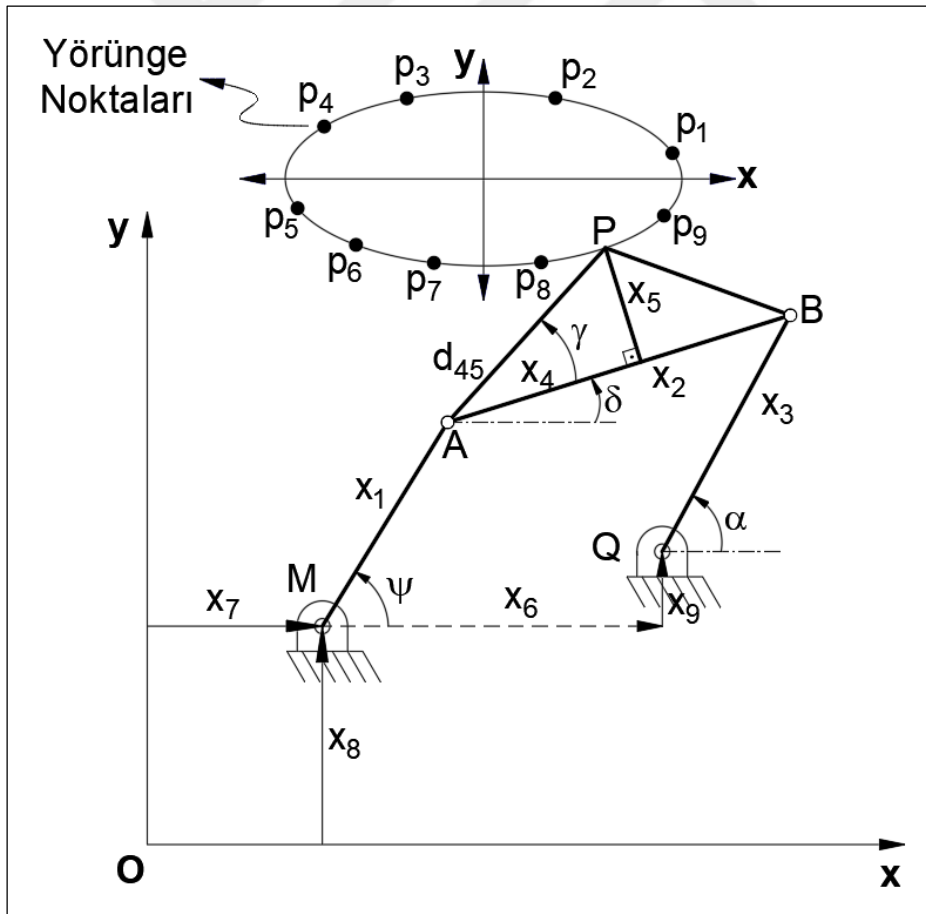
Şekil 3.17. Kol Sarkaç Mekanizması Programı Akış Diyagramı

3.7. Yörünge Sentezi Modülü Teorisi

Bu modül kapsamında biyel uzvu üzerindeki bir noktanın istenen bir yörüngeyi en az hata ile üretebilmesini sağlayan bir dört çubuk mekanizmasının tasarımı amaçlanmıştır. Mutlu (2023) referans alınarak, tasarımda öncelikle istenen yörünge için 5 farklı kesin nokta verilerek bunların dört çubuk mekanizmasına ait kinematik analiz denklemlerinde ardışık yok edilmesi ile yüksek dereceden polinom kökleri olarak mekanizmanın boyutlandırılması yapılmıştır. Sonrasında ise ön tasarım boyutları ve verilen ilk 5 noktaya ek olarak 4 farklı nokta belirlenmiş ve toplam 9 noktada yörüngeyi sağlanması için bir amaç fonksiyonu oluşturularak Newton-Raphson metodu ile en az hatalı yörüngeyi sağlayan mekanizma boyutları elde edilmiştir.

3.7.1. Yörünge Sentezi Ön Tasarım Süreci

Yörünge sentezi parametrelerinin görüldüğü Şekil 3.18’de giriş uzvu x_1 , biyel uzvu x_2 , çıkış uzvu x_3 , biyel uzvu üzerinde olan herhangi bir nokta ise P olarak gösterilmektedir.



Şekil 3.18. Yörünge Sentezi Tasarım Parametreleri

Uzuv boyların vektörel kabul ile biyel uzvu üzerindeki P noktasının orijine göre koordinatları şunlardır;

$$x = x_7 + x_1 \cos \psi + d_{45} \cos(\delta + \gamma) \quad (3.96)$$

$$y = x_8 + x_1 \sin \psi + d_{45} \sin(\delta + \gamma) \quad (3.97)$$

MABQ dörtgeninde ise;

$$x_1 \cos \psi + x_2 \cos \delta = x_6 + x_3 \cos \alpha \quad (3.98)$$

$$x_1 \sin \psi + x_2 \sin \delta = x_9 + x_3 \sin \alpha \quad (3.99)$$

eşitlikleri bulunmaktadır.

(3.96) ve (3.97) denklemlerinde $\cos(\delta + \gamma)$ ve $\sin(\delta + \gamma)$ ifadeleri bir tarafta bırakılarak her iki tarafın karelerinin toplamından (3.100) denklemi elde edilir.

$$x_1^2 + x_7^2 + x_8^2 - d_{45}^2 + x^2 + y^2 - 2x x_7 - 2y x_8 + \cos \psi (2x_1 x_7 - 2x x_1) + \sin \psi (2x_1 x_8 - 2y x_1) = 0 \quad (3.100)$$

(3.98) ve (3.99) denklemlerinde α değerini elemek üzere benzer işlemler yapıldığında (3.101) denklemi elde edilir.

$$x_1^2 + x_2^2 + x_6^2 + x_9^2 - x_3^2 - 2x_2 x_6 \cos \delta + 2x_1 x_2 \cos(\delta - \psi) - 2x_1 x_6 \cos \psi - 2x_2 x_9 \sin \delta - 2x_1 x_9 \sin \psi = 0 \quad (3.101)$$

Mekanizmanın giriş uzvunun yatay eksenle yaptığı ve ilk konumunda bilinmeyen ψ_0 değeri verilen (x, y) yörünge koordinatları ile değişkenlik göstermektedir. Mekanizmanın hareketi sırasında sonraki konumlarında ψ' kadar arttığı ($\psi = \psi_0 + \psi'$) şekilde olacağına göre, (3.100) denklemi yeniden düzenlendiğinde (3.102) ve (3.103) ulaşılır.

$$F_i(\psi_0, x_1, x_7, x_8, d_{45}; x_i, y_i, \psi'_i) = a_i + x_7(-b_i + d_{ci}x_1 \cos \psi_0 - d_{si}x_1 \sin \psi_0) + x_8(-c_i + d_{ci}x_1 \sin \psi_0 - d_{si}x_1 \cos \psi_0) + x_1(-v_{ci} \cos \psi_0 - v_{si} \sin \psi_0) + f_i P = 0, \quad i = 1, \dots, 5 \quad (3.102)$$

$$P = x_1^2 + x_7^2 + x_8^2 - d_{45}^2 \quad (3.103)$$

5 bilinmeyenli 5 denklemi olan (3.102) ifadesinde tasarımcı tarafından belirlenen 5 farklı yörünge noktası koordinatları ($x_i, y_i, i=1,2,3,4,5$), 5 farklı ψ' değeri ve bilinmeyen $\psi_0, x_1, x_7, x_8, d_{45}$ tasarım parametreleri bulunmakta olup katsayıları aşağıdaki şekildedir.

$$a_i = x_i^2 + y_i^2,$$

$$b_i = 2x_i,$$

$$c_i = 2y_i,$$

$$d_{ci} = 2\cos\psi'_i,$$

$$d_{si} = 2\sin\psi'_i, \quad (3.104)$$

$$v_{ci} = 2(x_i\cos\psi'_i + y_i\sin\psi'_i),$$

$$v_{si} = 2(x_i\sin\psi'_i + y_i\cos\psi'_i),$$

$$f_i = 1,$$

$$i = 1, 2, 3, 4$$

(3.102) ifadesinde sırasıyla P , x_7 , x_8 sıralı olarak yok edilerek ve ψ_0 değeri için yarım açı tanjant formülü kullanılırsa t ve 3. dereceden x_i e bağlı (3.105) denklem takımı elde edilir.

$$(a_{pk}t^4 + 2a_{pk}t^2 + a_{pk}) + x_1(-b_{pk}t^4 + 2c_{pk}t^3 + 2c_{pk}t + b_{pk}) + x_1^2((d_{pk} + f_{pk})t^4 - 4e_{pk}t^3 + (2d_{pk} - 6f_{pk})t^2 + 4e_{pk}t + (d_{pk} + f_{pk})) + x_1^3(-g_{pk}t^4 + 2h_{pk}t^3 + 2h_{pk}t + g_{pk}) = 0, \quad (3.105)$$

$$k = 1, 2$$

$$t = \tan \frac{\psi_0}{2} \quad (3.106)$$

Bu denklem takımından x_1 yok edildiğinde ise (3.107) numaralı 14. dereceden polinoma ulaşılır.

$$w_0(1 - t^{14}) + w_1(1 + t^{12})t + w_2(1 - t^{10})t^2 + w_3(1 + t^8)t^3 + w_4(1 - t^6)t^4 + w_5(1 + t^4)t^5 + w_6(1 - t^2)t^6 + w_7t^7 = 0 \quad (3.107)$$

(3.105) denklem takımının katsayıları (a_{pk} , b_{pk} , c_{pk} , d_{pk} , e_{pk} , f_{pk} , g_{pk} , h_{pk} ; $k=1,2$) ve (3.107) polinomuna ait (w_i , $i=0,2,3,..,7$) katsayıları uzun ifadeler olduklarından dolayı EK-3 ve EK-4'de verilmiştir. Bu polinomun çözümü ile ψ_0 değeri için t 'ye bağlı çözüm kümesi (3.108) deki gibi olur.

$$\psi_{0i} = 2 \tan^{-1} t_i; \quad i = 1, 2, \dots, 14 \quad (3.108)$$

Mekanizmanın diğer bilinmeyen parametrelerini bulmak için (3.105) denklem takımında x_i^3 yok edildiğinde (3.109) ifadesine ulaşılır.

$$(aa_2dd_1 - aa_1dd_2) + (bb_2dd_1 - bb_1dd_2)x_1 + (cc_2dd_1 - cc_1dd_2)x_1^2 = 0 \quad (3.109)$$

$$\begin{aligned}
 aa_k &= a_{pk}(t_i^4 + 2t_i^2 + 1) \\
 bb_k &= -b_{pk}t_i^4 + 2c_{pk}t_i^3 + 2c_{pk}t_i + b_{pk} \\
 cc_k &= (d_{pk} + f_{pk})t_i^4 - 4e_{pk}t_i^3 + (2d_{pk} - 6f_{pk})t_i^2 + 4e_{pk}t_i + (d_{pk} + f_{pk}) \\
 dd_k &= -g_{pk}t_i^4 + 2h_{pk}t_i^3 + 2h_{pk}t_i + g_{pk} \\
 i &= 1, 2, \dots, 14 \text{ ve } k = 1, 2
 \end{aligned} \tag{3.110}$$

(3.110)'de her t_i çözümünün karşılığı $x_{I\pm}$ şeklinde ikişer adet olmak üzere toplam 28 elemanlı bir çözüm kümesi elde edilir. ψ_{0i} ve $x_{I\pm i}$ ($i=1,2,3,\dots,14$) değerlerinin (3.7) ifadesinde kullanılmasıyla oluşan denklem takımının her üçü ile $(x_{7i\pm}, x_{8i\pm}, P_{i\pm} \quad i = 1, 2, 3, \dots, 14)$ ve (3.7) ifadesiyle de d_{45} bulunur.

$$d_{45i\pm} = \sqrt{x_{1i\pm}^2 + x_{7i\pm}^2 + x_{8i\pm}^2 - P} \quad , i = 1, 2, 3, \dots, 14 \tag{3.111}$$

Sanal ve gerçek çözümlerin tamamını içeren 28 elemanlı $(x_{1j}, x_{7j}, x_{8j}, d_{45j}, \psi_{0j}, j=1, 2, 3, \dots, 28)$ çözüm kümesindeki gerçek çözümler (3.102) denkleminde yerine konarak anlamlı çözümler elenir ve (3.96), (3.97) denklemlerinde kullanılarak mekanizmada dallanma tespiti yapılır. Bu amaçla tasarım başlangıcında kullanılan 5 yörünge noktasının $(\delta+\gamma)$ değeri (3.7) şeklinde olur ve bunlardan sadece biri istenen sonuca uygunluk gösterir.

$$\begin{aligned}
 (\delta + \gamma)_{i\pm} &= \frac{-B_i \pm \sqrt{B_i^2 - 4A_iC_i}}{2A_i}, \\
 i &= 1, 2, 3, 4, 5
 \end{aligned} \tag{3.112}$$

$$\begin{aligned}
 A_i &= (d_{45} - x_7 + x_i)^2 - (x_1 + x_8 - y_i)(x_1 - x_8 + y_i), \\
 B_i &= 4d_{45}(x_8 - y_i), \\
 C_i &= (d_{45} + x_7 + x_i)^2 - (x_1 + x_8 - y_i)(x_1 - x_8 + y_i), \\
 i &= 1, 2, 3, 4, 5
 \end{aligned} \tag{3.113}$$

Bu adımlar sonrasında mekanizmanın tasarım parametrelerinden olan x_7, x_8, x_1, d_{45} bulunmuş olur. Geriye kalan x_2, x_3, x_6, x_9 ve γ parametreleri ise (3.98), (3.99), (3.100) kullanılarak hesaplanır. Öncelikle α yok edilir ve ardından γ değerine ulaşabilmek için terimler $\delta=\delta+\gamma-\gamma$ şeklinde düzenlenir ve trigonometrik ifadeler $(\delta+\gamma)+(-\gamma)$ işlemine göre yazılırsa (3.114) numaralı denklem kümesi elde edilir.

$$G_i(x_2, x_3, x_6, x_9, \gamma; x_i, y_i, \psi'_i) = pa_i Z_1 + Z_3(pb_i Z_2 \sin \gamma - pe_i Z_2 \cos \gamma - pb_i \cos \gamma - pe_i \sin \gamma) + Z_4(cp_i \cos \gamma + pg_i \sin \gamma) - pf_i Z_2 - dp_i = 0, \quad (3.114)$$

$$i = 1, 2, 3, 4, 5$$

$$Z_1 = \frac{x_1^2 + x_2^2 + x_6^2 + x_9^2 - x_3^2}{2x_1 x_6},$$

$$Z_2 = \frac{x_9}{x_6},$$

$$Z_3 = \frac{x_2}{x_1},$$

$$Z_4 = \frac{x_2}{x_6}$$

$$pa_i = 1,$$

$$pb_i = \cos[(\delta + \gamma)_i]$$

$$pc_i = \cos[(\delta + \gamma)_i - (\psi_0 + \psi'_i)],$$

$$pd_i = \cos[\psi_0 + \psi'_i],$$

$$pe_i = \sin[(\delta + \gamma)_i],$$

$$pf_i = \sin[\psi_0 + \psi'_i],$$

$$pg_i = \sin[(\delta + \gamma)_i - (\psi_0 + \psi'_i)],$$

$$i = 1, 2, 3, 4$$

$((\delta + \gamma)_i, i=1,2,3,4,5)$ değerleri, tasarımda verilen noktalar $(x_i, y_i, i=1,2,3,4,5)$ ve $\psi'_i, i = 1,2,3,4,5$ değerleri ile (3.112) denklemi kullanılarak bulunur ve (3.116) ifadesindeki katsayılar hesaplanır.

(3.114) ifadesinde bilinmeyenleri bulmak için (3.115)'teki Z_1, Z_2, Z_4 sırasıyla yok edilip γ 'ya bağlı katsayılarla sahip Z_3 ün 2. derece polinomu olan (3.117) ifadesine ulaşılır ve burda Z_3 ifadesi yok edilirse 12. Dereceden (3.118) polinomu elde edilir.

$$(pa_{pk} \cos \gamma + pb_{pk} \sin \gamma) + Z_3(pe_{pk} + pf_{pk} \sin 2\gamma + pg_{pk} \cos 2\gamma)Z_3^2(pc_{pk} \cos \gamma + pd_{pk} \sin \gamma) = 0 \quad (3.117)$$

$$k = 1, 2$$

$$q_0(v^{12} + 1) + q_1(v^{10} - 1)v + q_2(v^8 + 1)v^2 + q_3(v^6 - 1)v^3 + q_4(v^4 + 1)v^4 + q_5(v^2 - 1)v^5 + p_6 v^6 = 0 \quad (3.118)$$

$$v = \tan \frac{\gamma}{2} \quad (3.119)$$

(3.117) denklem takımının katsayıları (p_{pk} , $p_{b_{pk}}$, $p_{c_{pk}}$, $p_{d_{pk}}$, $p_{e_{pk}}$, $p_{f_{pk}}$, $p_{g_{pk}}$ $k=1, 2$) ve (3.118) polinomuna ait (q_i , $i=0,2,3,..,6$) katsayıları uzun ifadeler olduklarından dolayı EK-5 ve EK-6'da verilmiştir.

(3.118) polinomunun v 'ye göre çözümü ile γ açısının gerçek ve sanal toplam 12 elemanlı çözüm kümesine ulaşılır (3.120).

$$\gamma_i = 2 \tan^{-1} v_i, i = 1,2,3,...,12 \quad (3.120)$$

(3.120) çözüm kümesi ile (3.117) denklem takımlarında Z_2^2 yok edilmesiyle ortaya çıkan denklemden (Z_{2i} , $i=1,2,3,...,12$) çözülür. Bulunan γ_i , Z_{2i} değerleri (3.114) denklem takımında yerine konarak ve herhangi üçünün kullanımıyla da bulunan (Z_{1i} , Z_{3i} , Z_{4i} , $i=1,2,3,...,12$) değerleri (3.115) denklem takımında yerlerine konur ve (x_{2i} , x_{3i} , x_{6i} , x_{9i} , $i=1,2,3,...,12$) büyüklükleri hesaplanır.

Elde edilen (x_{2j} , x_{3j} , x_{6j} , x_{9j} , γ_j , $j=1,2,3,...,12$) çözümlerinden gerçek olanlar (3.114) denklem takımında denkleminde yerine konarak anlamlı çözümler elenir ve (3.98), (3.99), (3.100) kullanılarak mekanizmada dallanma tespiti yapılır. Öncelikle α yok edilir ve verilen her yörünge noktasındaki δ açısı (3.121) kullanılarak çözülür. Bulunan değerler (3.96) ve (3.97) de yerine konur ve hesaplanan değerlerden yalnız biri verilen noktalarla uyumludur.

$$\delta_{i\pm} = \frac{-B'_i \pm \sqrt{B'^2_i - 4A'_i C'_i}}{2A'_i}, \quad (3.121)$$

$$i = 1, 2, 3, 4, 5$$

$$\begin{aligned} A_i &= x_1^2 - x_3^2 + x_9^2 + (x_2 - x_6)^2 + 2x_1([x_2 - x_6]\cos[\psi_0 + \psi'_i] - x_9\sin[\psi_0 + \psi'_i]) \\ B'_i &= -4x_2(x_9 - x_1\sin[\psi_0 + \psi'_i]) \\ C'_i &= x_1^2 - x_3^2 + x_9^2 + (x_2 - x_6)^2 - 2x_1([x_2 + x_6]\cos[\psi_0 + \psi'_i] + x_9\sin[\psi_0 + \psi'_i]) \end{aligned} \quad (3.122)$$

$$i = 1, 2, 3, 4, 5$$

3.7.2. Yörünge Sentezi Optimizasyon Süreci

Verilen 5 kesin noktadan geçen mekanizmanın ön tasarımından sonra biyel eğrisi üzerinde yörüngeyi geçebileceği 4 farklı nokta daha verilerek, yani maksimum 9 olan parametre sayısına ulaşarak tasarım sürecine devam edilir.

Ön tasarım aşamasının girişinde verilen vektörel eşitliklere OQBP açık döngüsüne ait şu denklemler eklenir;

$$x = x_6 + x_7 + x_3 \cos \alpha + (x_4 - x_2) \cos \delta - x_5 \sin \delta \quad (3.123)$$

$$y = x_8 + x_9 + x_3 \sin \alpha + (x_4 - x_2) \sin \delta + x_5 \cos \delta \quad (3.124)$$

Ön tasarım sürecinde olduğu gibi ψ , (3.96) ve (3.97) denklemlerinde ve α (3.123), (3.124) denklemlerinde yok edilerek, $x_{42}=(x_4-x_2)$ dönüşümü yapılarak ve yarım açı tanjant formülü kullanılarak düzenlenirse $\xi = \tan \frac{\delta}{2}$ olmak üzere (3.125) ve (3.126) denklemleri elde edilir.

$$x^2 + y^2 - x_1^2 + x_4^2 + x_5^2 + x_7^2 + x_8^2 - 2x(x_4 + x_7) - 2y(x_5 + x_8) + 2(x_4x_7 + x_5x_8) + 4\xi[xx_5 - yx_4 - x_5x_7 + x_4x_8] + \xi^2[x^2 + y^2 - x_1^2 + x_4^2 + x_5^2 + x_7^2 + x_8^2 + 2x(x_4 - x_7) + 2y(x_5 - x_8) - 2(x_4x_7 + x_5x_8)] = 0 \quad (3.125)$$

$$[x^2 + y^2 - x_3^2 + x_{42}^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2 + x_9^2 - 2x(x_{42} + x_6 + x_7) - 2y(x_5 + x_8 + x_9) + 2(x_{42}x_6 + x_{42}x_7 + x_6x_7 + x_5x_8 + x_5x_9 + x_8x_9)] + 4\xi[xx_5 - yx_{42} - x_5x_6 - x_5x_7 + x_{42}x_8 + x_{42}x_9] + \xi^2[x^2 + y^2 - x_3^2 + x_{42}^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2 + x_9^2 + 2x(x_{42} - x_6 - x_7) + 2y(x_5 - x_8 - x_9) - 2(x_{42}x_6 + x_{42}x_7 - x_6x_7 + x_5x_8 + x_5x_9 + x_8x_9)] = 0 \quad (3.126)$$

Bu eşitliklerde ξ ifadesi yok edilerek biyel uzvu eğrisine ait ve uzuv boylarını içeren katsayılarla sahip (σ_i , $i=0,1,3,4,5,\dots,15$), 6. derece kapalı (3.127) denklemi elde edilir.

$$H(x, y) = \sigma_0 + \sigma_1y + \sigma_2y^2 + \sigma_3y^3 + \sigma_4(y^4 + x^2y^2) + \sigma_5(x^4y + y^5 + 2x^2y^3) + \sigma_6(x^6 + 3x^2y^4 + 3x^4y^2 + y^6) + \sigma_7(x^5 + xy^4 + 2x^3y^2) + \sigma_8(x^4 + x^2y^2) + \sigma_9x^3 + \sigma_{10}(x^3y + xy^3) + \sigma_{11}x^2 + \sigma_{12}x^2y + \sigma_{13}x + \sigma_{14}xy + \sigma_{15}xy^2 = 0 \quad (3.127)$$

Ön tasarım aşamasında elde edilen boyutlar ve yörünge üzerinden geçmesi istenen 9 nokta (3.127) denkleminde yerine yazıldığında önceden kullanılan 5 kesin noktanın haricinde kalan noktaların denklemi sağlamayacağı beklenir. Tüm noktaların yörünge kapsamına alınması için (3.7) ifadesi 9 bilinmeyenli 9 denklem halinde (3.128) şeklinde yazılır ve çözümü için (3.129)'teki katsayılarla uygun normalize edilmiş amaç fonksiyonunun minimize edilmesi gerekir.

$$H_i(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9; x_i, y_i) = 0, \quad i = 1, 2, 3, \dots, 9 \quad (3.128)$$

$$E(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9) = \frac{\sum_{i=1}^9 H_i^2(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9; x_i, y_i)}{\sum_{i=0}^{15} \sqrt{\sigma_i^2}} = \text{Min!} \quad (3.129)$$

(3.129) denkleminde amaç fonksiyonunun minimum olması demek bu fonksiyonunun bilinmeyen parametre sayısı kadar kısmi türevinin sıfıra eşit olmasını gerektirir. Bu durum matris formunda (3.130) şekilde ifade edilir.

$$\underline{f}(\underline{X}) = \frac{\partial E(\underline{X})}{\partial \underline{X}} = 2 \sum_{i=1}^9 H_i \frac{\partial H_i}{\partial \underline{X}} = 0 \quad (3.130)$$

$$\underline{f}(\underline{X}) = [2 \sum_{i=1}^9 H_i \frac{\partial H_i}{\partial x_1}, 2 \sum_{i=1}^9 H_i \frac{\partial H_i}{\partial x_2}, \dots, 2 \sum_{i=1}^9 H_i \frac{\partial H_i}{\partial x_9}]^T = [0, 0, 0, \dots, 0]^T \quad (3.131)$$

$$\underline{X} = [x_1, x_2, \dots, x_9]^T \quad (3.132)$$

Biyel eğrisine ait (3.131) denkleminin $\underline{X}$ e göre çözümü, amaç fonksiyonu olan (3.129)'ün de çözüm kümesi olabileceği için Newton Rapson metoduna göre aşağıda verilen ardışık tekrarlama kullanılır.

$$\underline{X}_{k+1} = \underline{X}_k - \lambda \underline{J}[\underline{X}_k]^{-1} \underline{f}(\underline{X}_k), k = 0, 1, 2, 3, \dots \quad (3.133)$$

(3.133)'de $\underline{X}_{k+1}$ ifadesi bir sonraki tekrarda hesaplanan parametre değerlerini, (3.134) ise (3.131) denklem kümesine ait birinci, biyel eğrisi denkleminin ikinci türevini içeren Jakobi matrisinin tersidir.

$$\underline{J}[\underline{X}_k] = \begin{bmatrix} \sum_{i=1}^9 \left\{ \left(\frac{\partial H_i}{\partial x_1} \right)^2 + H_i \frac{\partial^2 H_i}{\partial x_1^2} \right\} & \dots & \sum_{i=1}^9 \left\{ \frac{\partial H_i}{\partial x_9} \frac{\partial H_i}{\partial x_1} + H_i \frac{\partial^2 H_i}{\partial x_9 \partial x_1} \right\} \\ \dots & \dots & \dots \\ \sum_{i=1}^9 \left\{ \frac{\partial H_i}{\partial x_1} \frac{\partial H_i}{\partial x_9} + H_i \frac{\partial^2 H_i}{\partial x_1 \partial x_9} \right\} & \dots & \sum_{i=1}^9 \left\{ \left(\frac{\partial H_i}{\partial x_9} \right)^2 + H_i \frac{\partial^2 H_i}{\partial x_9^2} \right\} \end{bmatrix} \quad (3.134)$$

(3.134) ifadesinin çözüm başlangıcında (3.133)'teki $\lambda=1$ olarak Newton-Raphson yöntemi ile başlayan hesaplarda yakınsama olmadığı, çözümünden uzaklaşma veya dallanma durumunda λ gevşetme faktörü, $n>1$ olmak üzere $\lambda=-\lambda/n$ şeklinde değiştirilir ve yakınsama sağlanır.

(3.7) ifadesi çözülürken yapılan adımlar şunlardır;

- a) Yapılacak maksimum ardışık tekrar (N) sınırlandırılır, sonlandırmak için küçük bir ϵ_s belirlenir.
- b) Yapılan ardışık tekrarların kaç adet olduğunun tespiti için $J=1$ olarak belirlenir.

- c) Ön tasarımdaki mekanizma boyutlar kullanılarak $\underline{X}_{j-1}$ başlangıç değeri yapılır ve bunlar kullanılarak (3.129) denkleminde $E(\underline{X}_{j-1})$, (3.131) denkleminde $f(\underline{X}_{j-1})$, (3.134) denkleminde $J[(\underline{X}_{j-1})]^{-1}$ hesaplanır.
- d) $\lambda=1$ yapılarak c. adımda hesaplanan ifadeler (3.133)'de yerine yazılarak $\underline{X}_j$ hesaplanır.
- e) (3.129) ifadesinden $E(\underline{X}_j)$ hesaplanır.
- f) $E(\underline{X}_j)$ ve $E(\underline{X}_{j-1})$ sonuçları karşılaştırılarak $E(\underline{X}_j) > E(\underline{X}_{j-1})$ olması durumunda $n>1$ olmak üzere $\lambda=-\lambda/n$ olarak düzenlenerek e adımına dönlür.
- g) $\left| \frac{E(\underline{X}_{j-1})-E(\underline{X}_j)}{E(\underline{X}_{j-1})} \right| < \varepsilon$ veya $j>N$ ise tasarım sonlandırılır değilse $j=j+1$ şeklinde arttırılarak c adımına dönlerek işlemler tekrarlanır.

3.8. Yörünge Sentezi Modülü Program Arayüzü

Bölüm 3.7'de açıklanan eşitlik ve çözüm adımları kullanılarak geliştirilen bilgisayar programı öncelikle yine metin tabanlı konsol uygulaması olarak yazılmış ve çeşitli kod hataları kontrol edilmiştir. Sonuçlar Şekil 3.19'te görüldüğü biçimde kullanıcıya incelemesi için sunulmaktadır.

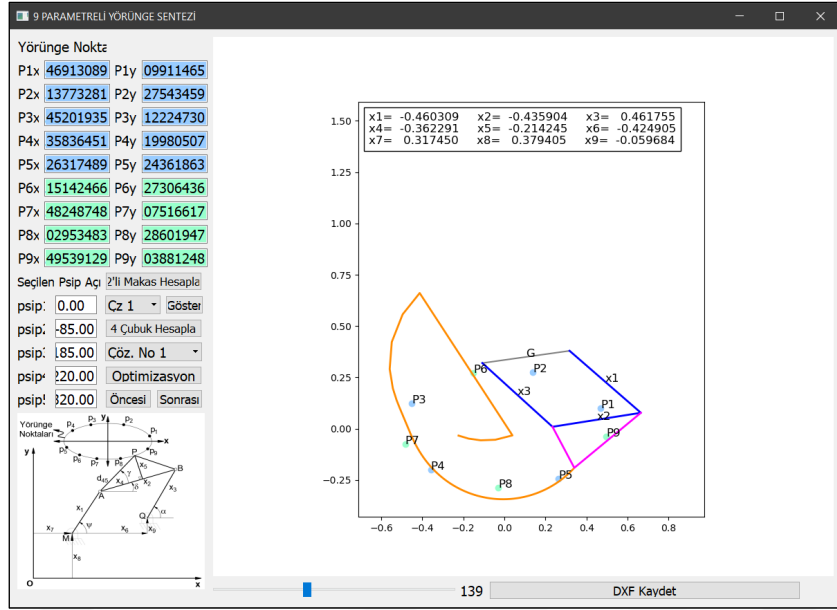
```

1 | Yörünge Noktaları:
2 | x1, y1 = 0.4691, 0.0991
3 | x2, y2 = 0.1377, 0.2754
4 | x3, y3 = -0.4520, 0.1222
5 | x4, y4 = -0.3584, -0.1998
6 | x5, y5 = 0.2632, -0.2436
7 |
8 | Secilen Psip Acilari:
9 | psip1= 0.0000
10 | psip2= -85.0000
11 | psip3= -185.0000
12 | psip4= -220.0000
13 | psip5= -320.0000
14 |
15 | coz.no      x1      x7      x8      ps0      d45
16 | ----
17 | 1      -0.461159    0.316600    0.378555    -178.938554    0.422063
18 | 2      -0.008614    -0.002652    -0.335621    -178.938554    0.635125
19 | 3      1.960606    -0.172967    0.259761    -120.439277    2.239319
20 | 4      -0.397707    0.315086    0.308176    -120.439277    0.553986
21 | 5      -0.415119    0.339707    0.366087    -171.017926    0.434533
22 | 6      -0.096526    0.002913    0.007114    -171.017926    0.378770
23 | 7      -0.594956    0.281727    0.418920    171.450737    0.462905
24 | 8      0.377131    0.158814    1.584896    171.450737    1.686454
25 | 9      -0.463669    -3.181572    -0.789424    -144.218866    3.332250
26 | 10     -0.341475    0.246937    0.301343    -144.218866    0.405609
27 | 11     -0.429140    2.364261    -1.009441    147.481944    2.624421
28 | 12     -0.539881    0.714641    0.045869    147.481944    0.780396
29 | 13     0.415119    0.339707    0.366087    8.982074    0.434533
30 | 14     0.096526    0.002913    0.007114    8.982074    0.378770
31 | 15     0.463669    -3.181572    -0.789424    35.781134    3.332250
32 | 16     0.341475    0.246937    0.301343    35.781134    0.405609
33 | 17     0.461159    0.315935    0.378639    1.061446    0.421634
34 | 18     0.008614    -0.002652    -0.335621    1.061446    0.635125
35 | 19     -1.960606    -0.172967    0.259761    59.560723    2.239319
36 | 20     0.397707    0.273944    0.279018    59.560723    0.522831
37 | 22     -1.193388    0.366711    0.306210    -75.515155    1.420308
38 | 24     1.193388    0.366711    0.306210    104.484845    1.420308
39 | 25     0.429140    2.364261    -1.009441    -32.518056    2.624421
40 | 26     0.539881    0.714641    0.045869    -32.518056    0.780396
41 | 27     0.594956    0.281727    0.418920    -8.549263    0.462905
42 | 28     -0.377131    0.158814    1.584896    -8.549263    1.686454
43 |
44 | x1= -0.461159250    x7= 0.316600428    x8= 0.378555222    ps0= -178.938554363    d45= 0.4220631

```

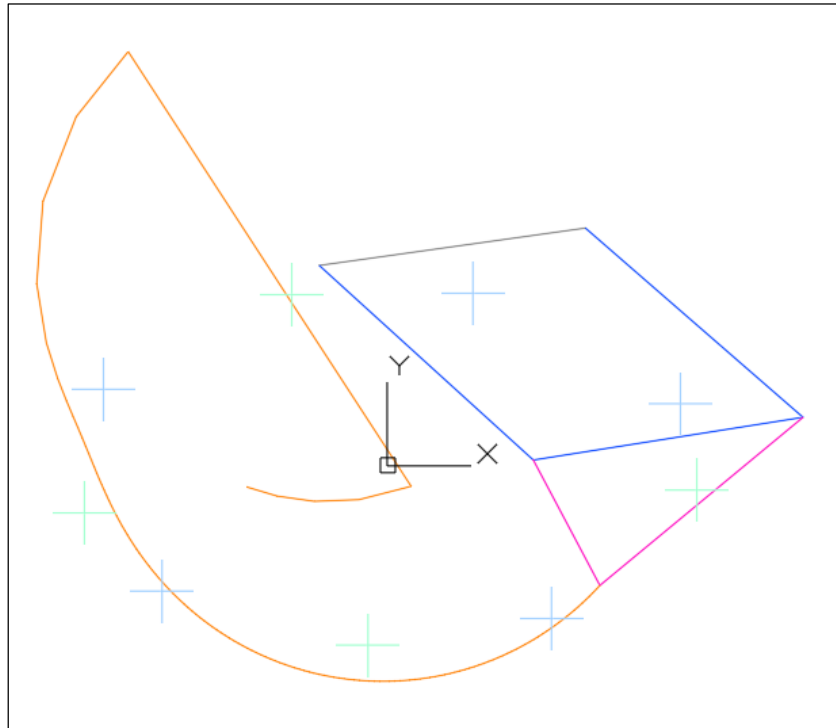
Şekil 3.19. Yörünge Sentezi Ön Tasarım Sonuçları

Ardından Şekil 3.20'deki kullanıcı arayüzü daha önce bahsedilen Python kütüphaneleri kullanılarak geliştirilmiştir.



Şekil 3.20. Yörünge Sentezi Modülü Programı Arayüzü

Kullanıcı arayüzünün solunda programa gömülü tasarım parametreleri ile parametre girişleri yer almaktadır. Açılır menüden kullanıcının seçeceği çözüm arayüze eklenen bir kayar çubuk vasıtasıyla istenen ψ değerinde durdurularak kontrol edilebilmektedir. İstenen çözüm optimizasyona tabi tutularak öncesi ve sonrası hareketli olarak kontrol edilebilmektedir. Ayrıca sonuçlar “dxf” dosya formatında 2D olarak Şekil 3.21’de görüldüğü gibi kullanıcıya verilerek CAD programlarında kullanma imkanı sağlanmaktadır.

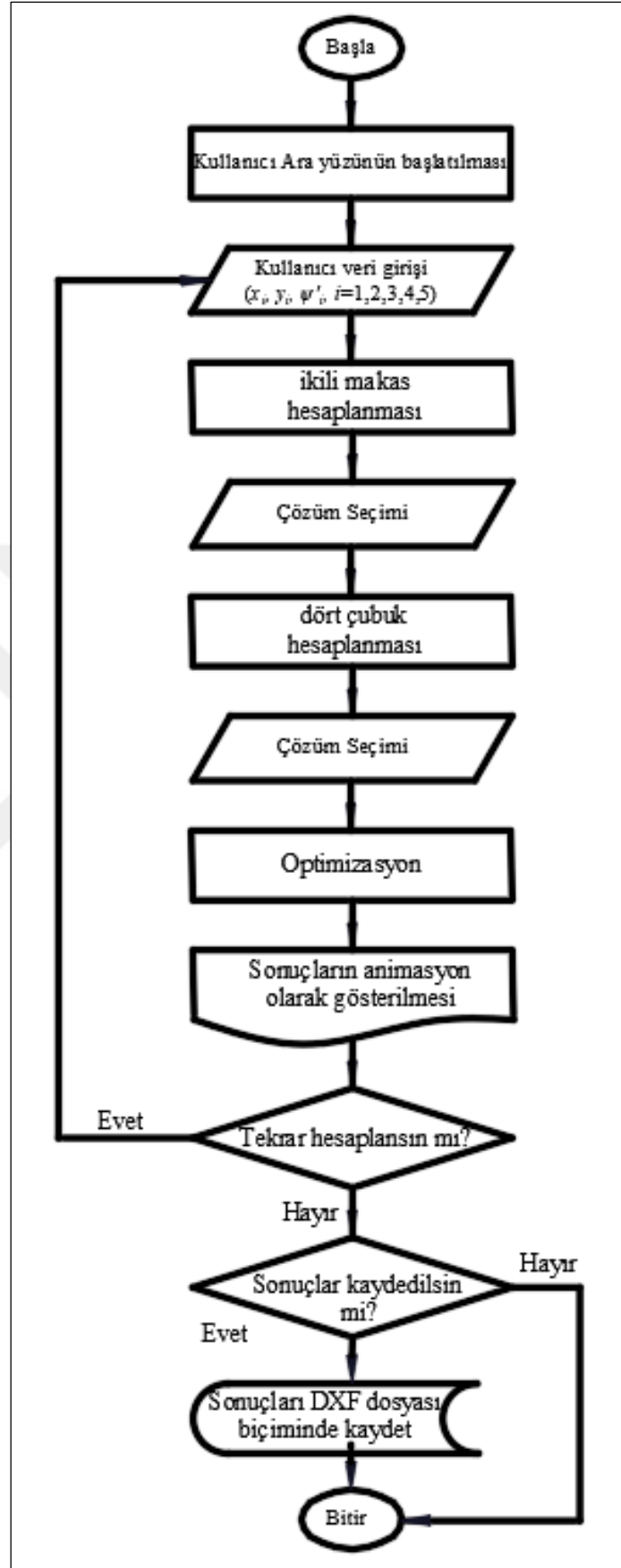


Şekil 3.21. Dxf Dosya Türünde Kaydedilen Yörünge Sentezi Tasarım Sonuçları

Programın hesap adımları Tablo 3.4’de, bu adımlara dayalı oluşturulan akış diyagramı Şekil 3.22’de sunulmuştur.

Tablo 3.5. Yörünge Sentezi Programı Hesaplama Algoritması

1.	Kullanıcıdan 5 adet yörünge noktası (x_i, y_i) ve 5 adet ψ' açısı değerlerinin alınması
2.	(3.102) denkleminin (3.104) denklemindeki katsayılarının hesaplanması
3.	(3.105) denkleminin katsayılarının hesaplanması
4.	(3.107) denkleminin katsayılarının hesaplanması ve çözülmesi ile (3.108) denkleminin çözüm kümesinin ψ_{0i} elde edilmesi
5.	x_{1+i} çözüm kümesinin (3.110) denklem katsayılarının denklem (3.109) kullanılarak hesaplanması
6.	Hesaplanan ψ_{0i} ve x_{1+i} değerlerinin (3.102) ve (3.103) denklemlerinde kullanılması ile $x_{7i\pm}, x_{8i\pm}, P_{i\pm}$ değerlerinin hesaplanması
7.	Hesaplanan $x_{7i\pm}, x_{8i\pm}, P_{i\pm}$ değerlerinin (3.111) denkleminde kullanılması ile $d_{45i\pm}$ değerlerinin hesaplanması
8.	Hesaplanan çözüm kümesinin gerçek sonuçlarının (3.102) ve (3.103) denkleminde yerine konarak sonuçların elenmesi
9.	Elenen sonuçlar için (3.112) ve (3.113) denklemlerinde hesaplanan değerlerin (3.96) ve (3.97) denklemlerinde kullanılarak dallanma tespiti yapılması
10.	(3.120) denklemi ile hesaplanan γ_i değerlerinin (3.117) denkleminde kullanılması ile Z_{2i} değerlerinin hesaplanması
11.	γ_i, Z_{2i} değerlerinin (3.114) denklem takımında yerine konarak kullanılmasıyla Z_{1i}, Z_{3i}, Z_{4i} değerlerinin hesaplanması.
12.	Hesaplanan Z_1, Z_2, Z_3, Z_4 değerlerinin (3.115) denklem takımında yerlerine konmasıyla $x_{2i}, x_{3i}, x_{6i}, x_{9i}$ büyüklüklerinin hesaplanması
13.	Hesaplanan çözüm kümesinin gerçek sonuçlarının (3.114) denklem takımında yerine konarak sonuçların elenmesi
14.	Elenen sonuçlar için (3.121) ve (3.122) denklemlerinde hesaplanan değerleri (3.98), (3.99), (3.100) denklemlerinde kullanılarak dallanma tespiti yapılması
15.	(3.121) denkleminde elde edilen $\delta_{i\pm}$ sonuçlarının (3.96) ve (3.97) denklemlerinde yerine konarak ve sonuçların elenmesi
16.	Kullanıcıdan 4 adet yörünge noktası (x_i, y_i) alınması
17.	N, ε_s ataması yapılması
18.	$J=1$ ataması yapılması
19.	Başlangıç değeri olarak ön tasarımdaki mekanizma boyutlarının $\underline{X}_{j-1}$ ’e atanması
20.	$\underline{X}_{j-1}$ kullanılarak (3.129) denkleminde $E(\underline{X}_{j-1})$ hesaplanması
21.	$\underline{X}_{j-1}$ kullanılarak (3.131) denkleminde $f(\underline{X}_{j-1})$ hesaplanması
22.	$\underline{X}_{j-1}$ kullanılarak (3.134) denkleminde $J[(\underline{X}_{j-1})]^{-1}$ hesaplanması
23.	$\lambda=1$ ataması yapılması
24.	20, 21, 22 adımlarında hesaplanan büyüklüklerin (3.133)’de denkleminde yerine konarak $\underline{X}_j$ hesaplanması
25.	(3.129) denklemi kullanılarak $E(\underline{X}_j)$ hesaplanması
26.	$E(\underline{X}_j)$ ve $E(\underline{X}_{j-1})$ sonuçlarının karşılaştırılması ve $E(\underline{X}_j) > E(\underline{X}_{j-1})$ olması durumunda, $n>1$ olmak üzere $\lambda=-\lambda/n$ ataması yapılarak 25. adıma geri dönülmesi
27.	$\left \frac{E(\underline{x}_{j-1})-E(\underline{x}_j)}{E(\underline{x}_{j-1})} \right < \varepsilon_s$ veya $j>N$ ise tasarım sonlandırılması, değilse $j=j+1$ şeklinde artırılarak 19. adıma dönülerek işlemlerin tekrarlanması



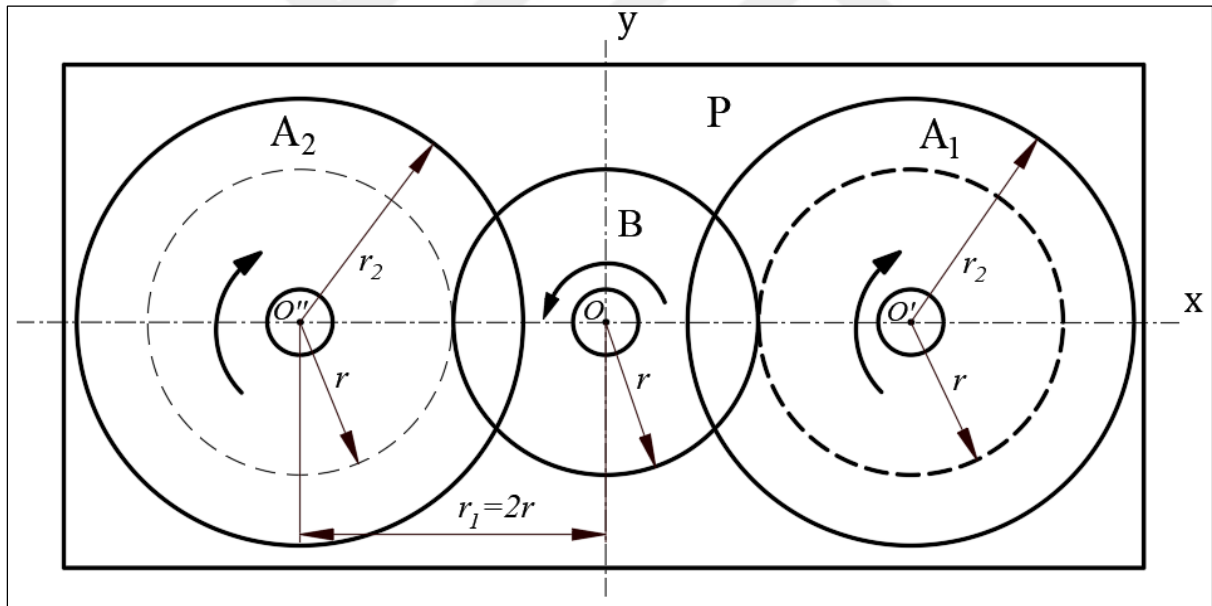
Şekil 3.22. Yörünge Sentezi Programı Akış Diyagramı

V_3 , V_4 , V_5 kesitli hacimsel bölge oluşturmaktadır. Bu sızdırmaz hacimler, rotorların eşit miktardaki dönmesi ile gövde içerisindeki sızdırmazlığını korurken, diğer yandan da gövde içerisindeki konumu ve büyüklüğü değişmektedir. Rotorların bir çevrimi süresince konumu ve büyüklüğü değişen hareketli bu hacimler sayesinde hava vb. sıkıştırılabilir gazların, gövde üzerine açılmış en az iki adet emme kanalı (i, iii) vasıtasıyla dış ortamdan emilmesi, yine gövde üzerine açılmış en az iki adet çıkış kanalından da (ii, iv) basılması sağlanmaktadır, Şekil 3.23(b).

3.9.1. Kompresörün Ön Geometrisinin Oluşturulması

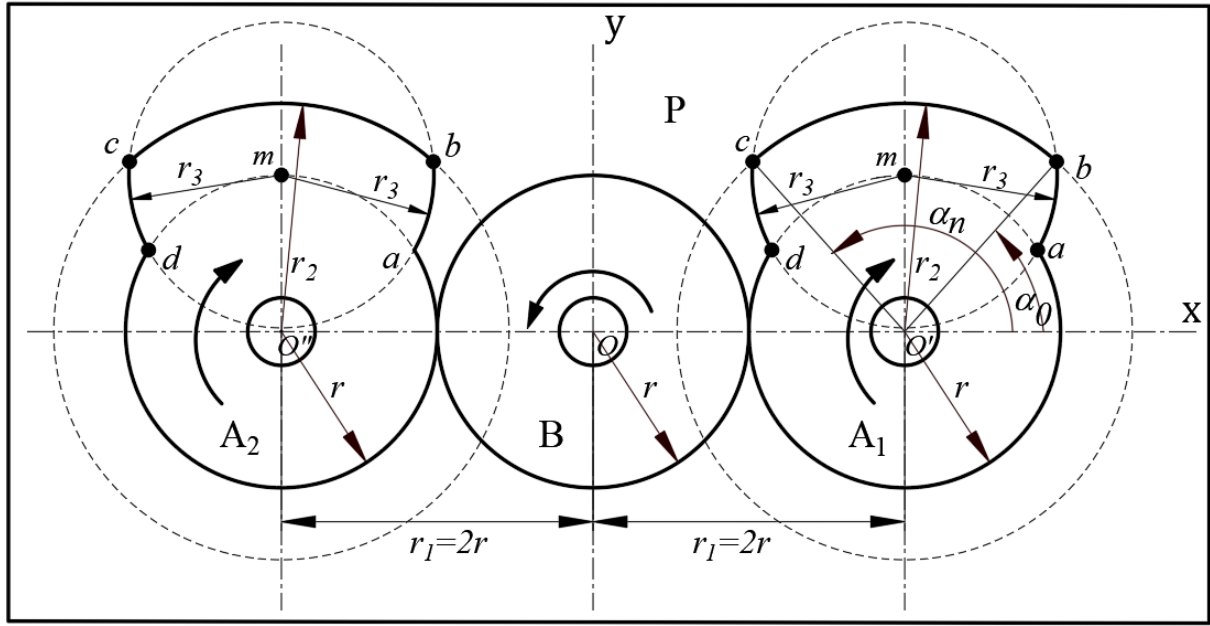
Yukarıda sözü edilen amaçlara ulaşmak için döngüsel kompresörün geometrik tasarımında ilk aşama olarak Şekil 3.24’ de görülen r_2 yarıçaplı A_1 ve A_2 ile r yarıçaplı B dairesel kesitli silindirik geometriler ele alınmıştır.

Kesitlerin yarıçapları (r_2 , r) kompresörün maksimum emme-basma hacimlerini belirlemekte olup seçimi keyfidir. Ayrıca, üç silindirin, Şekil 3.24’de görülen yönlerde, sabit P düzlemindeki O , O' , O'' eksenleri etrafında eşit miktarlarda döndüğü varsayılmıştır.



Şekil 3.24. Kompresör Rotor Kesitlerinin Gövde Üzerindeki Konumları

İkinci aşamada ise r_2 yarıçaplı A_1 ve A_2 kesitleri üzerinde (r_2-r) kalınlığında, r_2 yarıçapında bc yayı ile m ve n merkezli r_3 yarıçapında ab, cd yayları ile sınırlandırılmış kısmi setler Şekil 3.25’teki gibi seçilmiştir. Aynı zamanda seçilen bu kısmi setin oluşturduğu A_1 ve A_2 eş kesiti, kompresörün giriş geometrisi olarak ele alınmaktadır. Kompresörün kapasite, sıkıştırma oranı gibi performans karakteristikleri, seçilen bu kısmi setin geometrik parametreleri değiştirilerek ayarlanabileceğinden, setin geometrik özellikleri (r , r_1 , r_2) aynı zamanda kompresörün tasarım parametreleridir.



Şekil 3.25. Kompresör Rotor Kesitlerinin Tasarım Parametreleri ve Kinematik Düzenleme

3.9.2. Kompresörün Çıkış Geometrisinin Oluşturulması

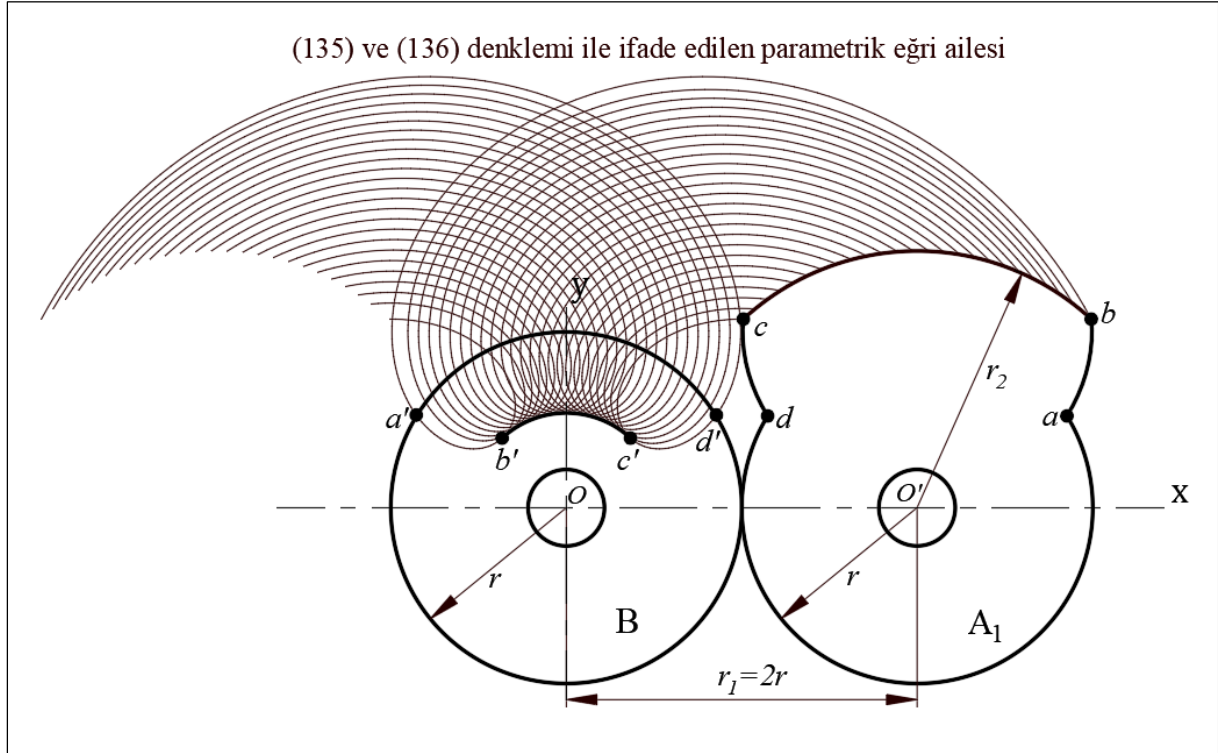
Üzerinde kısmi setlere sahip giriş geometrisi A_1 ve A_2 kesitleri ile B kesitinin birbirleri ile uyumlu çalışabileceği bir kompresör yapısının oluşturabilmesi için, çıkış geometrisi B kesitinin sahip olacağı bir profil eğrisinin tasarlanması gerekmektedir.

Bu profilin tasarlanması için, Şekil 3.25'deki geometrik düzenlemeden sadece A_1 ve B kesitlerinin göz önüne alınması ile Şekil 3.26'teki daha basit bir geometrik yapıdan faydalanılmıştır. Bu yapıda, r_3 yarıçapı sabit tutularak ve m ve n noktalarının r yarıçapı üzerindeki çakışık konumları bc yay uzunluğuna bağlı olarak değiştiği kabulüne göre m , n , r_3 parametreleri Şekil 3.26'te gösterilmemiş ve hesaplamalarda kullanılmamıştır. Bu yapıda, Şekil 3.25'deki kesikli çizgiler yok edilir ve m merkezli r_3 yarıçaplı çember ile O' merkezli r yarıçaplı çember ve aynı merkezli r_2 yarıçaplı çemberlerin oluşturduğu geometrik düzenlemelerden kosinüs teoremi kullanılarak α_0 ve α_n açıları sırası ile (3.137) ve (3.138)'deki gibi hesaplanır.

Bu yeni geometrik düzenlemede B rotörü üzerindeki uygun profilin tasarlanması için, hareketli olan B kesitini içeren düzlem sabit, sabit olan P düzleminin de hareketli yapılması ile Şekil 3.26'teki oluşturulan ters dönüşüm mekanizmasından faydalanılmıştır. B kesitinin uygun profil eğrisinin tasarımı için, söz konusu ters dönüşüm mekanizmasının hareketli düzlemlerinin (A_1 , P) istenilen aralıktaki hareketi süresince, A_1 kesitine ait kısmi setin ab , bc , cd yayları üzerindeki noktaların B kesiti düzleminde çizdiği yörünge eğri ailesinin bulunması gereklidir.

$$\alpha_0 = \csc^{-1}\left(\frac{2 r r_2}{r^2 + r_2^2 - r_3^2}\right) \quad (3.137)$$

$$\alpha_n = \pi - \csc^{-1}\left(\frac{2 r r_2}{r^2 + r_2^2 - r_3^2}\right) \quad (3.138)$$



Şekil 3.27. Ters Dönüşüm Mekanizmasında Yörünge Eğri Ailesi

Buna göre, Şekil 3.26 deki ters dönüşüm mekanizmasından elde edilen (3.135) ve (3.136) denklemi ile ifade edilen parametrik eğri ailesinin geometrisi Şekil 3.27'teki gibidir. B silindirik kesitinin profil eğrisinin tasarımı için bu eğri ailesinden faydalanılmıştır.

B silindirik kesitinin profil eğrisi, $d'c'$, $b'a'$ simetrik eğri parçası ile $c'b'$ eğri parçasının bileşiminden oluşmaktadır, Şekil 3.27. Bunlardan $c'b'$ eğri parçası (3.135), (3.136) parametrik eğri ailesinin zarfıdır. Diğer $d'c'$, $b'a'$ simetrik eğri parçası ise (3.135), (3.136) parametrik eğrisinin bc yayı üzerindeki b ve c noktalarına karşılık gelen eğrinin B silindirik kesitinin dış çemberi ile kesiştiği d' , a' noktası ile zarf eğrisinin teğeti olan c' , b' noktaları arasında kalan eğri parçalarıdır. Bu eğrilerden (3.135), (3.136) ile ifade edilen parametrik eğri denkleminin zarfı olan $c'b'$ eğrisini bulmak için, (3.135), (3.136) parametrik eğri denklemindeki eğri ailesinin zarfını ifade eden (3.139) denkleminde gerekli kısmi türevler alınıp yerine konulduğunda, (3.140) nolu ifadeye ulaşılır.

$$\frac{\partial P_x}{\partial \alpha} \frac{\partial P_y}{\partial \theta} - \frac{\partial P_x}{\partial \theta} \frac{\partial P_y}{\partial \alpha} = 0 \quad (3.139)$$

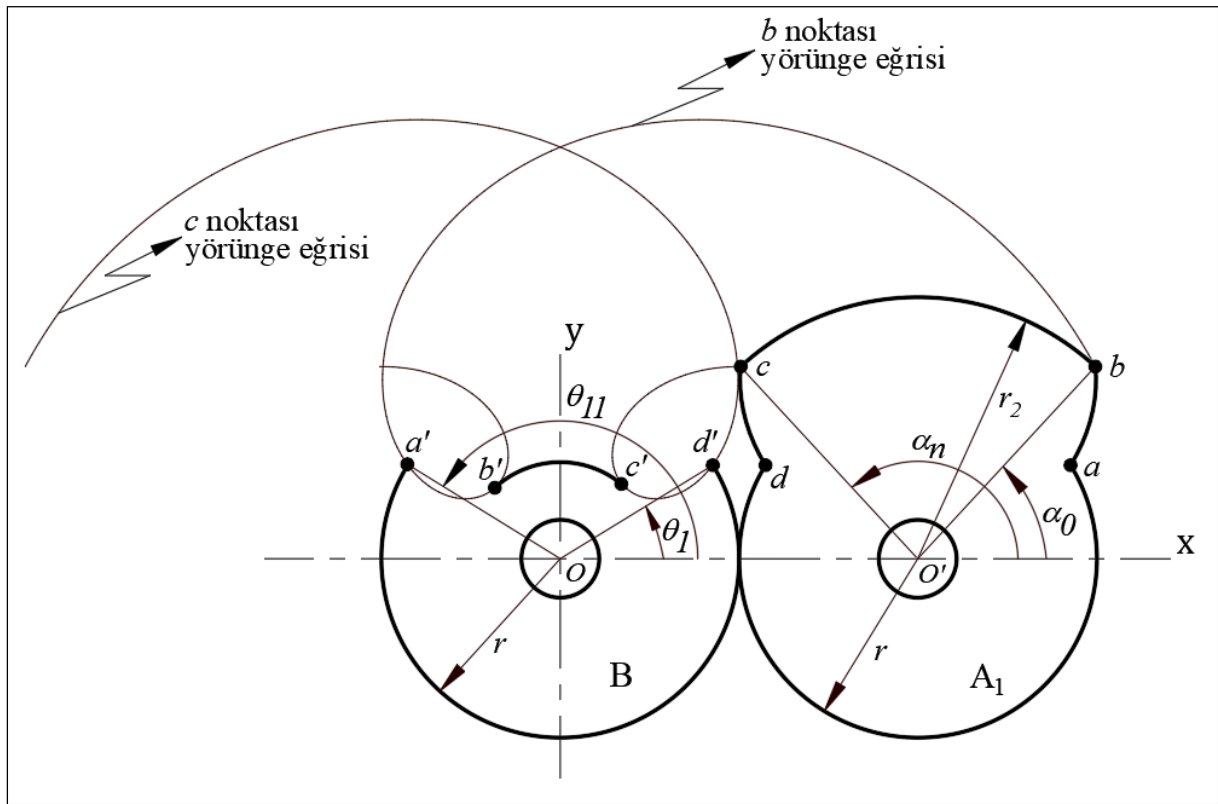
$$r_1 r_2 \sin(\alpha + \theta) = 0 \quad (3.140)$$

(3.140) numaralı periyodik denklemin çok sayıda kökleri arasında $\theta = \alpha + \pi$ kökü aranan çözümdür. Bu çözüm (3.135) ve (3.136) parametrik denklemlerinde yerine konursa, α 'ya bağlı (3.141) ve (3.142) zarf eğrisi denklemleri elde edilir.

$$z_x(\alpha) = r_1 \cos(\pi - \alpha) + r_2 \cos(2\pi - \alpha) \quad (3.141)$$

$$z_y(\alpha) = r_1 \sin(\pi - \alpha) + r_2 \sin(2\pi - \alpha) \quad (3.142)$$

(3.141), (3.142) denklemleri $\alpha_0 \leq \alpha \leq \alpha_n$ aralığında (3.135), (3.136) parametrik eğrinin zarfı olan ve Şekil 3.28'da görülen $c'b'$ eğri parçasını ifade etmektedir.



Şekil 3.28. a' ve d' noktasının yerini belirleyen bağımsız değişkenler

$d'c'$ ve $b'a'$ eğri parçalarını bulmak için ise, öncelikle (3.135), (3.136) denklemi ile temsil edilen parametrik eğri ailesinden elde edilen ve Şekil 3.28'daki A_1 kesiti üzerindeki b noktasının çizdiği $(P_x(\alpha_0, \theta), P_y(\alpha_0, \theta))$ parametrik eğri ile c noktasının çizdiği $(P_x(\alpha_n, \theta), P_y(\alpha_n, \theta))$ parametrik eğrinin (3.141), (3.142) ifadesindeki zarf eğrisi ile kesiştiği c' ve b' noktalarına karşılık gelen θ bağımsız değişkenin bulunması gerekir. Bunun için genel bir ifadeye ulaşmak amacıyla $(P_y(\alpha, \theta) = Z_y(\alpha))$ koşulu eşitlik (3.143)'de gösterildiği gibi yazılıp çözülmesi gerekir.

$$r_1 \sin \theta + r_2 \sin(\alpha + 2\theta) = r_1 \sin(\pi - \alpha) + r_2 \sin(2\pi - \alpha) \quad (3.143)$$

Çözüm için eşitlik (3.143)'deki trigonometrik ifadeler açılıp, yeniden düzenlenip sadeleştirilmesinden sonra, yarım açı tanjant formülleri kullanılarak, $(t = \tan \frac{\theta}{2}, \sin \theta = \frac{2t}{1+t^2}, \cos \theta = \frac{1-t^2}{1+t^2})$, formüllerdeki t dönüşümü yapılması sonrasında aşağıdaki t ye bağlı (3.144) numaraları dördüncü dereceden polinoma ulaşılır.

$$(r_1 \sin \alpha - 2r_2 \sin \alpha)t^4 + (4r_2 \cos \alpha - 2r_1)t^3 + (2r_1 \sin \alpha + 4r_2 t^2 \sin \alpha)t^2 + (-2r_1 - 4r_2 \cos \alpha)t + r_1 \sin \alpha - 2r_2 \sin \alpha = 0 \quad (3.144)$$

Sanal ve gerçek olmak üzere 4 adet kökü bulunan (3.9) polinomunun köklerinden tasarım amacına uygun olanı (3.145) ifadesindeki çözümdür.

$$t = \cot \frac{\alpha}{2} \quad (3.145)$$

(3.145) denkleminde θ parametresi $t = \tan \frac{\theta}{2}$ dönüşümünden faydalanılarak eşitlik (3.9) deki gibi bulunur

$$\theta = 2 \tan^{-1}(\cot \frac{\alpha}{2}) \quad (3.146)$$

(3.145) çözümü yardımı ile Şekil 3.27'deki b' ve c' noktasının yerini belirleyen (3.135), (3.136) denklemindeki bağımsız değişken (θ) ya karşılık gelen sırası ile θ_{33} ve θ_3 açısal büyüklükleri, (3.146) denkleminde sırasıyla $\alpha = \alpha_0$, $\alpha = \alpha_n$ konulması sonucunda, denklem (3.147), (3.148)'deki gibi bulunur.

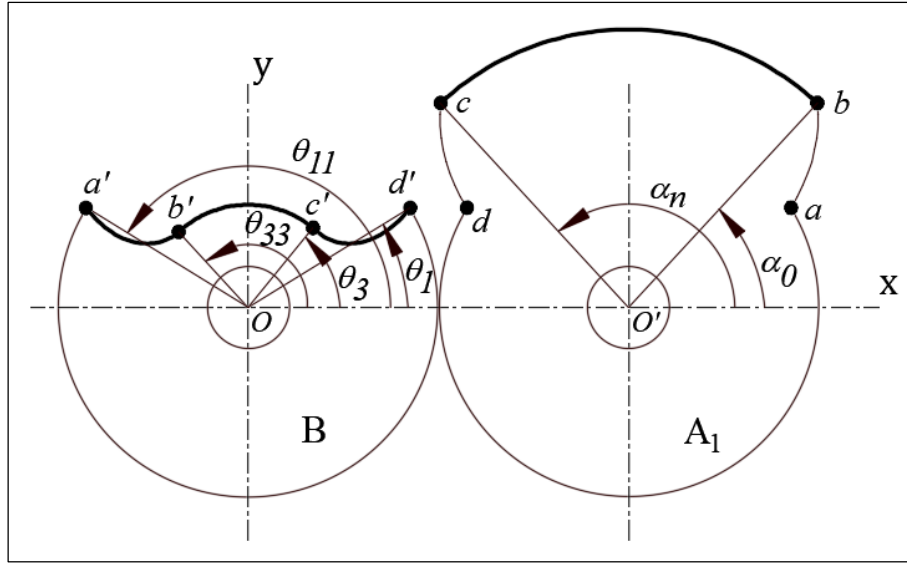
$$\theta_{33} = 2 \tan^{-1}(\cot \frac{\alpha_0}{2}) \quad (3.147)$$

$$\theta_3 = 2 \tan^{-1}(\cot \frac{\alpha_n}{2}) \quad (3.148)$$

Benzer olarak, a' , d' noktalarına karşılık gelen θ bağımsız değişkeninin bulunması için genel bir ifadeye ulaşmak amacıyla, Şekil 3.28'deki A_1 kesiti üzerindeki b noktasının çizdiği $(P_x(\alpha_0, \theta), P_y(\alpha_0, \theta))$ parametrik eğri ile c noktasının çizdiği $(P_x(\alpha_n, \theta), P_y(\alpha_n, \theta))$ parametrik eğrinin B kesitinin dış çemberi ile kesişimini ifade eden (3.148) denklemindeki koşuldan yararlanılır.

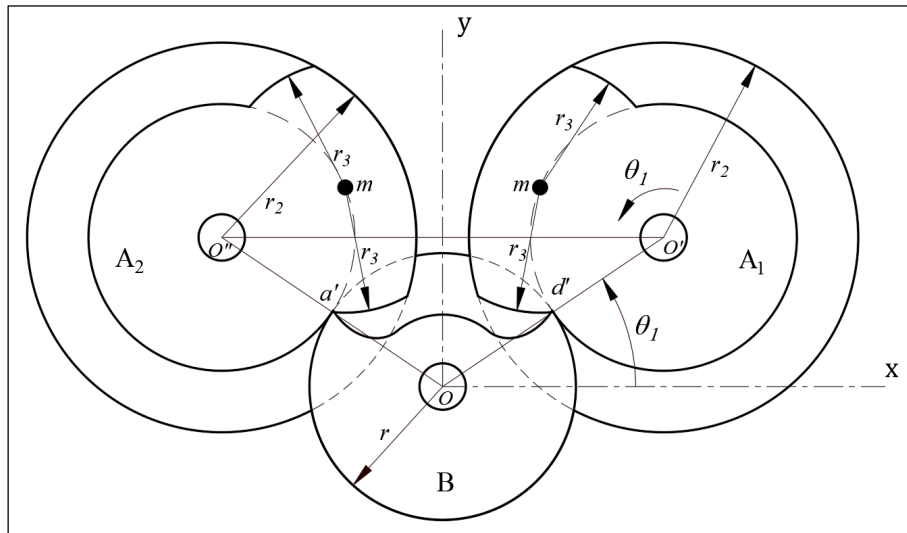
$$(P_x(\alpha, \theta))^2 + (P_y(\alpha, \theta))^2 - r^2 = 0 \quad (3.149)$$

(3.135), (3.136) ifadeleri (3.149) de yerine yazılıp, üslü ve trigonometrik ifadeler açılıp, yeniden düzenlenip ve sadeleştirilmesinden sonra yarım açı tanjant formülleri kullanılarak $(v = \tan \frac{\theta}{2}, \sin \theta =$



Şekil 3.30. B silindirik kesitinin hesaplanan büyüklükleri

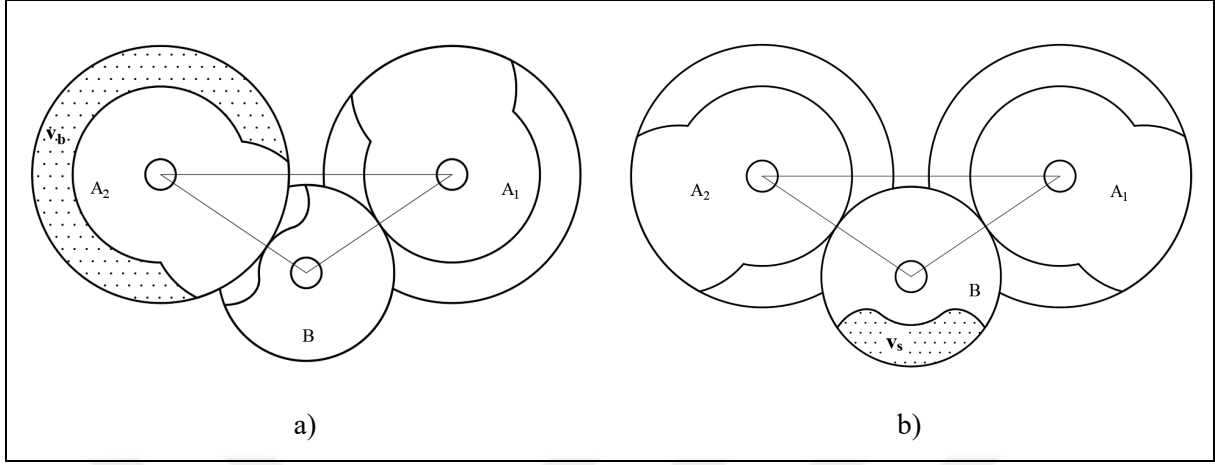
Giriş geometrisi A₁ rotoru ve çıkış geometrisi B rotoru yardımı ile bir kompresör yapısının oluşturulması Şekil 3.31’de gösterilmiştir. Bu düzenleme, A₁ rotoru önce O noktası etrafında θ_1 ve O’ noktası etrafında da aynı miktarda döndürüldükten sonraki konumundaki A₁ rotorunun y eksenine göre ayna görüntüsü alınarak elde edilir, Şekil 3.31. Kompresörü çevreleyen dış gövdesi, Şekil 3.31da görülen O merkezli r yarıçaplı ile O’ ve O’’ merkezli r_2 yarıçaplı çemberler çizilir ve çizilen çemberlerin, Şekil 3.31’deki kesikli çizgi ile gösterilen yayları budanarak elde edilir.



Şekil 3.31. Kompresör modelinin geometrik düzenlemesi

Elde edilen kompresör yapısında büyük V_b ve küçük V_s gibi iki tane emme basma yapabilen hacimsel bölgenin en büyük ve en küçük olduğu rotor konumları sırası ile Şekil 3.32a’ ve Şekil 3.32b’de gösterilmiştir. Bu iki hacmin toplamı kompresörün hacimsel kapasitesini belirlemektedir. Bu hacimlerin

büyüklüğü, Şekil 3.25’de görülen r , r_2 ve r_3 yarıçapı ile temsil edilen parametrelerin seçimi ile belirlenmektedir.



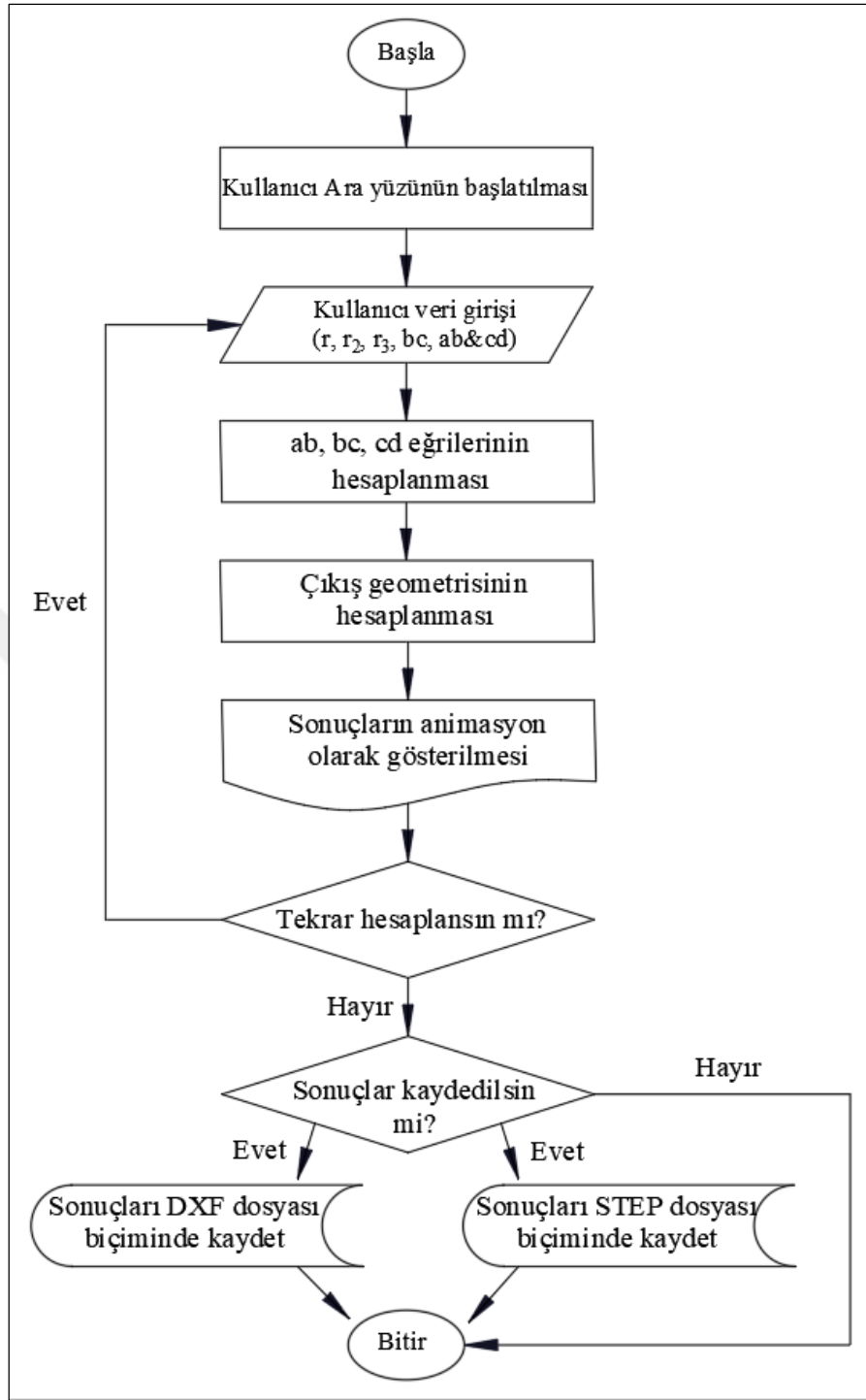
Şekil 3.32. Kompresör kapasitesini belirleyen büyük (a) ve küçük (b) hacimsel bölgeyi oluşturan rotor konumları

3.9.3. Kompresör Geometri Hesabı Program Arayüzü

Yukarıda verilen kompresörün matematik modeline dayalı bilgisayar programı öncelikle yine metin tabanlı konsol uygulaması olarak yazılmış ve çeşitli kod hataları kontrol edilmiştir. Programın hesap adımları Tablo 3.5’te, ve bu adımlara dayalı oluşturulan akış diyagramı Şekil 3.33’da sunulmuştur. Bu hesap adımı ve akış diyagramına göre geliştirilen program ve arayüzü daha önce bahsedilen Python kütüphaneleri kullanılarak geliştirilmiştir.

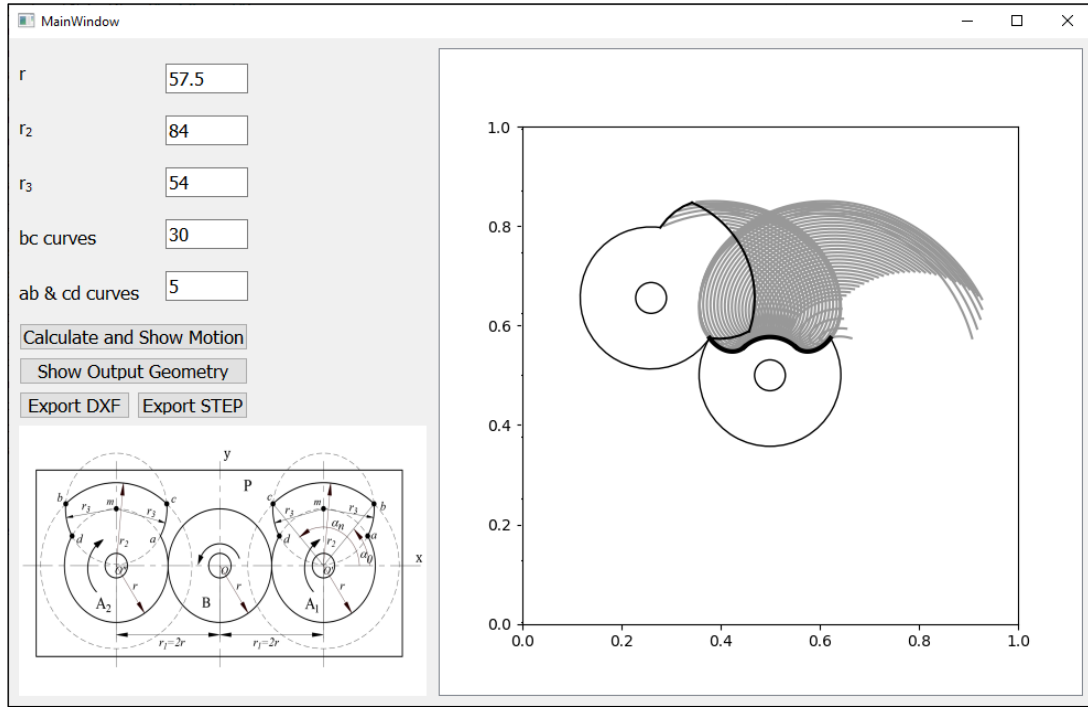
Tablo 3.6. Kompresör Çıkış Geometrisi Hesaplama Algoritması

1.	Kullanıcıdan $r, r_2, r_3, bc \text{ curves}, ab \& cd \text{ curves}$ parametrelerinin alınması
2.	Denklem (3.137) ve (3.138) kullanılarak α_0 ve α_n değerlerinin hesaplanması
3.	bc arasındaki istenen sıklıkta ($bc \text{ curves}$) noktanın yörünge eğri ailesinin θ ve α değişkenlerini denklem (3.135) ve (3.136)’da kullanılarak $0^\circ \leq \theta \leq 180^\circ$ ve $\alpha_0 < \alpha < \alpha_n$ aralıklarında hesaplanması.
4.	Eğri ailesinin zarf eğrisinin (çıkış geometrisindeki $c'b'$ eğri parçası) r_2, α değişkenlerini denklem (3.141) ve (3.142)’de kullanılarak hesaplanması.
5.	b' noktasının yerini belirleyen θ_{33} açısız büyüklüğünün denklem (3.147)’de α_0 değişkeni kullanılarak bulunması.
6.	c' noktasının yerini belirleyen θ_3 açısız büyüklüğünün denklem (3.148)’de α_n değişkeni kullanılarak bulunması.
7.	a' noktasının yerini belirleyen θ_1 açısız büyüklüğünün denklem (3.152)’de α_0, r, r_2 değişkenleri kullanılarak bulunması.
8.	d' noktasının yerini belirleyen θ_{11} açısız büyüklüğünün denklem (3.153)’de α_n, r, r_2 değişkenleri kullanılarak bulunması.



Şekil 3.33. Kompresör Geometri Hesabı Programı Akış Diyagramı

Geliştirilen programın kullanımı için verilen örnekteki sayısal değerlerde, izleyen Bulgular ve Tartışma kısmında verilen 1. örnek referans alınmıştır. Buna göre, programın Şekil 3.34’da gösterilen ana penceresinin Şekil 3.35’teki veri girişi kutularından kompresör geometrisi ile ilgili boyutsal parametreler girilmektedir. Bu kutulardan, kompresör pistonlarına ait kesit yarıçapları (r, r_2, r_3), giriş geometrisinin kısmi seti olan bc , ab ve cd yay parçalarının hassasiyetini belirleyen adım miktarları kullanıcı tarafından girilmektedir.

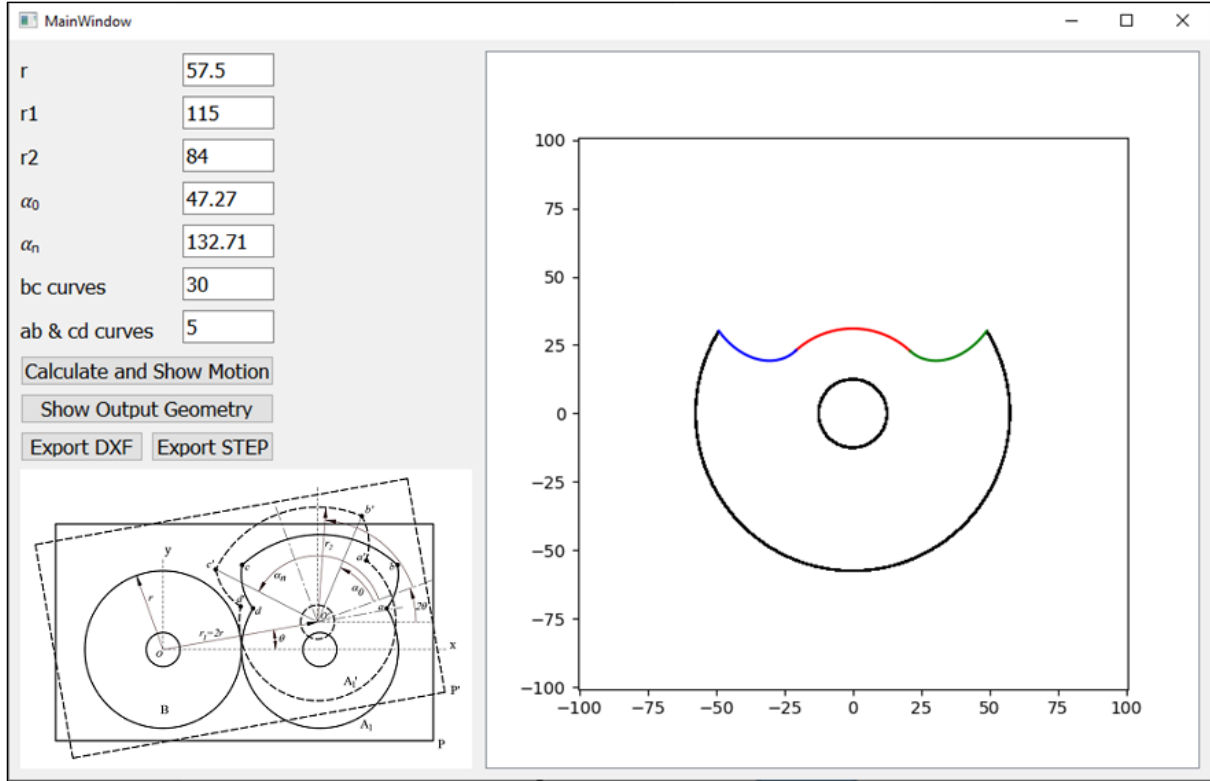


Şekil 3.34. Etkileşimli Kullanıcı Arayüzünün Ana ve Simülasyon Penceresi

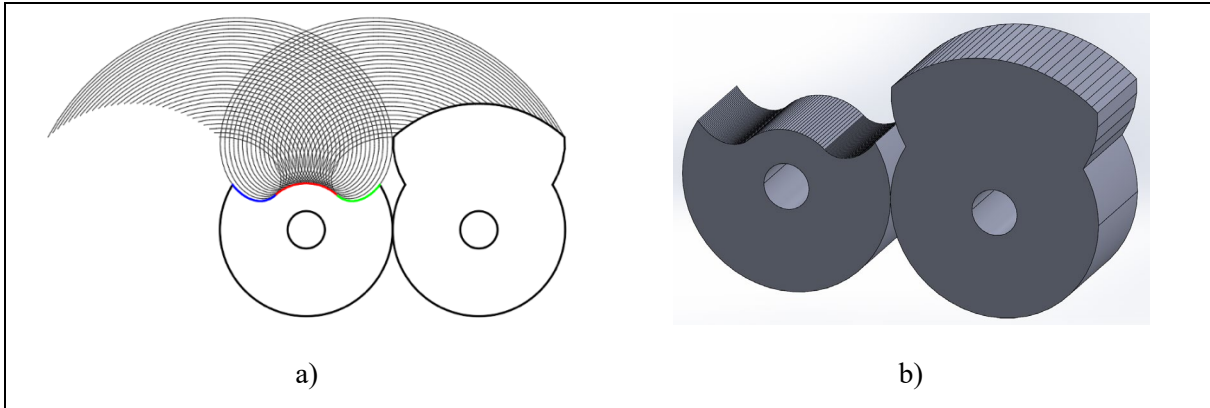
Şekil 3.35. Parametre Giriş Menüsü

Programın işletilmesi için “Calculate and Show Motion” butonuna basıldığında, menünün sağ kısmındaki pencerede eğri ailesi ile giriş ve çıkış geometrisinin bir animasyonu Şekil 3.34’deki gibi gösterilmektedir. Benzer biçimde “Show Output Geometry” butonuna basıldığında ise çıkış geometrisinin son hali Şekil 3.36’deki gibi görülmektedir. İsteğe bağlı olarak farklı CAD programlarında değerlendirilmek üzere, giriş ve çıkış geometrisinin 2D dxf formatında kaydetmek için

“Export DXF” butonuna, 3D STEP formatında kaydetmek için ise “Export STEP” butonuna basmak yeterlidir. Bununla ilgili dxf ve step çıktı örnekleri sırası ile Şekil 3.37a ve Şekil 3.37b de verilmiştir.



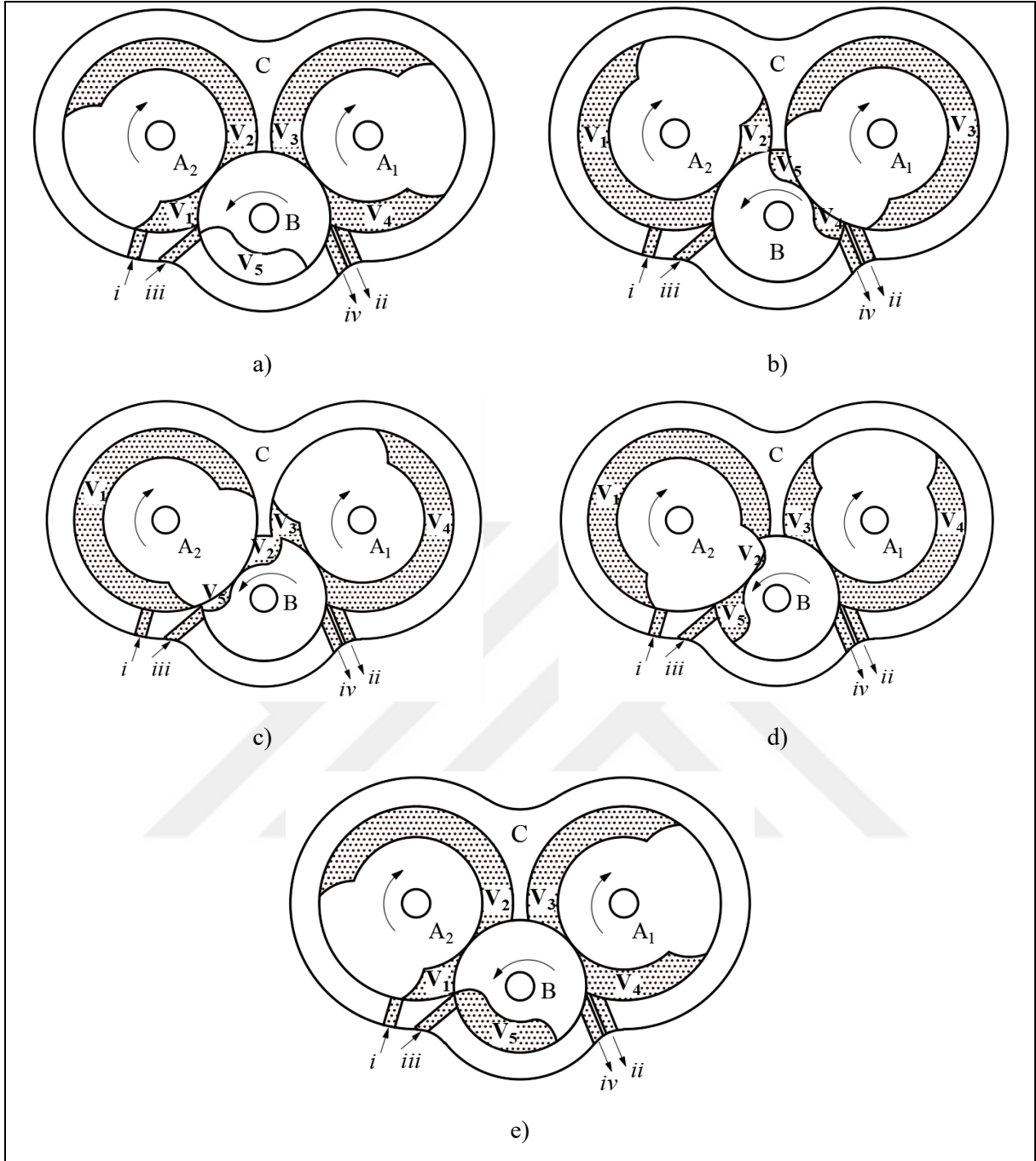
Şekil 3.36. Etkileşimli Kullanıcı Arayüzünün Ana ve Çıkış Geometri Penceresi



Şekil 3.37. Giriş ve Çıkış Geometrisinin Dxf Formatı (a) ve Step Formatı (b)

3.9.4. Kompresörün Çalışma Prensibi

Kompresörün çalışma prensibini gösterebilmek için Şekil 3.23(b) deki geometriden faydalanılmıştır. Aynı zamanda Bulgular ve Tartışma kısmında birinci örnek olan bu kompresör yapısının bir çevrimdeki çalışmasının tipik evreleri de Şekil 3.38’deki gibidir.



Şekil 3.38. Örnek 1’deki kompresör yapısının 1 çevrimdeki I. Evre (a), II. Evre (b), III. Evre (c), IV. Evre (d) ve V. Evre (e) kompresyon evreleri.

Buna göre, Şekil 3.38’deki kompresörün silindirik kesit ve gövdeden oluşan geometrik düzenlemesinde bağımsız emme basma yapabilen iki farklı hacim bulunmaktadır. Birincisi, Şekil 3.38 (a)’da V₁ ile gösterilen bölgenin oluşturduğu hacimdir. Bu hacim, dış ortamdaki gazın pistonların dönme hareketi ile *i* girişinden emilerek C gövdesi içerisinde A₁, A₂, B silindirik kesitlerinin hareketi ile V₄ hacmine aktararak *ii* çıkışından basılmasını gerçekleştirmektedir. İkincisi ise, Şekil 3.38 (a)’da görüldüğü üzere, piston hareketleri ile *iii* girişinden gazı emerek *iv* çıkışından doğrudan basan V₅ bölgesinin oluşturduğu hacimdir. Her iki hacim bölgesi yardımı ile gerçekleştirilen emme-basma işi, C

gövdesi içerisine 3 adet eş dişli ile yataklanan A_1 , A_2 , B döner silindirik kesitlerinin uyumlu geometrisi sayesinde ve silindirik kesitlerin tam bir devir dönmesi sonunda (bir çevrim) gerçekleşmektedir. Yukarıda sözü edilen emme-basma basma sürecinin ayrıntılı izahı şu şekildedir.

Şekil 3.38 referans alınarak, kompresörün A_1 , A_2 döner silindirik kesitlerinin saat yönünde, B silindirik kesitinin saatin ters yönünde eşit miktarda döndüğü göz önüne alınmaktadır. Buna göre, C gövdesi içerisindeki A_1 , A_2 , B silindirik kesitlerinin Şekil 3.38(a)'daki konumları başlangıç konumu olarak referans alınmıştır, (I. Evre). Bu konumda kompresörün içerisindeki V_1 , V_2 , V_3 , V_4 , V_5 ile gösterilen hacimler gaz ile doludur.

A_2 silindirik kesiti saat yönünde bir miktar dönerek Şekil 3.38(b)'deki konuma geldiğinde (II. Evre), V_1 hacmi artmaktadır. Bu artış sayesinde *i* giriş kanalından dış ortamdaki gaz emilmektedir. Bu esnada da diğer V_2 hacmi içerisindeki gaz sıkışmakta, V_3 hacmi içerisindeki gaz genişmekte, V_4 hacmi içerisindeki gaz *ii* çıkış kanalından ve V_5 hacmi içerisindeki gaz da diğer *iv* çıkış kanalından basılmaktadır. Ayrıca, bu süreçte B silindirik kesitinin silindirik yüzeyi *iii* nolu girişi, A_1 silindirik kesitinin silindirik yüzeyi ise *iv* nolu çıkışı kapatmıştır.

Silindirik kesitler Şekil 3.38(c)'deki konuma geldiğinde (III. Evre), V_1 hacmi içerisine gaz emilmeye devam edilirken, V_2 hacim bölgesinde sıkışan gaz, I Evredeki V_3 hacmi içerisindeki gaz ile birleşmeye başlamaktadır. Eş zamanlı olarak da V_4 hacmindeki genişmiş gaz A_1 silindirik kesiti tarafından kompresörün *ii* nolu çıkışına süpürülürken *iii* nolu giriş açılarak V_5 hacim bölgesine gaz emilmeye başlamıştır. Bu sırada *iv* çıkışı B silindirik kesitinin dış yüzeyi tarafından kapatılmış durumdadır.

Şekil 3.38(d)'deki IV. evrede ise V_1 hacmine maksimum miktarda hava emilmiş ve pistonun A_1 silindirik kesitinin dış yüzeyi ile *i* nolu giriş kanalı kapatılmış durumdadır. Aynı zamanda bir önceki evrede açılmaya başlayan *iii* nolu giriş kanalından V_5 hacim bölgesine gaz emilmeye devam ederken, V_3 hacim bölgesine aktarılan bir önceki çevrim gazı genişleme başlangıcındadır. V_4 hacim bölgesindeki gaz da *ii* nolu çıkış kanalından basılmaya devam etmektedir.

Son evre olan Şekil 3.38(e)'deki V. Evrede, döner pistonun A_2 silindirik kesitinin dış yüzeyi tarafından *i* nolu girişi kapatılmıştır. Bu anda, birinci emme hacmi olan V_2 bölgesindeki gaz pistonun A_2 silindirik kesitinin hareketi ile sıkışmakta, V_3 hacim bölgesindeki gaz ise pistonun A_1 silindirik kesitinin hareketi ile genişmekte, V_4 hacim bölgesindeki gaz ise yine A_1 silindirik kesitinin hareketi ile *ii* nolu çıkış kanalından basılmaya devam etmektedir. Ayrıca bu V. evrede ikincil V_5 emme hacim bölgesine maksimum miktarda gaz, ileride açılacak olan *iv* nolu çıkış kanalından basılmak üzere, emilmeye devam etmektedir. Sonuç olarak bu süreç periyodik olarak devam etmektedir.



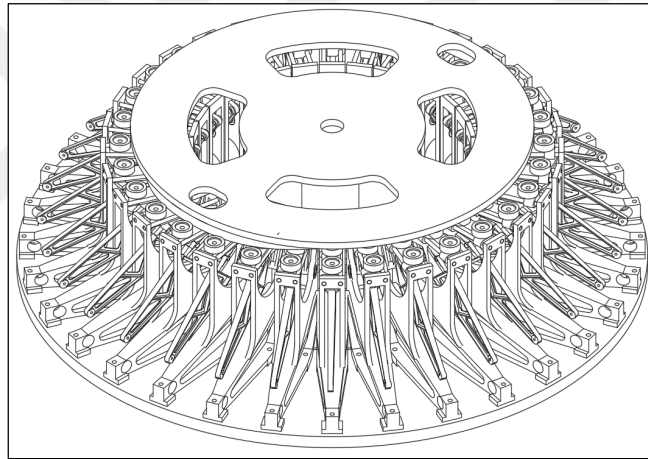
4. BULGULAR ve TARTIŞMA

4.1. Kam Mekanizması Modülü Örnek Uygulamaları

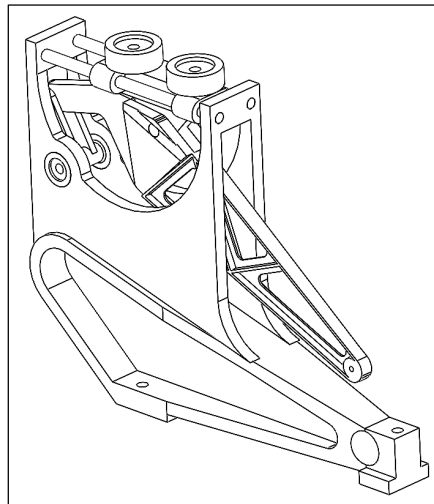
Bu bölümde bölüm 3’de açıklanan teori kullanılarak geliştirilen ve bölüm 3.2’de gösterilen bilgisayar programı kullanılarak endüstriyel uygulaması olan 2 örnek yapılmıştır.

4.1.1. Dairesel Dokuma Tezgâhı Kam Profili Tasarımı Örnek Uygulaması

Bu örnek kapsamında özel bir tekstil şirketinde kullanılmakta olan, Şekil 4.1’de gösterilen dairesele dokuma tezgâhlarında atkı ipleri arasından mekiğin geçmesinde kullanılan ve Şekil 4.2 gösterilen mekik ağız açma sistemlerinin bir parçası olan kam profili belirlenmiştir. Sistem tasarlanan kamın mekik ağız açma sistemi üstündeki silindirik iticiler arasında dönmesi sonucu çalışmakta olup tasarımı yapılan kam iç ve dış birlikte temaslı tasarıma örnektir.

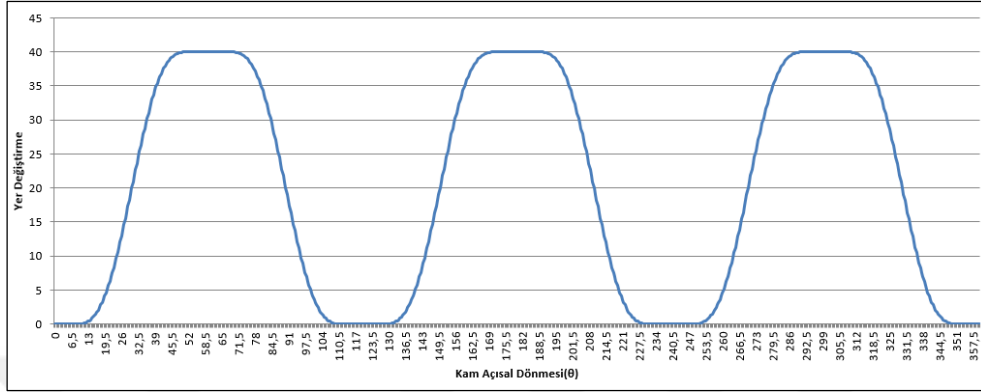


Şekil 4.1. Dairesel Dokuma Tezgahı



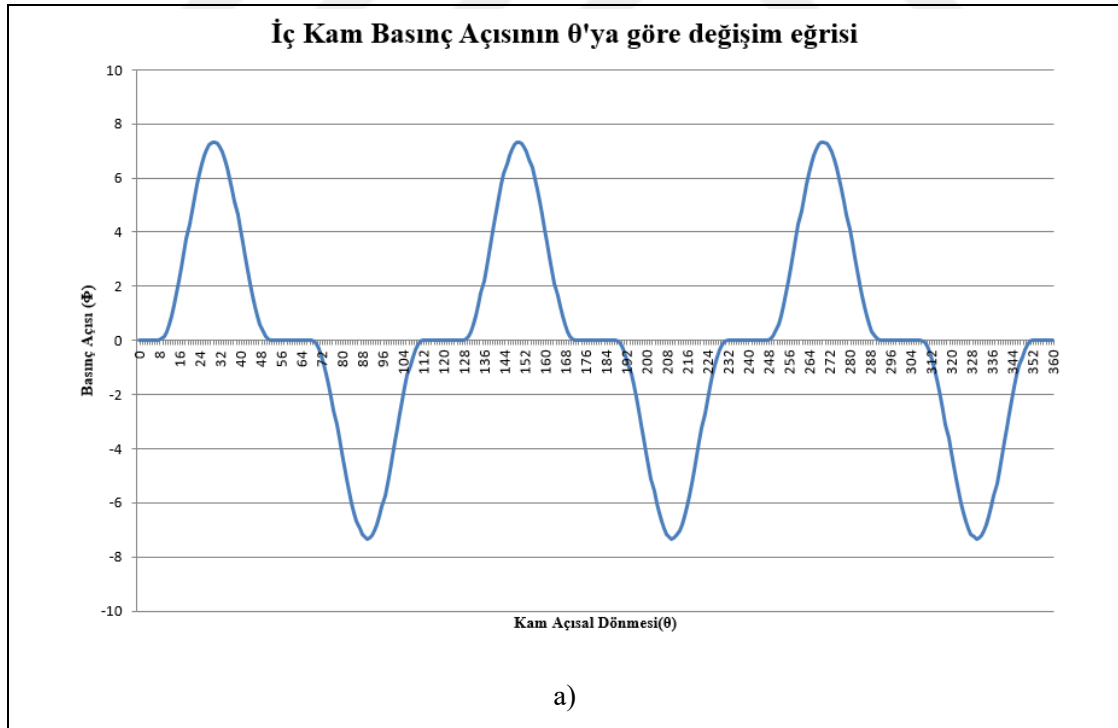
Şekil 4.2. Mekik Ağız Açma Sistemi

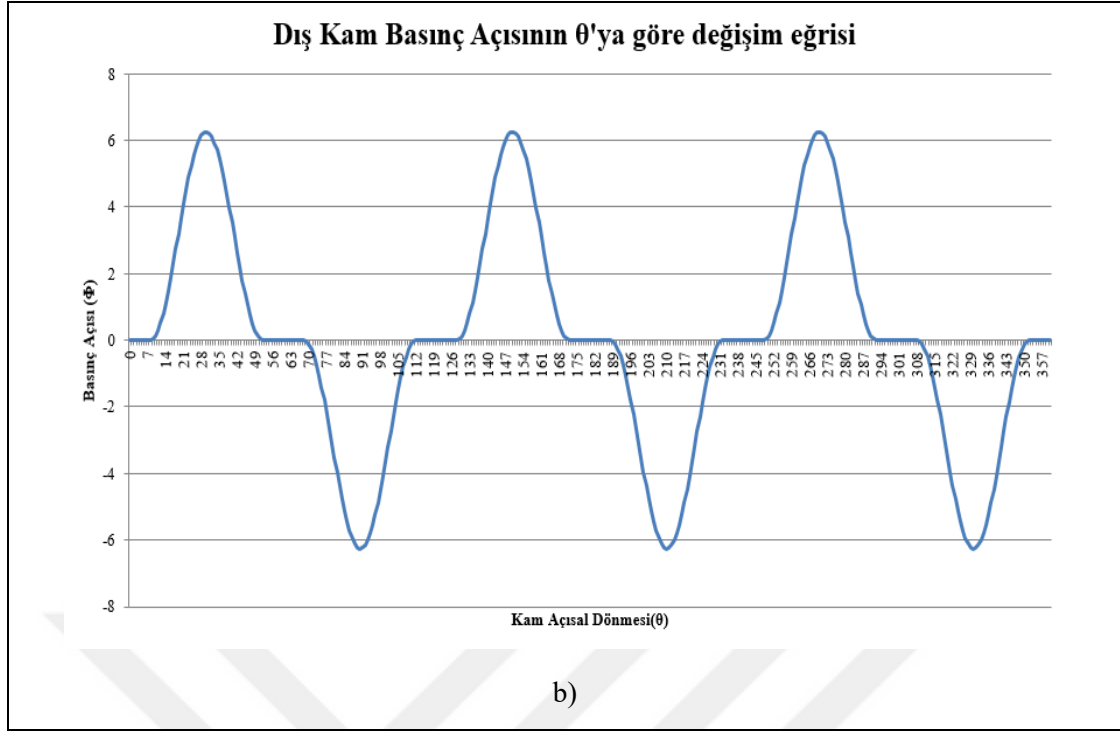
Tasarım parametreleri olarak hareket değişim sayısı 3, kam iç çapı 880 birim, kam kalınlığı 40 birim, izleyici yarıçapı 50 birim, hesaplama hassasiyeti $0,5^\circ$, 3D step dosyası oluşturulurken kullanılacak yükseklik 35 birim ve hesaplama tipi iç ve dış temaslı kam olarak belirlenmiş olup çıkış uzununun (izleyicilerin) kam dönüş açısına bağlı yer değiştirme eğrisi Şekil 4.3’da gösterilmiştir.



Şekil 4.3. Örnek 1 İçin İzleyicinin Kam Açısıl Dönmesine Bağlı Yer Değiştirme Eğrisi

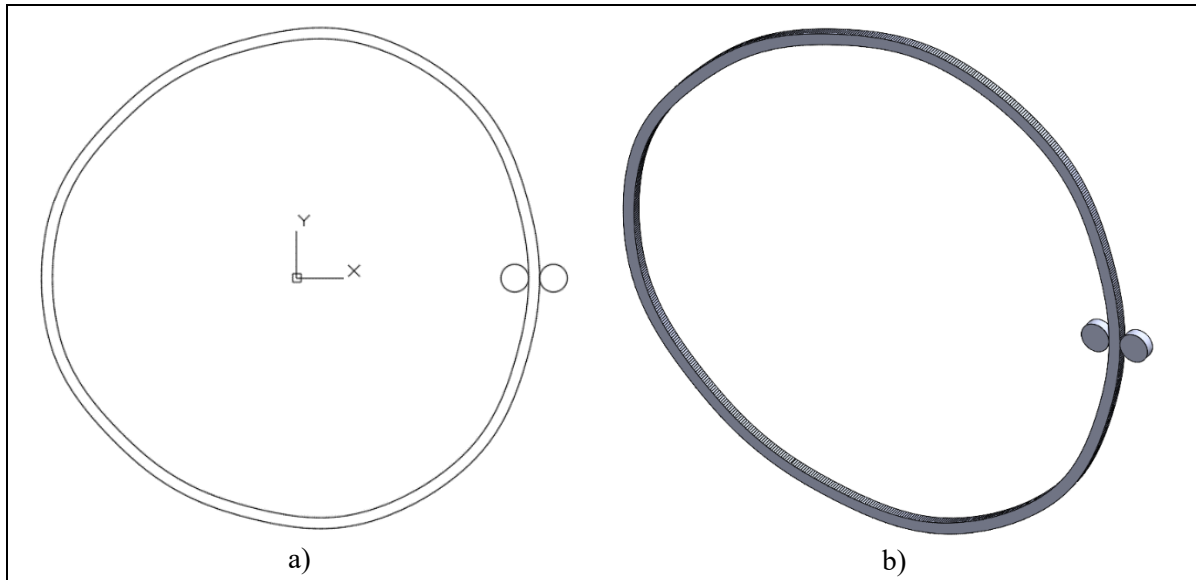
Tasarımda kam profiline ait basınç açılarının 30° den küçük olması hedeflenmiş ve iç-dış izleyicilere ait elde edilen basınç açısı grafikleri Şekil 4.4’de görüldüğü biçimdedir.





Şekil 4.4. Örnek 1'e Ait İç Kam(a) ve Dış Kam (b) Basınç Açıları

Geliştirilen programdan elde edilen kam profiline ait dxf ve step dosyaları sırasıyla Şekil 4.5'de görüldüğü biçimdedir.

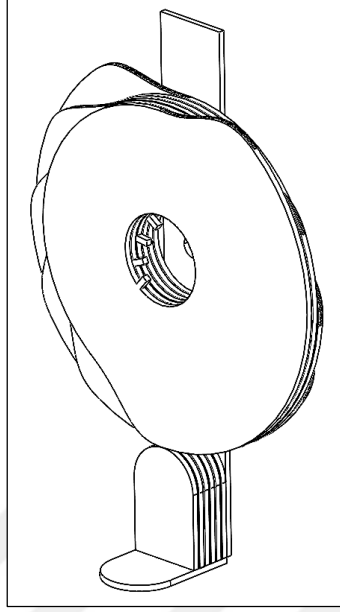


Şekil 4.5. Örnek 1'e Ait Programdan Elde Edilen Dxf (a) ve Step (b) Dosya Görüntüleri

4.1.2. Beklemeli Pim Basma Mekanizması Kam Profili Tasarımı Örnek Uygulaması

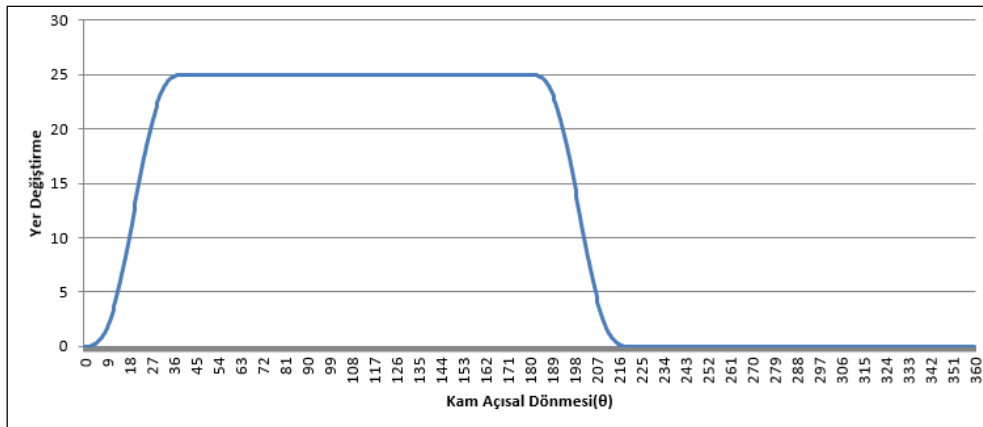
Bu örnek kapsamında özel bir araştırma şirketine ait bir projede ihtiyaç duyulan, Şekil 4.6 gösterilen bekleme pim basma mekanizmasına ait kam tasarımı yapılmıştır. Tasarım ihtiyacı olan

pimlerin sırasıyla 25 birim ilerlemesi ve hepsinin ilerlemiş halde beklemeleri sonucu başlangıç konumlarına dönmesi durumu, eş kamların aynı merkezde 28° döndürülmüş biçimde montajı ile sağlanmıştır.



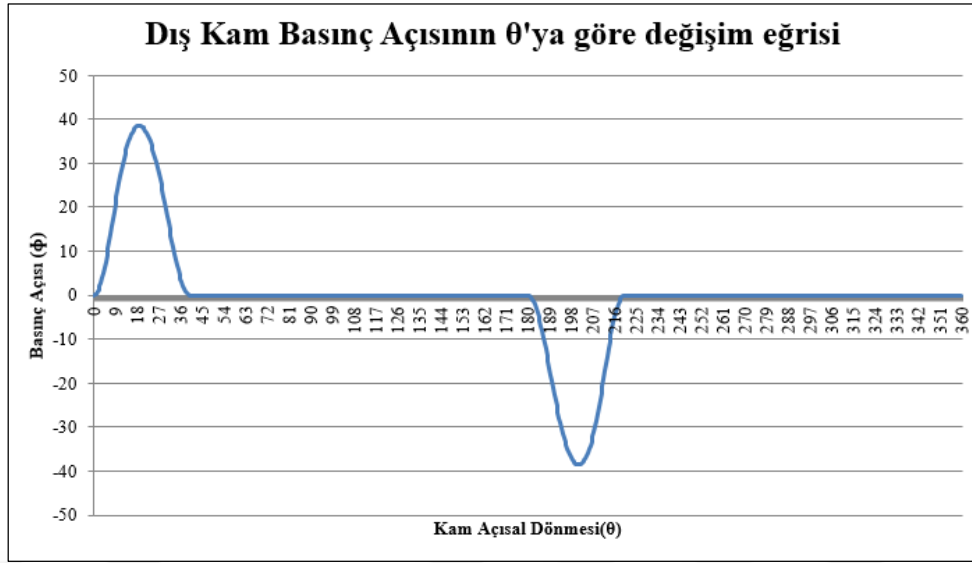
Şekil 4.6. Beklemeli Pim Basma Mekanizması

Eş kamların tasarım parametreleri olarak hareket değişim sayısı 1, kam çapı 100 birim, izleyici yarıçapı 25 birim, hesaplama hassasiyeti $0,5^\circ$, 3D step dosyası oluşturulurken kullanılacak yükseklik 40 birim ve hesaplama tipi dış temaslı kam olarak belirlenmiş olup çıkış uzvunun (izleyicilerin) kam dönüş açısına bağlı yer değiştirme eğrisi Şekil 4.7’de gösterilmiştir.



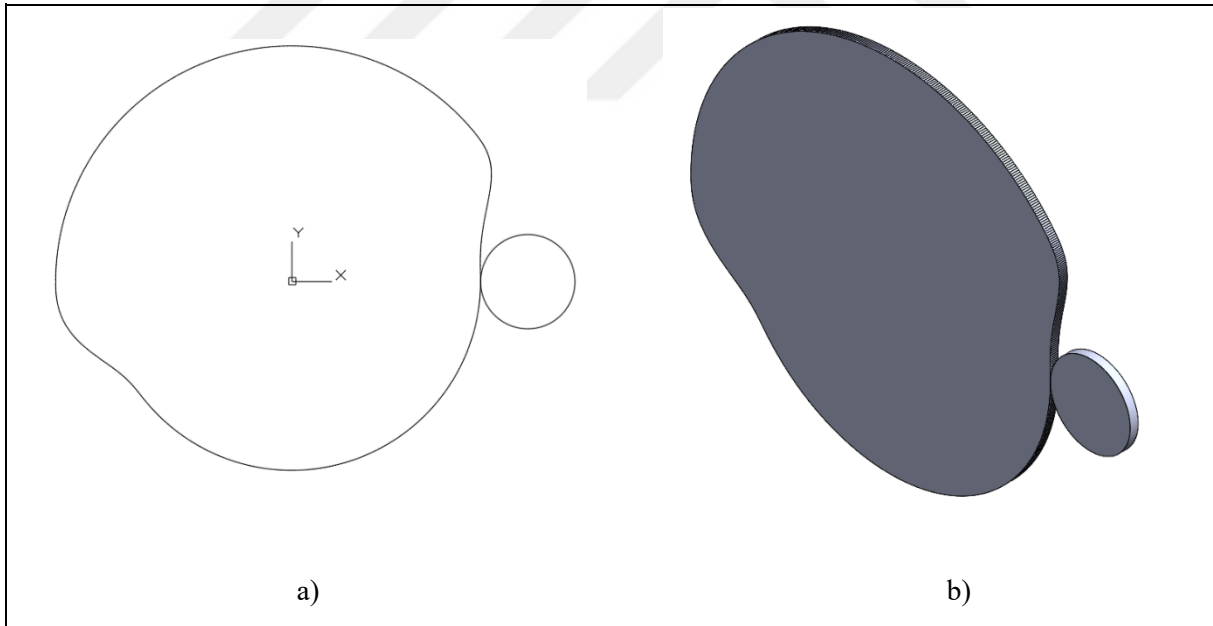
Şekil 4.7. Örnek 2 İçin İzleyicinin Kam Açısal Dönmesine Bağlı Yer Değiştirme Eğrisi

Tasarım sonucu elde edilen kam yapısına ait basınç açısı grafiği Şekil 4.8’de görüldüğü biçimdedir.



Şekil 4.8. Örnek 2'ye ait Dış Kam Basınç Açıları

Programdan elde edilen kam profiline ait dxf ve step dosyaları sırasıyla Şekil 4.9'de görüldüğü biçimdedir.



Şekil 4.9. Örnek 2'ye Ait Programdan Elde Edilen Dxf (a) ve Step (b) Dosya Görüntüleri

4.2. Fonksiyon Sentezi Modülü Örnek Uygulamaları

4.2.1. $y=e^x$ Fonksiyonu Örnek Uygulaması

Fonksiyon sentezi modülü uygulaması birinci örneği olarak $0 \leq x \leq 1$ aralığında $y=e^x$ fonksiyonunu üreten bir dört çubuk mekanizması tasarımı yapılmıştır. $x_6=1$, giriş uzvu dönüş açısı $\Delta\psi=90^\circ$ 'ye karşılık çıkış uzvu dönüş açısı $\Delta\alpha=-90^\circ$ olmak üzere uygulanan üç farklı hesap yöntemine ait diğer tasarım parametreleri şunlardır;

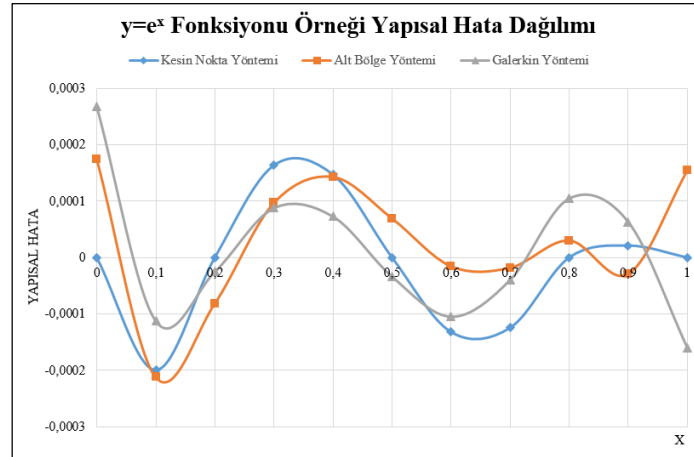
Kesin nokta yönteminde kullanılan kesin noktalar $x_{pi}= 0.0, 0.2, 0.5, 0.8, 1.0$ ($i=1, 2, \dots, 5$) olarak,

Alt bölge yönteminde kullanılan alt bölgeler $x_{si}= 0.0, 0.05, 0.5, 0.7, 0.95, 1.0$ ($i=1, 2, \dots, 6$) olarak,

Galerkin yönteminde kullanılan ağırlık fonksiyonları $w_i= 1, x, x^2, x^3, x^4$ ($i=1, 2, \dots, 5$) olarak yapılan hesaplamalardan bulunan sonuçlar karşılaştırmalı olarak Tablo 4.1'de verilmiştir. Yöntemler arası hata dağılımı karşılaştırmalı olarak Şekil 4.10'da gösterilmiştir.

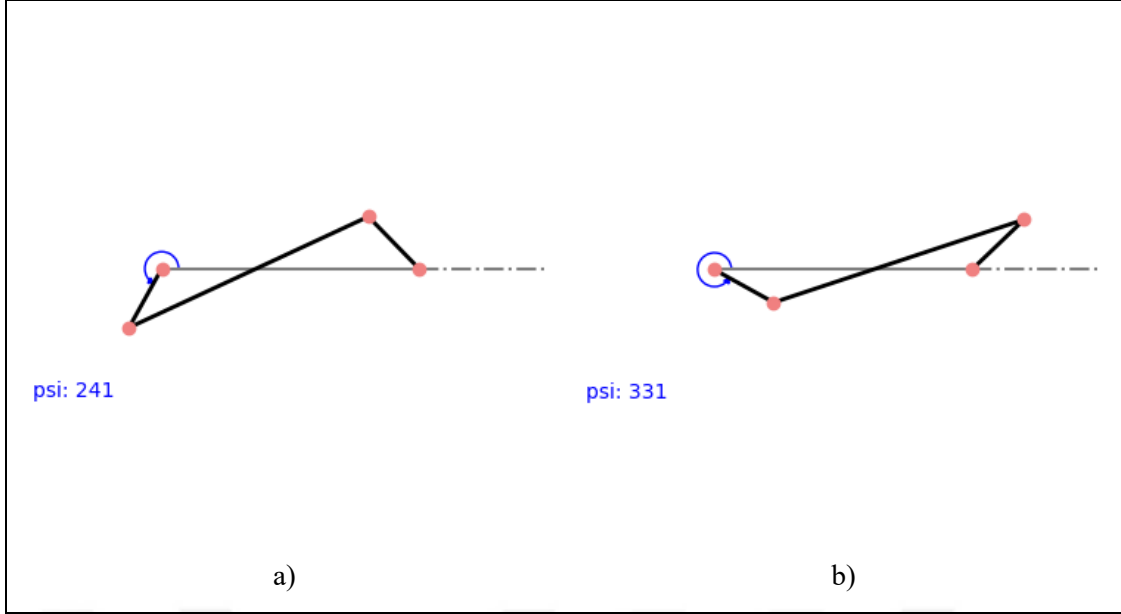
Tablo 4.1. Fonksiyon Sentezi Modülü $y=e^x$ Fonksiyonu Örneği Çözüm Sonuçlarının Karşılaştırılması

Hesap Yöntemi	x_1	x_2	x_3	ψ_0 (°)	α_0 (°)	Max. Yapısal Hata
Kesin Nokta	0.262840	1.026022	-0.282306	-119.432814	-45.530417	-0.00019938
Alt Bölge	0.259429	1.027546	-0.279788	-119.393751	-46.288960	-0.00021224
Galerkin	0.262051	1.026716	-0.282385	-119.209783	-45.936266	0.00026822



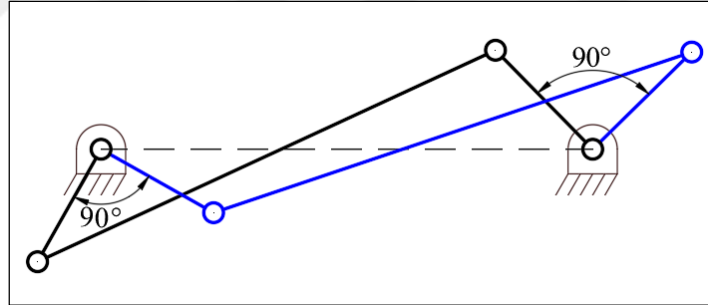
Şekil 4.10. Fonksiyon Sentezi Modülü $y=e^x$ Fonksiyonu Örneğine Ait Karşılaştırmalı Yapısal Hata Dağılımı

Tasarım sonucu hesaplanan mekanizmanın çalışma aralığında uzuvların başlangıç (a) ve bitiş (b) konumları, programdaki animasyon ekranında Şekil 4.11'deki biçimde görülmektedir.



Şekil 4.11. Fonksiyon Sentezi Modülü $y=e^x$ Fonksiyonu Örneğine Ait Tasarım Sonucunun Çalışma Aralığının Başlangıç (a) ve Bitiş (b) Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.12’de gösterilmiştir.



Şekil 4.12. Fonksiyon Sentezi Modülü $y=e^x$ Örneğine Ait Tasarım Sonucunun CAD Programı ile Yapılan Doğrulaması

4.2.2. $y=x^2$ Fonksiyonu Örnek Uygulaması

Fonksiyon sentezi modülü uygulaması ikinci örneği olarak $0 \leq x \leq 1$ aralığında $y=x^2$ fonksiyonunu üreten bir dört çubuk mekanizması tasarımı yapılmıştır. $x_6=1$, giriş uzvu dönüş açısı $\Delta\psi=90^\circ$ ’ye karşılık çıkış uzvu dönüş açısı $\Delta\alpha=60^\circ$ olmak üzere uygulanan üç farklı hesaplama yöntemine ait diğer tasarım parametreleri şunlardır;

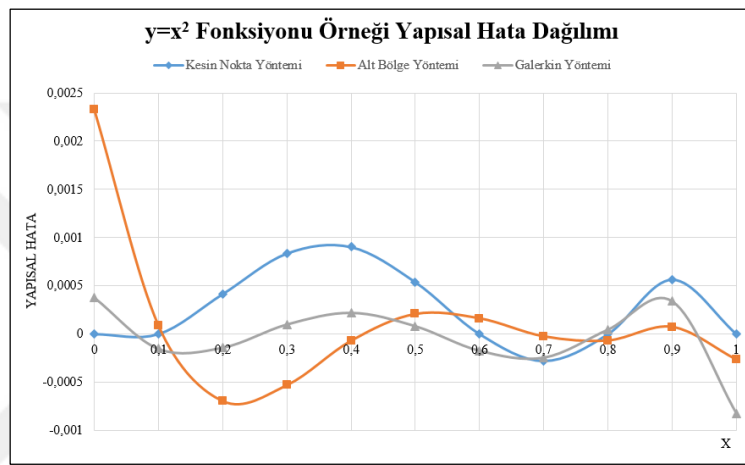
Kesin nokta yönteminde kullanılan kesin noktalar $x_{pi}=0.0, 0.1, 0.6, 0.8, 1.0$ ($i=1, 2, \dots, 5$) olarak,

Alt bölge yönteminde kullanılan alt bölgeler $x_{si}=0.0, 0.3, 0.6, 0.8, 0.9, 1.0$ ($i=1, 2, \dots, 6$) olarak,

Galerkin yönteminde kullanılan ağırlık fonksiyonları $w_i = 1, x, x^2, x^3, x^4$ ($i=1, 2, \dots, 5$) olarak yapılan hesaplamalardan bulunan sonuçlar karşılaştırmalı olarak Tablo 4.2’de verilmiştir. Yöntemler arası hata dağılımı karşılaştırmalı olarak Şekil 4.13’da gösterilmiştir.

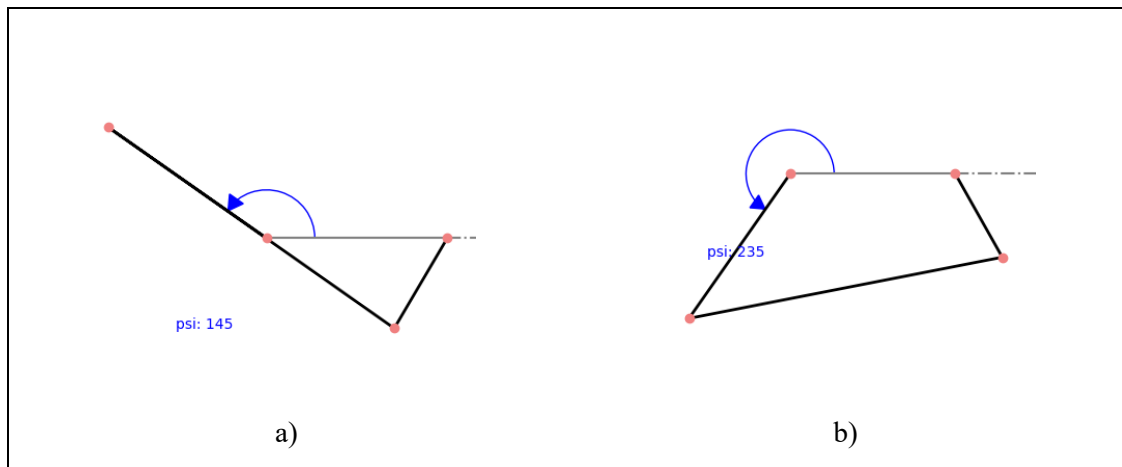
Tablo 4.2. Fonksiyon Sentezi Modülü $y=x^2$ Fonksiyonu Örneği Çözüm Sonuçlarının Karşılaştırılması

Hesap Yöntemi	x_1	x_2	x_3	ψ_0 (°)	α_0 (°)	Max. Yapısal Hata
Kesin Nokta	-0.991534	1.844513	-0.555468	-33.589538	58.469385	0.00090239
Alt Bölge	1.426488	2.248630	-0.662497	140.185798	54.988135	0,00232545
Galerkin	1.067542	1.932398	0.578846	145.136260	-120.449456	-0,00082401



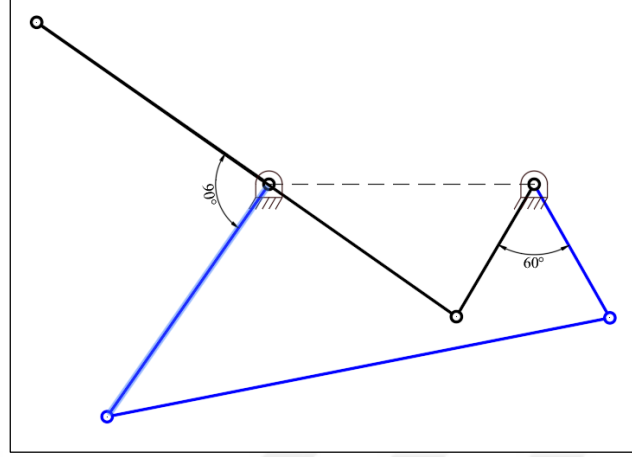
Şekil 4.13. Fonksiyon Sentezi Modülü $y=x^2$ Fonksiyonu Örneğine Ait Karşılaştırmalı Yapısal Hata Dağılımı

Tasarım sonucu hesaplanan mekanizmanın çalışma aralığında uzuvların başlangıç (a) ve bitiş (b) konumları, programdaki animasyon ekranında Şekil 4.14’deki biçimde görülmektedir.



Şekil 4.14. Fonksiyon Sentezi Modülü $y=x^2$ Fonksiyonu Örneğine Ait Tasarım Sonucunun Çalışma Aralığının Başlangıç (a) ve Bitiş (b) Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.15’de gösterilmiştir.



Şekil 4.15. Fonksiyon Sentezi Modülü $y=x^2$ Örneğine Ait Tasarım Sonucunun CAD Programı ile Yapılan Doğrulaması

4.2.3. $y=\sin(x)$ Fonksiyonu Örnek Uygulaması

Fonksiyon sentezi modülü uygulaması üçüncü örneği olarak $0^\circ \leq x \leq 90^\circ$ aralığında $y=\sin(x)$ fonksiyonunu üreten bir dört çubuk mekanizması tasarımı yapılmıştır. $x_6=1$, giriş uzvu dönüş açısı $\Delta\psi=90^\circ$ ’ye karşılık çıkış uzvu dönüş açısı $\Delta\alpha=90^\circ$ olmak üzere uygulanan üç farklı hesap yöntemine ait diğer tasarım parametreleri şunlardır;

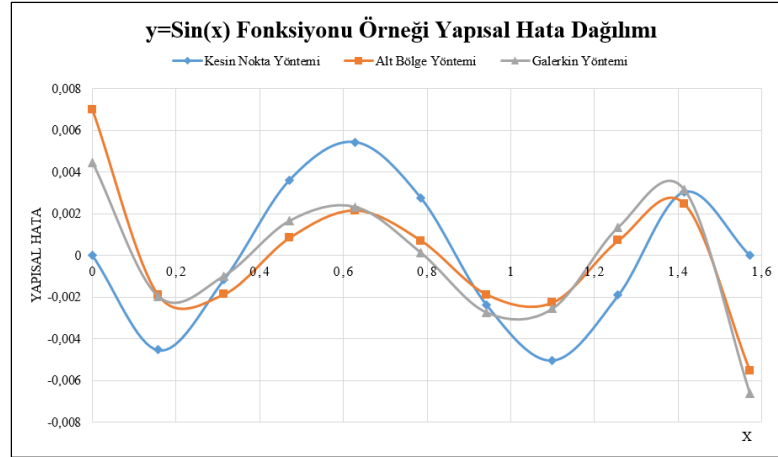
Kesin nokta yönteminde kullanılan kesin noktalar $x_{pi}=0.0, 0.35, 0.87, 1.31, 1.57$ ($i=1, 2, \dots, 5$) olarak,

Alt bölge yönteminde kullanılan alt bölgeler $x_{si}=0.0, 0.26, 0.61, 1.05, 1.39, 1.57$ ($i=1, 2, \dots, 6$) olarak,

Galerkin yönteminde kullanılan ağırlık fonksiyonları $w_i=1, x, x^2, x^3, x^4$ ($i=1, 2, \dots, 5$) olarak yapılan hesaplamalardan bulunan sonuçlar karşılaştırmalı olarak Tablo 4.3’de verilmiştir. Yöntemler arası hata dağılımı karşılaştırmalı olarak Şekil 4.16’de gösterilmiştir.

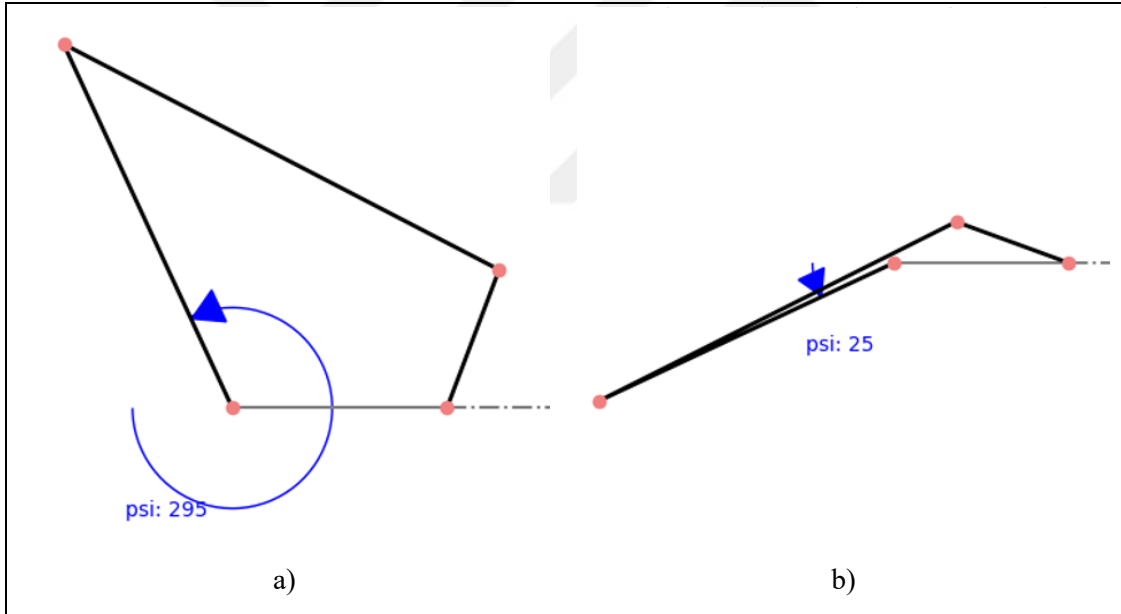
Tablo 4.3. Fonksiyon Sentezi Modülü $y=\sin(x)$ Fonksiyonu Örneği Çözüm Sonuçlarının Karşılaştırılması

Hesap Yöntemi	x_1	x_2	x_3	ψ_0 (°)	α_0 (°)	Max. Yapısal Hata
Kesin Nokta	-1.639879	2.062290	0.662115	-66.100988	71.918167	0,00542252
Alt Bölge	-1.861324	2.283709	0.685388	-64.940224	69.704334	0,00699312
Galerkin	-2.123757	2.476895	0.750987	-63.638874	72.395602	0,00446947



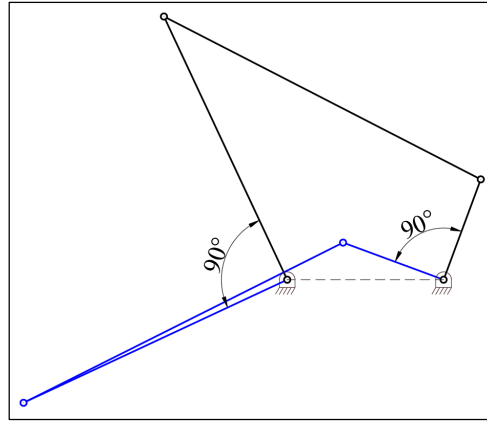
Şekil 4.16. Fonksiyon Sentezi Modülü $y=\sin(x)$ Fonksiyonu Örneğine Ait Karşılaştırmalı Yapısal Hata Dağılımı

Tasarım sonucu hesaplanan mekanizmanın çalışma aralığında uzuvların başlangıç (a) ve bitiş (b) konumları, programdaki animasyon ekranında Şekil 4.17'deki biçimde görülmektedir.



Şekil 4.17. Fonksiyon Sentezi Modülü $y=\sin(x)$ Fonksiyonu Örneğine Ait Tasarım Sonucunun Çalışma Aralığının Başlangıç (a) ve Bitiş (b) Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.18'de gösterilmiştir.



Şekil 4.18. Fonksiyon Sentezi Modülü $y=\sin(x)$ Örneğine Ait Tasarım Sonucunun CAD Programı ile Yapılan Doğrulaması

4.3. Kol Sarkaç Modülü Örnek Uygulamaları

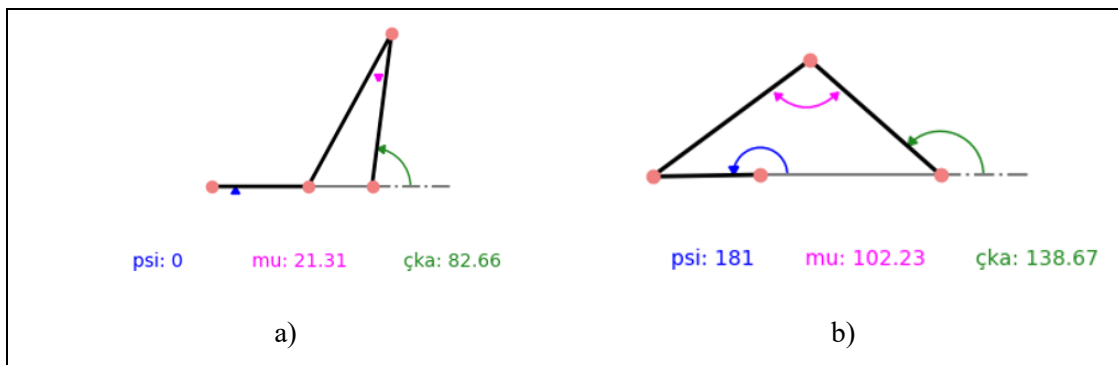
4.3.1. Durum 1 (α , ZO ve β kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 1. duruma ait uygulama olarak salınım miktarı $\alpha = 90^\circ$, ilk sınır konum açısı $\beta = 30^\circ$ ve zaman oranı $ZO = 1,6$ ($\eta = 41,53^\circ$) olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.4’da verilmiştir.

Tablo 4.4. Durum 1 Örnek Uygulaması Tasarım Sonuçları

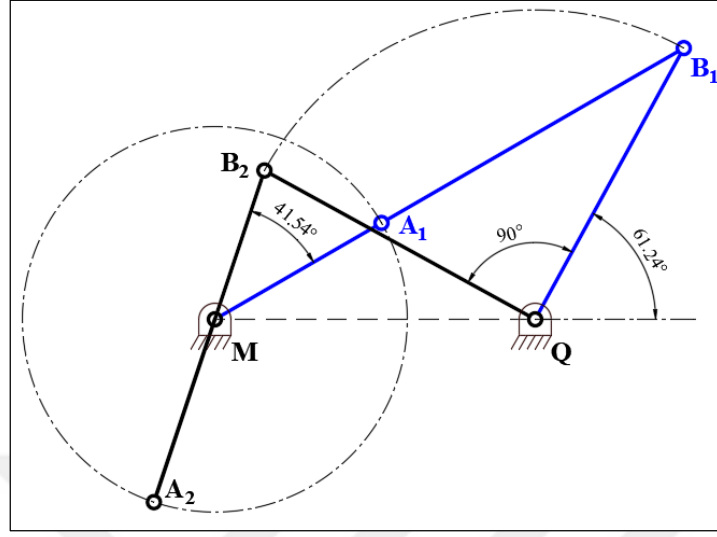
Çözüm No	Φ (°)	u/p (birim)	r/p (birim)	q/p (birim)
1	61,24088	0,96407	1,08965	0,60065

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.19’daki biçimde görülmektedir.



Şekil 4.19. Kol Sarkaç Modülü Durum 1’e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.20’de gösterilmiştir.



Şekil 4.20. Kol Sarkaç Modülü Durum 1’e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

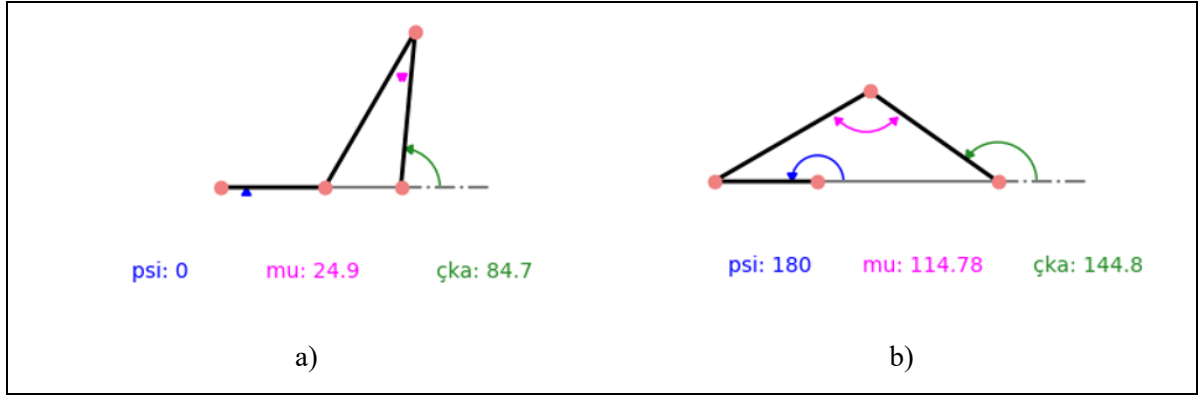
4.3.2. Durum 2 (α , ZO ve Φ kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 2. duruma ait uygulama olarak salınım miktarı $\alpha = 90^\circ$, ilk sınır konum çıkış kolu açısı $\Phi = 65^\circ$ ve zaman oranı $ZO = 1,4$ ($\eta = 30^\circ$) olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.5’de verilmiştir.

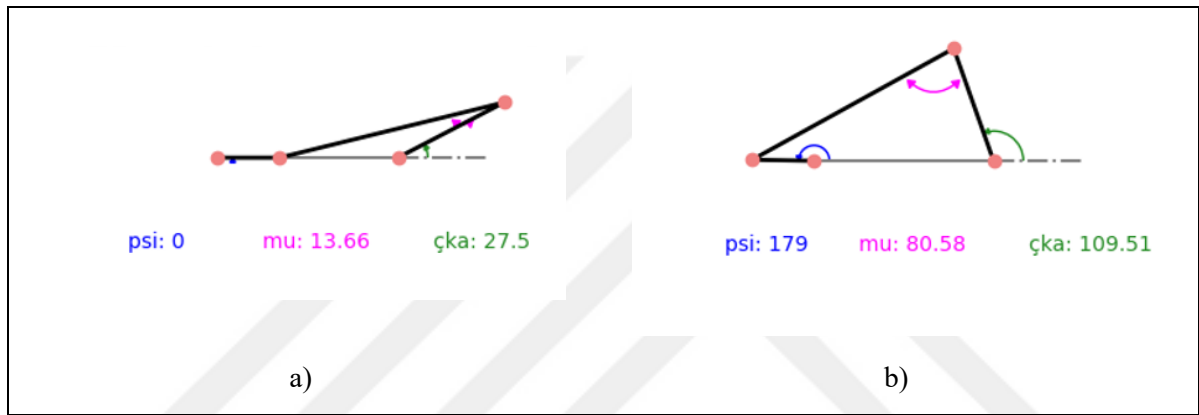
Tablo 4.5. Durum 2 Örnek Uygulaması Tasarım Sonuçları

Çözüm No	$\beta(^{\circ})$	u/p (birim)	r/p (birim)	q/p (birim)
1	-150.05890	0.86890	-1.00103	-0.57676
2	-39.94110	-0.66446	1.28233	-0.34431
3	29.94110	0.86890	1.00103	0.57676
4	140.05890	-0.66446	-1.28233	0.34431

Tasarım sonuçlarından boyutları farklı olan iki mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.21 ve Şekil 4.22’deki biçimde görülmektedir.

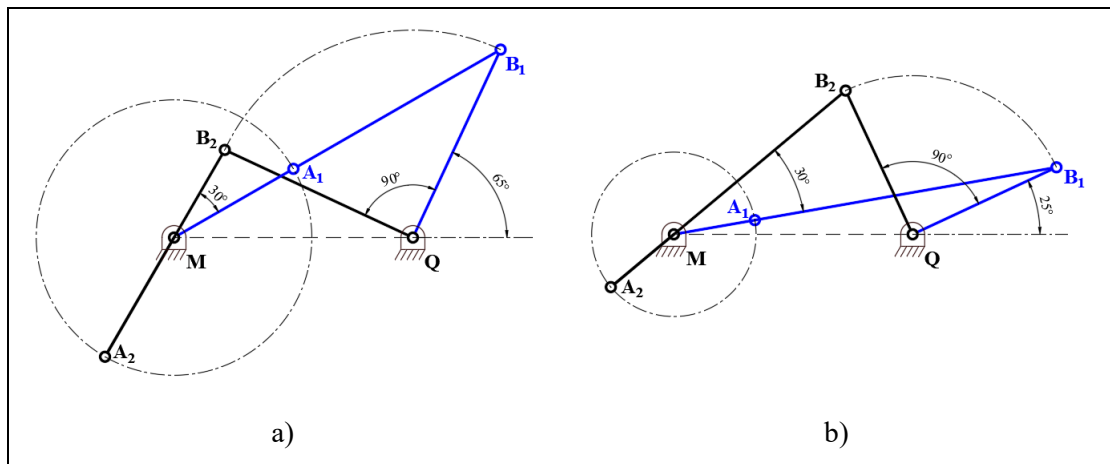


Şekil 4.21. Kol Sarkaç Modülü Durum 2'ye Ait Tasarım Sonuçlarından Birinci Çözümün Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları



Şekil 4.22. Kol Sarkaç Modülü Durum 2'ye Ait Tasarım Sonuçlarından İkinci Çözümün Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen iki farklı mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması sırasıyla Şekil 4.23'de gösterilmiştir.



Şekil 4.23. Kol Sarkaç Modülü Durum 2'ye Ait Tasarım Sonuçlarından Birinci (a) ve İkinci (b) Çözümün İlk ve Son Sınır Konumları

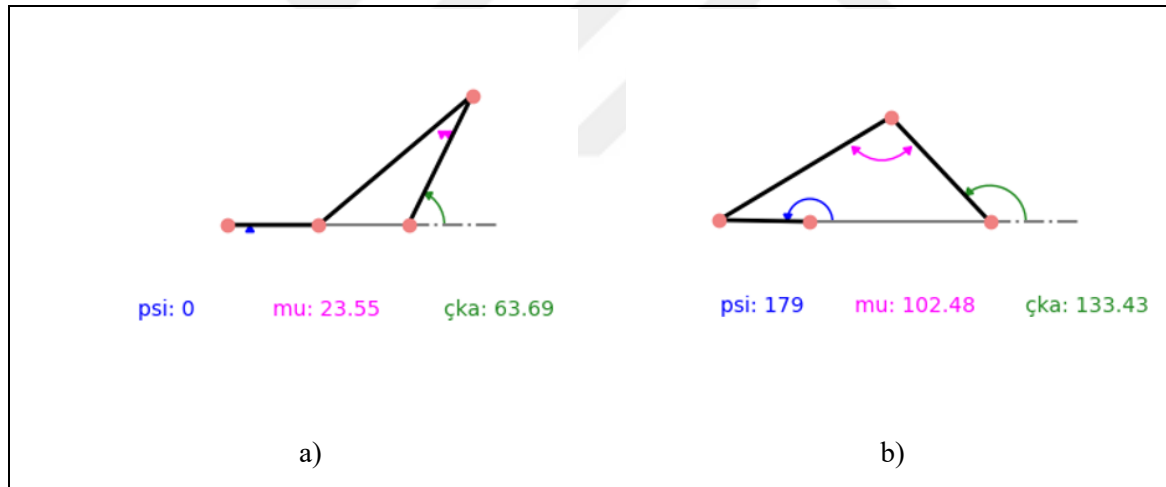
4.3.3. Durum 3 (α , ZO ve u/p kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 3. duruma ait uygulama olarak salınım miktarı $\alpha = 90^\circ$, zaman oranı $ZO = 1,4$ ($\eta = 30^\circ$) ve çıkış uzvunun gövde uzvuna oranı $u/p = 0,8$ olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.6’de verilmiştir.

Tablo 4.6. Durum 3 Örnek Uygulaması Tasarım Sonuçları

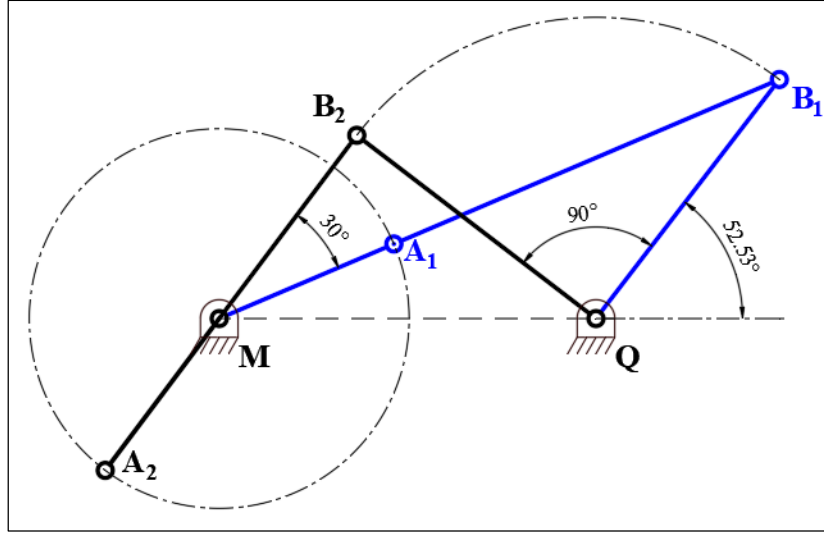
Çözüm No	$\beta(^{\circ})$	$\Phi(^{\circ})$	r/p (birim)	q/p (birim)
1	-156.87405	52.52847	-1.11250	-0.50410
2	-53.12595	37.47153	1.11250	-0.50410
3	23.12595	52.52847	1.11250	0.50410
4	126.87405	37.47153	-1.11250	0.50410

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.24’deki biçimde görülmektedir.



Şekil 4.24. Kol Sarkaç Modülü Durum 3’e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.25’de gösterilmiştir.



Şekil 4.25. Kol Sarkaç Modülü Durum 3'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

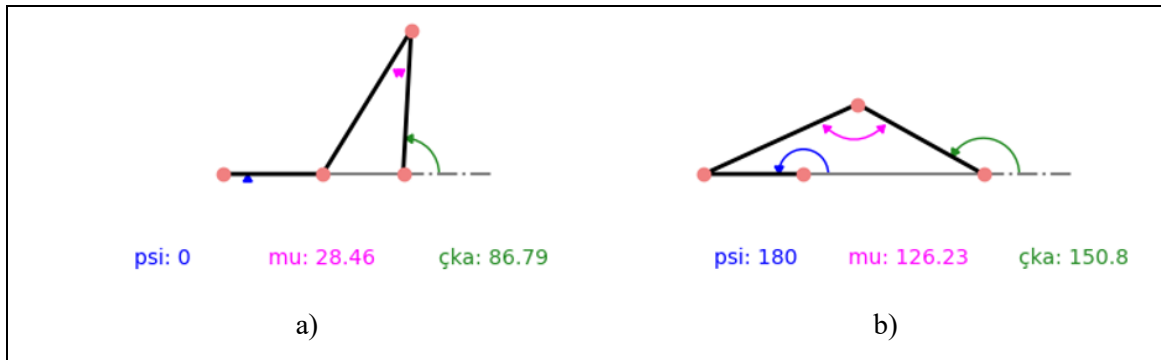
4.3.4. Durum 4 (α , β ve u/p kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 4. duruma ait uygulama olarak salınım miktarı $\alpha = 90^\circ$, ilk sınır konum açısı $\beta = 30^\circ$ ve çıkış uzvunun gövde uzvuna oranı $u/p=0,8$ olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.7'de verilmiştir.

Tablo 4.7. Durum 4 Örnek Uygulaması Tasarım Sonuçları

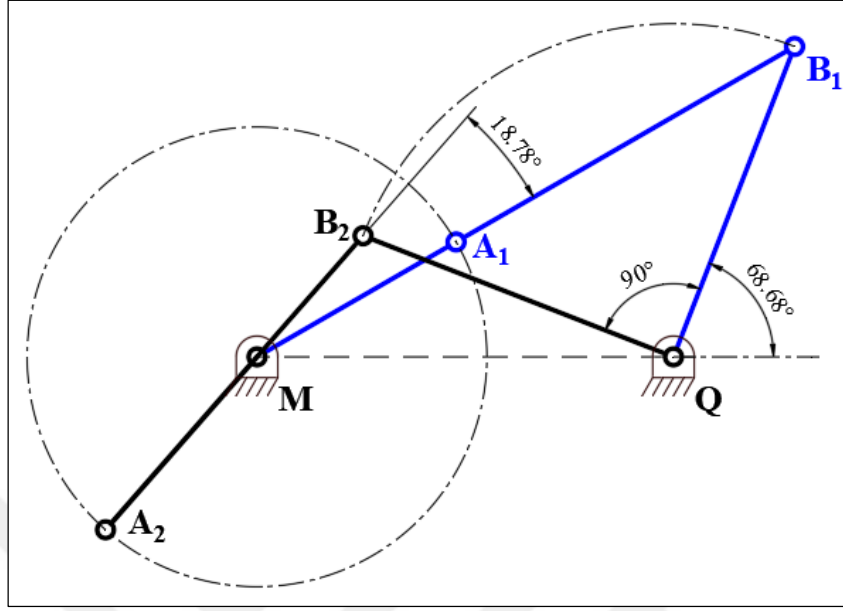
Çözüm No	Φ (°)	u/p (birim)	r/p (birim)	q/p (birim)
1	68,68219	0,8	0,93857	0,55195

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.26'deki biçimde görülmektedir.



Şekil 4.26. Kol Sarkaç Modülü Durum 4'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.27’te gösterilmiştir.



Şekil 4.27. Kol Sarkaç Modülü Durum 4’e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

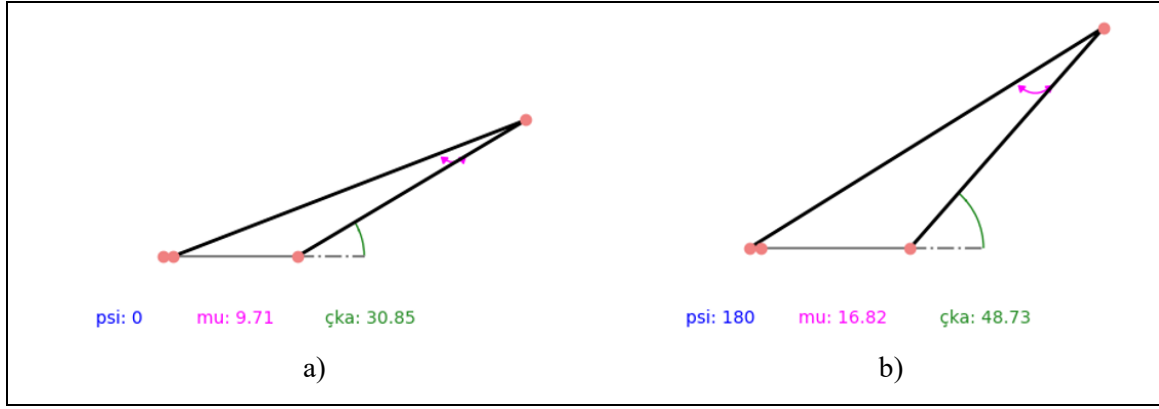
4.3.5. Durum 5 (α , β ve Φ kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 5. duruma ait uygulama olarak salınım miktarı $\alpha = 20^\circ$, ilk sınır konum açısı $\beta = 20^\circ$ ve ilk sınır konum çıkış uzvu açısı $\Phi = 30^\circ$ olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.8’de verilmiştir.

Tablo 4.8. Durum 5 Örnek Uygulaması Tasarım Sonuçları

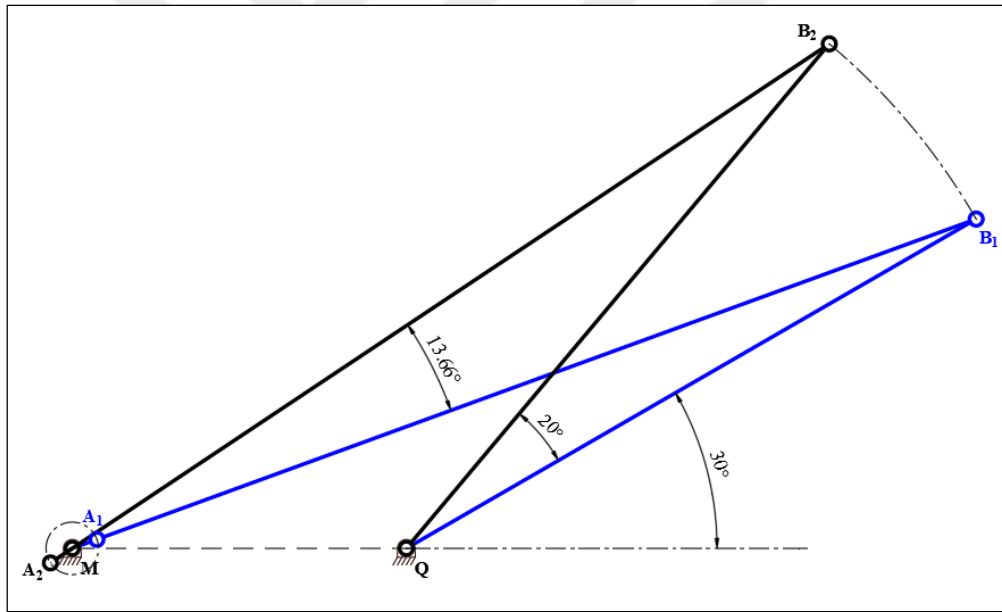
Çözüm No	η (°)	u/p (birim)	r/p (birim)	q/p (birim)
1	13,65709	1,96962	2,80089	0,07849

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının oluştuğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.28’teki biçimde görülmektedir.



Şekil 4.28. Kol Sarkaç Modülü Durum 5'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.29'da gösterilmiştir.



Şekil 4.29. Kol Sarkaç Modülü Durum 5'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

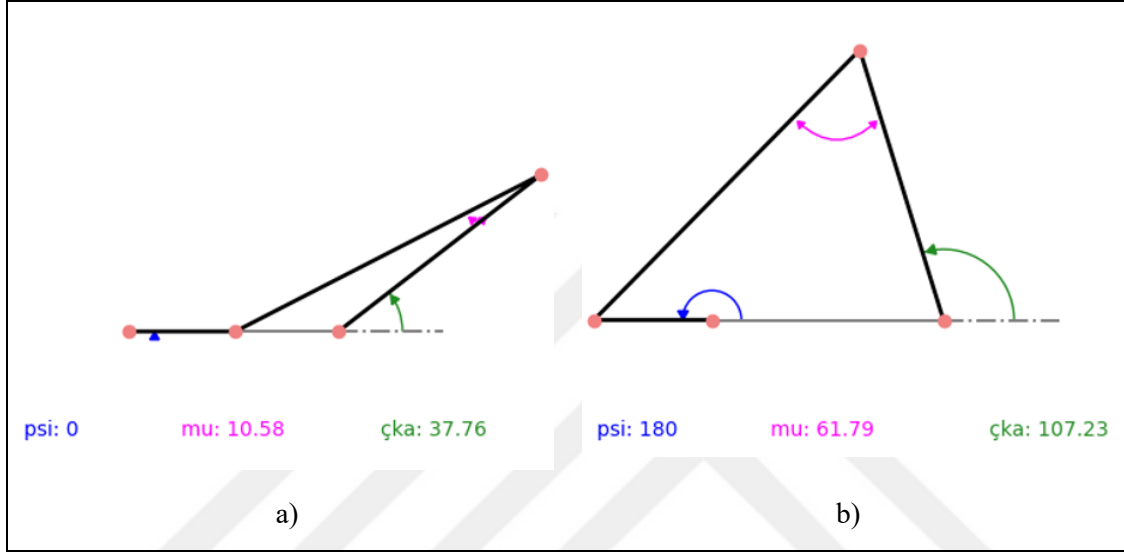
4.3.6. Durum 6 (α , Φ ve u/p kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 6. duruma ait uygulama olarak salınım miktarı $\alpha = 90^\circ$, ilk sınır konum çıkış uzvu açısı $\Phi = 30^\circ$ ve çıkış uzvunun gövde uzvuna oranı $u/p = 1,22056$ olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.9'de verilmiştir.

Tablo 4.9. Durum 6 Örnek Uygulaması Tasarım Sonuçları

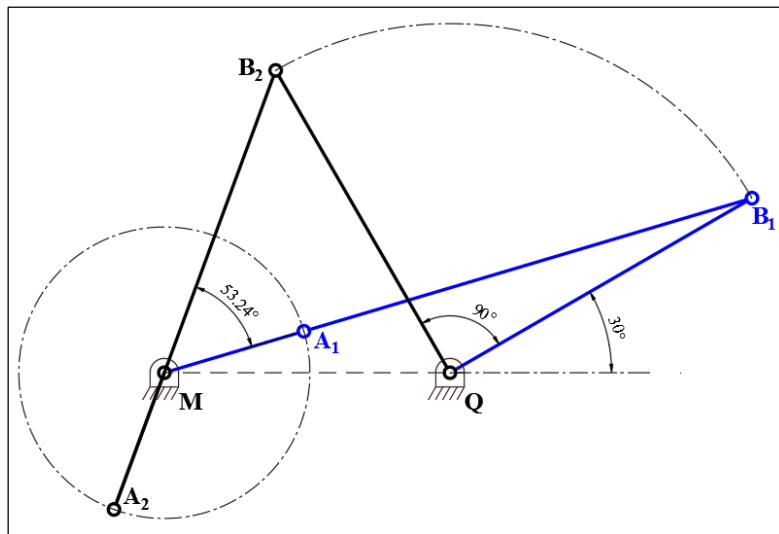
Çözüm No	$\beta(^{\circ})$	$\eta(^{\circ})$	u/p (birim)	r/p (birim)	q/p (birim)
1	16,52453	53,23697	1,22056	1,63612	0,50953

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.30'deki biçimde görülmektedir.



Şekil 4.30. Kol Sarkaç Modülü Durum 6'ya Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.31'de gösterilmiştir.



Şekil 4.31. Kol Sarkaç Modülü Durum 6'ya Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

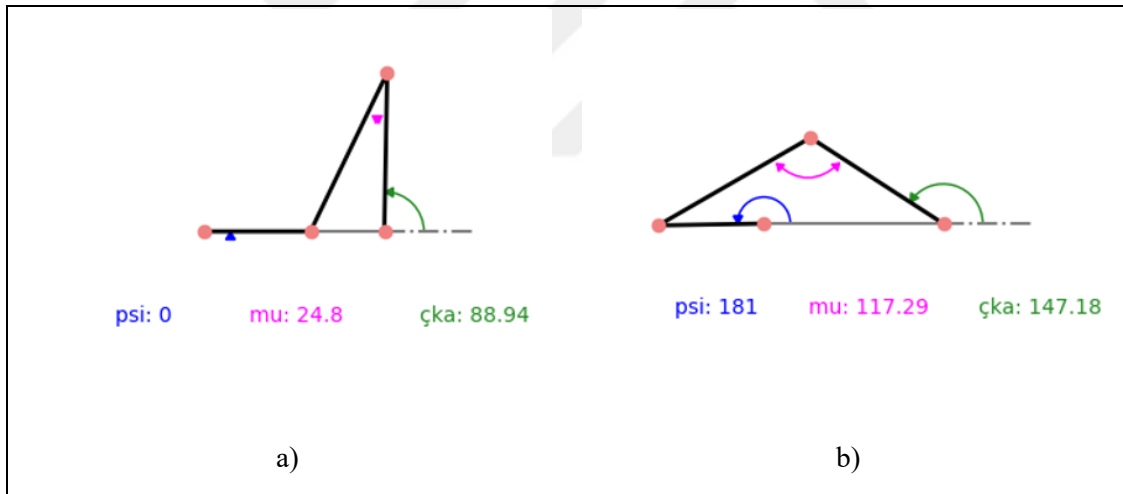
4.3.7. Durum 7 (α , ZO ve u/r kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 7. duruma ait uygulama olarak salınım miktarı $\alpha = 90^\circ$, zaman oranı $ZO = 1,4$ ($\eta = 30^\circ$) ve çıkış uzvunun biyel uzvuna oranı $u/p = 0,9$ olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.10'de verilmiştir.

Tablo 4.10. Durum 7 Örnek Uygulaması Tasarım Sonuçları

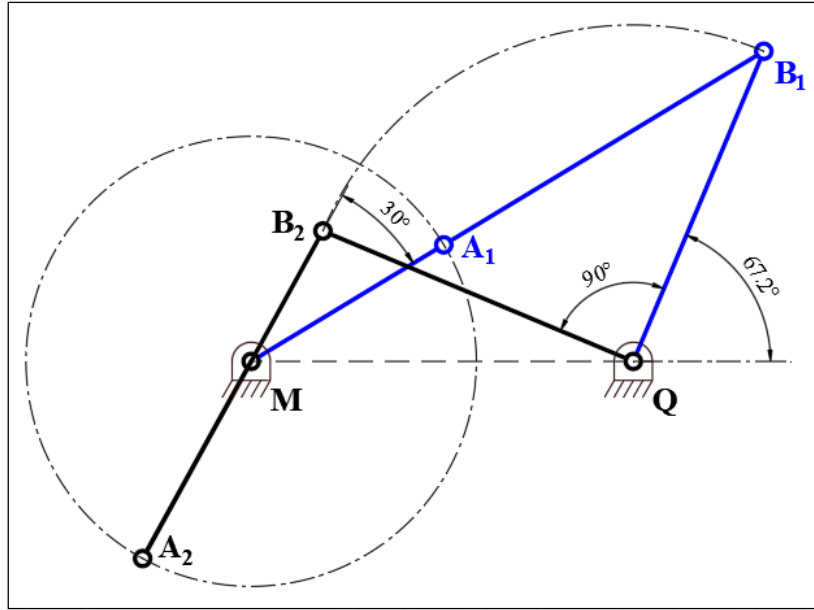
Çözüm No	$\beta(^{\circ})$	$\Phi(^{\circ})$	r/p (birim)	q/p (birim)
1	-148.80749	67.19513	-0.97897	-0.58924
2	-61.19251	22.80487	0.97897	-0.58924
3	31.19251	67.19513	0.97897	0.58924
4	118.80749	22.80487	-0.97897	0.58924

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.32'teki biçimde görülmektedir.



Şekil 4.32. Kol Sarkaç Modülü Durum 7'ye Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.33'de gösterilmiştir.



Şekil 4.33. Kol Sarkaç Modülü Durum 7'ye Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

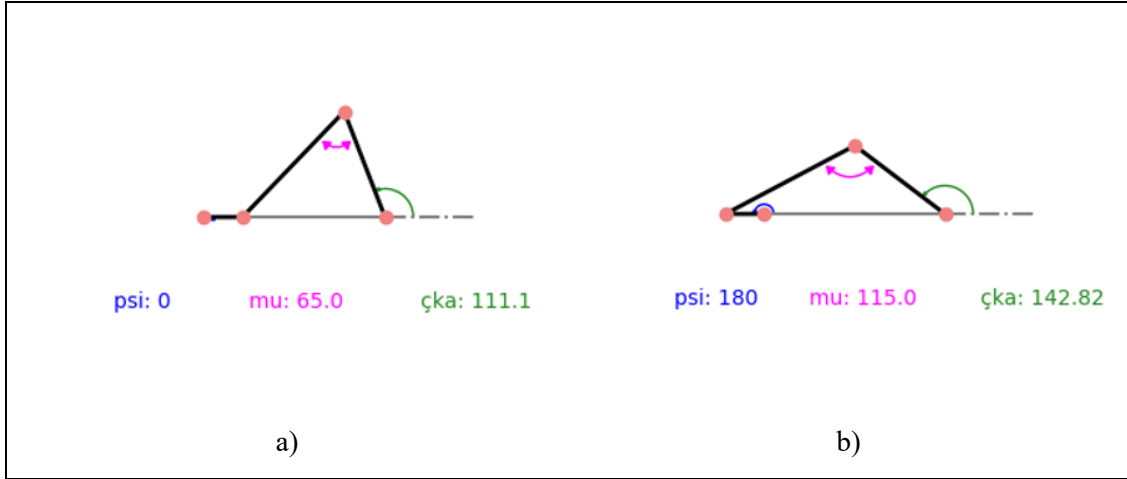
4.3.8. Durum 8 (Bağlama açısı kullanılarak) Örnek Uygulaması

Kol sarkaç modülü tasarım tiplerinden 8. duruma ait uygulama olarak salınım miktarı $\alpha = 40^\circ$, minimum bağlama açısı $\mu_{\min} = 115^\circ$ ve maksimum bağlama açısı $\mu_{\max} = 65^\circ$ olduğu durumda programdan elde edilen tasarım sonuçları Tablo 4.11'de verilmiştir.

Tablo 4.11. Durum 8 Örnek Uygulaması Tasarım Sonuçları

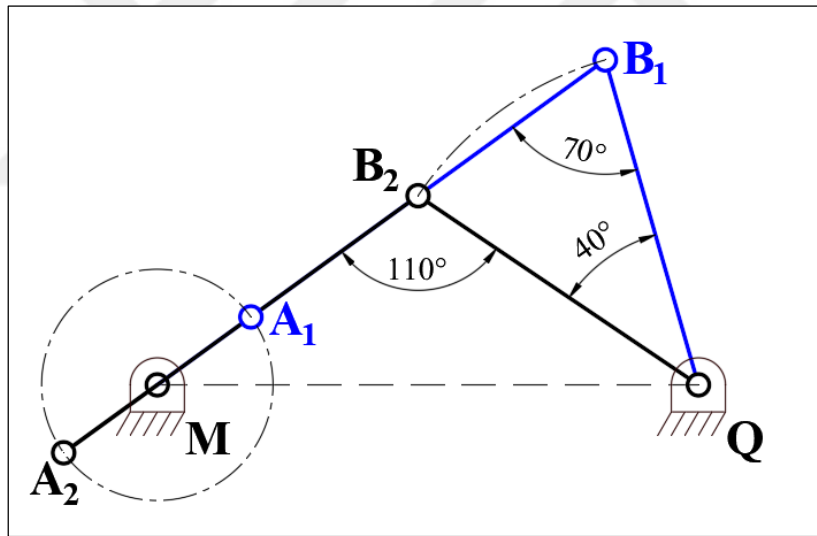
Çözüm No	$\beta(^{\circ})$	$\Phi(^{\circ})$	$\eta(^{\circ})$	u/p (birim)	r/p (birim)	q/p (birim)
1	-35,97352	1,00000	0,00000	-0,62511	0,80929	-0,21380
2	144,02648	1,00000	0,00000	-0,62511	-0,80929	0,21380
3	-144,02648	1,00000	0,00000	0,62511	-0,80929	-0,21380
4	35,97352	1,00000	0,00000	0,62511	0,80929	0,21380

Tasarım sonucu hesaplanan mekanizmanın minimum (a) ve maksimum (b) bağlama açılarının olduğu giriş uzvu konumları, programdaki animasyon ekranında Şekil 4.34'teki biçimde görülmektedir.



Şekil 4.34. Kol Sarkaç Modülü Durum 8'e Ait Tasarım Sonucunun Minimum (a) ve Maksimum (b) Bağlama Açılarında Giriş Uzvu Konumları

Tasarım sonucu elde edilen mekanizmanın ilk ve son sınır konumlarının CAD programı ile yapılan doğrulaması Şekil 4.35'de gösterilmiştir.



Şekil 4.35. Kol Sarkaç Modülü Durum 8'e Ait Tasarım Sonucunun İlk ve Son Sınır Konumları

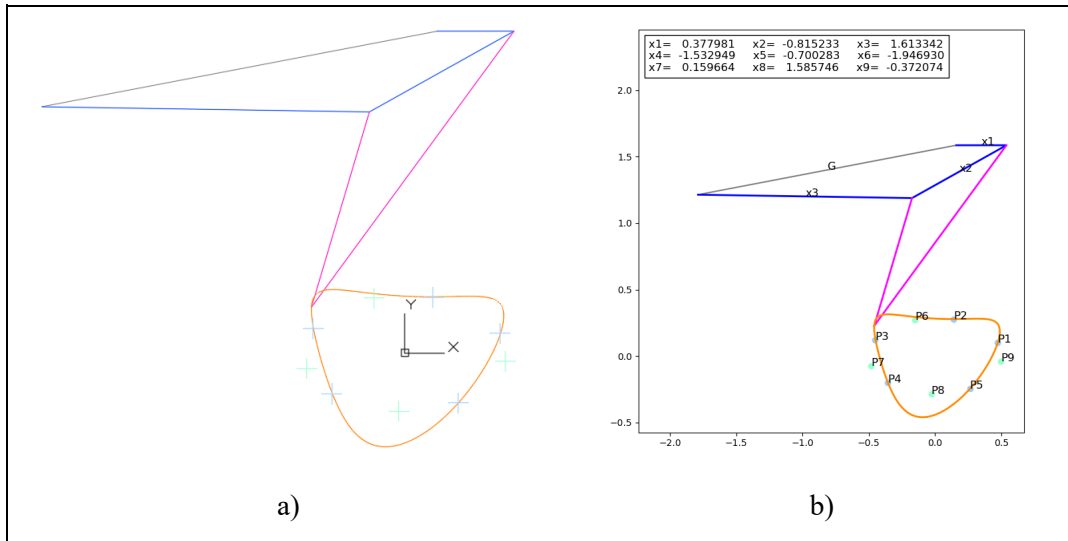
4.4. Yörünge Sentezi Modülü Örnek Uygulaması

Yörünge sentezi modülü uygulaması olarak 9 kesin noktadan geçen tam bir elips yörünge elde edilmesi amaçlanmıştır. Ön tasarım ile mekanizmanın ilk boyutlarının belirlenmesi amacıyla Şekil 3.18 esas alınarak kullanılacak ilk beş noktanın x ve y koordinatları ve giriş uzvu açıları sıralı biçimde şunlardır;

$$\begin{aligned}
 P_1 &= (0.46913089, 0.09911465), \\
 P_2 &= (0.13773281, 0.27543459), \\
 P_4 &= (-0.45201935, 0.12224730), \\
 P_6 &= (-0.35836451, -0.19980507), \\
 P_8 &= (0.26317489, -0.24361863), \\
 (\psi'_i, i=1, 2, \dots, 5) &= (0^\circ, -85^\circ, -185^\circ, -220^\circ, -320^\circ)
 \end{aligned}$$

Yukarıda verilen değerler kullanılarak programdan elde edilen ve excel dosya tipinde kaydedilen ön tasarım sonuçlarından uygun görülen bir tanesinin kinematik boyutları aşağıda verilmiştir. Mekanizmanın ön tasarım aşamasında verilen 5 noktadan geçtiği görülmekte, optimizasyon öncesi yörüngesi dxf formatında Şekil 4.36(a)'da, animasyon ekran görüntüsü Şekil 4.36(b)'de görüldüğü biçimdedir.

$$\begin{aligned}
 x_1 &= 0.377971 \\
 x_2 &= -0.815233 \\
 x_3 &= 1.613342 \\
 x_4 &= -1.532949 \\
 x_5 &= -0.700283 \\
 x_6 &= -1.946930 \\
 x_7 &= 0.159664 \\
 x_8 &= 1.585746 \\
 x_9 &= -0.372074 \\
 \psi_0 &= 171.450737
 \end{aligned}$$



Şekil 4.36. Yörünge Sentezi Modülü Örneği Optimizasyon Öncesi Sonuçları

Ön tasarım sonuçlarına ek olarak elips yörüngesine ait ilave dört nokta aşağıda verilmiş olup optimizasyon aşamasında kullanılarak mekanizmanın son boyutları hesaplanmıştır.

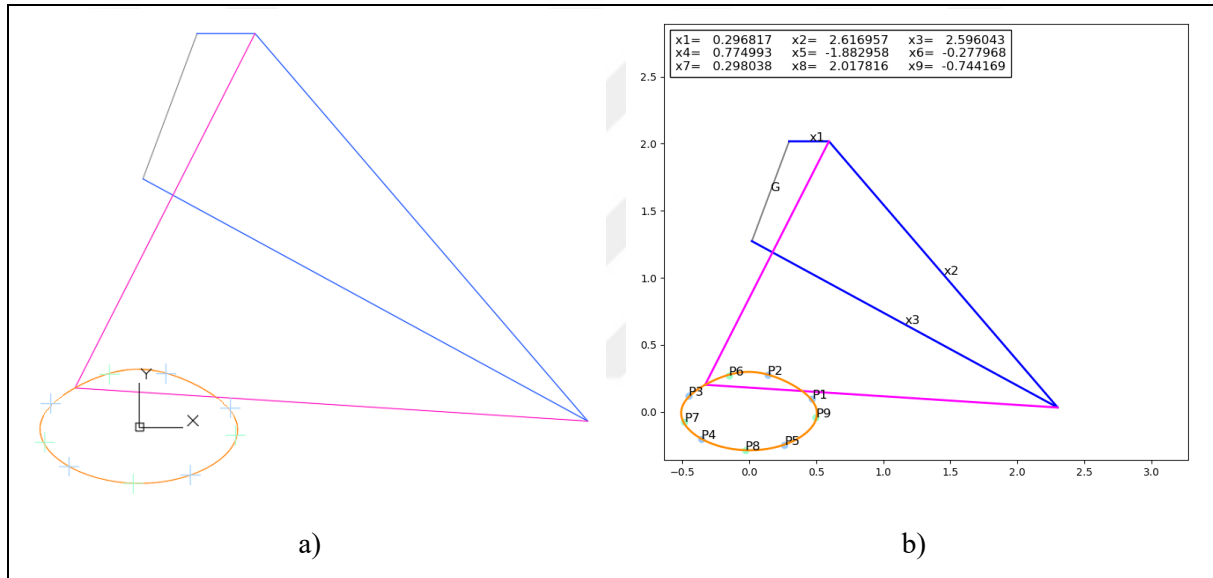
$$P_3 = (-0.15142466, 0.27306436),$$

$$P_5 = (-0.48248748, -0.07516617),$$

$$P_7 = (-0.02953483, -0.28601947),$$

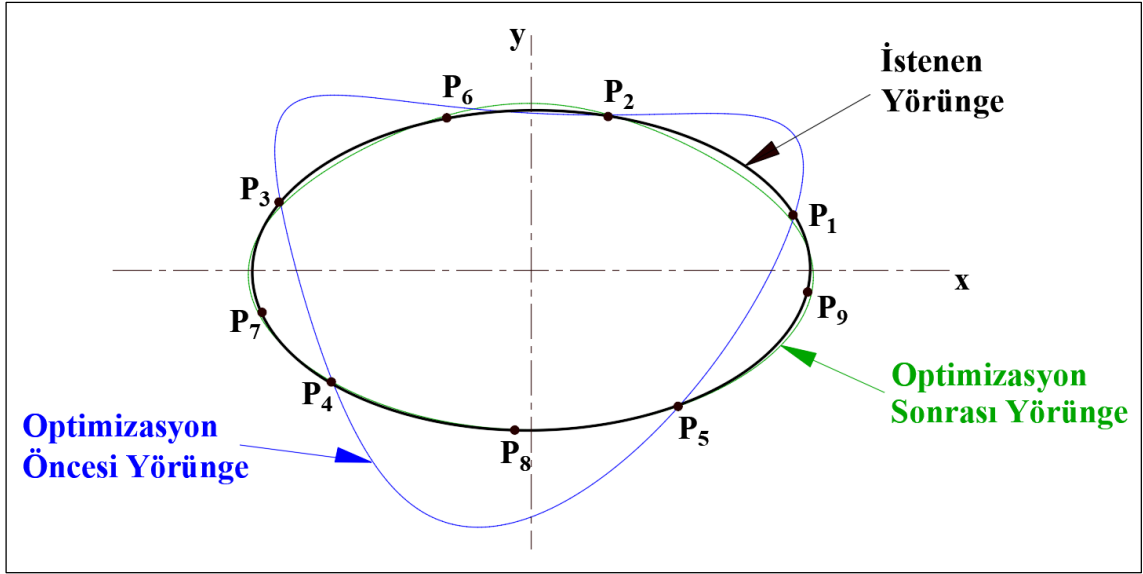
$$P_9 = (0.49539129, -0.03881248),$$

Ön tasarım sonucu elde edilen mekanizma boyutlarının ilave dört yörünge noktası ile optimize edilmesi sonucu elde edilen nihai tasarım sonuçları ve programdan elde edilen yörüngesi dxf formatında Şekil 4.37(a)'da, animasyon ekran görüntüsü Şekil 4.37 (b)'de görüldüğü biçimdedir.



Şekil 4.37. Yörünge Sentezi Modülü Örneği Son Tasarım Sonuçları

Mekanizmanın, optimizasyon öncesi verilen beş noktadan geçmesine karşılık diğer noktalardan uzak olan yörüngesinin optimizasyon sonrasında ilave dört noktaya ve istenen elips yörüngeye oldukça yakın olduğu Şekil 4.38'te görülmektedir.

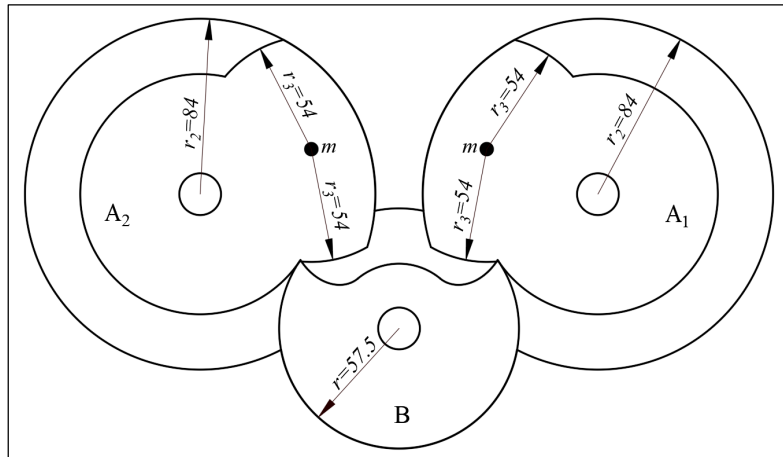


Şekil 4.38. Yörünge Sentezi Modülü Örneğine Ait Amaçlanan Elips Yörünge ile Ön Tasarım ve Son Tasarım Aşamalarındaki Yörüngelerin Karşılaştırmalı Görüntüsü

4.5. Kompresör Modülü Örnek Uygulamaları

Materyal ve Yöntem başlığı Bölüm 3.9'daki tasarım sürecinin etkinliği, geliştirilen bilgisayar programı kullanılarak verilen 3 adet sayısal örnekle gösterilmiştir. Örneklerde kıyaslama yapmak amacıyla, bir referans tasarım civarında, tasarım parametreleri (r, r_2, r_3) sistematik değiştirilerek kompresör geometrisi ve kapasite değişimi değerlendirilmiştir.

Öncelikle referans tasarım adı verilen, tasarımcının öngördüğü kompresör kapasitesine ve kompresör boyutlarına göre, başlangıç tasarım parametreleri r, r_2, r_3 belirlenir. Burada, öngörülen kompresör referans tasarımına örnek olarak $r=57.5$ cm, $r_2=84$ cm ve $r_3=54$ cm ve kompresör derinliği 57 cm alınarak Şekil 4.39'de gösterilen tasarım ele alınmış olup büyük hacim $V_b=493.4$ cm³, küçük hacim $V_s=130.9$ cm³, toplam hacim $V_t=624.4$ cm³ şeklinde bulunmuştur.

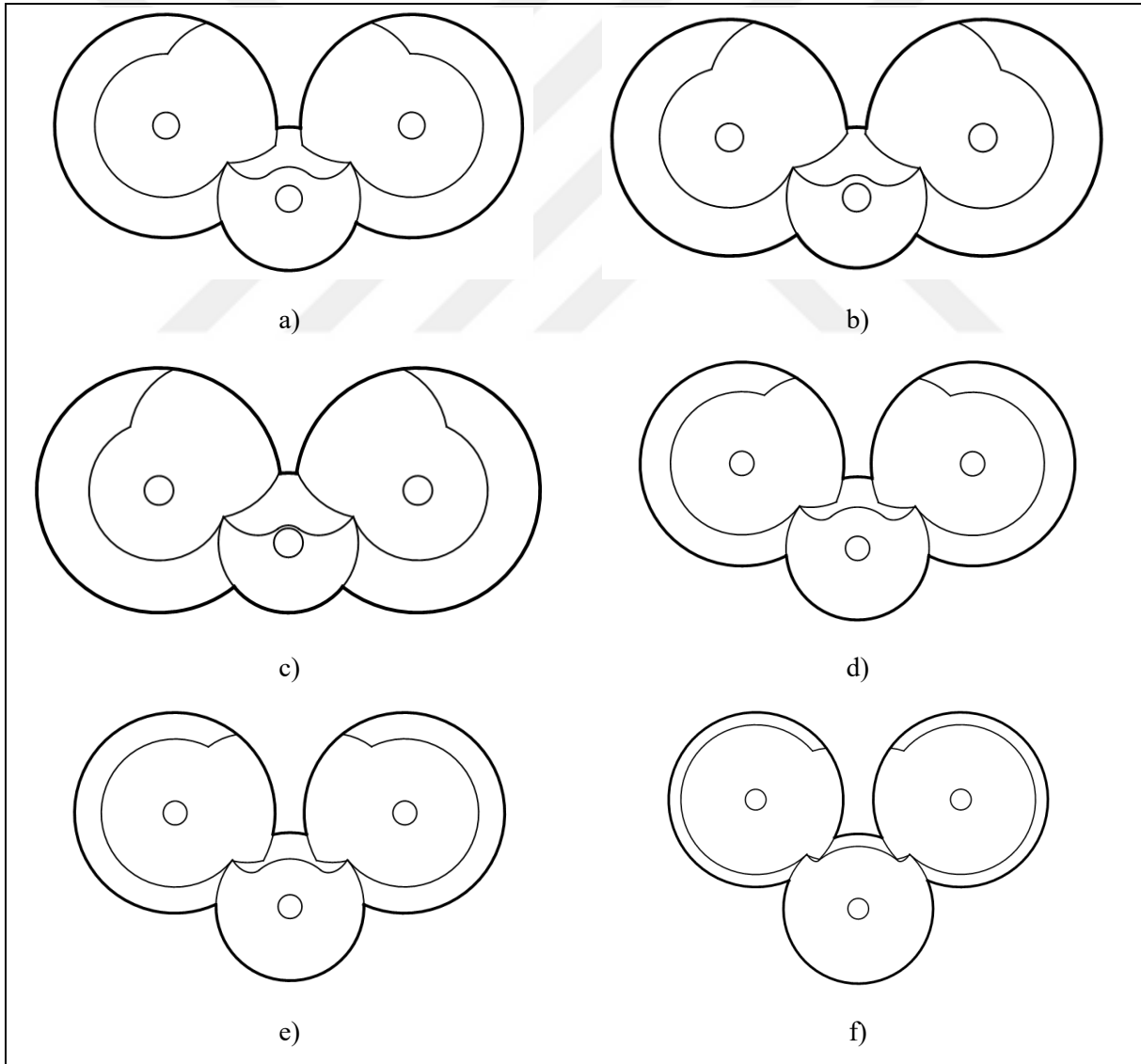


Şekil 4.39. Kompresör referans tasarımı

Tasarım parametrelerinden r yarıçapındaki değişimin karakteristik hacimlere etkileri Tablo 4.12’de sayısal olarak verilmiş olup Şekil 4.40’da görsel olarak verilmiştir.

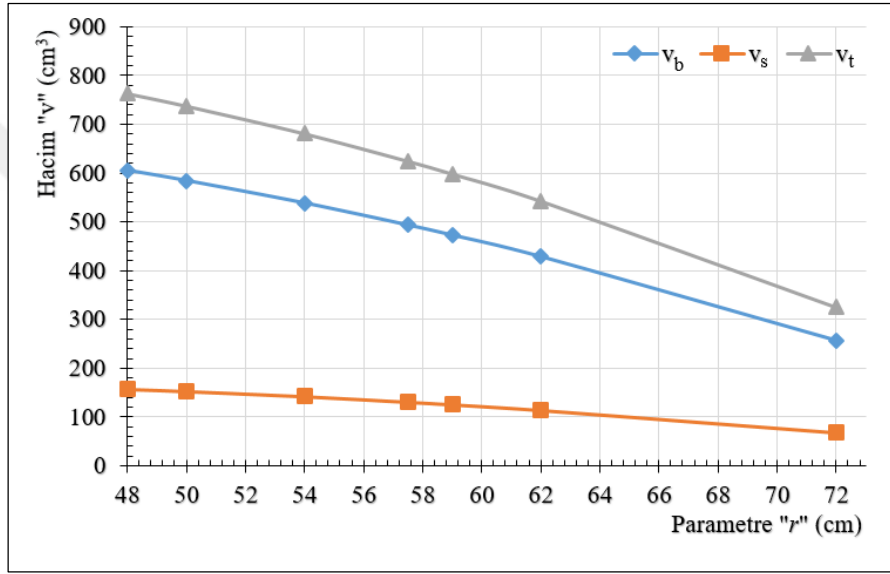
Tablo 4.12. Kompresör referans tasarıma göre r yarıçapındaki değişimin kompresör geometrisi ve kapasitesine etkilerinin sayısal sonuçları

	r (cm)	V_b (cm ³)	V_s (cm ³)	V_t (cm ³)
a)	54	538.4	142.3	680.7
b)	50	584.3	152.9	737.2
c)	48	605.5	157.4	762.9
d)	59	472.7	125.6	598.3
e)	62	428.8	114.0	542.7
f)	72	257.3	68.0	325.3



Şekil 4.40. Kompresör referans tasarıma göre r yarıçapındaki değişimin kompresör geometrisi ve kapasitedeki etkileri

Şekil 4.40 (a),(b) ve (c) de görüldüğü gibi, r yarı çapı küçüldükçe rotor geometrileri Şekil 4.39'deki referans tasarıma göre rotor geometrileri çembersellikten uzaklaşmaktadır. r yarı çapı arttıkça bunun aksine rotor geometrilerinin çemberselliğe yaklaştığı söylenebilir, Şekil 4.40 (d),(e),(f). Rotor geometrilerinin çemberselliğe yaklaşması tasarımın imalat kalitesini artırmakta ve rotorların dengelenmesini kolaylaştırmaktadır. Ayrıca, r yarı çapının artması ile kompresör kapasitesini belirleyen büyük (V_b) ve küçük (V_s) ile toplam hacmin (V_t) azaldığı Şekil 4.41 deki grafikten görülebilir. Bunların dışında, r yarıçapının Şekil 4.40 (b) ve Şekil 4.40 (f) deki değerlerinin sınır değerleri olduğu ve seçilen Şekil 4.39' deki referansa göre bu sınır değerlerin aşılması halinde kompresör geometrisinin elde edilemeyeceği görülmektedir.

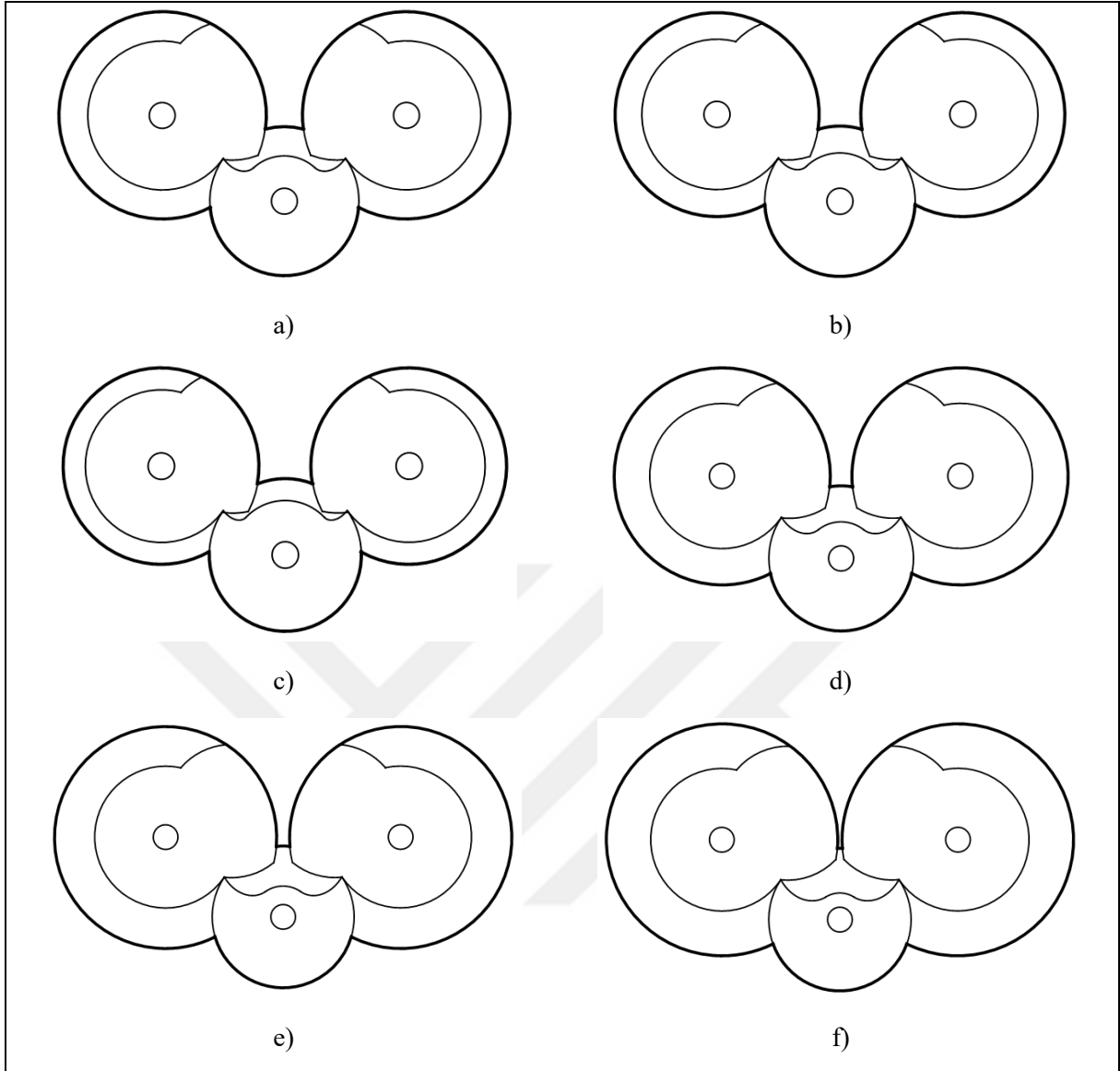


Şekil 4.41. Kompresör kapasitesinin belirleyen V_b , V_s ve V_t hacimlerinin r yarıçapına göre değişim grafiği

Tasarım parametrelerinden r_2 yarıçapındaki değişimin karakteristik hacimlere etkileri Tablo 4.13'de sayısal olarak verilmiş olup Şekil 4.42'de görsel olarak verilmiştir.

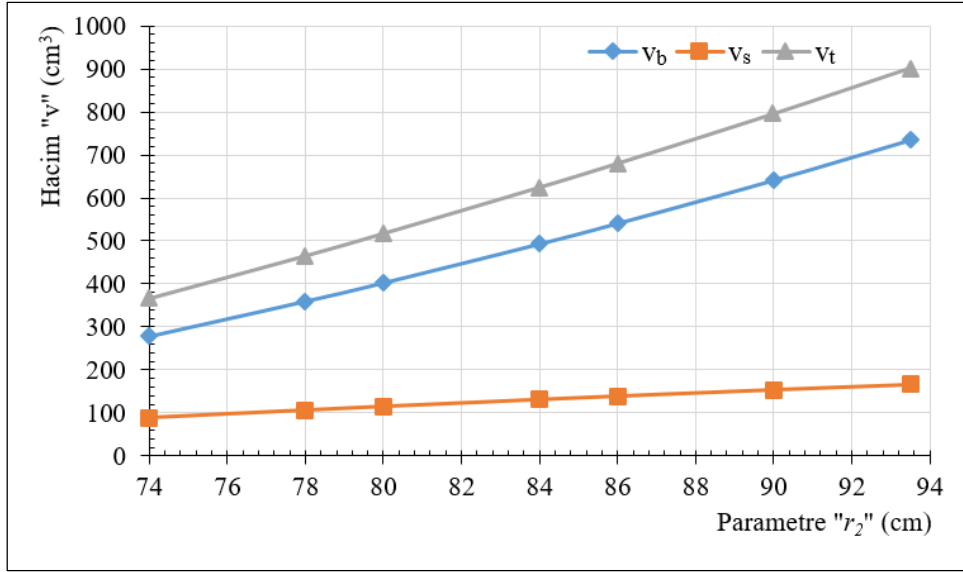
Tablo 4.13. Kompresör referans tasarıma göre r_2 yarıçapındaki değişimin kompresör geometrisi ve kapasitesine etkilerinin sayısal sonuçları

	r_2 (cm)	V_b (cm ³)	V_s (cm ³)	V_t (cm ³)
a)	80	402.7	114.5	517.2
b)	78	359.7	105.9	465.3
c)	74	278.0	88.1	366.2
d)	86	541.2	138.8	680.0
e)	90	641.7	153.8	795.5
f)	93.5	735.4	166.1	901.5



Şekil 4.42. Kompresör referans tasarıma göre r_2 yarıçapındaki değişimin kompresör geometrisi ve kapasitedeki etkileri

Şekil 4.42 (a), (b) ve (c) den, parametre r_2 yarıçapı küçüldükçe rotor geometrilerinin Şekil 4.39'deki referans tasarıma göre çemberselliğe yaklaşmakta olduğu görülmektedir. r_2 yarıçapı arttıkça tersi olarak rotor geometrilerinin çembersellikten uzaklaştığı Şekil 4.42 (d), (e), (f)'den görülebilir. Öte yandan, r_2 yarıçapının artması ile kompresör kapasitesini belirleyen büyük (V_b) ve küçük (V_s) ile toplam hacmin (V_t) arttığı Şekil 4.43deki grafikten görülebilir. Ayrıca, r_2 yarıçapının Şekil 4.42 (c) de görüldenden küçük olması ve Şekil 4.42 (f) de görüldenden büyük olması halinde ise pratik değeri olan bir kompresör geometrisinin oluşturulamayacağı söylenebilir.

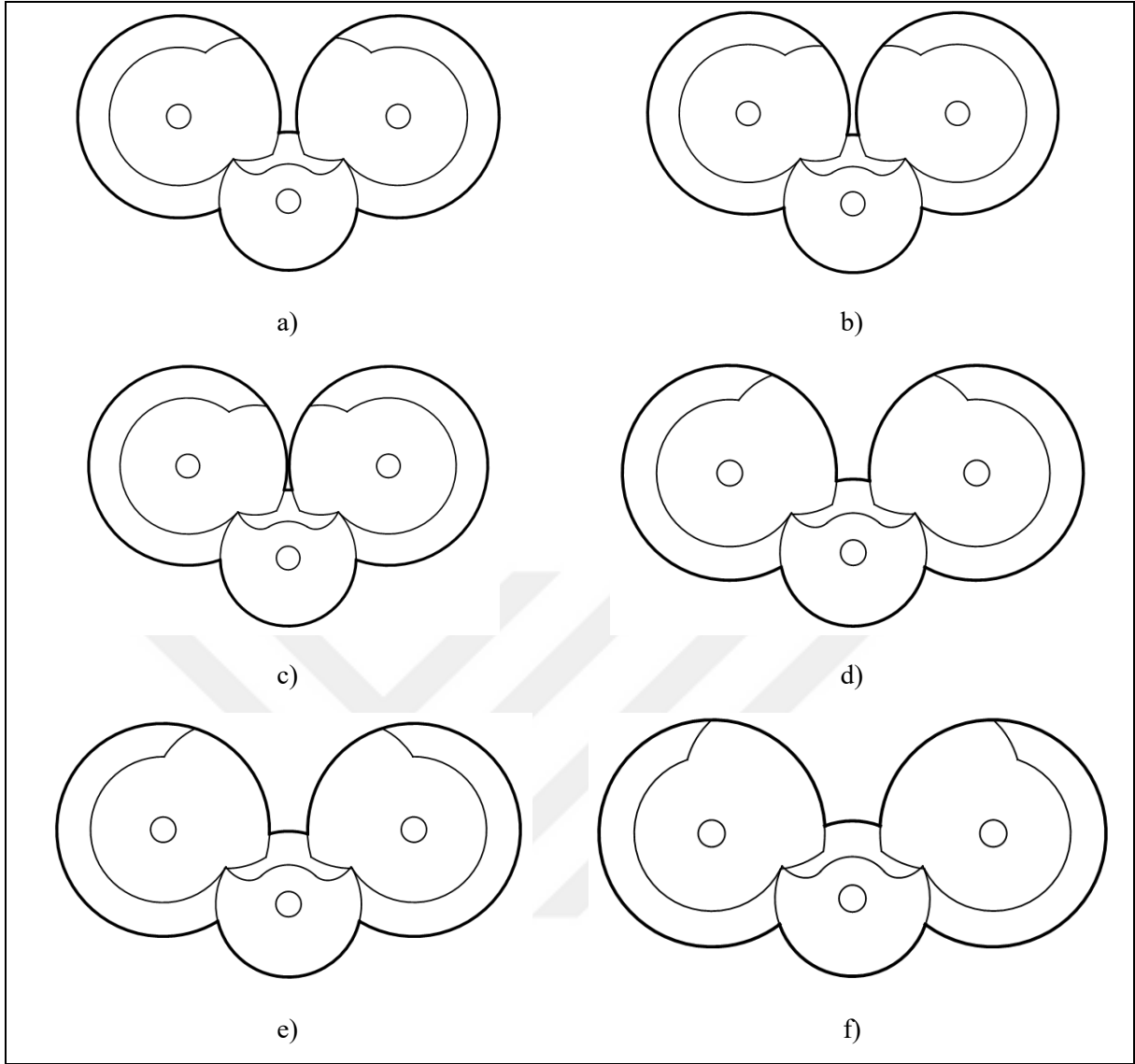


Şekil 4.43. Kompresör kapasitesinin belirleyen V_b , V_s ve V_t hacimlerinin r_2 yarıçapına göre değişim grafiği

Tasarım parametrelerinden r_3 yarıçapındaki değişimin karakteristik hacimlere etkileri Tablo 4.14’da sayısal olarak verilmiş olup Şekil 4.44’de görsel olarak verilmiştir.

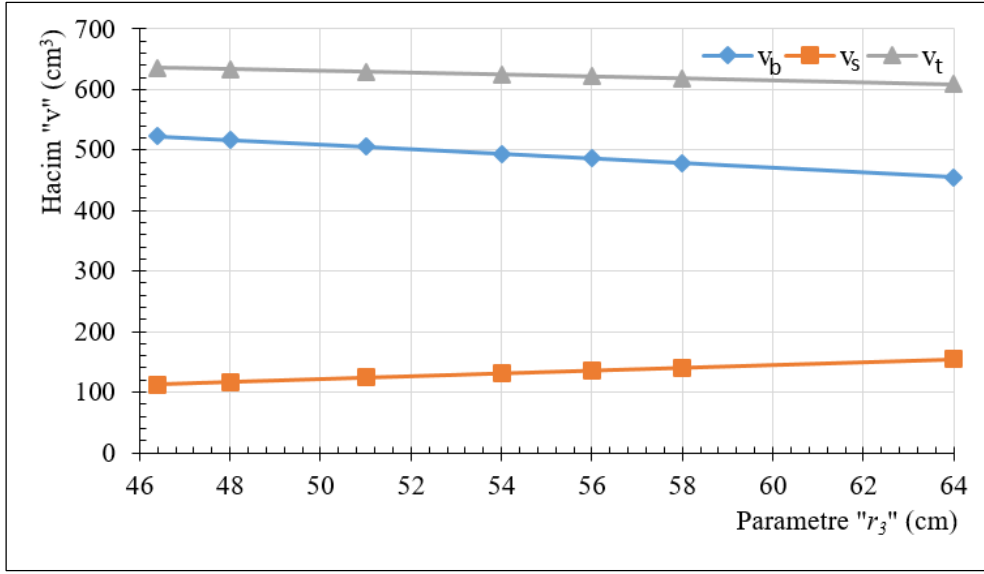
Tablo 4.14. Kompresör referans tasarıma göre r_3 yarıçapındaki değişimin kompresör geometrisi ve kapasitesine etkilerinin sayısal sonuçları

	r_3 (cm)	V_b (cm³)	V_s (cm³)	V_t (cm³)
a)	51	505.1	123.9	628.8
b)	48	516.3	116.7	633.0
c)	46.4	522.4	112.8	635.2
d)	56	485.7	135.6	621.3
e)	58	478.0	140.3	618.3
f)	64	454.5	154.2	608.7



Şekil 4.44. Kompresör referans tasarıma göre r_3 yarıçapındaki değişimin kompresör geometrisi ve kapasitedeki etkileri

Son olarak, Şekil 4.44 (a), (b) ve (c) den, parametre r_3 yarıçapı küçüldükçe rotor geometrilerinde Şekil 4.39'deki referans tasarıma göre yan rotorların üzerindeki kısmi set daralır iken tersi olarak da Şekil 4.44 (d),(e),(f).den r_3 arttıkça kısmi setin genişlediği görülmektedir. r_3 yarıçapının artması ile kompresör kapasitesini belirleyen büyük (V_b) hacim azalırken, küçük (V_s) hacmi atmakta buna karşın toplam hacmin (V_t) azaldığı Şekil 4.45 deki grafikten görülebilir. Öte yandan, r_3 yarıçapının değişmesi sonucunda kompresör kapasitesindeki değişime etkisi diğer r ve r_2 parametresine göre çok daha azdır. Ayrıca r_3 yarıçapının Şekil 4.44 (c) de görüldenden küçük olması halinde ise pratik değeri olan bir kompresör geometrisinin oluşturulamayacağı görülmektedir.



Şekil 4.45. Kompresör kapasitesinin belirleyen V_b , V_s ve V_t hacimlerinin r_3 yarıçapına göre değişim grafiği

5. SONUÇLAR VE ÖNERİLER

Bu çalışma kapsamında makine ve teçhizat tasarımında en önemli adımlardan biri olan mekanizmaların kinematik sentezi kapsamında 4 çubuk mekanizmalarının fonksiyon üretimi, kol-sarkaç tasarımı, yörünge tasarımı, kam mekanizması tasarımı ve özgün bir döngüsel kompresör yapısının geometrik modellenmesi yapılmıştır. Sonuçların hızla elenerek tasarımcıya görsel ve hareketli olarak sunulması ve farklı dosya tiplerinde raporlanması amaçlanmıştır.

Kam mekanizması modülü kapsamında sikloid hareket eğrisi kullanılarak, 4 hareket değişimiyle kısıtlanmış, iç ve/veya dıştan temaslı kam profili oluşturan bilgisayar programı geliştirilmiştir. Geliştirilen program ile mekanizmanın hareketi, harekete bağlı yer değiştirme ve basınç açısı grafikleri kullanıcıya hareketli animasyon olarak sunulmakta ve sonuçlar xlsx, dxf ve step dosya türlerinde sunulmaktadır.

Kam mekanizması modülü teorisinde değinilen fakat programın tez kapsamında yapılan halinde hız ve ivme hesaplamaları kullanıcıya sonuç olarak verilmemekte olup programın geliştirilmesi için yapılacak ilerleyen çalışmalarda dâhil edilmelidir. Ayrıca program için yapılacak sonraki çalışmalarda, hareket değişim sayısının da bir parametre olarak alınması ve bu sayede istenen sayıda değişime izin verilmesi sağlanmalıdır.

Fonksiyon sentezi modülü kapsamında açıklanan ve kesin noktalar, alt bölge ve galerkin yöntemleri kullanılarak yapılan örnek uygulamalarda en fazla yapısal hata $y=\sin(x)$ fonksiyonu örneğinde 0.00699312 olarak hesaplanmıştır. Elde edilen karşılaştırmalı hata grafiklerinde görüldüğü biçimde her üç yöntem sonuçları birbirine yakın ve tutarlıdır. Fakat hesap yöntemleri arasında yapısal hatanın kullanıcı tarafından tanımlanan kesin noktalar ve alt bölgelerden, galerkin yönteminde tanımlanan ağırlık fonksiyonundan daha fazla etkilendiği görülmüş olup bu durumun galerkin metodunun tercih edilmesinde etkili olacağı söylenebilir.

Fonksiyon sentezi modülü kapsamında geliştirilen bilgisayar programı, uygun sonuçları eleyerek kullanıcıya görsel ve metin dosyası olarak sunmaktadır, fakat tez sonrasında yapılacak çalışmalar, sonuçların dxf ve step dosya türlerinde de sunulması için geliştirilmesi yönünde olmalıdır.

Kol sarkaç mekanizması modülü kapsamında açıklanan yöntem kullanılarak yapılan örnek uygulamalarda durum 1, 4, 5, 6'ya ait örneklerde boyutları ve sınır konumlarıyla tek sonuç bulunması yanında, 3, 7, 8 örneklerinde farklı sınır konumları farklı fakat boyutları aynı olan tek mekanizmanın 4'er sonucu elde edilmiştir. Tez kapsamında sekiz farklı durumda çözüm yapılmasına karşın, bilinmeyen parametreler veya parametre sayılarının azaltılmasıyla yeni sentez durumları geliştirmeye açıktır.

Kol sarkaç mekanizması modülü kapsamında geliştirilen bilgisayar programı, uygun sonuçları eleyerek kullanıcıya görsel ve metin dosyası olarak sunmakla birlikte, CAD programlarında açılabilir dxf dosya türünde de sunulmaktadır. Tez sonrasında yapılacak çalışmalar, sonuçların step dosya türlerinde de sunulması için geliştirilmesi yönünde olmalıdır.

Yörünge sentezi modülü kapsamında; kinematik sentez hesaplarındaki matematiksel ifadelerde kullanılan denklemler içerisinde bulunan çok sayıdaki fonksiyon veya ifadelerin süreksiz veya tanımsız olduğu noktalarda ortaya çıkan ve ne zaman karşılaşılabileceği önceden bilinmeyen hesaplama tekillikleri ve yakınsamayı sağlayıcı ilk tahmin değerlerinin uygun seçilmesi ihtiyaçlarından kaynaklanan sorunların çözümüne yönelik iki aşamalı bir yaklaşım gerçekleştirilerek, sonuçların elenerek kullanıcıya görsel ve CAD programlarında açılabilir dxf dosya türünde de sunulması sağlanmıştır.

Bu amaçla istenen yörüngeyi üreten bir 4 çubuk mekanizmasının sentezinde öncelikle, ilk tahmin gerektirmeyen, verilen 5 yörünge noktası için kapalı çözüm ile 14. ve 12. dereceden polinomların mümkün olan gerçek ve kompleks tüm kökleri bulunarak ön tasarım yapılmıştır. Sonrasında verilen 4 yeni nokta ile biyel eğrisin genel denkleminde elde edilen ve bilinmeyen parametre sayısı 9 olan bir amaç fonksiyonunun iteratif yaklaşımla minimizasyonu gerçekleştirilmiştir. Minimizasyon sürecinde belirlenen amaç fonksiyonunun tüm noktalarda sürekli ve istenilen dereceden türevi alınabildiğinden, geliştirilen bilgisayar programında hesaplama imkânsızlıkları ile karşılaşılmamıştır. İterasyon sürecini başlatacak ilk tahmin değerleri olarak, bir önceki aşamada bulunan 4-çubuk mekanizmasının çözüm değerleri kullanıldığından, tasarımcının ilk tahmin değerleri bulmasına gerek kalmamış ve aranan optimum çözüme yakın olması nedeniyle son tasarımda iterasyonda yakınsama sorunları olmamıştır. İkinci aşamada kullanılan bilinmeyen parametre sayısı 9 olduğundan, bu aşamada elde edilen çözüm her zaman verilen yörüngeye daha yakın olduğu ve ön tasarımda olası tüm çözümler elde edildiğinden, verilen yörüngeyi en iyi üretecek farklı mekanizma sayısının bir den fazla olabildiği örnek ile gösterilmiştir.

Özgün bir döngüsel kompresör yapısının geometrik modellenmesi modülü kapsamında üç rotorlu bir kompresör geometrisinin tasarımı için gerekli parametrik eğri ailesi ve eğriye ait zarf eğrileri kullanılarak bir matematiksel model geliştirilmiştir. Bu modeli kullanmak üzere geliştirilen bilgisayar programı ile az sayıda geometrik parametre değiştirilerek hızlı ve alternatif kompresör yapılarına ulaşılmakta, sonuçlar hareketli animasyon ve CAD programlarında açılabilir dxf ve step dosya türlerinde kullanıcıya sunulabilmektedir.

Özgün bir döngüsel kompresör yapısı için gelecekte yapılacak çalışmalarda, buradan elde edilen katı model geometrilerin Hesaplamalı Akışkanlar Dinamiği (HAD) yöntemleri ile kompresör karakteristikleri (verim, sızdırmazlık performansı vb.) belirlenerek daha akılcı geometrik iyileştirmelerin yapılması mümkün olacaktır. Ayrıca bilgisayar destekli tasarım, üretim, yazılım ve katı model teknolojileri ile bütünleştirilmiş bu çalışmada elde edilen geometrik tasarımın, dengeleme, gerilme, titreşim gibi yapısal analizlerle desteklenerek, geliştirilmesi için gerekli verimli ve sezgisel yaklaşımlar da gelecekte olanaklı hale getirilmiştir.

Tez kapsamında geliştirilen bilgisayar programlarına yönelik tez sonrasında yapılacak çalışmalar, kodların daha verimli çalışması ve mevcut hataların giderilmesi, yukarıda bahsedilen özelliklerin modüllere eklenmesi, yeni durum ve farklı ihtiyaçlara göre geliştirilmesine yönelik olmalıdır.

Ayrıca bu tez kapsamında geliştirilen mekanizma modüllerinin ticari bir ürüne dönüştürülmesi amacıyla İMC Mekatronik Sanayi ve Ticaret Limited Şirketi ile ortaklaşa sanayi projesine başvuru yapılmıştır ve proje “Makine Tasarımında Kullanılacak Kinematik Sentezin Modellenmesi ve Simulasyona Yönelik Modüler Masa Üstü-Web Tabanlı Etkileşimli Yazılım Geliştirilmesi” başlıklı 7230960 proje numarası ile 1507 TÜBİTAK KOBİ Ar-Ge Başlangıç Destek Programı kapsamında 01.07.2024 tarihi itibarı ile desteklenmektedir.





KAYNAKLAR

Akcali, I. D. ve Dittrich, G. (1989). "Function generation by Galerkin's method." *Mechanism and Machine Theory*, Vol. 24, No. 1, pp. 39–43.

Akcali, I. D. ve Mutlu, H. (2014). *Mekanizma Tekniği*, Karahan Kitabevi, Adana, Türkiye.

Akçali, I.D. ve Lindholm, I. C. (1979). "A Novel Method for The 4-Bar Function Generation Problem: Subdomain Method." , V. World Congren On Theory Of Machines and Mechanisms.

Akcalı, I. D. (1994). *Mekanizmaların Sentezi*, Cukurova Universitesi Muhendislik Mimarlık Fakultesi, Adana, Türkiye.

Bureerat, S. ve Slesongsom, S. (2021). "Constraint handling technique for four-bar linkage path generation using self-adaptive teaching–learning-based optimization with a diversity archive." *Engineering Optimization*, Vol. 53, No. 3, pp. 513–530.

Buśkiewicz, J. (2015). "A method for optimal path synthesis of four-link planar mechanisms." *Inverse Problems in Science and Engineering*, Vol. 23, No. 5, pp. 818–850.

Flores, P. (2013). "A computational approach for cam size optimization of disc cam-follower mechanisms with translating roller followers." *Journal of Mechanisms and Robotics*, Vol. 5, No. 4, pp. 1–6.

Hartenberg, R. S. ve Denavit, J. (1964). "Kinematic Synthesis of Linkages." .

Kafash, S. H. ve Nahvi, A. (2017). "Optimal synthesis of four-bar path generator linkages using Circular Proximity Function." *Mechanism and Machine Theory*, Vol. 115, pp. 18–34.

Li, X., Wei, S., Liao, Q. ve Zhang, Y. (2016). "A novel analytical method for function generation synthesis of planar four-bar linkages." *Mechanism and Machine Theory*, Vol. 101, pp. 222–235.

Megep (2011). *Makine Teknolojisi Hareket Dönüştürme Elemanları*, T.C. Milli Eğitim Bakanlığı, Ankara, Türkiye.

Mundo, D., Danieli, G. A. ve Yan, H. S. (2006). "Kinematic optimization of mechanical presses by optimal synthesis of cam-integrated linkages." *Transactions of the Canadian Society for Mechanical*

Engineering, Vol. 30, No. 4, pp. 519–532.

Mutlu, H. (1994). Düzlemde Tek Serbestlik Derecesine Sahip Mekanizmaların Kinematik Sentezi, Cukurova Üniversitesi, Adana, Türkiye.

Mutlu, H. (2011). "Dört-Çubuk Mekanizmasının Kapalı Çözüme Dayalı 5 Parametrelili Fonksiyon Sentezi." İçinde : 15. Ulusal Makina Teorisi Sempozyumu, pp. 21–32.

Mutlu, H. (2021). "Design of the crank–rocker mechanism for various design cases based on the closed-form solution." Sadhana - Academy Proceedings in Engineering Sciences, Vol. 46, No. 1

Mutlu, H. (2023). "Nine-Parameter Path Synthesis of Planar Four-Bar Mechanism With the Aid of the General Equation of Coupler Curve." Thermal Science, Vol. 27, No. 4, pp. 3349–3360.

Nadal, F., Cabrera, J. A., Bataller, A., Castillo, J. J. ve Ortiz, A. (2015). "Turning functions in optimal synthesis of mechanisms." Journal of Mechanical Design, Vol. 137, No. 6

Norton, R. L. (2009). Cam Design and Manufacturing Handbook, Industrial Press Inc., New York, USA

Norton, R. L. (2020). Design of Machinery, McGraw-Hill. New York, USA

Pan, T. H., Chang, C. H., Lin, P. T. ve Hsu, K. L. (2022). "A Computational Approach for Cam Design Parameters Optimization of Disk Cam Mechanisms with Oscillating Roller Follower." International Conference on Advanced Robotics and Intelligent Systems, ARIS, Vol. 2022-Augus, No. January

Petropoulou, A., Dimopoulos, S., Mourtzis, D. ve Chondros, T. G. (2009). "A Computer Aided Method for Cam Profile Design." .

Rothbart, H. A. (2004). Cam Design Handbook, McGraw-Hill, USA

Shala, A., Likaj, R. ve Bruqi, M. (2013). "Synthesis of Cam mechanism based on given program." IFAC Proceedings Volumes (IFAC-PapersOnline), Vol. 15, No. PART 1, pp. 60–63.

Shariati, M. ve Norouzi, M. (2011). "Optimal synthesis of function generator of four-bar linkages based on distribution of precision points." Meccanica, Vol. 46, No. 5, pp. 1007–1021.

Shiakolas, P. S., Koladiya, D. ve Kebrle, J. (2002). "On the optimum synthesis of four-bar linkages

using differential evolution and the geometric centroid of precision positions." *Inverse Problems in Engineering*, Vol. 10, No. 6, pp. 485–502.

Simionescu, P. A. ve Neagoe, M. (2020). "Cam Profiles Generation as Follower Envelopes with MATLAB Programs." *Mechanisms and Machine Science*, Vol. 83, pp. 20–30.

Simionescu, P. A. ve Ortega, S. (2020). "Analysis and Synthesis of Planar Cam Mechanisms Using Working Model 2D." *Mechanisms and Machine Science*, Vol. 83, pp. 10–19.

Soong, R. C. ve Chang, S. Bin (2011). "Synthesis of function-generation mechanisms using variable length driving links." *Mechanism and Machine Theory*, Vol. 46, No. 11, pp. 1696–1706.

Soylemez, E. (2007). *Mekanizma Tekniği*, Birsen Yayınevi, Ankara, Türkiye.

Stoilov Todorov, T. (2015). "Synthesis of Four Bar Mechanisms as Function Generators by Freudenstein - Chebyshev." *Journal of Robotics and Mechanical Engineering Research*, Vol. 1, No. 1, pp. 1–6.

Trajcevski, L. ve Geramitchioski, T. (2013). "Cams-software program to design and synthesis of cams mechanisms." *Mechanisms and Machine Science*, Vol. 13, pp. 427–434.

Wang, B., Du, X., Jian, D., Dong, Y., Wang, C. ve Liu, X. (2021). "The Synthesis of Planar Four-Bar Linkage for Mixed Motion and Function Generation." *Sensors*.

Zarkandi, S. (2021). "A novel optimization-based method to find multiple solutions for path synthesis of planar four-bar and slider-crank mechanisms." *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 235, No. 21, pp. 5385–5405.

Zhang, Y. ve Shin, J. H. (2004). "A Computational Approach to Profile Generation of Planar Cam Mechanisms." *Journal of Mechanical Design*.

Zhu, G., Wang, Y., Zhu, G. N., Weng, M., Liu, J., Zhou, J. ve Lu, B. (2023). "Parameter Optimization of Large-Size High-Speed Cam-Linkage Mechanism for Kinematic Performance." *Actuators*, Vol. 12, No. 1, .

Zhu, W. W. ve Lan, Z. H. (2017). "Kinematic synthesis of planar cam mechanism with translating roller follower by kinematic analysis of binary link chain." *Lecture Notes in Electrical Engineering*, Vol. 408, pp. 1557–1565.



EKLER**EK 1. Fonksiyon Sentezi Modülü Denklem (3.3) Katsayıları**

$$\begin{aligned}
a_{pi} = & a[2]*(c[k]*e[3]*f[1]-c[3]*e[k]*f[1]-c[k]*e[1]*f[3]+c[1]*e[k]*f[3]+b[3]*d[1]*f[k]- \\
& b[1]*d[3]*f[k]+c[3]*e[1]*f[k]-c[1]*e[3]*f[k]-c[k]*d[3]*g[1]- \\
& b[3]*e[k]*g[1]+c[k]*d[1]*g[3]+b[1]*e[k]*g[3]+d[k]*(-(b[3]*f[1])+b[1]*f[3]+c[3]*g[1]- \\
& c[1]*g[3])+b[k]*(d[3]*f[1]-d[1]*f[3]+e[3]*g[1]-e[1]*g[3]))+a[k]*(c[3]*e[2]*f[1]- \\
& c[2]*e[3]*f[1]+b[1]*d[3]*f[2]-c[3]*e[1]*f[2]+c[1]*e[3]*f[2]-b[1]*d[2]*f[3]+c[2]*e[1]*f[3]- \\
& c[1]*e[2]*f[3]-c[3]*d[2]*g[1]+c[2]*d[3]*g[1]+c[3]*d[1]*g[2]- \\
& c[1]*d[3]*g[2]+b[1]*e[3]*g[2]+b[3]*(d[2]*f[1]-d[1]*f[2]+e[2]*g[1]-e[1]*g[2])- \\
& c[2]*d[1]*g[3]+c[1]*d[2]*g[3]-b[1]*e[2]*g[3]+b[2]*(-(d[3]*f[1])+d[1]*f[3]- \\
& e[3]*g[1]+e[1]*g[3]))+a[1]*(-(c[k]*e[3]*f[2])+c[3]*e[k]*f[2]+c[k]*e[2]*f[3]-c[2]*e[k]*f[3]- \\
& b[3]*d[2]*f[k]+b[2]*d[3]*f[k]-c[3]*e[2]*f[k]+c[2]*e[3]*f[k]+c[k]*d[3]*g[2]+b[3]*e[k]*g[2]- \\
& c[k]*d[2]*g[3]-b[2]*e[k]*g[3]+d[k]*(b[3]*f[2]-b[2]*f[3]-c[3]*g[2]+c[2]*g[3])+b[k]*(- \\
& (d[3]*f[2])+d[2]*f[3]-e[3]*g[2]+e[2]*g[3]))- \\
& a[2]*c[3]*d[1]*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[2]*b[3]*e[1]*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]+a[3]*(-(c[k]*e[2]*f[1])+c[2]*e[k]*f[1]+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]+c[k]*d[2]*g[1]+b[2]*e[k]*g[1]-c[k]*d[1]*g[2]- \\
& b[1]*e[k]*g[2]+d[k]*(b[2]*f[1]-b[1]*f[2]-c[2]*g[1]+c[1]*g[2])+b[k]*(-(d[2]*f[1])+d[1]*f[2]- \\
& e[2]*g[1]+e[1]*g[2])+c[2]*d[1]*g[k]-c[1]*d[2]*g[k]-b[2]*e[1]*g[k]+b[1]*e[2]*g[k]) \\
\\
b_{pi} = & a[2]*(-(c[k]*e[3]*f[1])+c[3]*e[k]*f[1]+c[k]*e[1]*f[3]-c[1]*e[k]*f[3]+b[3]*d[1]*f[k]- \\
& b[1]*d[3]*f[k]-c[3]*e[1]*f[k]+c[1]*e[3]*f[k]+c[k]*d[3]*g[1]-b[3]*e[k]*g[1]- \\
& c[k]*d[1]*g[3]+b[1]*e[k]*g[3]+d[k]*(-(b[3]*f[1])+b[1]*f[3]-c[3]*g[1]+c[1]*g[3])+b[k]*(d[3]*f[1]- \\
& d[1]*f[3]+e[3]*g[1]-e[1]*g[3]))+a[k]*(- \\
& (c[3]*e[2]*f[1])+c[2]*e[3]*f[1]+b[1]*d[3]*f[2]+c[3]*e[1]*f[2]-c[1]*e[3]*f[2]-b[1]*d[2]*f[3]- \\
& c[2]*e[1]*f[3]+c[1]*e[2]*f[3]+c[3]*d[2]*g[1]-c[2]*d[3]*g[1]- \\
& c[3]*d[1]*g[2]+c[1]*d[3]*g[2]+b[1]*e[3]*g[2]+b[3]*(d[2]*f[1]-d[1]*f[2]+e[2]*g[1]- \\
& e[1]*g[2])+c[2]*d[1]*g[3]-c[1]*d[2]*g[3]-b[1]*e[2]*g[3]+b[2]*(-(d[3]*f[1])+d[1]*f[3]- \\
& e[3]*g[1]+e[1]*g[3]))+a[1]*(c[k]*e[3]*f[2]-c[3]*e[k]*f[2]-c[k]*e[2]*f[3]+c[2]*e[k]*f[3]- \\
& b[3]*d[2]*f[k]+b[2]*d[3]*f[k]+c[3]*e[2]*f[k]-c[2]*e[3]*f[k]- \\
& c[k]*d[3]*g[2]+b[3]*e[k]*g[2]+c[k]*d[2]*g[3]-b[2]*e[k]*g[3]+d[k]*(b[3]*f[2]-b[2]*f[3]+c[3]*g[2]- \\
& c[2]*g[3])+b[k]*(-(d[3]*f[2])+d[2]*f[3]-e[3]*g[2]+e[2]*g[3]))+a[2]*c[3]*d[1]*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[2]*b[3]*e[1]*g[k]-
\end{aligned}$$

$$\begin{aligned} & a[1]*b[3]*e[2]*g[k]-a[2]*b[1]*e[3]*g[k]+a[1]*b[2]*e[3]*g[k]+a[3]*(c[k]*e[2]*f[1]-c[2]*e[k]*f[1]- \\ & c[k]*e[1]*f[2]+c[1]*e[k]*f[2]-b[2]*d[1]*f[k]+b[1]*d[2]*f[k]+c[2]*e[1]*f[k]-c[1]*e[2]*f[k]- \\ & c[k]*d[2]*g[1]+b[2]*e[k]*g[1]+c[k]*d[1]*g[2]-b[1]*e[k]*g[2]+d[k]*(b[2]*f[1]-b[1]*f[2]+c[2]*g[1]- \\ & c[1]*g[2])+b[k]*(-(d[2]*f[1])+d[1]*f[2]-e[2]*g[1]+e[1]*g[2])-c[2]*d[1]*g[k]+c[1]*d[2]*g[k]- \\ & b[2]*e[1]*g[k]+b[1]*e[2]*g[k]) \end{aligned}$$

$$\begin{aligned} \mathbf{c}_{pi} &= a[2]*(c[k]*e[3]*f[1]-c[3]*e[k]*f[1]-c[k]*e[1]*f[3]+c[1]*e[k]*f[3]- \\ & b[3]*d[1]*f[k]+b[1]*d[3]*f[k]+c[3]*e[1]*f[k]-c[1]*e[3]*f[k]+c[k]*d[3]*g[1]-b[3]*e[k]*g[1]- \\ & c[k]*d[1]*g[3]+b[1]*e[k]*g[3]+d[k]*(b[3]*f[1]-b[1]*f[3]-c[3]*g[1]+c[1]*g[3])+b[k]*(- \\ & (d[3]*f[1])+d[1]*f[3]+e[3]*g[1]-e[1]*g[3]))+a[k]*(c[3]*e[2]*f[1]-c[2]*e[3]*f[1]-b[1]*d[3]*f[2]- \\ & c[3]*e[1]*f[2]+c[1]*e[3]*f[2]+b[1]*d[2]*f[3]+c[2]*e[1]*f[3]-c[1]*e[2]*f[3]+c[3]*d[2]*g[1]- \\ & c[2]*d[3]*g[1]-c[3]*d[1]*g[2]+c[1]*d[3]*g[2]+b[1]*e[3]*g[2]+b[3]*(- \\ & (d[2]*f[1])+d[1]*f[2]+e[2]*g[1]-e[1]*g[2]))+c[2]*d[1]-c[1]*d[2]-b[1]*e[2])*g[3]+b[2]*(d[3]*f[1]- \\ & d[1]*f[3]-e[3]*g[1]+e[1]*g[3]))+a[1]*(-(c[k]*e[3]*f[2])+c[3]*e[k]*f[2]+c[k]*e[2]*f[3]- \\ & c[2]*e[k]*f[3]+b[3]*d[2]*f[k]-b[2]*d[3]*f[k]-c[3]*e[2]*f[k]+c[2]*e[3]*f[k]- \\ & c[k]*d[3]*g[2]+b[3]*e[k]*g[2]+c[k]*d[2]*g[3]-b[2]*e[k]*g[3]+d[k]*(- \\ & (b[3]*f[2])+b[2]*f[3]+c[3]*g[2]-c[2]*g[3])+b[k]*(d[3]*f[2]-d[2]*f[3]- \\ & e[3]*g[2]+e[2]*g[3]))+a[2]*c[3]*d[1]*g[k]-a[1]*c[3]*d[2]*g[k]- \\ & a[2]*c[1]*d[3]*g[k]+a[1]*c[2]*d[3]*g[k]+a[2]*b[3]*e[1]*g[k]-a[1]*b[3]*e[2]*g[k]- \\ & a[2]*b[1]*e[3]*g[k]+a[1]*b[2]*e[3]*g[k]+a[3]*(-(c[k]*e[2]*f[1])+c[2]*e[k]*f[1]+c[k]*e[1]*f[2]- \\ & c[1]*e[k]*f[2]+(b[2]*d[1]-b[1]*d[2])*f[k]-c[2]*e[1]*f[k]+c[1]*e[2]*f[k]- \\ & c[k]*d[2]*g[1]+b[2]*e[k]*g[1]+c[k]*d[1]*g[2]-b[1]*e[k]*g[2]+d[k]*(- \\ & (b[2]*f[1])+b[1]*f[2]+c[2]*g[1]-c[1]*g[2])+b[k]*(d[2]*f[1]-d[1]*f[2]-e[2]*g[1]+e[1]*g[2]))+(- \\ & (c[2]*d[1])+c[1]*d[2]-b[2]*e[1]+b[1]*e[2])*g[k]) \end{aligned}$$

$$\begin{aligned} \mathbf{d}_{pi} &= a[2]*(-(c[k]*e[3]*f[1])+c[3]*e[k]*f[1]+c[k]*e[1]*f[3]-c[1]*e[k]*f[3]- \\ & b[3]*d[1]*f[k]+b[1]*d[3]*f[k]-c[3]*e[1]*f[k]+c[1]*e[3]*f[k]-c[k]*d[3]*g[1]- \\ & b[3]*e[k]*g[1]+c[k]*d[1]*g[3]+b[1]*e[k]*g[3]+d[k]*(b[3]*f[1]-b[1]*f[3]+c[3]*g[1]- \\ & c[1]*g[3])+b[k]*(-(d[3]*f[1])+d[1]*f[3]+e[3]*g[1]-e[1]*g[3]))+a[k]*(- \\ & (c[3]*e[2]*f[1])+c[2]*e[3]*f[1]-b[1]*d[3]*f[2]+c[3]*e[1]*f[2]-c[1]*e[3]*f[2]+b[1]*d[2]*f[3]- \\ & c[2]*e[1]*f[3]+c[1]*e[2]*f[3]-c[3]*d[2]*g[1]+c[2]*d[3]*g[1]+c[3]*d[1]*g[2]- \\ & c[1]*d[3]*g[2]+b[1]*e[3]*g[2]+b[3]*(-(d[2]*f[1])+d[1]*f[2]+e[2]*g[1]-e[1]*g[2]))-(c[2]*d[1]- \\ & c[1]*d[2]+b[1]*e[2])*g[3]+b[2]*(d[3]*f[1]-d[1]*f[3]-e[3]*g[1]+e[1]*g[3]))+a[1]*(c[k]*e[3]*f[2]- \\ & c[3]*e[k]*f[2]-c[k]*e[2]*f[3]+c[2]*e[k]*f[3]+b[3]*d[2]*f[k]-b[2]*d[3]*f[k]+c[3]*e[2]*f[k]- \\ & c[2]*e[3]*f[k]+c[k]*d[3]*g[2]+b[3]*e[k]*g[2]-(c[k]*d[2]+b[2]*e[k])*g[3]+d[k]*(- \\ & (b[3]*f[2])+b[2]*f[3]-c[3]*g[2]+c[2]*g[3])+b[k]*(d[3]*f[2]-d[2]*f[3]-e[3]*g[2]+e[2]*g[3]))- \\ & a[2]*c[3]*d[1]*g[k]+a[1]*c[3]*d[2]*g[k]+a[2]*c[1]*d[3]*g[k]- \end{aligned}$$

$$\begin{aligned} & a[1]*c[2]*d[3]*g[k]+a[2]*b[3]*e[1]*g[k]-a[1]*b[3]*e[2]*g[k]- \\ & a[2]*b[1]*e[3]*g[k]+a[1]*b[2]*e[3]*g[k]+a[3]*(c[k]*e[2]*f[1]-c[2]*e[k]*f[1]- \\ & c[k]*e[1]*f[2]+c[1]*e[k]*f[2]+b[2]*d[1]*f[k]-b[1]*d[2]*f[k]+c[2]*e[1]*f[k]- \\ & c[1]*e[2]*f[k]+c[k]*d[2]*g[1]+b[2]*e[k]*g[1]-c[k]*d[1]*g[2]-b[1]*e[k]*g[2]+d[k]*(- \\ & (b[2]*f[1])+b[1]*f[2]-c[2]*g[1]+c[1]*g[2])+b[k]*(d[2]*f[1]-d[1]*f[2]- \\ & e[2]*g[1]+e[1]*g[2])+(c[2]*d[1]-c[1]*d[2]-b[2]*e[1]+b[1]*e[2])*g[k]) \end{aligned}$$

$$\begin{aligned} \mathbf{e}_{pi} = & a[k]*(-(b[3]*e[2]*f[1])+b[2]*e[3]*f[1]-c[1]*d[3]*f[2]+b[3]*e[1]*f[2]- \\ & b[1]*e[3]*f[2]+c[1]*d[2]*f[3]-b[2]*e[1]*f[3]+b[1]*e[2]*f[3]+b[3]*d[2]*g[1]-b[2]*d[3]*g[1]- \\ & b[3]*d[1]*g[2]+b[1]*d[3]*g[2]-c[1]*e[3]*g[2]+c[3]*(-(d[2]*f[1])+d[1]*f[2]- \\ & e[2]*g[1]+e[1]*g[2])+(b[2]*d[1]-b[1]*d[2]+c[1]*e[2])*g[3]+c[2]*(d[3]*f[1]-d[1]*f[3]+e[3]*g[1]- \\ & e[1]*g[3]))+a[2]*(-(b[k]*e[3]*f[1])+b[3]*e[k]*f[1]+b[k]*e[1]*f[3]-b[1]*e[k]*f[3]- \\ & c[3]*d[1]*f[k]+c[1]*d[3]*f[k]-b[3]*e[1]*f[k]+b[1]*e[3]*f[k]+b[k]*d[3]*g[1]+c[3]*e[k]*g[1]- \\ & (b[k]*d[1]+c[1]*e[k])*g[3]+d[k]*(c[3]*f[1]-c[1]*f[3]-b[3]*g[1]+b[1]*g[3])+c[k]*(- \\ & (d[3]*f[1])+d[1]*f[3]-e[3]*g[1]+e[1]*g[3]))+a[1]*(b[k]*e[3]*f[2]-b[3]*e[k]*f[2]- \\ & b[k]*e[2]*f[3]+b[2]*e[k]*f[3]+c[3]*d[2]*f[k]-c[2]*d[3]*f[k]+b[3]*e[2]*f[k]-b[2]*e[3]*f[k]- \\ & b[k]*d[3]*g[2]-c[3]*e[k]*g[2]+b[k]*d[2]*g[3]+c[2]*e[k]*g[3]+d[k]*(- \\ & (c[3]*f[2])+c[2]*f[3]+b[3]*g[2]-b[2]*g[3])+c[k]*(d[3]*f[2]-d[2]*f[3]+e[3]*g[2]- \\ & e[2]*g[3]))+a[2]*b[3]*d[1]*g[k]-a[1]*b[3]*d[2]*g[k]-a[2]*b[1]*d[3]*g[k]+a[1]*b[2]*d[3]*g[k]- \\ & a[2]*c[3]*e[1]*g[k]+a[1]*c[3]*e[2]*g[k]+a[2]*c[1]*e[3]*g[k]- \\ & a[1]*c[2]*e[3]*g[k]+a[3]*(b[k]*e[2]*f[1]-b[2]*e[k]*f[1]- \\ & b[k]*e[1]*f[2]+b[1]*e[k]*f[2]+c[2]*d[1]*f[k]-c[1]*d[2]*f[k]+b[2]*e[1]*f[k]-b[1]*e[2]*f[k]- \\ & b[k]*d[2]*g[1]-c[2]*e[k]*g[1]+b[k]*d[1]*g[2]+c[1]*e[k]*g[2]+d[k]*(- \\ & (c[2]*f[1])+c[1]*f[2]+b[2]*g[1]-b[1]*g[2])+c[k]*(d[2]*f[1]-d[1]*f[2]+e[2]*g[1]-e[1]*g[2])+(- \\ & (b[2]*d[1])+b[1]*d[2]+c[2]*e[1]-c[1]*e[2])*g[k]) \end{aligned}$$

$$\begin{aligned} \mathbf{f}_{pi} = & a[k]*(b[3]*e[2]*f[1]-b[2]*e[3]*f[1]+c[1]*d[3]*f[2]-b[3]*e[1]*f[2]+b[1]*e[3]*f[2]- \\ & c[1]*d[2]*f[3]+b[2]*e[1]*f[3]-b[1]*e[2]*f[3]+b[3]*d[2]*g[1]-b[2]*d[3]*g[1]- \\ & b[3]*d[1]*g[2]+b[1]*d[3]*g[2]-c[1]*e[3]*g[2]+c[3]*(d[2]*f[1]-d[1]*f[2]- \\ & e[2]*g[1]+e[1]*g[2])+(b[2]*d[1]-b[1]*d[2]+c[1]*e[2])*g[3]+c[2]*(-(d[3]*f[1])+d[1]*f[3]+e[3]*g[1]- \\ & e[1]*g[3]))+a[2]*(b[k]*e[3]*f[1]-b[3]*e[k]*f[1]-b[k]*e[1]*f[3]+b[1]*e[k]*f[3]+c[3]*d[1]*f[k]- \\ & c[1]*d[3]*f[k]+b[3]*e[1]*f[k]-b[1]*e[3]*f[k]+b[k]*d[3]*g[1]+c[3]*e[k]*g[1]- \\ & (b[k]*d[1]+c[1]*e[k])*g[3]+d[k]*(-(c[3]*f[1])+c[1]*f[3]-b[3]*g[1]+b[1]*g[3])+c[k]*(d[3]*f[1]- \\ & d[1]*f[3]-e[3]*g[1]+e[1]*g[3]))+a[1]*(-(b[k]*e[3]*f[2])+b[3]*e[k]*f[2]+b[k]*e[2]*f[3]- \\ & b[2]*e[k]*f[3]-c[3]*d[2]*f[k]+c[2]*d[3]*f[k]-b[3]*e[2]*f[k]+b[2]*e[3]*f[k]-b[k]*d[3]*g[2]- \\ & c[3]*e[k]*g[2]+b[k]*d[2]*g[3]+c[2]*e[k]*g[3]+d[k]*(c[3]*f[2]-c[2]*f[3]+b[3]*g[2]- \\ & b[2]*g[3])+c[k]*(-(d[3]*f[2])+d[2]*f[3]+e[3]*g[2]-e[2]*g[3]))+a[2]*b[3]*d[1]*g[k]- \end{aligned}$$

$$\begin{aligned}
 & a[1]*b[3]*d[2]*g[k]-a[2]*b[1]*d[3]*g[k]+a[1]*b[2]*d[3]*g[k]- \\
 & a[2]*c[3]*e[1]*g[k]+a[1]*c[3]*e[2]*g[k]+a[2]*c[1]*e[3]*g[k]-a[1]*c[2]*e[3]*g[k]+a[3]*(- \\
 & (b[k]*e[2]*f[1])+b[2]*e[k]*f[1]+b[k]*e[1]*f[2]-b[1]*e[k]*f[2]-c[2]*d[1]*f[k]+c[1]*d[2]*f[k]- \\
 & b[2]*e[1]*f[k]+b[1]*e[2]*f[k]-b[k]*d[2]*g[1]- \\
 & c[2]*e[k]*g[1]+b[k]*d[1]*g[2]+c[1]*e[k]*g[2]+d[k]*(c[2]*f[1]-c[1]*f[2]+b[2]*g[1]- \\
 & b[1]*g[2])+c[k]*(-(d[2]*f[1])+d[1]*f[2]+e[2]*g[1]-e[1]*g[2])+(-(b[2]*d[1])+b[1]*d[2]+c[2]*e[1]- \\
 & c[1]*e[2])*g[k])
 \end{aligned}$$

$$\begin{aligned}
 g_{pi}= & a[k]*(-(b[3]*e[2]*f[1])+b[2]*e[3]*f[1]+c[1]*d[3]*f[2]+b[3]*e[1]*f[2]-b[1]*e[3]*f[2]- \\
 & c[1]*d[2]*f[3]-b[2]*e[1]*f[3]+b[1]*e[2]*f[3]-b[3]*d[2]*g[1]+b[2]*d[3]*g[1]+b[3]*d[1]*g[2]- \\
 & b[1]*d[3]*g[2]-c[1]*e[3]*g[2]+c[3]*(d[2]*f[1]-d[1]*f[2]-e[2]*g[1]+e[1]*g[2])+(- \\
 & (b[2]*d[1])+b[1]*d[2]+c[1]*e[2])*g[3]+c[2]*(-(d[3]*f[1])+d[1]*f[3]+e[3]*g[1]-e[1]*g[3]))+a[2]*(- \\
 & (b[k]*e[3]*f[1])+b[3]*e[k]*f[1]+b[k]*e[1]*f[3]-b[1]*e[k]*f[3]+c[3]*d[1]*f[k]-c[1]*d[3]*f[k]- \\
 & b[3]*e[1]*f[k]+b[1]*e[3]*f[k]-b[k]*d[3]*g[1]+c[3]*e[k]*g[1]+b[k]*d[1]*g[3]-c[1]*e[k]*g[3]+d[k]*(- \\
 & (c[3]*f[1])+c[1]*f[3]+b[3]*g[1]-b[1]*g[3])+c[k]*(d[3]*f[1]-d[1]*f[3]- \\
 & e[3]*g[1]+e[1]*g[3]))+a[1]*(b[k]*e[3]*f[2]-b[3]*e[k]*f[2]-b[k]*e[2]*f[3]+b[2]*e[k]*f[3]- \\
 & c[3]*d[2]*f[k]+c[2]*d[3]*f[k]+b[3]*e[2]*f[k]-b[2]*e[3]*f[k]+b[k]*d[3]*g[2]-c[3]*e[k]*g[2]- \\
 & b[k]*d[2]*g[3]+c[2]*e[k]*g[3]+d[k]*(c[3]*f[2]-c[2]*f[3]-b[3]*g[2]+b[2]*g[3])+c[k]*(- \\
 & (d[3]*f[2])+d[2]*f[3]+e[3]*g[2]-e[2]*g[3]))- \\
 & a[2]*b[3]*d[1]*g[k]+a[1]*b[3]*d[2]*g[k]+a[2]*b[1]*d[3]*g[k]-a[1]*b[2]*d[3]*g[k]- \\
 & a[2]*c[3]*e[1]*g[k]+a[1]*c[3]*e[2]*g[k]+a[2]*c[1]*e[3]*g[k]- \\
 & a[1]*c[2]*e[3]*g[k]+a[3]*(b[k]*e[2]*f[1]-b[2]*e[k]*f[1]-b[k]*e[1]*f[2]+b[1]*e[k]*f[2]- \\
 & c[2]*d[1]*f[k]+c[1]*d[2]*f[k]+b[2]*e[1]*f[k]-b[1]*e[2]*f[k]+b[k]*d[2]*g[1]-c[2]*e[k]*g[1]- \\
 & b[k]*d[1]*g[2]+c[1]*e[k]*g[2]+d[k]*(c[2]*f[1]-c[1]*f[2]-b[2]*g[1]+b[1]*g[2])+c[k]*(- \\
 & (d[2]*f[1])+d[1]*f[2]+e[2]*g[1]-e[1]*g[2])+((b[2]*d[1]-b[1]*d[2]+c[2]*e[1]-c[1]*e[2])*g[k])
 \end{aligned}$$

$i=1, 2$

$k=i+3$

EK 2. Fonksiyon Sentezi Modülü Denklem (3.39) Katsayıları

$x_1=ap[1]*ap[1]$, $x_2=ap[2]*ap[2]$, $x_3=bp[1]*bp[1]$, $x_4=bp[2]*bp[2]$, $x_5=cp[1]*cp[1]$, $x_6=cp[2]*cp[2]$,
 $x_7=dp[1]*dp[1]$, $x_8=dp[2]*dp[2]$, $x_9=ep[1]*ep[1]$, $x_{10}=ep[2]*ep[2]$, $x_{11}=fp[1]*fp[1]$,
 $x_{12}=fp[2]*fp[2]$, $x_{13}=gp[1]*gp[1]$, $x_{14}=gp[2]*gp[2]$ olmak üzere;

$$\begin{aligned} w_0 = & 64*(x_3*(x_8-x_{10})+2*bp[1]*cp[1]*(x_8-x_{10})+x_5*(x_8-x_{10})+sqr(dp[1]*ep[2]-dp[2]*(ep[1]+fp[1]))- \\ & 2*(bp[1]+cp[1])*cp[2]*(dp[1]*dp[2]-ep[2]*(ep[1]+fp[1]))+x_4*(x_7-sqr(ep[1]+fp[1]))+x_6*(x_7- \\ & sqr(ep[1]+fp[1]))+x_2*(sqr(bp[1]+cp[1])+sqr(ep[1]+fp[1]))+2*bp[2]*((-bp[1]-cp[1])*(dp[1]*dp[2]- \\ & ep[2]*(ep[1]+fp[1]))+cp[2]*(x_7-sqr(ep[1]+fp[1])))+2*(cp[1]*cp[2]*ep[1]-dp[1]*dp[2]*ep[1]- \\ & x_3*ep[2]-x_5*ep[2]+x_7*ep[2]+(cp[1]*cp[2]-dp[1]*dp[2])*fp[1]+bp[2]*cp[1]*(ep[1]+fp[1])+bp[1]*(- \\ & 2*cp[1]*ep[2]+(bp[2]+cp[2])*(ep[1]+fp[1])))*fp[2]-(sqr(bp[1]+cp[1])- \\ & x_7)*x_{12}+x_1*(sqr(bp[2]+cp[2])+sqr(ep[2]+fp[2]))+2*ap[1]*(x_4*dp[1]+x_6*dp[1]- \\ & (bp[1]+cp[1])*cp[2]*dp[2]+bp[2]*(2*cp[2]*dp[1]-(bp[1]+cp[1])*dp[2])+(ep[2]+fp[2])*(- \\ & dp[2]*(ep[1]+fp[1])+dp[1]*(ep[2]+fp[2])))+2*ap[2]*(-bp[2]*cp[1]*dp[1]- \\ & cp[1]*cp[2]*dp[1]+x_3*dp[2]+x_5*dp[2]+bp[1]*((-bp[2]-cp[2])*dp[1]+2*cp[1]*dp[2])+dp[2]*x_9- \\ & dp[1]*ep[1]*ep[2]+2*dp[2]*ep[1]*fp[1]-dp[1]*ep[2]*fp[1]+dp[2]*x_{11}-dp[1]*(ep[1]+fp[1])*fp[2]- \\ & ap[1]*((bp[1]+cp[1])*(bp[2]+cp[2])+(ep[1]+fp[1])*(ep[2]+fp[2])))) \end{aligned}$$

$$\begin{aligned} w_1 = & -512*((-cp[2]*(ap[1]+dp[1])+cp[1]*dp[2])*(- \\ & dp[2]*ep[1]+(ap[1]+dp[1])*ep[2])+ep[2]*(cp[2]*ep[1]-cp[1]*ep[2])*fp[1]+cp[2]*ep[2]*x_{11}+x_2*(- \\ & cp[1]*ep[1]+bp[1]*fp[1])-(cp[1]*ep[2]*(- \\ & ep[1]+fp[1])+cp[2]*ep[1]*(ep[1]+fp[1]))*fp[2]+cp[1]*ep[1]*x_{12}+x_3*cp[2]*(ep[2]+fp[2])-(x_6*dp[1]- \\ & cp[1]*cp[2]*dp[2]+(ep[2]+fp[2])*(- \\ & dp[2]*(ep[1]+fp[1])+dp[1]*(ep[2]+fp[2]))+ap[1]*(x_6+sqr(ep[2]+fp[2])))*gp[1]+x_4*(cp[1]*(ep[1]+fp[1])- \\ & (ap[1]+dp[1])*gp[1])+bp[1]*(x_8*fp[1]-x_6*(ep[1]+fp[1])-(ap[1]+dp[1])*dp[2]*fp[2]- \\ & (ep[2]+fp[2])*(ep[2]*fp[1]- \\ & ep[1]*fp[2])+cp[2]*(cp[1]*(ep[2]+fp[2])+dp[2]*gp[1]))+(cp[1]*cp[2]*dp[1]-x_3*dp[2]- \\ & x_5*dp[2]+bp[1]*(cp[2]*dp[1]-2*cp[1]*dp[2])-dp[2]*x_9+dp[1]*ep[1]*ep[2]- \\ & 2*dp[2]*ep[1]*fp[1]+dp[1]*ep[2]*fp[1]- \\ & dp[2]*x_{11}+dp[1]*(ep[1]+fp[1])*fp[2]+ap[1]*((bp[1]+cp[1])*cp[2]+(ep[1]+fp[1])*(ep[2]+fp[2])))*gp[\\ & 2]+ap[2]*(cp[1]*(-2*dp[2]*ep[1]+dp[1]*ep[2])-bp[2]*dp[1]*fp[1]+ap[1]*(cp[2]*ep[1]+cp[1]*ep[2]- \\ & bp[2]*fp[1]-bp[1]*fp[2])+bp[1]*(2*dp[2]*fp[1]- \\ & dp[1]*fp[2])+bp[2]*(bp[1]+cp[1])+(ep[1]+fp[1])*(ep[2]+fp[2]))*gp[1]+cp[2]*(dp[1]*ep[1]+(bp[1]+c \\ & p[1])*gp[1])-(sqr(bp[1]+cp[1])+sqr(ep[1]+fp[1]))*gp[2]+bp[2]*(fp[1]*((-ap[1]- \\ & dp[1])*dp[2]+ep[2]*(ep[1]+fp[1]))+(sqr(ap[1]+dp[1])-cp[1]*(ep[1]+fp[1]))*fp[2]-x_5*(ep[2]+fp[2])- \\ & 2*cp[2]*(ap[1]+dp[1])*gp[1]+cp[1]*(cp[2]*(ep[1]+fp[1])+dp[2]*gp[1]+(ap[1]+dp[1])*gp[2])+bp[1]* \\ & (-cp[2]*(ep[1]+fp[1])-cp[1]*(ep[2]+fp[2])+dp[2]*gp[1]+(ap[1]+dp[1])*gp[2])))) \end{aligned}$$

$$\begin{aligned} w_2 = & 512 * (-\text{sqr}(\text{dp}[2] * \text{ep}[1] - \text{dp}[1] * \text{ep}[2]) + 2 * \text{cp}[1] * \text{cp}[2] * (\text{dp}[1] * \text{dp}[2] - \\ & \text{ep}[1] * \text{ep}[2]) + x1 * (x4 + x6 + x10) + x5 * (-x8 + x10) - 4 * \text{dp}[2] * (\text{dp}[2] * \text{ep}[1] - \text{dp}[1] * \text{ep}[2]) * \text{fp}[1] - \\ & (x8 + 2 * x10) * x11 - x6 * (x7 - x9 + x11) + x2 * (x3 + x5 + x9 + x11) - 2 * (2 * x7 * \text{ep}[2] - \\ & (\text{cp}[1] * \text{cp}[2] + 2 * \text{ep}[1] * \text{ep}[2]) * \text{fp}[1] - \text{dp}[1] * \text{dp}[2] * (2 * \text{ep}[1] + \text{fp}[1])) * \text{fp}[2] - (-x1 + x5 + x7 + 2 * x9) * x12 - \\ & x3 * (2 * x6 + x8 + x10 - x12) + \text{bp}[1] * (-4 * \text{cp}[1] * x8 + \text{cp}[2] * (4 * \text{dp}[1] * \text{dp}[2] + 6 * \text{cp}[2] * \text{fp}[1] - 6 * \text{cp}[1] * \text{fp}[2])) - \\ & 4 * (\text{cp}[2] * (\text{dp}[2] * \text{ep}[1] - 2 * \text{dp}[1] * \text{ep}[2]) + \text{dp}[2] * (\text{cp}[1] * \text{ep}[2] - \\ & \text{bp}[1] * \text{fp}[2])) * \text{gp}[1] + 2 * (x6 + \text{sqr}(\text{ep}[2] + \text{fp}[2])) * x13 - x4 * (2 * x5 + x7 + x9 - x11 - \\ & 2 * x13) + 2 * \text{bp}[2] * (\text{cp}[1] * (2 * \text{dp}[1] * \text{dp}[2] - \\ & 3 * \text{ep}[2] * \text{fp}[1] + 3 * \text{ep}[1] * \text{fp}[2]) + \text{bp}[1] * (2 * \text{cp}[1] * \text{cp}[2] + \text{dp}[1] * \text{dp}[2] + \text{ep}[1] * \text{ep}[2] - \\ & \text{fp}[1] * \text{fp}[2]) + 2 * (\text{dp}[2] * \text{fp}[1] - 2 * \text{dp}[1] * \text{fp}[2]) * \text{gp}[1] - 2 * \text{cp}[2] * (x7 - x13)) - 4 * (\text{cp}[2] * \text{dp}[1] * \text{ep}[1] - \\ & 2 * \text{cp}[1] * \text{dp}[2] * \text{ep}[1] + \text{cp}[1] * \text{dp}[1] * \text{ep}[2] - \text{bp}[2] * \text{dp}[1] * \text{fp}[1] + 2 * \text{bp}[1] * \text{dp}[2] * \text{fp}[1] - \\ & \text{bp}[1] * \text{dp}[1] * \text{fp}[2] + ((\text{bp}[1] + \text{cp}[1]) * (\text{bp}[2] + \text{cp}[2]) + (\text{ep}[1] + \text{fp}[1]) * (\text{ep}[2] + \text{fp}[2])) * \text{gp}[1]) * \text{gp}[2] + 2 * (\text{sqr}(\text{bp}[1] + \text{cp}[1]) + \text{sqr}(\text{ep}[1] + \text{fp}[1])) * x14 - 2 * \text{ap}[2] * (-\text{bp}[2] * \text{cp}[1] * \text{dp}[1] - \\ & \text{bp}[1] * \text{cp}[2] * \text{dp}[1] + 2 * \text{bp}[1] * \text{cp}[1] * \text{dp}[2] + 2 * \text{dp}[2] * \text{ep}[1] * \text{fp}[1] - \text{dp}[1] * \text{ep}[2] * \text{fp}[1] - \\ & \text{dp}[1] * \text{ep}[1] * \text{fp}[2] + \text{ap}[1] * (\text{bp}[1] * \text{bp}[2] + \text{cp}[1] * \text{cp}[2] + \text{ep}[1] * \text{ep}[2] + \text{fp}[1] * \text{fp}[2]) + 2 * \text{cp}[2] * \text{ep}[1] * \text{gp}[1] + \\ & 2 * \text{cp}[1] * \text{ep}[2] * \text{gp}[1] - 2 * \text{bp}[2] * \text{fp}[1] * \text{gp}[1] - 2 * \text{bp}[1] * \text{fp}[2] * \text{gp}[1] - \\ & 4 * \text{cp}[1] * \text{ep}[1] * \text{gp}[2] + 4 * \text{bp}[1] * \text{fp}[1] * \text{gp}[2]) + 2 * \text{ap}[1] * (\text{bp}[1] * \text{cp}[2] * \text{dp}[2] + \text{dp}[2] * \text{cp}[2] * \text{fp}[1] + \text{dp}[2] * \text{ep}[1] * \text{fp}[2] - 2 * \text{dp}[1] * \text{ep}[2] * \text{fp}[2] + 4 * \text{cp}[2] * \text{ep}[2] * \text{gp}[1] - 2 * (\text{cp}[2] * \text{ep}[1] + \text{cp}[1] * \text{ep}[2] - \\ & \text{bp}[1] * \text{fp}[2]) * \text{gp}[2] + \text{bp}[2] * (-2 * \text{cp}[2] * \text{dp}[1] + \text{cp}[1] * \text{dp}[2] - 4 * \text{fp}[2] * \text{gp}[1] + 2 * \text{fp}[1] * \text{gp}[2])))) \end{aligned}$$

$$\begin{aligned} w_3 = & -512 * (11 * \text{cp}[1] * \text{dp}[2] * (\text{dp}[2] * \text{ep}[1] - \text{dp}[1] * \text{ep}[2]) + \text{cp}[2] * (-11 * \text{dp}[1] * \text{dp}[2] * \text{ep}[1] + (- \\ & 5 * x1 + 5 * x3 + 11 * x7) * \text{ep}[2]) - 3 * \text{ap}[1] * (\text{cp}[1] * \text{dp}[2] * \text{ep}[2] + \text{cp}[2] * (\text{dp}[2] * \text{ep}[1] - 2 * \text{dp}[1] * \text{ep}[2])) - \\ & 3 * \text{cp}[2] * \text{ep}[1] * \text{ep}[2] * \text{fp}[1] + 3 * \text{cp}[1] * x10 * \text{fp}[1] + 5 * \text{cp}[2] * \text{ep}[2] * x11 + x2 * (- \\ & 5 * \text{cp}[1] * \text{ep}[1] + 5 * \text{bp}[1] * \text{fp}[1]) - (3 * x3 * \text{cp}[2] + \text{cp}[2] * \text{ep}[1]) * (- \\ & 3 * \text{ep}[1] + 5 * \text{fp}[1]) + \text{cp}[1] * \text{ep}[2] * (3 * \text{ep}[1] + 5 * \text{fp}[1])) * \text{fp}[2] + 5 * \text{cp}[1] * \text{ep}[1] * x12 - (- \\ & 3 * x6 * \text{dp}[1] + 3 * \text{cp}[1] * \text{cp}[2] * \text{dp}[2] + \text{ep}[2] * (-3 * \text{dp}[1] * \text{ep}[2] + \text{dp}[2] * (3 * \text{ep}[1] + 11 * \text{fp}[1])) + (- \\ & 22 * \text{dp}[1] * \text{ep}[2] + \text{dp}[2] * (11 * \text{ep}[1] + 3 * \text{fp}[1])) * \text{fp}[2] - 3 * \text{dp}[1] * x12 + \text{ap}[1] * (5 * (x6 + x10) - \\ & 6 * \text{ep}[2] * \text{fp}[2] + 5 * x12)) * \text{gp}[1] - 16 * \text{cp}[2] * \text{ep}[2] * x13 + x4 * (\text{cp}[1] * (5 * \text{ep}[1] - 3 * \text{fp}[1]) + (- \\ & 5 * \text{ap}[1] + 3 * \text{dp}[1]) * \text{gp}[1]) + \text{bp}[1] * (x6 * (3 * \text{ep}[1] - 5 * \text{fp}[1]) - \\ & 11 * x8 * \text{fp}[1] + (3 * \text{ap}[1] + 11 * \text{dp}[1]) * \text{dp}[2] * \text{fp}[2] - (5 * \text{ep}[2] - 3 * \text{fp}[2]) * (\text{ep}[2] * \text{fp}[1] - \\ & \text{ep}[1] * \text{fp}[2]) + \text{cp}[2] * (\text{cp}[1] * (-3 * \text{ep}[2] + 5 * \text{fp}[2]) - 11 * \text{dp}[2] * \text{gp}[1])) + (- \\ & 3 * \text{cp}[1] * \text{cp}[2] * \text{dp}[1] + 3 * x3 * \text{dp}[2] + 3 * x5 * \text{dp}[2] + 3 * \text{dp}[2] * x9 - \\ & 3 * \text{dp}[1] * \text{ep}[1] * \text{ep}[2] + 22 * \text{dp}[2] * \text{ep}[1] * \text{fp}[1] - 11 * \text{dp}[1] * \text{ep}[2] * \text{fp}[1] + 3 * \text{dp}[2] * x11 - \\ & 11 * \text{dp}[1] * \text{ep}[1] * \text{fp}[2] - 3 * \text{dp}[1] * \text{fp}[1] * \text{fp}[2] + \text{ap}[1] * ((-3 * \text{bp}[1] + 5 * \text{cp}[1]) * \text{cp}[2] + \text{ep}[1] * (5 * \text{ep}[2] - \\ & 3 * \text{fp}[2]) + \text{fp}[1] * (-3 * \text{ep}[2] + 5 * \text{fp}[2])) + 16 * (\text{cp}[2] * \text{ep}[1] + \text{cp}[1] * \text{ep}[2]) * \text{gp}[1] + \text{bp}[1] * (- \\ & 11 * \text{cp}[2] * \text{dp}[1] + 22 * \text{cp}[1] * \text{dp}[2] - 16 * \text{fp}[2] * \text{gp}[1])) * \text{gp}[2] - 16 * (\text{cp}[1] * \text{ep}[1] - \end{aligned}$$

$$\begin{aligned} & bp[1]*fp[1]*x14+ap[2]*(6*cp[1]*dp[2]*ep[1]-3*cp[1]*dp[1]*ep[2]+3*bp[2]*dp[1]*fp[1]- \\ & 6*bp[1]*dp[2]*fp[1]+3*bp[1]*dp[1]*fp[2]+5*ap[1]*(cp[2]*ep[1]+cp[1]*ep[2]-bp[2]*fp[1]- \\ & bp[1]*fp[2])+5*bp[1]*bp[2]*gp[1]-3*bp[2]*cp[1]*gp[1]+5*ep[1]*ep[2]*gp[1]-3*ep[2]*fp[1]*gp[1]- \\ & 3*ep[1]*fp[2]*gp[1]+5*fp[1]*fp[2]*gp[1]+cp[2]*(-3*dp[1]*ep[1]+(-3*bp[1]+5*cp[1])*gp[1])-(5*x3- \\ & 6*bp[1]*cp[1]+5*(x5+x9)- \\ & 6*ep[1]*fp[1]+5*x11)*gp[2])+bp[2]*(3*ap[1]*dp[2]*fp[1]+11*dp[1]*dp[2]*fp[1]+5*ep[1]*ep[2]*fp[1]- \\ & 3*ep[2]*x11+x5*(3*ep[2]-5*fp[2])+5*x1*fp[2]-6*ap[1]*dp[1]*fp[2]-11*x7*fp[2]- \\ & 5*x9*fp[2]+3*ep[1]*fp[1]*fp[2]+6*ap[1]*cp[2]*gp[1]+22*cp[2]*dp[1]*gp[1]+16*fp[2]*x13- \\ & 16*fp[1]*gp[1]*gp[2]+bp[1]*(cp[2]*(-5*ep[1]+3*fp[1])+cp[1]*(-5*ep[2]+3*fp[2])- \\ & 3*dp[2]*gp[1]+(5*ap[1]-3*dp[1])*gp[2])-cp[1]*(cp[2]*(3*ep[1]- \\ & 5*fp[1])+11*dp[2]*gp[1]+(3*ap[1]+11*dp[1])*gp[2]))) \end{aligned}$$

$$\begin{aligned} w_4= & 256*(-\text{sqrt}(dp[2]*ep[1]-dp[1]*ep[2])+7*x1*(x4+x6+x10)+x5*(-x8+x10)+38*dp[2]*(dp[2]*ep[1]- \\ & dp[1]*ep[2])*fp[1]-(x8+8*x10)*x11+x6*(-x7+x9+10*ep[1]*fp[1]-7*x11)+x2*(7*x3- \\ & 10*bp[1]*cp[1]+7*(x5+x9)-10*ep[1]*fp[1]+7*x11)+2*cp[1]*cp[2]*(dp[1]*dp[2]- \\ & ep[2]*(ep[1]+5*fp[1]))+2*(5*x5*ep[2]+19*x7*ep[2]+dp[1]*dp[2]*(-19*ep[1]+fp[1])+cp[1]*cp[2]*(- \\ & 5*ep[1]+7*fp[1])+ep[2]*(-5*x1+8*ep[1]*fp[1]))*fp[2]-(-7*x1+7*x5+x7+8*x9)*x12+x3*(-8*x6-x8- \\ & 7*x10+10*ep[2]*fp[2]+x12)-2*bp[1]*(cp[2]*(19*dp[1]*dp[2]+5*ep[2]*(ep[1]- \\ & 3*fp[1])+(9*ep[1]+5*fp[1])*fp[2])-cp[1]*(19*x8+5*x10- \\ & 6*ep[2]*fp[2]+5*x12))+48*(cp[2]*(dp[2]*ep[1]-2*dp[1]*ep[2])+dp[2]*(cp[1]*ep[2]- \\ & bp[1]*fp[2]))*gp[1]+8*(x6+x10-6*ep[2]*fp[2]+x12)*x13+x4*(-8*x5-x7- \\ & 7*x9+10*ep[1]*fp[1]+x11+8*x13)-2*bp[2]*(5*x1*cp[2]-cp[2]*(19*x7+5*x9- \\ & 6*ep[1]*fp[1]+5*x11)+bp[1]*(-8*cp[1]*cp[2]-dp[1]*dp[2]+fp[1]*(5*ep[2]+fp[2])+ep[1]*(- \\ & 7*ep[2]+5*fp[2]))+cp[1]*(19*dp[1]*dp[2]+5*ep[1]*(ep[2]- \\ & 3*fp[2])+fp[1]*(9*ep[2]+5*fp[2]))+24*(dp[2]*fp[1]- \\ & 2*dp[1]*fp[2])*gp[1]+24*cp[2]*x13)+16*(3*(cp[2]*dp[1]*ep[1]+cp[1]*(- \\ & 2*dp[2]*ep[1]+dp[1]*ep[2])-bp[2]*dp[1]*fp[1]+2*bp[1]*dp[2]*fp[1]-bp[1]*dp[1]*fp[2])- \\ & (bp[1]*(bp[2]-3*cp[2])+cp[1]*(-3*bp[2]+cp[2])+ep[1]*(ep[2]-3*fp[2])+fp[1]*(- \\ & 3*ep[2]+fp[2]))*gp[1])*gp[2]+8*(x3-6*bp[1]*cp[1]+x5+x9-6*ep[1]*fp[1]+x11)*x14+2*ap[1]*(- \\ & 5*(x4+x6)*dp[1]+ep[2]*(-5*dp[1]*ep[2]+dp[2]*(5*ep[1]+fp[1]))+(- \\ & 2*dp[1]*ep[2]+dp[2]*(ep[1]+5*fp[1]))*fp[2]-5*dp[1]*x12-8*(cp[1]*ep[2]- \\ & bp[1]*fp[2])*gp[2]+cp[2]*((bp[1]+5*cp[1])*dp[2]+16*ep[2]*gp[1]-8*ep[1]*gp[2])+bp[2]*(- \\ & 2*cp[2]*dp[1]+(5*bp[1]+cp[1])*dp[2]+8*(-2*fp[2]*gp[1]+fp[1]*gp[2])))-2*ap[2]*(- \\ & bp[2]*cp[1]*dp[1]+5*(-cp[1]*cp[2]*dp[1]+x5*dp[2]+dp[2]*(x3+x9)- \\ & dp[1]*ep[1]*ep[2])+(2*dp[2]*ep[1]-dp[1]*ep[2])*fp[1]+5*dp[2]*x11- \\ & dp[1]*(ep[1]+5*fp[1])*fp[2]+ap[1]*(bp[1]*(7*bp[2]-5*cp[2])+cp[1]*(- \\ & 5*bp[2]+7*cp[2])+ep[1]*(7*ep[2]-5*fp[2])+fp[1]*(- \end{aligned}$$

$$5*ep[2]+7*fp[2]))+8*cp[2]*ep[1]*gp[1]+8*cp[1]*ep[2]*gp[1]-8*bp[2]*fp[1]*gp[1]-16*cp[1]*ep[1]*gp[2]-bp[1]*((5*bp[2]+cp[2])*dp[1]-2*(cp[1]*dp[2]-4*fp[2]*gp[1]+8*fp[1]*gp[2]))))$$

$$\begin{aligned} w_5 = & -512*(25*cp[1]*dp[2]*(-dp[2]*ep[1]+dp[1]*ep[2])- \\ & 15*ap[1]*(cp[1]*dp[2]*ep[2]+cp[2]*(dp[2]*ep[1]-2*dp[1]*cp[2]))+cp[2]*(9*(- \\ & x1+x3)*ep[2]+25*dp[1]*(dp[2]*ep[1]-dp[1]*ep[2]))- \\ & 15*cp[2]*ep[1]*ep[2]*fp[1]+15*cp[1]*x10*fp[1]+9*cp[2]*ep[2]*x11-9*x2*(cp[1]*ep[1]- \\ & bp[1]*fp[1])-3*(5*x3*cp[2]+cp[2]*ep[1]*(- \\ & 5*ep[1]+3*fp[1])+cp[1]*ep[2]*(5*ep[1]+3*fp[1]))*fp[2]+9*cp[1]*ep[1]*x12-(3*ap[1]*(3*x6+(ep[2]- \\ & 3*fp[2])*(3*ep[2]-fp[2])))+5*(-3*x6*dp[1]+3*cp[1]*cp[2]*dp[2]-dp[1]*(ep[2]-3*fp[2])*(3*ep[2]- \\ & fp[2])+dp[2]*(ep[1]*(3*ep[2]-5*fp[2])+fp[1]*(- \\ & 5*ep[2]+3*fp[2])))*gp[1]+16*cp[2]*ep[2]*x13+3*x4*(cp[1]*(3*ep[1]-5*fp[1])+(- \\ & 3*ap[1]+5*dp[1])*gp[1])+bp[1]*(3*x6*(5*ep[1]-3*fp[1])+25*x8*fp[1]+5*(3*ap[1]- \\ & 5*dp[1])*dp[2]*fp[2]-3*(3*ep[2]-5*fp[2])*(ep[2]*fp[1]-ep[1]*fp[2])+cp[2]*(3*cp[1]*(- \\ & 5*ep[2]+3*fp[2])+25*dp[2]*gp[1]))+(5*(- \\ & 3*cp[1]*cp[2]*dp[1]+3*x3*dp[2]+3*x5*dp[2]+5*bp[1]*(cp[2]*dp[1]-2*cp[1]*dp[2])+3*dp[2]*x9- \\ & 3*dp[1]*ep[1]*cp[2]-10*dp[2]*ep[1]*fp[1]+5*dp[1]*ep[2]*fp[1]+3*dp[2]*x11+dp[1]*(5*ep[1]- \\ & 3*fp[1])*fp[2])-3*ap[1]*((5*bp[1]-3*cp[1])*cp[2]+fp[1]*(5*ep[2]-3*fp[2])+ep[1]*(- \\ & 3*ep[2]+5*fp[2]))-16*(cp[2]*ep[1]+cp[1]*ep[2]-bp[1]*fp[2])*gp[1])*gp[2]+16*(cp[1]*ep[1]- \\ & bp[1]*fp[1])*x14+3*ap[2]*(3*ap[1]*(cp[2]*ep[1]+cp[1]*ep[2]-bp[2]*fp[1]- \\ & bp[1]*fp[2])+5*(cp[1]*(2*dp[2]*ep[1]-dp[1]*ep[2])+bp[2]*dp[1]*fp[1]+bp[1]*(- \\ & 2*dp[2]*fp[1]+dp[1]*fp[2]))+3*bp[1]*bp[2]*gp[1]-5*bp[2]*cp[1]*gp[1]+3*ep[1]*ep[2]*gp[1]- \\ & 5*ep[2]*fp[1]*gp[1]-5*cp[1]*fp[2]*gp[1]+3*fp[1]*fp[2]*gp[1]+cp[2]*(-5*dp[1]*ep[1]+(- \\ & 5*bp[1]+3*cp[1])*gp[1])-((bp[1]-3*cp[1])*(3*bp[1]-cp[1])+(ep[1]-3*fp[1])*(3*ep[1]- \\ & fp[1]))*gp[2])+bp[2]*((5*(3*ap[1]-5*dp[1])*dp[2]+3*ep[2]*(3*ep[1]- \\ & 5*fp[1]))*fp[1]+3*x5*(5*ep[2]-3*fp[2])+9*x1*fp[2]-30*ap[1]*dp[1]*fp[2]+25*x7*fp[2]- \\ & 9*x9*fp[2]+15*ep[1]*fp[1]*fp[2]+30*ap[1]*cp[2]*gp[1]-50*cp[2]*dp[1]*gp[1]- \\ & 16*fp[2]*x13+16*fp[1]*gp[1]*gp[2]-3*bp[1]*(cp[2]*(3*ep[1]-5*fp[1])+cp[1]*(3*ep[2]- \\ & 5*fp[2])+5*dp[2]*gp[1]+(-3*ap[1]+5*dp[1])*gp[2])+cp[1]*(3*cp[2]*(- \\ & 5*ep[1]+3*fp[1])+5*(5*dp[2]*gp[1]+(-3*ap[1]+5*dp[1])*gp[2])))) \end{aligned}$$

$$\begin{aligned} w_6 = & 512*(x3*(2*x6+9*x8-7*x10)+9*(x6*(x7-x9)+sqr(dp[2]*ep[1]-dp[1]*ep[2]))+2*cp[1]*cp[2]*(- \\ & dp[1]*dp[2]+ep[1]*ep[2])+x5*(x8-x10))+x4*(2*x5+9*x7-(7*ep[1]-9*fp[1])*(ep[1]- \\ & fp[1]))+4*(4*x6*ep[1]-4*cp[1]*cp[2]*ep[2]+3*dp[2]*(-dp[2]*ep[1]+dp[1]*ep[2]))*fp[1]-7*x6- \\ & 9*x8-2*x10)*x11+x2*(7*x3-16*bp[1]*cp[1]+7*(x5+x9)- \\ & 16*ep[1]*fp[1]+7*x11)+2*bp[1]*(2*cp[1]*(-3*x8+4*x10)+cp[2]*(6*dp[1]*dp[2]+ep[2]*(- \end{aligned}$$

$$\begin{aligned}
 & 8*ep[1]+5*fp[1]))-2*bp[2]*(cp[1]*(-6*dp[1]*dp[2]+ep[2]*(8*ep[1]-11*fp[1]))+2*cp[2]*(3*x7- \\
 & 4*sqr(ep[1]-fp[1]))+bp[1]*(2*cp[1]*cp[2]+9*dp[1]*dp[2]+ep[2]*(-7*ep[1]+8*fp[1])))- \\
 & 16*bp[1]*bp[2]*ep[1]*fp[2]+10*bp[2]*cp[1]*ep[1]*fp[2]+22*bp[1]*cp[2]*ep[1]*fp[2]- \\
 & 16*cp[1]*cp[2]*ep[1]*fp[2]+12*dp[1]*dp[2]*ep[1]*fp[2]+16*x3*ep[2]*fp[2]- \\
 & 32*bp[1]*cp[1]*ep[2]*fp[2]+16*x5*ep[2]*fp[2]-12*x7*ep[2]*fp[2]+18*bp[1]*bp[2]*fp[1]*fp[2]- \\
 & 16*bp[2]*cp[1]*fp[1]*fp[2]-16*bp[1]*cp[2]*fp[1]*fp[2]+14*cp[1]*cp[2]*fp[1]*fp[2]- \\
 & 18*dp[1]*dp[2]*fp[1]*fp[2]-4*ep[1]*ep[2]*fp[1]*fp[2]-9*x3*x12+16*bp[1]*cp[1]*x12- \\
 & 7*x5*x12+9*x7*x12+2*x9*x12+x1*(7*x4-16*bp[2]*cp[2]+7*(x6+x10)- \\
 & 16*ep[2]*fp[2]+7*x12)+4*cp[2]*dp[2]*ep[1]*gp[1]- \\
 & 8*cp[2]*dp[1]*ep[2]*gp[1]+4*cp[1]*dp[2]*ep[2]*gp[1]- \\
 & 4*bp[2]*dp[2]*fp[1]*gp[1]+8*bp[2]*dp[1]*fp[2]*gp[1]-4*bp[1]*dp[2]*fp[2]*gp[1]-2*x4*x13- \\
 & 4*bp[2]*cp[2]*x13-2*x6*x13-2*x10*x13-4*ep[2]*fp[2]*x13-2*x12*x13+4*(cp[2]*dp[1]*ep[1]- \\
 & 2*cp[1]*dp[2]*ep[1]+cp[1]*dp[1]*ep[2]-bp[2]*dp[1]*fp[1]+2*bp[1]*dp[2]*fp[1]- \\
 & bp[1]*dp[1]*fp[2]+((bp[1]+cp[1])*(bp[2]+cp[2])+(ep[1]+fp[1])*(ep[2]+fp[2]))*gp[1]*gp[2]- \\
 & 2*(sqr(bp[1]+cp[1])+sqr(ep[1]+fp[1]))*x14-2*ap[2]*(9*bp[2]*cp[1]*dp[1]+8*(- \\
 & cp[1]*cp[2]*dp[1]+x5*dp[2]+dp[2]*(x3+x9)-dp[1]*ep[1]*ep[2])-9*(2*dp[2]*ep[1]- \\
 & dp[1]*ep[2])*fp[1]+8*dp[2]*x11+dp[1]*(9*ep[1]-8*fp[1])*fp[2]+ap[1]*(bp[1]*(7*bp[2]- \\
 & 8*cp[2])+cp[1]*(-8*bp[2]+7*cp[2])+ep[1]*(7*ep[2]-8*fp[2])+fp[1]*(-8*ep[2]+7*fp[2]))- \\
 & 2*cp[2]*ep[1]*gp[1]-2*cp[1]*ep[2]*gp[1]+2*bp[2]*fp[1]*gp[1]+4*cp[1]*ep[1]*gp[2]+bp[1]*((- \\
 & 8*bp[2]+9*cp[2])*dp[1]-18*cp[1]*dp[2]+2*fp[2]*gp[1]-4*fp[1]*gp[2]))+2*ap[1]*(-8*x4*dp[1]- \\
 & 8*x6*dp[1]+8*dp[2]*ep[1]*ep[2]-8*dp[1]*x10-9*dp[2]*ep[2]*fp[1]- \\
 & 9*dp[2]*ep[1]*fp[2]+18*dp[1]*ep[2]*fp[2]+8*dp[2]*fp[1]*fp[2]-8*dp[1]*x12+2*(cp[1]*ep[2]- \\
 & bp[1]*fp[2])*gp[2]+cp[2]*((-9*bp[1]+8*cp[1])*dp[2]- \\
 & 4*ep[2]*gp[1]+2*ep[1]*gp[2])+bp[2]*(18*cp[2]*dp[1]+(8*bp[1]-9*cp[1])*dp[2]+4*fp[2]*gp[1]- \\
 & 2*fp[1]*gp[2]))))
 \end{aligned}$$

EK 3. Yörünge Sentezi Modülü Denklem (3.7) Katsayıları

$$a_{pi}=2*(a[2]*(c[k]*(b[3]*f[1]+b[1]*f[3])+b[k]*(c[3]*f[1]-c[1]*f[3]))+a[k]*(b[3]*(c[2]*f[1]-c[1]*f[2])+b[2]*(-c[3]*f[1]+c[1]*f[3])+b[1]*(c[3]*f[2]-c[2]*f[3]))+a[1]*(c[k]*(b[3]*f[2]-b[2]*f[3])+b[k]*(-c[3]*f[2]+c[2]*f[3]))+a[2]*b[3]*c[1]*f[k]-a[1]*b[3]*c[2]*f[k]-a[2]*b[1]*c[3]*f[k]+a[1]*b[2]*c[3]*f[k]+a[3]*(b[k]*(-c[2]*f[1]+c[1]*f[2])+b[2]*(c[k]*f[1]-c[1]*f[k])+b[1]*(-c[k]*f[2]+c[2]*f[k]))))$$

$$b_{pi}=-2*(a[k]*(c[2]*dc[3]*f[1]+b[3]*ds[2]*f[1]-b[2]*ds[3]*f[1]-c[1]*dc[3]*f[2]-b[3]*ds[1]*f[2]+b[1]*ds[3]*f[2]+c[3]*(-dc[2]*f[1]+dc[1]*f[2])+(-c[2]*dc[1]+c[1]*dc[2]+b[2]*ds[1]-b[1]*ds[2])*f[3])+a[2]*((-c[k]*dc[3]+c[3]*dc[k]+b[k]*ds[3]-b[3]*ds[k])*f[1]+(c[k]*dc[1]-c[1]*dc[k]-b[k]*ds[1]+b[1]*ds[k])*f[3])+a[1]*((c[k]*dc[3]-c[3]*dc[k]-b[k]*ds[3]+b[3]*ds[k])*f[2]+(-c[k]*dc[2]+c[2]*dc[k]+b[k]*ds[2]-b[2]*ds[k])*f[3])-a[2]*c[3]*dc[1]*f[k]+a[1]*c[3]*dc[2]*f[k]+a[2]*c[1]*dc[3]*f[k]-a[1]*c[2]*dc[3]*f[k]+a[2]*b[3]*ds[1]*f[k]-a[1]*b[3]*ds[2]*f[k]-a[2]*b[1]*ds[3]*f[k]+a[1]*b[2]*ds[3]*f[k]+a[3]*(-c[2]*dc[k]*f[1]-b[k]*ds[2]*f[1]+b[2]*ds[k]*f[1]+c[1]*dc[k]*f[2]+b[k]*ds[1]*f[2]-b[1]*ds[k]*f[2]+c[k]*(dc[2]*f[1]-dc[1]*f[2])+(c[2]*dc[1]-c[1]*dc[2]-b[2]*ds[1]+b[1]*ds[2])*f[k])-b[k]*c[3]*f[2]*vc[1]+b[3]*c[k]*f[2]*vc[1]+b[k]*c[2]*f[3]*vc[1]-b[2]*c[k]*f[3]*vc[1]-b[3]*c[2]*f[k]*vc[1]+b[2]*c[3]*f[k]*vc[1]+b[k]*c[3]*f[1]*vc[2]-b[3]*c[k]*f[1]*vc[2]-b[k]*c[1]*f[3]*vc[2]+b[1]*c[k]*f[3]*vc[2]+b[3]*c[1]*f[k]*vc[2]-b[1]*c[3]*f[k]*vc[2]-b[k]*c[2]*f[1]*vc[3]+b[2]*c[k]*f[1]*vc[3]+b[k]*c[1]*f[2]*vc[3]-b[1]*c[k]*f[2]*vc[3]-b[2]*c[1]*f[k]*vc[3]+b[1]*c[2]*f[k]*vc[3]+b[3]*c[2]*f[1]*vc[k]-b[2]*c[3]*f[1]*vc[k]-b[3]*c[1]*f[2]*vc[k]+b[1]*c[3]*f[2]*vc[k]+b[2]*c[1]*f[3]*vc[k]-b[1]*c[2]*f[3]*vc[k])$$

$$c_{pi}=-2*(a[k]*(-b[2]*dc[3]*f[1]+c[3]*ds[2]*f[1]-c[2]*ds[3]*f[1]+b[1]*dc[3]*f[2]-c[3]*ds[1]*f[2]+c[1]*ds[3]*f[2]+b[3]*(dc[2]*f[1]-dc[1]*f[2])+(b[2]*dc[1]-b[1]*dc[2]+c[2]*ds[1]-c[1]*ds[2])*f[3])+a[2]*((b[k]*dc[3]-b[3]*dc[k]+c[k]*ds[3]-c[3]*ds[k])*f[1]+(-b[k]*dc[1]+b[1]*dc[k]-c[k]*ds[1]+c[1]*ds[k])*f[3])+a[1]*((-b[k]*dc[3]+b[3]*dc[k]-c[k]*ds[3]+c[3]*ds[k])*f[2]+(b[k]*dc[2]-b[2]*dc[k]+c[k]*ds[2]-c[2]*ds[k])*f[3])+a[2]*b[3]*dc[1]*f[k]-a[1]*b[3]*dc[2]*f[k]-a[2]*b[1]*dc[3]*f[k]+a[1]*b[2]*dc[3]*f[k]+a[2]*c[3]*ds[1]*f[k]-a[1]*c[3]*ds[2]*f[k]-a[2]*c[1]*ds[3]*f[k]+a[1]*c[2]*ds[3]*f[k]+a[3]*(b[2]*dc[k]*f[1]-c[k]*ds[2]*f[1]+c[2]*ds[k]*f[1]-b[1]*dc[k]*f[2]+c[k]*ds[1]*f[2]-c[1]*ds[k]*f[2]+b[k]*(-dc[2]*f[1]+dc[1]*f[2])-(b[2]*dc[1]-b[1]*dc[2]+c[2]*ds[1]-c[1]*ds[2])*f[k])-b[k]*c[3]*f[2]*vs[1]+b[3]*c[k]*f[2]*vs[1]+b[k]*c[2]*f[3]*vs[1]-b[2]*c[k]*f[3]*vs[1]-b[3]*c[2]*f[k]*vs[1]+b[2]*c[3]*f[k]*vs[1]+b[k]*c[3]*f[1]*vs[2]-b[3]*c[k]*f[1]*vs[2]-b[k]*c[1]*f[3]*vs[2]+b[1]*c[k]*f[3]*vs[2]+b[3]*c[1]*f[k]*vs[2]-b[1]*c[3]*f[k]*vs[2]-$$

$$\begin{aligned} & b[k]*c[2]*f[1]*vs[3]+b[2]*c[k]*f[1]*vs[3]+b[k]*c[1]*f[2]*vs[3]-b[1]*c[k]*f[2]*vs[3]- \\ & b[2]*c[1]*f[k]*vs[3]+b[1]*c[2]*f[k]*vs[3]+b[3]*c[2]*f[1]*vs[k]-b[2]*c[3]*f[1]*vs[k]- \\ & b[3]*c[1]*f[2]*vs[k]+b[1]*c[3]*f[2]*vs[k]+b[2]*c[1]*f[3]*vs[k]-b[1]*c[2]*f[3]*vs[k] \end{aligned}$$

$$\begin{aligned} d_{pr} = & 2*a[2]*dc[k]*ds[3]*f[1]-2*a[2]*dc[3]*ds[k]*f[1]- \\ & 2*a[1]*dc[k]*ds[3]*f[2]+2*a[1]*dc[3]*ds[k]*f[2]- \\ & 2*a[2]*dc[k]*ds[1]*f[3]+2*a[1]*dc[k]*ds[2]*f[3]+2*a[2]*dc[1]*ds[k]*f[3]- \\ & 2*a[1]*dc[2]*ds[k]*f[3]+2*a[k]*(dc[3]*(ds[2]*f[1]-ds[1]*f[2])+dc[2]*(- \\ & ds[3]*f[1]+ds[1]*f[3])+dc[1]*(ds[3]*f[2]-ds[2]*f[3]))+2*a[2]*dc[3]*ds[1]*f[k]- \\ & 2*a[1]*dc[3]*ds[2]*f[k]-2*a[2]*dc[1]*ds[3]*f[k]+2*a[1]*dc[2]*ds[3]*f[k]+2*a[3]*(dc[k]*(- \\ & ds[2]*f[1]+ds[1]*f[2])+dc[2]*(ds[k]*f[1]-ds[1]*f[k])+dc[1]*(- \\ & ds[k]*f[2]+ds[2]*f[k]))+c[k]*dc[3]*f[2]*vc[1]-c[3]*dc[k]*f[2]*vc[1]- \\ & b[k]*ds[3]*f[2]*vc[1]+b[3]*ds[k]*f[2]*vc[1]- \\ & c[k]*dc[2]*f[3]*vc[1]+c[2]*dc[k]*f[3]*vc[1]+b[k]*ds[2]*f[3]*vc[1]- \\ & b[2]*ds[k]*f[3]*vc[1]+c[3]*dc[2]*f[k]*vc[1]-c[2]*dc[3]*f[k]*vc[1]- \\ & b[3]*ds[2]*f[k]*vc[1]+b[2]*ds[3]*f[k]*vc[1]- \\ & c[k]*dc[3]*f[1]*vc[2]+c[3]*dc[k]*f[1]*vc[2]+b[k]*ds[3]*f[1]*vc[2]- \\ & b[3]*ds[k]*f[1]*vc[2]+c[k]*dc[1]*f[3]*vc[2]-c[1]*dc[k]*f[3]*vc[2]- \\ & b[k]*ds[1]*f[3]*vc[2]+b[1]*ds[k]*f[3]*vc[2]- \\ & c[3]*dc[1]*f[k]*vc[2]+c[1]*dc[3]*f[k]*vc[2]+b[3]*ds[1]*f[k]*vc[2]- \\ & b[1]*ds[3]*f[k]*vc[2]+c[k]*dc[2]*f[1]*vc[3]-c[2]*dc[k]*f[1]*vc[3]- \\ & b[k]*ds[2]*f[1]*vc[3]+b[2]*ds[k]*f[1]*vc[3]- \\ & c[k]*dc[1]*f[2]*vc[3]+c[1]*dc[k]*f[2]*vc[3]+b[k]*ds[1]*f[2]*vc[3]- \\ & b[1]*ds[k]*f[2]*vc[3]+c[2]*dc[1]*f[k]*vc[3]-c[1]*dc[2]*f[k]*vc[3]- \\ & b[2]*ds[1]*f[k]*vc[3]+b[1]*ds[2]*f[k]*vc[3]- \\ & c[3]*dc[2]*f[1]*vc[k]+c[2]*dc[3]*f[1]*vc[k]+b[3]*ds[2]*f[1]*vc[k]- \\ & b[2]*ds[3]*f[1]*vc[k]+c[3]*dc[1]*f[2]*vc[k]-c[1]*dc[3]*f[2]*vc[k]- \\ & b[3]*ds[1]*f[2]*vc[k]+b[1]*ds[3]*f[2]*vc[k]- \\ & c[2]*dc[1]*f[3]*vc[k]+c[1]*dc[2]*f[3]*vc[k]+b[2]*ds[1]*f[3]*vc[k]-b[1]*ds[2]*f[3]*vc[k]- \\ & b[k]*dc[3]*f[2]*vs[1]+b[3]*dc[k]*f[2]*vs[1]- \\ & c[k]*ds[3]*f[2]*vs[1]+c[3]*ds[k]*f[2]*vs[1]+b[k]*dc[2]*f[3]*vs[1]- \\ & b[2]*dc[k]*f[3]*vs[1]+c[k]*ds[2]*f[3]*vs[1]-c[2]*ds[k]*f[3]*vs[1]- \\ & b[3]*dc[2]*f[k]*vs[1]+b[2]*dc[3]*f[k]*vs[1]- \\ & c[3]*ds[2]*f[k]*vs[1]+c[2]*ds[3]*f[k]*vs[1]+b[k]*dc[3]*f[1]*vs[2]- \\ & b[3]*dc[k]*f[1]*vs[2]+c[k]*ds[3]*f[1]*vs[2]-c[3]*ds[k]*f[1]*vs[2]- \\ & b[k]*dc[1]*f[3]*vs[2]+b[1]*dc[k]*f[3]*vs[2]- \\ & c[k]*ds[1]*f[3]*vs[2]+c[1]*ds[k]*f[3]*vs[2]+b[3]*dc[1]*f[k]*vs[2]- \end{aligned}$$

$$\begin{aligned}
 & b[1]*dc[3]*f[k]*vs[2]+c[3]*ds[1]*f[k]*vs[2]-c[1]*ds[3]*f[k]*vs[2]- \\
 & b[k]*dc[2]*f[1]*vs[3]+b[2]*dc[k]*f[1]*vs[3]- \\
 & c[k]*ds[2]*f[1]*vs[3]+c[2]*ds[k]*f[1]*vs[3]+b[k]*dc[1]*f[2]*vs[3]- \\
 & b[1]*dc[k]*f[2]*vs[3]+c[k]*ds[1]*f[2]*vs[3]-c[1]*ds[k]*f[2]*vs[3]- \\
 & b[2]*dc[1]*f[k]*vs[3]+b[1]*dc[2]*f[k]*vs[3]-c[2]*ds[1]*f[k]*vs[3]+c[1]*ds[2]*f[k]*vs[3]+(- \\
 & b[2]*dc[3]*f[1]+c[3]*ds[2]*f[1]-c[2]*ds[3]*f[1]+b[1]*dc[3]*f[2]- \\
 & c[3]*ds[1]*f[2]+c[1]*ds[3]*f[2]+b[3]*(dc[2]*f[1]-dc[1]*f[2])+(b[2]*dc[1]-b[1]*dc[2]+c[2]*ds[1]- \\
 & c[1]*ds[2])*f[3])*vs[k]
 \end{aligned}$$

$$\begin{aligned}
 e_{pi}= & b[1]*(dc[k]*(f[3]*vc[2]-f[2]*vc[3])+dc[3]*(-f[k]*vc[2]+f[2]*vc[k])+dc[2]*(f[k]*vc[3]- \\
 & f[3]*vc[k]))+c[1]*(ds[k]*(f[3]*vc[2]-f[2]*vc[3])+ds[3]*(-f[k]*vc[2]+f[2]*vc[k])+ds[2]*(f[k]*vc[3]- \\
 & f[3]*vc[k]))+c[2]*(-ds[1]*f[k]*vc[3]+ds[k]*(- \\
 & f[3]*vc[1]+f[1]*vc[3])+ds[1]*f[3]*vc[k]+ds[3]*(f[k]*vc[1]-f[1]*vc[k])+dc[k]*f[3]*vs[1]- \\
 & dc[3]*f[k]*vs[1])+b[2]*(-dc[1]*f[k]*vc[3]+dc[k]*(- \\
 & f[3]*vc[1]+f[1]*vc[3])+dc[1]*f[3]*vc[k]+dc[3]*(f[k]*vc[1]-f[1]*vc[k])- \\
 & ds[k]*f[3]*vs[1]+ds[3]*f[k]*vs[1])- \\
 & c[1]*dc[k]*f[3]*vs[2]+b[1]*ds[k]*f[3]*vs[2]+c[1]*dc[3]*f[k]*vs[2]- \\
 & b[1]*ds[3]*f[k]*vs[2]+c[3]*(ds[1]*f[k]*vc[2]+ds[k]*(f[2]*vc[1]-f[1]*vc[2])- \\
 & ds[1]*f[2]*vc[k]+ds[2]*(-f[k]*vc[1]+f[1]*vc[k])-dc[k]*f[2]*vs[1]+dc[2]*f[k]*vs[1]+(dc[k]*f[1]- \\
 & dc[1]*f[k])*vs[2])-(c[2]*dc[k]*f[1]-b[2]*ds[k]*f[1]-c[1]*dc[k]*f[2]+b[1]*ds[k]*f[2]+(- \\
 & c[2]*dc[1]+c[1]*dc[2]+b[2]*ds[1]-b[1]*ds[2])*f[k])*vs[3]+b[k]*(-dc[1]*f[3]*vc[2]+dc[3]*(- \\
 & f[2]*vc[1]+f[1]*vc[2])+dc[1]*f[2]*vc[3]+dc[2]*(f[3]*vc[1]-f[1]*vc[3])- \\
 & ds[3]*f[2]*vs[1]+ds[2]*f[3]*vs[1]+ds[3]*f[1]*vs[2]-ds[1]*f[3]*vs[2]- \\
 & ds[2]*f[1]*vs[3]+ds[1]*f[2]*vs[3]+c[k]*(-ds[1]*f[3]*vc[2]+ds[3]*(- \\
 & f[2]*vc[1]+f[1]*vc[2])+ds[1]*f[2]*vc[3]+ds[2]*(f[3]*vc[1]-f[1]*vc[3])+dc[3]*f[2]*vs[1]- \\
 & dc[2]*f[3]*vs[1]-dc[3]*f[1]*vs[2]+dc[1]*f[3]*vs[2]+(dc[2]*f[1]-dc[1]*f[2])*vs[3])-(- \\
 & c[2]*dc[3]*f[1]+b[2]*ds[3]*f[1]+c[1]*dc[3]*f[2]-b[1]*ds[3]*f[2]+c[3]*(dc[2]*f[1]- \\
 & dc[1]*f[2])+(c[2]*dc[1]-c[1]*dc[2]- \\
 & b[2]*ds[1]+b[1]*ds[2])*f[3])*vs[k]+b[3]*(dc[1]*f[k]*vc[2]+dc[k]*(f[2]*vc[1]-f[1]*vc[2])- \\
 & dc[1]*f[2]*vc[k]+dc[2]*(-f[k]*vc[1]+f[1]*vc[k])+ds[k]*f[2]*vs[1]-ds[2]*f[k]*vs[1]- \\
 & ds[k]*f[1]*vs[2]+ds[1]*f[k]*vs[2]+(ds[2]*f[1]-ds[1]*f[2])*vs[k])
 \end{aligned}$$

$$\begin{aligned}
 f_{pi}= & -c[1]*dc[k]*f[3]*vc[2]+b[1]*ds[k]*f[3]*vc[2]+c[1]*dc[3]*f[k]*vc[2]- \\
 & b[1]*ds[3]*f[k]*vc[2]+c[1]*dc[k]*f[2]*vc[3]-b[1]*ds[k]*f[2]*vc[3]- \\
 & c[1]*dc[2]*f[k]*vc[3]+b[1]*ds[2]*f[k]*vc[3]- \\
 & c[1]*dc[3]*f[2]*vc[k]+b[1]*ds[3]*f[2]*vc[k]+c[1]*dc[2]*f[3]*vc[k]-b[1]*ds[2]*f[3]*vc[k]+b[2]*(- \\
 & ds[1]*f[k]*vc[3]+ds[k]*(-f[3]*vc[1]+f[1]*vc[3])+ds[1]*f[3]*vc[k]+ds[3]*(f[k]*vc[1]-
 \end{aligned}$$

$$\begin{aligned}
 & f[1]*vc[k]+dc[k]*f[3]*vs[1]-dc[3]*f[k]*vs[1]+c[2]*(dc[k]*(f[3]*vc[1]-f[1]*vc[3])+dc[3]*(- \\
 & f[k]*vc[1]+f[1]*vc[k])+dc[1]*(f[k]*vc[3]-f[3]*vc[k])+(ds[k]*f[3]-ds[3]*f[k])*vs[1])- \\
 & ((b[1]*dc[k]+c[1]*ds[k])*f[3]- \\
 & (b[1]*dc[3]+c[1]*ds[3])*f[k])*vs[2]+b[3]*(ds[1]*f[k]*vc[2]+ds[k]*(f[2]*vc[1]-f[1]*vc[2])- \\
 & ds[1]*f[2]*vc[k]+ds[2]*(-f[k]*vc[1]+f[1]*vc[k])-dc[k]*f[2]*vs[1]+dc[2]*f[k]*vs[1]+(dc[k]*f[1]- \\
 & dc[1]*f[k])*vs[2])-(b[2]*dc[k]*f[1]+c[2]*ds[k]*f[1]-b[1]*dc[k]*f[2]-c[1]*ds[k]*f[2]+(- \\
 & b[2]*dc[1]+b[1]*dc[2]-c[2]*ds[1]+c[1]*ds[2])*f[k])*vs[3]+b[k]*(-ds[1]*f[3]*vc[2]+ds[3]*(- \\
 & f[2]*vc[1]+f[1]*vc[2])+ds[1]*f[2]*vc[3]+ds[2]*(f[3]*vc[1]-f[1]*vc[3])+dc[3]*f[2]*vs[1]- \\
 & dc[2]*f[3]*vs[1]-dc[3]*f[1]*vs[2]+dc[1]*f[3]*vs[2]+(dc[2]*f[1]- \\
 & dc[1]*f[2])*vs[3])+c[k]*(dc[1]*f[3]*vc[2]+dc[3]*(f[2]*vc[1]-f[1]*vc[2])-dc[1]*f[2]*vc[3]+dc[2]*(- \\
 & f[3]*vc[1]+f[1]*vc[3])+ds[3]*f[2]*vs[1]-ds[2]*f[3]*vs[1]- \\
 & ds[3]*f[1]*vs[2]+ds[1]*f[3]*vs[2]+(ds[2]*f[1]-ds[1]*f[2])*vs[3])+(b[2]*dc[3]*f[1]+c[2]*ds[3]*f[1]- \\
 & b[1]*dc[3]*f[2]-c[1]*ds[3]*f[2]+b[3]*(-dc[2]*f[1]+dc[1]*f[2])+(b[2]*dc[1]+b[1]*dc[2]- \\
 & c[2]*ds[1]+c[1]*ds[2])*f[3])*vs[k]+c[3]*(-dc[1]*f[k]*vc[2]+dc[k]*(- \\
 & f[2]*vc[1]+f[1]*vc[2])+dc[1]*f[2]*vc[k]+dc[2]*(f[k]*vc[1]-f[1]*vc[k])- \\
 & ds[k]*f[2]*vs[1]+ds[2]*f[k]*vs[1]+ds[k]*f[1]*vs[2]-ds[1]*f[k]*vs[2]- \\
 & ds[2]*f[1]*vs[k]+ds[1]*f[2]*vs[k])
 \end{aligned}$$

$$\begin{aligned}
 g_{pi}=2*((ds[k]*f[3]-ds[3]*f[k])*(dc[2]*vc[1]-dc[1]*vc[2])+(dc[2]*(- \\
 ds[k]*f[1]+ds[1]*f[k])+dc[1]*(ds[k]*f[2]-ds[2]*f[k]))*vc[3]+dc[k]*(ds[3]*(f[2]*vc[1]- \\
 f[1]*vc[2])+ds[2]*(-f[3]*vc[1]+f[1]*vc[3])+ds[1]*(f[3]*vc[2]-f[2]*vc[3]))+(dc[2]*(ds[3]*f[1]- \\
 ds[1]*f[3])+dc[1]*(-ds[3]*f[2]+ds[2]*f[3]))*vc[k]+dc[3]*(ds[k]*(- \\
 f[2]*vc[1]+f[1]*vc[2])+ds[2]*(f[k]*vc[1]-f[1]*vc[k])+ds[1]*(-f[k]*vc[2]+f[2]*vc[k]))))
 \end{aligned}$$

$$\begin{aligned}
 h_{pi}=2*(dc[2]*(f[k]*(-ds[3]*vs[1]+ds[1]*vs[3])+ds[k]*(f[3]*vs[1]- \\
 f[1]*vs[3]))+dc[k]*(ds[3]*(f[2]*vs[1]-f[1]*vs[2])+ds[2]*(-f[3]*vs[1]+f[1]*vs[3])+ds[1]*(f[3]*vs[2]- \\
 f[2]*vs[3]))+dc[1]*(f[k]*(ds[3]*vs[2]-ds[2]*vs[3])+ds[k]*(- \\
 f[3]*vs[2]+f[2]*vs[3]))+dc[2]*ds[3]*f[1]*vs[k]-dc[1]*ds[3]*f[2]*vs[k]- \\
 dc[2]*ds[1]*f[3]*vs[k]+dc[1]*ds[2]*f[3]*vs[k]+dc[3]*(ds[k]*(- \\
 f[2]*vs[1]+f[1]*vs[2])+ds[2]*(f[k]*vs[1]-f[1]*vs[k])+ds[1]*(-f[k]*vs[2]+f[2]*vs[k]))))
 \end{aligned}$$

$$i=1, 2$$

$$k=i+3$$

EK 4. Yörünge Sentezi Modülü Denklem (3.7) Katsayıları

$$\begin{aligned}
 w_0 = & ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*gp[1]-ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*gp[1]- \\
 & ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*gp[1]- \\
 & ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*gp[1]+ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*gp[1]+2*ap[1]*ap[2]*dp[1]* \\
 & dp[2]*dp[2]*gp[1]-ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*gp[1]+ap[2]*bp[1]*bp[2]*dp[2]*fp[1]*gp[1]- \\
 & ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*gp[1]- \\
 & 2*ap[2]*ap[2]*dp[1]*dp[2]*fp[1]*gp[1]+2*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*gp[1]- \\
 & ap[2]*ap[2]*dp[2]*fp[1]*fp[1]*gp[1]+ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*gp[1]- \\
 & ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*gp[1]-ap[2]*ap[2]*dp[1]*dp[1]*fp[2]*gp[1]- \\
 & 2*ap[2]*bp[1]*bp[1]*dp[2]*fp[2]*gp[1]+2*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*gp[1]+4*ap[1]*ap[2]*dp[\\
 & 1]*dp[2]*fp[2]*gp[1]-3*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*gp[1]+ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*gp[1]- \\
 & ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*gp[1]- \\
 & 2*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*gp[1]+4*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*gp[1]- \\
 & ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*gp[1]- \\
 & ap[2]*bp[1]*bp[1]*fp[2]*fp[2]*gp[1]+ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*gp[1]+2*ap[1]*ap[2]*dp[1]*fp \\
 & [2]*fp[2]*gp[1]-3*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*gp[1]+2*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*gp[1]- \\
 & ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*gp[1]- \\
 & ap[2]*bp[1]*bp[2]*bp[2]*gp[1]*gp[1]+ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*gp[1]+ap[2]*ap[2]*bp[2]*dp[\\
 & 1]*gp[1]*gp[1]+2*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*gp[1]- \\
 & 3*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*gp[1]+ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*gp[1]+2*ap[2]*ap[2]*bp[1] \\
 & *fp[2]*gp[1]*gp[1]-3*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*gp[1]-ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*gp[1]- \\
 & ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*gp[2]+ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*gp[2]+ap[2]*ap[2]*dp[1]*dp[\\
 & 1]*dp[1]*gp[2]+ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*gp[2]-ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*gp[2]- \\
 & 2*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*gp[2]+ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*gp[2]- \\
 & 2*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*gp[2]+2*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*gp[2]+3*ap[2]*ap[2]*dp[\\
 & 1]*dp[1]*fp[1]*gp[2]+ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*gp[2]-ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*gp[2]- \\
 & 4*ap[1]*ap[2]*dp[1]*dp[2]*fp[1]*gp[2]+ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*gp[2]- \\
 & ap[2]*bp[1]*bp[2]*fp[1]*fp[1]*gp[2]+ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*gp[2]+3*ap[2]*ap[2]*dp[1]*fp \\
 & [1]*fp[1]*gp[2]- \\
 & 2*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*gp[2]+ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*gp[2]+ap[2]*bp[1]*bp[1]*dp \\
 & [1]*fp[2]*gp[2]-ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*gp[2]- \\
 & 2*ap[1]*ap[2]*dp[1]*dp[1]*fp[2]*gp[2]+2*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*gp[2]+ap[2]*bp[1]*bp[1] \\
 & *fp[1]*fp[2]*gp[2]-ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*gp[2]- \\
 & 4*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*gp[2]+2*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*gp[2]- \\
 & 2*ap[1]*ap[2]*fp[1]*fp[1]*fp[2]*gp[2]+ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*gp[2]+ap[1]*ap[1]*fp[1]*fp[\\
 & 2]*fp[2]*gp[2]+2*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*gp[2]-2*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*gp[2]- \\
 & 3*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*gp[2]+ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*gp[2]+3*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*gp[2]- \\
 & 3*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*gp[2]+ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*gp[2]- \\
 & ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*gp[2]+3*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*gp[2]+3*ap[1]*ap[2]*ap[2] \\
 & *gp[1]*gp[1]*gp[2]- \\
 & ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*gp[2]+ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*gp[2]+3*ap[1]*ap[2]*bp[1]* \\
 & dp[1]*gp[2]*gp[2]-2*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*gp[2]- \\
 & ap[1]*ap[1]*bp[1]*dp[2]*gp[2]*gp[2]+3*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*gp[2]- \\
 & 2*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*gp[2]-ap[1]*ap[1]*bp[1]*fp[2]*gp[2]*gp[2]- \\
 & 3*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*gp[2]+ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*gp[2]
 \end{aligned}$$

$$\begin{aligned}
 w_1 = & 2*ap[2]*bp[2]*cp[1]*dp[1]*dp[2]*gp[1]+2*ap[2]*bp[1]*cp[2]*dp[1]*dp[2]*gp[1]- \\
 & 4*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*gp[1]- \\
 & 4*ap[2]*bp[1]*cp[1]*dp[2]*dp[2]*gp[1]+2*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*gp[1]+2*ap[1]*bp[1]*cp \\
 & [2]*dp[2]*dp[2]*gp[1]+4*ap[2]*bp[1]*bp[2]*dp[2]*cp[1]*gp[1]- \\
 & 4*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*gp[1]- \\
 & 8*ap[2]*ap[2]*dp[1]*dp[2]*ep[1]*gp[1]+8*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*gp[1]+4*ap[2]*bp[1]*bp \\
 & [2]*dp[1]*ep[2]*gp[1]-4*ap[1]*bp[2]*bp[2]*dp[1]*ep[2]*gp[1]- \\
 & 4*ap[2]*ap[2]*dp[1]*dp[1]*ep[2]*gp[1]- \\
 & 8*ap[2]*bp[1]*bp[1]*dp[2]*ep[2]*gp[1]+8*ap[1]*bp[1]*bp[2]*dp[2]*ep[2]*gp[1]+16*ap[1]*ap[2]*d \\
 & p[1]*dp[2]*ep[2]*gp[1]- \\
 & 12*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*gp[1]+2*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*gp[1]+2*ap[2]*bp[1]*c \\
 & p[2]*dp[2]*fp[1]*gp[1]-4*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*gp[1]- \\
 & 8*ap[2]*ap[2]*dp[2]*ep[1]*fp[1]*gp[1]+4*ap[2]*bp[1]*bp[2]*ep[2]*fp[1]*gp[1]- \\
 & 4*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*gp[1]- \\
 & 8*ap[2]*ap[2]*dp[1]*ep[2]*fp[1]*gp[1]+16*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*gp[1]- \\
 & 4*ap[2]*ap[2]*ep[2]*fp[1]*fp[1]*gp[1]+2*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*gp[1]+2*ap[2]*bp[1]*cp \\
 & [2]*dp[1]*fp[2]*gp[1]-4*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*gp[1]- \\
 & 8*ap[2]*bp[1]*cp[1]*dp[2]*fp[2]*gp[1]+4*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*gp[1]+4*ap[1]*bp[1]*cp \\
 & [2]*dp[2]*fp[2]*gp[1]+4*ap[2]*bp[1]*bp[2]*ep[1]*fp[2]*gp[1]- \\
 & 4*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*gp[1]- \\
 & 8*ap[2]*ap[2]*dp[1]*ep[1]*fp[2]*gp[1]+16*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*gp[1]- \\
 & 8*ap[2]*bp[1]*bp[1]*ep[2]*fp[2]*gp[1]+8*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*gp[1]+16*ap[1]*ap[2]*d \\
 & p[1]*ep[2]*fp[2]*gp[1]- \\
 & 24*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*gp[1]+2*ap[2]*bp[2]*cp[1]*fp[1]*fp[2]*gp[1]+2*ap[2]*bp[1]*cp \\
 & [2]*fp[1]*fp[2]*gp[1]-4*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*gp[1]- \\
 & 8*ap[2]*ap[2]*ep[1]*fp[1]*fp[2]*gp[1]+16*ap[1]*ap[2]*ep[2]*fp[1]*fp[2]*gp[1]- \\
 & 4*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*gp[1]+2*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*gp[1]+2*ap[1]*bp[1]*cp[
 \end{aligned}$$

$$\begin{aligned}
 & 2] * fp[2] * fp[2] * gp[1] + 8 * ap[1] * ap[2] * ep[1] * fp[2] * fp[2] * gp[1] - \\
 & 12 * ap[1] * ap[1] * ep[2] * fp[2] * fp[2] * gp[1] - 2 * ap[2] * bp[2] * bp[2] * cp[1] * gp[1] * gp[1] - \\
 & 4 * ap[2] * bp[1] * bp[2] * cp[2] * gp[1] * gp[1] + 6 * ap[1] * bp[2] * bp[2] * cp[2] * gp[1] * gp[1] + 2 * ap[2] * ap[2] * cp \\
 & [2] * dp[1] * gp[1] * gp[1] + 4 * ap[2] * ap[2] * cp[1] * dp[2] * gp[1] * gp[1] - \\
 & 6 * ap[1] * ap[2] * cp[2] * dp[2] * gp[1] * gp[1] + 4 * ap[2] * ap[2] * bp[2] * cp[1] * gp[1] * gp[1] + 8 * ap[2] * ap[2] * bp \\
 & [1] * ep[2] * gp[1] * gp[1] - \\
 & 12 * ap[1] * ap[2] * bp[2] * ep[2] * gp[1] * gp[1] + 2 * ap[2] * ap[2] * cp[2] * fp[1] * gp[1] * gp[1] + 4 * ap[2] * ap[2] * c \\
 & p[1] * fp[2] * gp[1] * gp[1] - 6 * ap[1] * ap[2] * cp[2] * fp[2] * gp[1] * gp[1] - \\
 & 2 * ap[2] * bp[2] * cp[1] * dp[1] * dp[1] * gp[2] - \\
 & 2 * ap[2] * bp[1] * cp[2] * dp[1] * dp[1] * gp[2] + 4 * ap[1] * bp[2] * cp[2] * dp[1] * dp[1] * gp[2] + 4 * ap[2] * bp[1] * cp \\
 & [1] * dp[1] * dp[2] * gp[2] - 2 * ap[1] * bp[2] * cp[1] * dp[1] * dp[2] * gp[2] - \\
 & 2 * ap[1] * bp[1] * cp[2] * dp[1] * dp[2] * gp[2] - \\
 & 8 * ap[2] * bp[1] * bp[2] * dp[1] * ep[1] * gp[2] + 8 * ap[1] * bp[2] * bp[2] * dp[1] * ep[1] * gp[2] + 12 * ap[2] * ap[2] * d \\
 & p[1] * dp[1] * ep[1] * gp[2] + 4 * ap[2] * bp[1] * bp[1] * dp[2] * ep[1] * gp[2] - \\
 & 4 * ap[1] * bp[1] * bp[2] * dp[2] * ep[1] * gp[2] - \\
 & 16 * ap[1] * ap[2] * dp[1] * dp[2] * ep[1] * gp[2] + 4 * ap[1] * ap[1] * dp[2] * dp[2] * ep[1] * gp[2] + 4 * ap[2] * bp[1] * b \\
 & p[1] * dp[1] * ep[2] * gp[2] - 4 * ap[1] * bp[1] * bp[2] * dp[1] * ep[2] * gp[2] - \\
 & 8 * ap[1] * ap[2] * dp[1] * dp[1] * ep[2] * gp[2] + 8 * ap[1] * ap[1] * dp[1] * dp[2] * ep[2] * gp[2] - \\
 & 4 * ap[2] * bp[2] * cp[1] * dp[1] * fp[1] * gp[2] - \\
 & 4 * ap[2] * bp[1] * cp[2] * dp[1] * fp[1] * gp[2] + 8 * ap[1] * bp[2] * cp[2] * dp[1] * fp[1] * gp[2] + 4 * ap[2] * bp[1] * cp[\\
 & 1] * dp[2] * fp[1] * gp[2] - 2 * ap[1] * bp[2] * cp[1] * dp[2] * fp[1] * gp[2] - \\
 & 2 * ap[1] * bp[1] * cp[2] * dp[2] * fp[1] * gp[2] - \\
 & 8 * ap[2] * bp[1] * bp[2] * ep[1] * fp[1] * gp[2] + 8 * ap[1] * bp[2] * bp[2] * ep[1] * fp[1] * gp[2] + 24 * ap[2] * ap[2] * d \\
 & p[1] * ep[1] * fp[1] * gp[2] - \\
 & 16 * ap[1] * ap[2] * dp[2] * ep[1] * fp[1] * gp[2] + 4 * ap[2] * bp[1] * bp[1] * ep[2] * fp[1] * gp[2] - \\
 & 4 * ap[1] * bp[1] * bp[2] * ep[2] * fp[1] * gp[2] - \\
 & 16 * ap[1] * ap[2] * dp[1] * ep[2] * fp[1] * gp[2] + 8 * ap[1] * ap[1] * dp[2] * ep[2] * fp[1] * gp[2] - \\
 & 2 * ap[2] * bp[2] * cp[1] * fp[1] * fp[1] * gp[2] - \\
 & 2 * ap[2] * bp[1] * cp[2] * fp[1] * fp[1] * gp[2] + 4 * ap[1] * bp[2] * cp[2] * fp[1] * fp[1] * gp[2] + 12 * ap[2] * ap[2] * ep[\\
 & 1] * fp[1] * fp[1] * gp[2] - \\
 & 8 * ap[1] * ap[2] * ep[2] * fp[1] * fp[1] * gp[2] + 4 * ap[2] * bp[1] * cp[1] * dp[1] * fp[2] * gp[2] - \\
 & 2 * ap[1] * bp[2] * cp[1] * dp[1] * fp[2] * gp[2] - \\
 & 2 * ap[1] * bp[1] * cp[2] * dp[1] * fp[2] * gp[2] + 4 * ap[2] * bp[1] * bp[1] * ep[1] * fp[2] * gp[2] - \\
 & 4 * ap[1] * bp[1] * bp[2] * ep[1] * fp[2] * gp[2] - \\
 & 16 * ap[1] * ap[2] * dp[1] * ep[1] * fp[2] * gp[2] + 8 * ap[1] * ap[1] * dp[2] * ep[1] * fp[2] * gp[2] + 8 * ap[1] * ap[1] * dp \\
 & [1] * ep[2] * fp[2] * gp[2] + 4 * ap[2] * bp[1] * cp[1] * fp[1] * fp[2] * gp[2] - \\
 & 2 * ap[1] * bp[2] * cp[1] * fp[1] * fp[2] * gp[2] - 2 * ap[1] * bp[1] * cp[2] * fp[1] * fp[2] * gp[2] -
 \end{aligned}$$

$$\begin{aligned}
 & 16*ap[1]*ap[2]*cp[1]*fp[1]*fp[2]*gp[2]+8*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*gp[2]+4*ap[1]*ap[1]*ep[1]*fp[2]*fp[2]*gp[2]+8*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*gp[2]- \\
 & 4*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*gp[2]+4*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*gp[2]- \\
 & 8*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*gp[2]- \\
 & 6*ap[2]*ap[2]*cp[1]*dp[1]*gp[1]*gp[2]+2*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*gp[2]- \\
 & 2*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*gp[2]+6*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*gp[2]- \\
 & 12*ap[2]*ap[2]*bp[1]*ep[1]*gp[1]*gp[2]+4*ap[1]*ap[2]*bp[2]*ep[1]*gp[1]*gp[2]- \\
 & 4*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*gp[2]+12*ap[1]*ap[1]*bp[2]*ep[2]*gp[1]*gp[2]- \\
 & 6*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*gp[2]+2*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*gp[2]- \\
 & 2*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*gp[2]+6*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*gp[2]- \\
 & 6*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*gp[2]+4*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*gp[2]+2*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*gp[2]+6*ap[1]*ap[2]*cp[1]*dp[1]*gp[2]*gp[2]- \\
 & 4*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*gp[2]- \\
 & 2*ap[1]*ap[1]*cp[1]*dp[2]*gp[2]*gp[2]+12*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*gp[2]- \\
 & 8*ap[1]*ap[1]*bp[2]*ep[1]*gp[2]*gp[2]- \\
 & 4*ap[1]*ap[1]*bp[1]*ep[2]*gp[2]*gp[2]+6*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*gp[2]- \\
 & 4*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*gp[2]- \\
 & 2*ap[1]*ap[1]*cp[1]*fp[2]*gp[2]*gp[2]+2*ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*hp[1]- \\
 & 2*ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*hp[1]-2*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*hp[1]- \\
 & 2*ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*hp[1]+2*ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*hp[1]+4*ap[1]*ap[2]*dp[1]*dp[2]*dp[2]*hp[1]- \\
 & 2*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*hp[1]+2*ap[2]*bp[1]*bp[2]*dp[2]*fp[1]*hp[1]- \\
 & 2*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*hp[1]- \\
 & 4*ap[2]*ap[2]*dp[1]*dp[2]*fp[1]*hp[1]+4*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*hp[1]- \\
 & 2*ap[2]*ap[2]*dp[2]*fp[1]*fp[1]*hp[1]+2*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*hp[1]- \\
 & 2*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*hp[1]-2*ap[2]*ap[2]*dp[1]*dp[1]*fp[2]*hp[1]- \\
 & 4*ap[2]*bp[1]*bp[1]*dp[2]*fp[2]*hp[1]+4*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*hp[1]+8*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*hp[1]- \\
 & 6*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*hp[1]+2*ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*hp[1]- \\
 & 2*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*hp[1]- \\
 & 4*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*hp[1]+8*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*hp[1]- \\
 & 2*ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*hp[1]- \\
 & 2*ap[2]*bp[1]*bp[1]*fp[2]*fp[2]*hp[1]+2*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*hp[1]+4*ap[1]*ap[2]*dp[1]*fp[2]*fp[2]*hp[1]- \\
 & 6*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*hp[1]+4*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*hp[1]- \\
 & 2*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*hp[1]- \\
 & 4*ap[2]*bp[1]*bp[2]*bp[2]*gp[1]*hp[1]+4*ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*hp[1]+4*ap[2]*ap[2]*bp
 \end{aligned}$$

$$\begin{aligned}
 & [2]*dp[1]*gp[1]*hp[1]+8*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*hp[1]- \\
 & 12*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*hp[1]+4*ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*hp[1]+8*ap[2]*ap[2]*b \\
 & p[1]*fp[2]*gp[1]*hp[1]-12*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*hp[1]- \\
 & 6*ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*hp[1]+4*ap[2]*bp[1]*bp[1]*bp[2]*gp[2]*hp[1]- \\
 & 4*ap[1]*bp[1]*bp[2]*bp[2]*gp[2]*hp[1]- \\
 & 6*ap[2]*ap[2]*bp[1]*dp[1]*gp[2]*hp[1]+2*ap[1]*ap[2]*bp[2]*dp[1]*gp[2]*hp[1]- \\
 & 2*ap[1]*ap[2]*bp[1]*dp[2]*gp[2]*hp[1]+6*ap[1]*ap[1]*bp[2]*dp[2]*gp[2]*hp[1]- \\
 & 6*ap[2]*ap[2]*bp[1]*fp[1]*gp[2]*hp[1]+2*ap[1]*ap[2]*bp[2]*fp[1]*gp[2]*hp[1]- \\
 & 2*ap[1]*ap[2]*bp[1]*fp[2]*gp[2]*hp[1]+6*ap[1]*ap[1]*bp[2]*fp[2]*gp[2]*hp[1]+12*ap[1]*ap[2]*ap \\
 & [2]*gp[1]*gp[2]*hp[1]-6*ap[1]*ap[1]*ap[2]*gp[2]*gp[2]*hp[1]- \\
 & 2*ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*hp[2]+2*ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*hp[2]+2*ap[2]*ap[2]*dp \\
 & [1]*dp[1]*dp[1]*hp[2]+2*ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*hp[2]- \\
 & 2*ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*hp[2]- \\
 & 4*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*hp[2]+2*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*hp[2]- \\
 & 4*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*hp[2]+4*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*hp[2]+6*ap[2]*ap[2]*dp[\\
 & 1]*dp[1]*fp[1]*hp[2]+2*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*hp[2]- \\
 & 2*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*hp[2]- \\
 & 8*ap[1]*ap[2]*dp[1]*dp[2]*fp[1]*hp[2]+2*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*hp[2]- \\
 & 2*ap[2]*bp[1]*bp[2]*fp[1]*fp[1]*hp[2]+2*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*hp[2]+6*ap[2]*ap[2]*dp[\\
 & 1]*fp[1]*fp[1]*hp[2]- \\
 & 4*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*hp[2]+2*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*hp[2]+2*ap[2]*bp[1]*bp[1] \\
 &]*dp[1]*fp[2]*hp[2]-2*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*hp[2]- \\
 & 4*ap[1]*ap[2]*dp[1]*dp[1]*fp[2]*hp[2]+4*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*hp[2]+2*ap[2]*bp[1]*bp[\\
 & 1]*fp[1]*fp[2]*hp[2]-2*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*hp[2]- \\
 & 8*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*hp[2]+4*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*hp[2]- \\
 & 4*ap[1]*ap[2]*fp[1]*fp[1]*fp[2]*hp[2]+2*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*hp[2]+2*ap[1]*ap[1]*fp[1] \\
 &]*fp[2]*fp[2]*hp[2]+4*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*hp[2]- \\
 & 4*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*hp[2]- \\
 & 6*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*hp[2]+2*ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*hp[2]- \\
 & 2*ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*hp[2]+6*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*hp[2]- \\
 & 6*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*hp[2]+2*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*hp[2]- \\
 & 2*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*hp[2]+6*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*hp[2]+6*ap[1]*ap[2]*ap[\\
 & 2]*gp[1]*gp[1]*hp[2]- \\
 & 4*ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*hp[2]+4*ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*hp[2]+12*ap[1]*ap[2]*b \\
 & p[1]*dp[1]*gp[2]*hp[2]-8*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*hp[2]- \\
 & 4*ap[1]*ap[1]*bp[1]*dp[2]*gp[2]*hp[2]+12*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*hp[2]-
 \end{aligned}$$

$$8*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*hp[2]-4*ap[1]*ap[1]*bp[1]*fp[2]*gp[2]*hp[2]-$$

$$12*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*hp[2]+6*ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*hp[2]$$

$$w_2=ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*gp[1]-$$

$$ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*gp[1]+4*ap[2]*cp[1]*cp[2]*dp[1]*dp[2]*gp[1]-$$

$$4*ap[1]*cp[2]*cp[2]*dp[1]*dp[2]*gp[1]-5*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*gp[1]-$$

$$ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*gp[1]+ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*gp[1]-$$

$$4*ap[2]*cp[1]*cp[1]*dp[2]*dp[2]*gp[1]+4*ap[1]*cp[1]*cp[2]*dp[2]*dp[2]*gp[1]+10*ap[1]*ap[2]*d$$

$$p[1]*dp[2]*dp[2]*gp[1]-$$

$$5*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*gp[1]+8*ap[2]*bp[2]*cp[1]*dp[2]*ep[1]*gp[1]+8*ap[2]*bp[1]*cp$$

$$[2]*dp[2]*ep[1]*gp[1]-16*ap[1]*bp[2]*cp[2]*dp[2]*ep[1]*gp[1]-$$

$$16*ap[2]*ap[2]*dp[2]*ep[1]*ep[1]*gp[1]+8*ap[2]*bp[2]*cp[1]*dp[1]*ep[2]*gp[1]+8*ap[2]*bp[1]*c$$

$$p[2]*dp[1]*ep[2]*gp[1]-16*ap[1]*bp[2]*cp[2]*dp[1]*ep[2]*gp[1]-$$

$$32*ap[2]*bp[1]*cp[1]*dp[2]*ep[2]*gp[1]+16*ap[1]*bp[2]*cp[1]*dp[2]*ep[2]*gp[1]+16*ap[1]*bp[1]$$

$$*cp[2]*dp[2]*ep[2]*gp[1]+16*ap[2]*bp[1]*bp[2]*ep[1]*ep[2]*gp[1]-$$

$$16*ap[1]*bp[2]*bp[2]*ep[1]*ep[2]*gp[1]-$$

$$32*ap[2]*ap[2]*dp[1]*ep[1]*ep[2]*gp[1]+64*ap[1]*ap[2]*dp[2]*ep[1]*ep[2]*gp[1]-$$

$$16*ap[2]*bp[1]*bp[1]*ep[2]*ep[2]*gp[1]+16*ap[1]*bp[1]*bp[2]*cp[2]*ep[2]*gp[1]+32*ap[1]*ap[2]$$

$$*dp[1]*ep[2]*cp[2]*gp[1]-48*ap[1]*ap[1]*dp[2]*ep[2]*ep[2]*gp[1]-$$

$$7*ap[2]*bp[1]*bp[2]*dp[2]*fp[1]*gp[1]+7*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*gp[1]+4*ap[2]*cp[1]*cp[$$

$$2]*dp[2]*fp[1]*gp[1]-$$

$$4*ap[1]*cp[2]*cp[2]*dp[2]*fp[1]*gp[1]+6*ap[2]*ap[2]*dp[1]*dp[2]*fp[1]*gp[1]-$$

$$6*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*gp[1]+8*ap[2]*bp[2]*cp[1]*ep[2]*fp[1]*gp[1]+8*ap[2]*bp[1]*cp[$$

$$2]*ep[2]*fp[1]*gp[1]-16*ap[1]*bp[2]*cp[2]*ep[2]*fp[1]*gp[1]-$$

$$32*ap[2]*ap[2]*ep[1]*ep[2]*fp[1]*gp[1]+32*ap[1]*ap[2]*ep[2]*ep[2]*fp[1]*gp[1]+11*ap[2]*ap[2]*$$

$$dp[2]*fp[1]*fp[1]*gp[1]-$$

$$7*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*gp[1]+7*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*gp[1]+4*ap[2]*cp[1]*cp[$$

$$2]*dp[1]*fp[2]*gp[1]-$$

$$4*ap[1]*cp[2]*cp[2]*dp[1]*fp[2]*gp[1]+3*ap[2]*ap[2]*dp[1]*dp[1]*fp[2]*gp[1]+14*ap[2]*bp[1]*b$$

$$p[1]*dp[2]*fp[2]*gp[1]-14*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*gp[1]-$$

$$8*ap[2]*cp[1]*cp[1]*dp[2]*fp[2]*gp[1]+8*ap[1]*cp[1]*cp[2]*dp[2]*fp[2]*gp[1]-$$

$$12*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*gp[1]+9*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*gp[1]+8*ap[2]*bp[2]*c$$

$$p[1]*ep[1]*fp[2]*gp[1]+8*ap[2]*bp[1]*cp[2]*ep[1]*fp[2]*gp[1]-$$

$$16*ap[1]*bp[2]*cp[2]*ep[1]*fp[2]*gp[1]-16*ap[2]*ap[2]*ep[1]*ep[1]*fp[2]*gp[1]-$$

$$32*ap[2]*bp[1]*cp[1]*ep[2]*fp[2]*gp[1]+16*ap[1]*bp[2]*cp[1]*ep[2]*fp[2]*gp[1]+16*ap[1]*bp[1]*$$

$$cp[2]*ep[2]*fp[2]*gp[1]+64*ap[1]*ap[2]*ep[1]*ep[2]*fp[2]*gp[1]-$$

$$48*ap[1]*ap[1]*ep[2]*ep[2]*fp[2]*gp[1]-$$

$$\begin{aligned}
 &15*ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*gp[1]+15*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*gp[1]+4*ap[2]*cp[1]*c \\
 &p[2]*fp[1]*fp[2]*gp[1]- \\
 &4*ap[1]*cp[2]*cp[2]*fp[1]*fp[2]*gp[1]+22*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*gp[1]- \\
 &44*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*gp[1]+19*ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*gp[1]+15*ap[2]*bp[1]* \\
 &bp[1]*fp[2]*fp[2]*gp[1]-15*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*gp[1]- \\
 &4*ap[2]*cp[1]*cp[1]*fp[2]*fp[2]*gp[1]+4*ap[1]*cp[1]*cp[2]*fp[2]*fp[2]*gp[1]- \\
 &22*ap[1]*ap[2]*dp[1]*fp[2]*fp[2]*gp[1]+33*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*gp[1]- \\
 &38*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*gp[1]+19*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*gp[1]+3*ap[2]*bp[1]*bp \\
 &[2]*bp[2]*gp[1]*gp[1]-3*ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*gp[1]- \\
 &8*ap[2]*bp[2]*cp[1]*cp[2]*gp[1]*gp[1]- \\
 &4*ap[2]*bp[1]*cp[2]*cp[2]*gp[1]*gp[1]+12*ap[1]*bp[2]*cp[2]*cp[2]*gp[1]*gp[1]+ap[2]*ap[2]*bp[\\
 &2]*dp[1]*gp[1]*gp[1]+2*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*gp[1]- \\
 &3*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*gp[1]+8*ap[2]*ap[2]*cp[2]*cp[1]*gp[1]*gp[1]+16*ap[2]*ap[2]*c \\
 &p[1]*ep[2]*gp[1]*gp[1]-24*ap[1]*ap[2]*cp[2]*cp[2]*gp[1]*gp[1]- \\
 &7*ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*gp[1]- \\
 &14*ap[2]*ap[2]*bp[1]*fp[2]*gp[1]*gp[1]+21*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*gp[1]- \\
 &ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*gp[1]- \\
 &ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*gp[2]+ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*gp[2]- \\
 &4*ap[2]*cp[1]*cp[2]*dp[1]*dp[1]*gp[2]+4*ap[1]*cp[2]*cp[2]*dp[1]*dp[1]*gp[2]+5*ap[2]*ap[2]*dp \\
 &[1]*dp[1]*dp[1]*gp[2]+ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*gp[2]- \\
 &ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*gp[2]+4*ap[2]*cp[1]*cp[1]*dp[1]*dp[2]*gp[2]- \\
 &4*ap[1]*cp[1]*cp[2]*dp[1]*dp[2]*gp[2]- \\
 &10*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*gp[2]+5*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*gp[2]- \\
 &16*ap[2]*bp[2]*cp[1]*dp[1]*ep[1]*gp[2]- \\
 &16*ap[2]*bp[1]*cp[2]*dp[1]*ep[1]*gp[2]+32*ap[1]*bp[2]*cp[2]*dp[1]*ep[1]*gp[2]+16*ap[2]*bp[1] \\
 &*cp[1]*dp[2]*ep[1]*gp[2]-8*ap[1]*bp[2]*cp[1]*dp[2]*ep[1]*gp[2]- \\
 &8*ap[1]*bp[1]*cp[2]*dp[2]*ep[1]*gp[2]- \\
 &16*ap[2]*bp[1]*bp[2]*ep[1]*ep[1]*gp[2]+16*ap[1]*bp[2]*bp[2]*ep[1]*ep[1]*gp[2]+48*ap[2]*ap[2] \\
 &*dp[1]*ep[1]*ep[1]*gp[2]- \\
 &32*ap[1]*ap[2]*dp[2]*ep[1]*ep[1]*gp[2]+16*ap[2]*bp[1]*cp[1]*dp[1]*ep[2]*gp[2]- \\
 &8*ap[1]*bp[2]*cp[1]*dp[1]*ep[2]*gp[2]- \\
 &8*ap[1]*bp[1]*cp[2]*dp[1]*ep[2]*gp[2]+16*ap[2]*bp[1]*bp[1]*cp[1]*ep[2]*gp[2]- \\
 &16*ap[1]*bp[1]*bp[2]*ep[1]*ep[2]*gp[2]- \\
 &64*ap[1]*ap[2]*dp[1]*ep[1]*ep[2]*gp[2]+32*ap[1]*ap[1]*dp[2]*ep[1]*ep[2]*gp[2]+16*ap[1]*ap[1] \\
 &*dp[1]*ep[2]*ep[2]*gp[2]+14*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*gp[2]- \\
 &14*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*gp[2]- \\
 &8*ap[2]*cp[1]*cp[2]*dp[1]*fp[1]*gp[2]+8*ap[1]*cp[2]*cp[2]*dp[1]*fp[1]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &9*ap[2]*ap[2]*dp[1]*dp[1]*fp[1]*gp[2]- \\
 &7*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*gp[2]+7*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*gp[2]+4*ap[2]*cp[1]*cp[1]*dp[2]*fp[1]*gp[2]- \\
 &4*ap[1]*cp[1]*cp[2]*dp[2]*fp[1]*gp[2]+12*ap[1]*ap[2]*dp[1]*dp[2]*fp[1]*gp[2]- \\
 &3*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*gp[2]-16*ap[2]*bp[2]*cp[1]*ep[1]*fp[1]*gp[2]- \\
 &16*ap[2]*bp[1]*cp[2]*ep[1]*fp[1]*gp[2]+32*ap[1]*bp[2]*cp[2]*ep[1]*fp[1]*gp[2]+48*ap[2]*ap[2]*ep[1]*ep[1]*fp[1]*gp[2]+16*ap[2]*bp[1]*cp[1]*cp[2]*fp[1]*gp[2]- \\
 &8*ap[1]*bp[2]*cp[1]*ep[2]*fp[1]*gp[2]-8*ap[1]*bp[1]*cp[2]*ep[2]*fp[1]*gp[2]- \\
 &64*ap[1]*ap[2]*ep[1]*ep[2]*fp[1]*gp[2]+16*ap[1]*ap[1]*ep[2]*ep[2]*fp[1]*gp[2]+15*ap[2]*bp[1]*bp[2]*fp[1]*fp[1]*gp[2]-15*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*gp[2]- \\
 &4*ap[2]*cp[1]*cp[2]*fp[1]*fp[1]*gp[2]+4*ap[1]*cp[2]*cp[2]*fp[1]*fp[1]*gp[2]- \\
 &33*ap[2]*ap[2]*dp[1]*fp[1]*fp[1]*gp[2]+22*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*gp[2]- \\
 &19*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*gp[2]- \\
 &7*ap[2]*bp[1]*bp[1]*dp[1]*fp[2]*gp[2]+7*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*gp[2]+4*ap[2]*cp[1]*cp[1]*dp[1]*fp[2]*gp[2]- \\
 &4*ap[1]*cp[1]*cp[2]*dp[1]*fp[2]*gp[2]+6*ap[1]*ap[2]*dp[1]*dp[1]*fp[2]*gp[2]- \\
 &6*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*gp[2]+16*ap[2]*bp[1]*cp[1]*ep[1]*fp[2]*gp[2]- \\
 &8*ap[1]*bp[2]*cp[1]*ep[1]*fp[2]*gp[2]-8*ap[1]*bp[1]*cp[2]*ep[1]*fp[2]*gp[2]- \\
 &32*ap[1]*ap[2]*ep[1]*ep[1]*fp[2]*gp[2]+32*ap[1]*ap[1]*ep[1]*ep[2]*fp[2]*gp[2]- \\
 &15*ap[2]*bp[1]*bp[1]*fp[1]*fp[2]*gp[2]+15*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*gp[2]+4*ap[2]*cp[1]*cp[1]*fp[1]*fp[2]*gp[2]- \\
 &4*ap[1]*cp[1]*cp[2]*fp[1]*fp[2]*gp[2]+44*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*gp[2]- \\
 &22*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*gp[2]+38*ap[1]*ap[2]*fp[1]*fp[1]*fp[2]*gp[2]- \\
 &11*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*gp[2]-19*ap[1]*ap[1]*fp[1]*fp[2]*fp[2]*gp[2]- \\
 &6*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*gp[2]+6*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*gp[2]+8*ap[2]*bp[2]*cp[1]*cp[1]*gp[1]*gp[2]+16*ap[2]*bp[1]*cp[1]*cp[2]*gp[1]*gp[2]- \\
 &16*ap[1]*bp[2]*cp[1]*cp[2]*gp[1]*gp[2]-8*ap[1]*bp[1]*cp[2]*cp[2]*gp[1]*gp[2]- \\
 &3*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*gp[2]+ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*gp[2]- \\
 &ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*gp[2]+3*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*gp[2]- \\
 &24*ap[2]*ap[2]*cp[1]*ep[1]*gp[1]*gp[2]+8*ap[1]*ap[2]*cp[2]*ep[1]*gp[1]*gp[2]- \\
 &8*ap[1]*ap[2]*cp[1]*ep[2]*gp[1]*gp[2]+24*ap[1]*ap[1]*cp[2]*ep[2]*gp[1]*gp[2]+21*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*gp[2]- \\
 &7*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*gp[2]+7*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*gp[2]- \\
 &21*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*gp[2]+3*ap[1]*ap[2]*ap[2]*gp[1]*gp[1]*gp[2]+3*ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*gp[2]-3*ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*gp[2]- \\
 &12*ap[2]*bp[1]*cp[1]*cp[1]*gp[2]*gp[2]+4*ap[1]*bp[2]*cp[1]*cp[1]*gp[2]*gp[2]+8*ap[1]*bp[1]*cp[1]*cp[2]*gp[2]*gp[2]+3*ap[1]*ap[2]*bp[1]*dp[1]*gp[2]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 2*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*gp[2]- \\
 & ap[1]*ap[1]*bp[1]*dp[2]*gp[2]*gp[2]+24*ap[1]*ap[2]*cp[1]*ep[1]*gp[2]*gp[2]- \\
 & 16*ap[1]*ap[1]*cp[2]*ep[1]*gp[2]*gp[2]-8*ap[1]*ap[1]*cp[1]*ep[2]*gp[2]*gp[2]- \\
 & 21*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*gp[2]+14*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*gp[2]+7*ap[1]*ap[1]* \\
 & bp[1]*fp[2]*gp[2]*gp[2]- \\
 & 3*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*gp[2]+ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*gp[2]+4*ap[2]*bp[2]*cp[1] \\
 & *dp[1]*dp[2]*hp[1]+4*ap[2]*bp[1]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 & 8*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 & 8*ap[2]*bp[1]*cp[1]*dp[2]*dp[2]*hp[1]+4*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*hp[1]+4*ap[1]*bp[1]*cp \\
 & [2]*dp[2]*dp[2]*hp[1]+8*ap[2]*bp[1]*bp[2]*dp[2]*ep[1]*hp[1]- \\
 & 8*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*hp[1]- \\
 & 16*ap[2]*ap[2]*dp[1]*dp[2]*ep[1]*hp[1]+16*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*hp[1]+8*ap[2]*bp[1]* \\
 & bp[2]*dp[1]*ep[2]*hp[1]-8*ap[1]*bp[2]*bp[2]*dp[1]*ep[2]*hp[1]- \\
 & 8*ap[2]*ap[2]*dp[1]*dp[1]*ep[2]*hp[1]- \\
 & 16*ap[2]*bp[1]*bp[1]*dp[2]*ep[2]*hp[1]+16*ap[1]*bp[1]*bp[2]*dp[2]*ep[2]*hp[1]+32*ap[1]*ap[2] \\
 & *dp[1]*dp[2]*ep[2]*hp[1]- \\
 & 24*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*hp[1]+4*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*hp[1]+4*ap[2]*bp[1]*c \\
 & p[2]*dp[2]*fp[1]*hp[1]-8*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*hp[1]- \\
 & 16*ap[2]*ap[2]*dp[2]*ep[1]*fp[1]*hp[1]+8*ap[2]*bp[1]*bp[2]*ep[2]*fp[1]*hp[1]- \\
 & 8*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*hp[1]- \\
 & 16*ap[2]*ap[2]*dp[1]*ep[2]*fp[1]*hp[1]+32*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*hp[1]- \\
 & 8*ap[2]*ap[2]*ep[2]*fp[1]*fp[1]*hp[1]+4*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*hp[1]+4*ap[2]*bp[1]*cp[\\
 & 2]*dp[1]*fp[2]*hp[1]-8*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*hp[1]- \\
 & 16*ap[2]*bp[1]*cp[1]*dp[2]*fp[2]*hp[1]+8*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*hp[1]+8*ap[1]*bp[1]*c \\
 & p[2]*dp[2]*fp[2]*hp[1]+8*ap[2]*bp[1]*bp[2]*ep[1]*fp[2]*hp[1]- \\
 & 8*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*hp[1]- \\
 & 16*ap[2]*ap[2]*dp[1]*ep[1]*fp[2]*hp[1]+32*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*hp[1]- \\
 & 16*ap[2]*bp[1]*bp[1]*ep[2]*fp[2]*hp[1]+16*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*hp[1]+32*ap[1]*ap[2] \\
 & *dp[1]*ep[2]*fp[2]*hp[1]- \\
 & 48*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*hp[1]+4*ap[2]*bp[2]*cp[1]*fp[1]*fp[2]*hp[1]+4*ap[2]*bp[1]*cp \\
 & [2]*fp[1]*fp[2]*hp[1]-8*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*hp[1]- \\
 & 16*ap[2]*ap[2]*ep[1]*fp[1]*fp[2]*hp[1]+32*ap[1]*ap[2]*ep[2]*fp[1]*fp[2]*hp[1]- \\
 & 8*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*hp[1]+4*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*hp[1]+4*ap[1]*bp[1]*cp[\\
 & 2]*fp[2]*fp[2]*hp[1]+16*ap[1]*ap[2]*ep[1]*fp[2]*fp[2]*hp[1]- \\
 & 24*ap[1]*ap[1]*ep[2]*fp[2]*fp[2]*hp[1]-8*ap[2]*bp[2]*bp[2]*cp[1]*gp[1]*hp[1]- \\
 & 16*ap[2]*bp[1]*bp[2]*cp[2]*gp[1]*hp[1]+24*ap[1]*bp[2]*bp[2]*cp[2]*gp[1]*hp[1]+8*ap[2]*ap[2]* \\
 & cp[2]*dp[1]*gp[1]*hp[1]+16*ap[2]*ap[2]*cp[1]*dp[2]*gp[1]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 24*ap[1]*ap[2]*cp[2]*dp[2]*gp[1]*hp[1]+16*ap[2]*ap[2]*bp[2]*cp[1]*gp[1]*hp[1]+32*ap[2]*ap[2]*bp[1]*ep[2]*gp[1]*hp[1]- \\
 & 48*ap[1]*ap[2]*bp[2]*ep[2]*gp[1]*hp[1]+8*ap[2]*ap[2]*cp[2]*fp[1]*gp[1]*hp[1]+16*ap[2]*ap[2]*cp[1]*fp[2]*gp[1]*hp[1]- \\
 & 24*ap[1]*ap[2]*cp[2]*fp[2]*gp[1]*hp[1]+16*ap[2]*bp[1]*bp[2]*cp[1]*gp[2]*hp[1]- \\
 & 8*ap[1]*bp[2]*bp[2]*cp[1]*gp[2]*hp[1]+8*ap[2]*bp[1]*bp[1]*cp[2]*gp[2]*hp[1]- \\
 & 16*ap[1]*bp[1]*bp[2]*cp[2]*gp[2]*hp[1]- \\
 & 12*ap[2]*ap[2]*cp[1]*dp[1]*gp[2]*hp[1]+4*ap[1]*ap[2]*cp[2]*dp[1]*gp[2]*hp[1]- \\
 & 4*ap[1]*ap[2]*cp[1]*dp[2]*gp[2]*hp[1]+12*ap[1]*ap[1]*cp[2]*dp[2]*gp[2]*hp[1]- \\
 & 24*ap[2]*ap[2]*bp[1]*ep[1]*gp[2]*hp[1]+8*ap[1]*ap[2]*bp[2]*ep[1]*gp[2]*hp[1]- \\
 & 8*ap[1]*ap[2]*bp[1]*ep[2]*gp[2]*hp[1]+24*ap[1]*ap[1]*bp[2]*ep[2]*gp[2]*hp[1]- \\
 & 12*ap[2]*ap[2]*cp[1]*fp[1]*gp[2]*hp[1]+4*ap[1]*ap[2]*cp[2]*fp[1]*gp[2]*hp[1]- \\
 & 4*ap[1]*ap[2]*cp[1]*fp[2]*gp[2]*hp[1]+12*ap[1]*ap[1]*cp[2]*fp[2]*gp[2]*hp[1]- \\
 & 4*ap[2]*bp[1]*bp[2]*bp[2]*hp[1]*hp[1]+4*ap[1]*bp[2]*bp[2]*bp[2]*hp[1]*hp[1]+4*ap[2]*ap[2]*bp[2]*dp[1]*hp[1]*hp[1]+8*ap[2]*ap[2]*bp[1]*dp[2]*hp[1]*hp[1]- \\
 & 12*ap[1]*ap[2]*bp[2]*dp[2]*hp[1]*hp[1]+4*ap[2]*ap[2]*bp[2]*fp[1]*hp[1]*hp[1]+8*ap[2]*ap[2]*bp[1]*fp[2]*hp[1]*hp[1]-12*ap[1]*ap[2]*bp[2]*fp[2]*hp[1]*hp[1]- \\
 & 12*ap[2]*ap[2]*ap[2]*gp[1]*hp[1]*hp[1]+12*ap[1]*ap[2]*ap[2]*gp[2]*hp[1]*hp[1]- \\
 & 4*ap[2]*bp[2]*cp[1]*dp[1]*dp[1]*hp[2]- \\
 & 4*ap[2]*bp[1]*cp[2]*dp[1]*dp[1]*hp[2]+8*ap[1]*bp[2]*cp[2]*dp[1]*dp[1]*hp[2]+8*ap[2]*bp[1]*cp[1]*dp[1]*dp[2]*hp[2]-4*ap[1]*bp[2]*cp[1]*dp[1]*dp[2]*hp[2]- \\
 & 4*ap[1]*bp[1]*cp[2]*dp[1]*dp[2]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[2]*dp[1]*ep[1]*hp[2]+16*ap[1]*bp[2]*bp[2]*dp[1]*ep[1]*hp[2]+24*ap[2]*ap[2]*dp[1]*dp[1]*ep[1]*hp[2]+8*ap[2]*bp[1]*bp[1]*dp[2]*ep[1]*hp[2]- \\
 & 8*ap[1]*bp[1]*bp[2]*dp[2]*ep[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*dp[1]*dp[2]*ep[1]*hp[2]+8*ap[1]*ap[1]*dp[2]*dp[2]*ep[1]*hp[2]+8*ap[2]*bp[1]*bp[1]*dp[1]*ep[2]*hp[2]-8*ap[1]*bp[1]*bp[2]*dp[1]*ep[2]*hp[2]- \\
 & 16*ap[1]*ap[2]*dp[1]*dp[1]*ep[2]*hp[2]+16*ap[1]*ap[1]*dp[1]*dp[2]*ep[2]*hp[2]- \\
 & 8*ap[2]*bp[2]*cp[1]*dp[1]*fp[1]*hp[2]- \\
 & 8*ap[2]*bp[1]*cp[2]*dp[1]*fp[1]*hp[2]+16*ap[1]*bp[2]*cp[2]*dp[1]*fp[1]*hp[2]+8*ap[2]*bp[1]*cp[1]*dp[2]*fp[1]*hp[2]-4*ap[1]*bp[2]*cp[1]*dp[2]*fp[1]*hp[2]- \\
 & 4*ap[1]*bp[1]*cp[2]*dp[2]*fp[1]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[2]*ep[1]*fp[1]*hp[2]+16*ap[1]*bp[2]*bp[2]*ep[1]*fp[1]*hp[2]+48*ap[2]*ap[2]*dp[1]*ep[1]*fp[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*dp[2]*ep[1]*fp[1]*hp[2]+8*ap[2]*bp[1]*bp[1]*ep[2]*fp[1]*hp[2]- \\
 & 8*ap[1]*bp[1]*bp[2]*ep[2]*fp[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*dp[1]*ep[2]*fp[1]*hp[2]+16*ap[1]*ap[1]*dp[2]*ep[2]*fp[1]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &4*ap[2]*bp[2]*cp[1]*fp[1]*fp[1]*hp[2]- \\
 &4*ap[2]*bp[1]*cp[2]*fp[1]*fp[1]*hp[2]+8*ap[1]*bp[2]*cp[2]*fp[1]*fp[1]*hp[2]+24*ap[2]*ap[2]*ep[1]*fp[1]*fp[1]*hp[2]- \\
 &16*ap[1]*ap[2]*ep[2]*fp[1]*fp[1]*hp[2]+8*ap[2]*bp[1]*cp[1]*dp[1]*fp[2]*hp[2]- \\
 &4*ap[1]*bp[2]*cp[1]*dp[1]*fp[2]*hp[2]- \\
 &4*ap[1]*bp[1]*cp[2]*dp[1]*fp[2]*hp[2]+8*ap[2]*bp[1]*bp[1]*cp[1]*fp[2]*hp[2]- \\
 &8*ap[1]*bp[1]*bp[2]*cp[1]*fp[2]*hp[2]- \\
 &32*ap[1]*ap[2]*dp[1]*ep[1]*fp[2]*hp[2]+16*ap[1]*ap[1]*dp[2]*ep[1]*fp[2]*hp[2]+16*ap[1]*ap[1]*dp[1]*ep[2]*fp[2]*hp[2]+8*ap[2]*bp[1]*cp[1]*fp[1]*fp[2]*hp[2]- \\
 &4*ap[1]*bp[2]*cp[1]*fp[1]*fp[2]*hp[2]-4*ap[1]*bp[1]*cp[2]*fp[1]*fp[2]*hp[2]- \\
 &32*ap[1]*ap[2]*ep[1]*fp[1]*fp[2]*hp[2]+16*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*hp[2]+8*ap[1]*ap[1]*ep[1]*fp[2]*fp[2]*hp[2]+16*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*hp[2]- \\
 &8*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*hp[2]+8*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*hp[2]- \\
 &16*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*hp[2]- \\
 &12*ap[2]*ap[2]*cp[1]*dp[1]*gp[1]*hp[2]+4*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*hp[2]- \\
 &4*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*hp[2]+12*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*hp[2]- \\
 &24*ap[2]*ap[2]*bp[1]*ep[1]*gp[1]*hp[2]+8*ap[1]*ap[2]*bp[2]*ep[1]*gp[1]*hp[2]- \\
 &8*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*hp[2]+24*ap[1]*ap[1]*bp[2]*cp[2]*gp[1]*hp[2]- \\
 &12*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*hp[2]+4*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*hp[2]- \\
 &4*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*hp[2]+12*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*hp[2]- \\
 &24*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*hp[2]+16*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*hp[2]+8*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*hp[2]+24*ap[1]*ap[2]*cp[1]*dp[1]*gp[2]*hp[2]- \\
 &16*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*hp[2]- \\
 &8*ap[1]*ap[1]*cp[1]*dp[2]*gp[2]*hp[2]+48*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*hp[2]- \\
 &32*ap[1]*ap[1]*bp[2]*ep[1]*gp[2]*hp[2]- \\
 &16*ap[1]*ap[1]*bp[1]*ep[2]*gp[2]*hp[2]+24*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*hp[2]- \\
 &16*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*hp[2]- \\
 &8*ap[1]*ap[1]*cp[1]*fp[2]*gp[2]*hp[2]+8*ap[2]*bp[1]*bp[1]*bp[2]*hp[1]*hp[2]- \\
 &8*ap[1]*bp[1]*bp[2]*bp[2]*hp[1]*hp[2]- \\
 &12*ap[2]*ap[2]*bp[1]*dp[1]*hp[1]*hp[2]+4*ap[1]*ap[2]*bp[2]*dp[1]*hp[1]*hp[2]- \\
 &4*ap[1]*ap[2]*bp[1]*dp[2]*hp[1]*hp[2]+12*ap[1]*ap[1]*bp[2]*dp[2]*hp[1]*hp[2]- \\
 &12*ap[2]*ap[2]*bp[1]*fp[1]*hp[1]*hp[2]+4*ap[1]*ap[2]*bp[2]*fp[1]*hp[1]*hp[2]- \\
 &4*ap[1]*ap[2]*bp[1]*fp[2]*hp[1]*hp[2]+12*ap[1]*ap[1]*bp[2]*fp[2]*hp[1]*hp[2]+24*ap[1]*ap[2]*ap[2]*gp[1]*hp[1]*hp[2]-24*ap[1]*ap[1]*ap[2]*gp[2]*hp[1]*hp[2]- \\
 &4*ap[2]*bp[1]*bp[1]*bp[1]*hp[2]*hp[2]+4*ap[1]*bp[1]*bp[1]*bp[2]*hp[2]*hp[2]+12*ap[1]*ap[2]*bp[1]*dp[1]*hp[2]*hp[2]-8*ap[1]*ap[1]*bp[2]*dp[1]*hp[2]*hp[2]- \\
 &4*ap[1]*ap[1]*bp[1]*dp[2]*hp[2]*hp[2]+12*ap[1]*ap[2]*bp[1]*fp[1]*hp[2]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &8*ap[1]*ap[1]*bp[2]*fp[1]*hp[2]*hp[2]-4*ap[1]*ap[1]*bp[1]*fp[2]*hp[2]*hp[2]- \\
 &12*ap[1]*ap[1]*ap[2]*gp[1]*hp[2]*hp[2]+12*ap[1]*ap[1]*ap[1]*gp[2]*hp[2]*hp[2] \\
 \\
 &w_3=4*ap[2]*bp[2]*cp[1]*dp[1]*dp[2]*gp[1]+4*ap[2]*bp[1]*cp[2]*dp[1]*dp[2]*gp[1]- \\
 &8*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*gp[1]- \\
 &8*ap[2]*bp[1]*cp[1]*dp[2]*dp[2]*gp[1]+4*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*gp[1]+4*ap[1]*bp[1]*cp \\
 &[2]*dp[2]*dp[2]*gp[1]- \\
 &8*ap[2]*bp[1]*bp[2]*dp[2]*ep[1]*gp[1]+8*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*gp[1]+16*ap[2]*cp[1]*c \\
 &p[2]*dp[2]*ep[1]*gp[1]-16*ap[1]*cp[2]*cp[2]*dp[2]*ep[1]*gp[1]- \\
 &16*ap[2]*ap[2]*dp[1]*dp[2]*ep[1]*gp[1]+16*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*gp[1]- \\
 &8*ap[2]*bp[1]*bp[2]*dp[1]*ep[2]*gp[1]+8*ap[1]*bp[2]*bp[2]*dp[1]*ep[2]*gp[1]+16*ap[2]*cp[1]*c \\
 &p[2]*dp[1]*ep[2]*gp[1]-16*ap[1]*cp[2]*cp[2]*dp[1]*ep[2]*gp[1]- \\
 &8*ap[2]*ap[2]*dp[1]*dp[1]*ep[2]*gp[1]+16*ap[2]*bp[1]*bp[1]*dp[2]*cp[2]*gp[1]- \\
 &16*ap[1]*bp[1]*bp[2]*dp[2]*cp[2]*gp[1]- \\
 &32*ap[2]*cp[1]*cp[1]*dp[2]*ep[2]*gp[1]+32*ap[1]*cp[1]*cp[2]*dp[2]*ep[2]*gp[1]+32*ap[1]*ap[2] \\
 &*dp[1]*dp[2]*ep[2]*gp[1]- \\
 &24*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*gp[1]+32*ap[2]*bp[2]*cp[1]*ep[1]*ep[2]*gp[1]+32*ap[2]*bp[1] \\
 &*cp[2]*ep[1]*ep[2]*gp[1]-64*ap[1]*bp[2]*cp[2]*ep[1]*ep[2]*gp[1]- \\
 &64*ap[2]*ap[2]*ep[1]*ep[1]*ep[2]*gp[1]- \\
 &64*ap[2]*bp[1]*cp[1]*ep[2]*ep[2]*gp[1]+32*ap[1]*bp[2]*cp[1]*ep[2]*ep[2]*gp[1]+32*ap[1]*bp[1] \\
 &*cp[2]*ep[2]*ep[2]*gp[1]+128*ap[1]*ap[2]*ep[1]*ep[2]*ep[2]*gp[1]- \\
 &64*ap[1]*ap[1]*ep[2]*ep[2]*ep[2]*gp[1]-12*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*gp[1]- \\
 &12*ap[2]*bp[1]*cp[2]*dp[2]*fp[1]*gp[1]+24*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*gp[1]+48*ap[2]*ap[2] \\
 &*dp[2]*ep[1]*fp[1]*gp[1]- \\
 &40*ap[2]*bp[1]*bp[2]*ep[2]*fp[1]*gp[1]+40*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*gp[1]+16*ap[2]*cp[1] \\
 &*cp[2]*ep[2]*fp[1]*gp[1]- \\
 &16*ap[1]*cp[2]*cp[2]*ep[2]*fp[1]*gp[1]+48*ap[2]*ap[2]*dp[1]*ep[2]*fp[1]*gp[1]- \\
 &96*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*gp[1]+56*ap[2]*ap[2]*ep[2]*fp[1]*fp[1]*gp[1]- \\
 &12*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*gp[1]- \\
 &12*ap[2]*bp[1]*cp[2]*dp[1]*fp[2]*gp[1]+24*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*gp[1]+48*ap[2]*bp[1] \\
 &*cp[1]*dp[2]*fp[2]*gp[1]-24*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*gp[1]- \\
 &24*ap[1]*bp[1]*cp[2]*dp[2]*fp[2]*gp[1]- \\
 &40*ap[2]*bp[1]*bp[2]*ep[1]*fp[2]*gp[1]+40*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*gp[1]+16*ap[2]*cp[1] \\
 &*cp[2]*ep[1]*fp[2]*gp[1]- \\
 &16*ap[1]*cp[2]*cp[2]*ep[1]*fp[2]*gp[1]+48*ap[2]*ap[2]*dp[1]*ep[1]*fp[2]*gp[1]- \\
 &96*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*gp[1]+80*ap[2]*bp[1]*bp[1]*ep[2]*fp[2]*gp[1]- \\
 &80*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*gp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 32*ap[2]*cp[1]*cp[1]*ep[2]*fp[2]*gp[1]+32*ap[1]*cp[1]*cp[2]*ep[2]*fp[2]*gp[1]- \\
 & 96*ap[1]*ap[2]*dp[1]*ep[2]*fp[2]*gp[1]+144*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*gp[1]- \\
 & 28*ap[2]*bp[2]*cp[1]*fp[1]*fp[2]*gp[1]- \\
 & 28*ap[2]*bp[1]*cp[2]*fp[1]*fp[2]*gp[1]+56*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*gp[1]+112*ap[2]*ap[2] \\
 & *ep[1]*fp[1]*fp[2]*gp[1]- \\
 & 224*ap[1]*ap[2]*ep[2]*fp[1]*fp[2]*gp[1]+56*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*gp[1]- \\
 & 28*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*gp[1]-28*ap[1]*bp[1]*cp[2]*fp[2]*fp[2]*gp[1]- \\
 & 112*ap[1]*ap[2]*ep[1]*fp[2]*fp[2]*gp[1]+168*ap[1]*ap[1]*ep[2]*fp[2]*fp[2]*gp[1]+4*ap[2]*bp[2] \\
 & *bp[2]*cp[1]*gp[1]*gp[1]+8*ap[2]*bp[1]*bp[2]*cp[2]*gp[1]*gp[1]- \\
 & 12*ap[1]*bp[2]*bp[2]*cp[2]*gp[1]*gp[1]- \\
 & 8*ap[2]*cp[1]*cp[2]*cp[2]*gp[1]*gp[1]+8*ap[1]*cp[2]*cp[2]*cp[2]*gp[1]*gp[1]+4*ap[2]*ap[2]*cp[\\
 & 2]*dp[1]*gp[1]*gp[1]+8*ap[2]*ap[2]*cp[1]*dp[2]*gp[1]*gp[1]- \\
 & 12*ap[1]*ap[2]*cp[2]*dp[2]*gp[1]*gp[1]-8*ap[2]*ap[2]*bp[2]*ep[1]*gp[1]*gp[1]- \\
 & 16*ap[2]*ap[2]*bp[1]*ep[2]*gp[1]*gp[1]+24*ap[1]*ap[2]*bp[2]*ep[2]*gp[1]*gp[1]- \\
 & 12*ap[2]*ap[2]*cp[2]*fp[1]*gp[1]*gp[1]- \\
 & 24*ap[2]*ap[2]*cp[1]*fp[2]*gp[1]*gp[1]+36*ap[1]*ap[2]*cp[2]*fp[2]*gp[1]*gp[1]- \\
 & 4*ap[2]*bp[2]*cp[1]*dp[1]*dp[1]*gp[2]- \\
 & 4*ap[2]*bp[1]*cp[2]*dp[1]*dp[1]*gp[2]+8*ap[1]*bp[2]*cp[2]*dp[1]*dp[1]*gp[2]+8*ap[2]*bp[1]*cp \\
 & [1]*dp[1]*dp[2]*gp[2]-4*ap[1]*bp[2]*cp[1]*dp[1]*dp[2]*gp[2]- \\
 & 4*ap[1]*bp[1]*cp[2]*dp[1]*dp[2]*gp[2]+16*ap[2]*bp[1]*bp[2]*dp[1]*ep[1]*gp[2]- \\
 & 16*ap[1]*bp[2]*bp[2]*dp[1]*ep[1]*gp[2]- \\
 & 32*ap[2]*cp[1]*cp[2]*dp[1]*ep[1]*gp[2]+32*ap[1]*cp[2]*cp[2]*dp[1]*ep[1]*gp[2]+24*ap[2]*ap[2] \\
 & *dp[1]*dp[1]*ep[1]*gp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*dp[2]*ep[1]*gp[2]+8*ap[1]*bp[1]*bp[2]*dp[2]*ep[1]*gp[2]+16*ap[2]*cp[1]*c \\
 & p[1]*dp[2]*ep[1]*gp[2]-16*ap[1]*cp[1]*cp[2]*dp[2]*ep[1]*gp[2]- \\
 & 32*ap[1]*ap[2]*dp[1]*dp[2]*ep[1]*gp[2]+8*ap[1]*ap[1]*dp[2]*dp[2]*ep[1]*gp[2]- \\
 & 32*ap[2]*bp[2]*cp[1]*ep[1]*ep[1]*gp[2]- \\
 & 32*ap[2]*bp[1]*cp[2]*ep[1]*ep[1]*gp[2]+64*ap[1]*bp[2]*cp[2]*ep[1]*ep[1]*gp[2]+64*ap[2]*ap[2] \\
 & *ep[1]*ep[1]*ep[1]*gp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*dp[1]*ep[2]*gp[2]+8*ap[1]*bp[1]*bp[2]*dp[1]*ep[2]*gp[2]+16*ap[2]*cp[1]*c \\
 & p[1]*dp[1]*ep[2]*gp[2]-16*ap[1]*cp[1]*cp[2]*dp[1]*ep[2]*gp[2]- \\
 & 16*ap[1]*ap[2]*dp[1]*dp[1]*ep[2]*gp[2]+16*ap[1]*ap[1]*dp[1]*dp[2]*ep[2]*gp[2]+64*ap[2]*bp[1] \\
 & *cp[1]*ep[1]*ep[2]*gp[2]-32*ap[1]*bp[2]*cp[1]*ep[1]*ep[2]*gp[2]- \\
 & 32*ap[1]*bp[1]*cp[2]*ep[1]*ep[2]*gp[2]- \\
 & 128*ap[1]*ap[2]*ep[1]*ep[1]*ep[2]*gp[2]+64*ap[1]*ap[1]*ep[1]*ep[2]*ep[2]*gp[2]+24*ap[2]*bp[2] \\
 & *cp[1]*dp[1]*fp[1]*gp[2]+24*ap[2]*bp[1]*cp[2]*dp[1]*fp[1]*gp[2]- \\
 & 48*ap[1]*bp[2]*cp[2]*dp[1]*fp[1]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 24*ap[2]*bp[1]*cp[1]*dp[2]*fp[1]*gp[2]+12*ap[1]*bp[2]*cp[1]*dp[2]*fp[1]*gp[2]+12*ap[1]*bp[1]* \\
 & *cp[2]*dp[2]*fp[1]*gp[2]+80*ap[2]*bp[1]*bp[2]*ep[1]*fp[1]*gp[2]- \\
 & 80*ap[1]*bp[2]*bp[2]*ep[1]*fp[1]*gp[2]- \\
 & 32*ap[2]*cp[1]*cp[2]*ep[1]*fp[1]*gp[2]+32*ap[1]*cp[2]*cp[2]*ep[1]*fp[1]*gp[2]- \\
 & 144*ap[2]*ap[2]*dp[1]*ep[1]*fp[1]*gp[2]+96*ap[1]*ap[2]*dp[2]*ep[1]*fp[1]*gp[2]- \\
 & 40*ap[2]*bp[1]*bp[1]*ep[2]*fp[1]*gp[2]+40*ap[1]*bp[1]*bp[2]*ep[2]*fp[1]*gp[2]+16*ap[2]*cp[1]* \\
 & *cp[1]*ep[2]*fp[1]*gp[2]- \\
 & 16*ap[1]*cp[1]*cp[2]*ep[2]*fp[1]*gp[2]+96*ap[1]*ap[2]*dp[1]*ep[2]*fp[1]*gp[2]- \\
 & 48*ap[1]*ap[1]*dp[2]*ep[2]*fp[1]*gp[2]+28*ap[2]*bp[2]*cp[1]*fp[1]*fp[1]*gp[2]+28*ap[2]*bp[1]* \\
 & cp[2]*fp[1]*fp[1]*gp[2]-56*ap[1]*bp[2]*cp[2]*fp[1]*fp[1]*gp[2]- \\
 & 168*ap[2]*ap[2]*ep[1]*fp[1]*fp[1]*gp[2]+112*ap[1]*ap[2]*ep[2]*fp[1]*fp[1]*gp[2]- \\
 & 24*ap[2]*bp[1]*cp[1]*dp[1]*fp[2]*gp[2]+12*ap[1]*bp[2]*cp[1]*dp[1]*fp[2]*gp[2]+12*ap[1]*bp[1]* \\
 & *cp[2]*dp[1]*fp[2]*gp[2]- \\
 & 40*ap[2]*bp[1]*bp[1]*cp[1]*fp[2]*gp[2]+40*ap[1]*bp[1]*bp[2]*cp[1]*fp[2]*gp[2]+16*ap[2]*cp[1]* \\
 & *cp[1]*ep[1]*fp[2]*gp[2]- \\
 & 16*ap[1]*cp[1]*cp[2]*ep[1]*fp[2]*gp[2]+96*ap[1]*ap[2]*dp[1]*ep[1]*fp[2]*gp[2]- \\
 & 48*ap[1]*ap[1]*dp[2]*ep[1]*fp[2]*gp[2]-48*ap[1]*ap[1]*dp[1]*ep[2]*fp[2]*gp[2]- \\
 & 56*ap[2]*bp[1]*cp[1]*fp[1]*fp[2]*gp[2]+28*ap[1]*bp[2]*cp[1]*fp[1]*fp[2]*gp[2]+28*ap[1]*bp[1]* \\
 & cp[2]*fp[1]*fp[2]*gp[2]+224*ap[1]*ap[2]*ep[1]*fp[1]*fp[2]*gp[2]- \\
 & 112*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*gp[2]-56*ap[1]*ap[1]*ep[1]*fp[2]*fp[2]*gp[2]- \\
 & 16*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*gp[2]+8*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*gp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*gp[2]+16*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*gp[2]+16*ap[2]*cp[1]* \\
 & cp[1]*cp[2]*gp[1]*gp[2]-16*ap[1]*cp[1]*cp[2]*cp[2]*gp[1]*gp[2]- \\
 & 12*ap[2]*ap[2]*cp[1]*dp[1]*gp[1]*gp[2]+4*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*gp[2]- \\
 & 4*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*gp[2]+12*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*gp[2]+24*ap[2]*ap[2]* \\
 & bp[1]*ep[1]*gp[1]*gp[2]- \\
 & 8*ap[1]*ap[2]*bp[2]*cp[1]*gp[1]*gp[2]+8*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*gp[2]- \\
 & 24*ap[1]*ap[1]*bp[2]*ep[2]*gp[1]*gp[2]+36*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*gp[2]- \\
 & 12*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*gp[2]+12*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*gp[2]- \\
 & 36*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*gp[2]+12*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*gp[2]- \\
 & 8*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*gp[2]-8*ap[2]*cp[1]*cp[1]*cp[1]*gp[2]*gp[2]- \\
 & 4*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*gp[2]+8*ap[1]*cp[1]*cp[1]*cp[2]*gp[2]*gp[2]+12*ap[1]*ap[2]*c \\
 & p[1]*dp[1]*gp[2]*gp[2]-8*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*gp[2]- \\
 & 4*ap[1]*ap[1]*cp[1]*dp[2]*gp[2]*gp[2]- \\
 & 24*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*gp[2]+16*ap[1]*ap[1]*bp[2]*ep[1]*gp[2]*gp[2]+8*ap[1]*ap[1]* \\
 & bp[1]*ep[2]*gp[2]*gp[2]- \\
 & 36*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*gp[2]+24*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*gp[2]+12*ap[1]*ap[1]*
 \end{aligned}$$

$$\begin{aligned}
 & cp[1]*fp[2]*gp[2]*gp[2]+4*ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*hp[1]- \\
 & 4*ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*hp[1]+8*ap[2]*cp[1]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 & 8*ap[1]*cp[2]*cp[2]*dp[1]*dp[2]*hp[1]-12*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*hp[1]- \\
 & 4*ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*hp[1]+4*ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*hp[1]- \\
 & 8*ap[2]*cp[1]*cp[1]*dp[2]*dp[2]*hp[1]+8*ap[1]*cp[1]*cp[2]*dp[2]*dp[2]*hp[1]+24*ap[1]*ap[2]*d \\
 & p[1]*dp[2]*dp[2]*hp[1]- \\
 & 12*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*hp[1]+16*ap[2]*bp[2]*cp[1]*dp[2]*ep[1]*hp[1]+16*ap[2]*bp[1] \\
 & *cp[2]*dp[2]*ep[1]*hp[1]-32*ap[1]*bp[2]*cp[2]*dp[2]*ep[1]*hp[1]- \\
 & 32*ap[2]*ap[2]*dp[2]*ep[1]*ep[1]*hp[1]+16*ap[2]*bp[2]*cp[1]*dp[1]*ep[2]*hp[1]+16*ap[2]*bp[1] \\
 & *cp[2]*dp[1]*ep[2]*hp[1]-32*ap[1]*bp[2]*cp[2]*dp[1]*ep[2]*hp[1]- \\
 & 64*ap[2]*bp[1]*cp[1]*dp[2]*ep[2]*hp[1]+32*ap[1]*bp[2]*cp[1]*dp[2]*ep[2]*hp[1]+32*ap[1]*bp[1] \\
 & *cp[2]*dp[2]*ep[2]*hp[1]+32*ap[2]*bp[1]*bp[2]*ep[1]*ep[2]*hp[1]- \\
 & 32*ap[1]*bp[2]*bp[2]*ep[1]*ep[2]*hp[1]- \\
 & 64*ap[2]*ap[2]*dp[1]*ep[1]*ep[2]*hp[1]+128*ap[1]*ap[2]*dp[2]*ep[1]*ep[2]*hp[1]- \\
 & 32*ap[2]*bp[1]*bp[1]*ep[2]*ep[2]*hp[1]+32*ap[1]*bp[1]*bp[2]*ep[2]*ep[2]*hp[1]+64*ap[1]*ap[2] \\
 & *dp[1]*ep[2]*ep[2]*hp[1]-96*ap[1]*ap[1]*dp[2]*ep[2]*ep[2]*hp[1]- \\
 & 12*ap[2]*bp[1]*bp[2]*dp[2]*fp[1]*hp[1]+12*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*hp[1]+8*ap[2]*cp[1]* \\
 & cp[2]*dp[2]*fp[1]*hp[1]- \\
 & 8*ap[1]*cp[2]*cp[2]*dp[2]*fp[1]*hp[1]+8*ap[2]*ap[2]*dp[1]*dp[2]*fp[1]*hp[1]- \\
 & 8*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*hp[1]+16*ap[2]*bp[2]*cp[1]*ep[2]*fp[1]*hp[1]+16*ap[2]*bp[1]*c \\
 & p[2]*ep[2]*fp[1]*hp[1]-32*ap[1]*bp[2]*cp[2]*ep[2]*fp[1]*hp[1]- \\
 & 64*ap[2]*ap[2]*ep[1]*ep[2]*fp[1]*hp[1]+64*ap[1]*ap[2]*ep[2]*ep[2]*fp[1]*hp[1]+20*ap[2]*ap[2]* \\
 & dp[2]*fp[1]*fp[1]*hp[1]- \\
 & 12*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*hp[1]+12*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*hp[1]+8*ap[2]*cp[1]* \\
 & cp[2]*dp[1]*fp[2]*hp[1]- \\
 & 8*ap[1]*cp[2]*cp[2]*dp[1]*fp[2]*hp[1]+4*ap[2]*ap[2]*dp[1]*dp[1]*fp[2]*hp[1]+24*ap[2]*bp[1]*b \\
 & p[1]*dp[2]*fp[2]*hp[1]-24*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*hp[1]- \\
 & 16*ap[2]*cp[1]*cp[1]*dp[2]*fp[2]*hp[1]+16*ap[1]*cp[1]*cp[2]*dp[2]*fp[2]*hp[1]- \\
 & 16*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*hp[1]+12*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*hp[1]+16*ap[2]*bp[2] \\
 & *cp[1]*ep[1]*fp[2]*hp[1]+16*ap[2]*bp[1]*cp[2]*ep[1]*fp[2]*hp[1]- \\
 & 32*ap[1]*bp[2]*cp[2]*ep[1]*fp[2]*hp[1]-32*ap[2]*ap[2]*ep[1]*ep[1]*fp[2]*hp[1]- \\
 & 64*ap[2]*bp[1]*cp[1]*ep[2]*fp[2]*hp[1]+32*ap[1]*bp[2]*cp[1]*ep[2]*fp[2]*hp[1]+32*ap[1]*bp[1]* \\
 & cp[2]*ep[2]*fp[2]*hp[1]+128*ap[1]*ap[2]*ep[1]*ep[2]*fp[2]*hp[1]- \\
 & 96*ap[1]*ap[1]*ep[2]*ep[2]*fp[2]*hp[1]- \\
 & 28*ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*hp[1]+28*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*hp[1]+8*ap[2]*cp[1]*c \\
 & p[2]*fp[1]*fp[2]*hp[1]- \\
 & 8*ap[1]*cp[2]*cp[2]*fp[1]*fp[2]*hp[1]+40*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 &80*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*hp[1]+36*ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*hp[1]+28*ap[2]*bp[1]* \\
 &bp[1]*fp[2]*fp[2]*hp[1]-28*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*hp[1]- \\
 &8*ap[2]*cp[1]*cp[1]*fp[2]*fp[2]*hp[1]+8*ap[1]*cp[1]*cp[2]*fp[2]*fp[2]*hp[1]- \\
 &40*ap[1]*ap[2]*dp[1]*fp[2]*fp[2]*hp[1]+60*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*hp[1]- \\
 &72*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*hp[1]+36*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*hp[1]+8*ap[2]*bp[1]*bp \\
 &[2]*bp[2]*gp[1]*hp[1]-8*ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*hp[1]- \\
 &32*ap[2]*bp[2]*cp[1]*cp[2]*gp[1]*hp[1]- \\
 &16*ap[2]*bp[1]*cp[2]*cp[2]*gp[1]*hp[1]+48*ap[1]*bp[2]*cp[2]*cp[2]*gp[1]*hp[1]+8*ap[2]*ap[2]* \\
 &bp[2]*dp[1]*gp[1]*hp[1]+16*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*hp[1]- \\
 &24*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*hp[1]+32*ap[2]*ap[2]*cp[2]*ep[1]*gp[1]*hp[1]+64*ap[2]*ap[2] \\
 &*cp[1]*cp[2]*gp[1]*hp[1]-96*ap[1]*ap[2]*cp[2]*ep[2]*gp[1]*hp[1]- \\
 &24*ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*hp[1]- \\
 &48*ap[2]*ap[2]*bp[1]*fp[2]*gp[1]*hp[1]+72*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*hp[1]- \\
 &12*ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*hp[1]- \\
 &8*ap[2]*bp[1]*bp[1]*bp[2]*gp[2]*hp[1]+8*ap[1]*bp[1]*bp[2]*bp[2]*gp[2]*hp[1]+16*ap[2]*bp[2]*c \\
 &p[1]*cp[1]*gp[2]*hp[1]+32*ap[2]*bp[1]*cp[1]*cp[2]*gp[2]*hp[1]- \\
 &32*ap[1]*bp[2]*cp[1]*cp[2]*gp[2]*hp[1]-16*ap[1]*bp[1]*cp[2]*cp[2]*gp[2]*hp[1]- \\
 &12*ap[2]*ap[2]*bp[1]*dp[1]*gp[2]*hp[1]+4*ap[1]*ap[2]*bp[2]*dp[1]*gp[2]*hp[1]- \\
 &4*ap[1]*ap[2]*bp[1]*dp[2]*gp[2]*hp[1]+12*ap[1]*ap[1]*bp[2]*dp[2]*gp[2]*hp[1]- \\
 &48*ap[2]*ap[2]*cp[1]*ep[1]*gp[2]*hp[1]+16*ap[1]*ap[2]*cp[2]*ep[1]*gp[2]*hp[1]- \\
 &16*ap[1]*ap[2]*cp[1]*ep[2]*gp[2]*hp[1]+48*ap[1]*ap[1]*cp[2]*ep[2]*gp[2]*hp[1]+36*ap[2]*ap[2] \\
 &*bp[1]*fp[1]*gp[2]*hp[1]- \\
 &12*ap[1]*ap[2]*bp[2]*fp[1]*gp[2]*hp[1]+12*ap[1]*ap[2]*bp[1]*fp[2]*gp[2]*hp[1]- \\
 &36*ap[1]*ap[1]*bp[2]*fp[2]*gp[2]*hp[1]+24*ap[1]*ap[2]*ap[2]*gp[1]*gp[2]*hp[1]- \\
 &12*ap[1]*ap[1]*ap[2]*gp[2]*gp[2]*hp[1]-8*ap[2]*bp[2]*bp[2]*cp[1]*hp[1]*hp[1]- \\
 &16*ap[2]*bp[1]*bp[2]*cp[2]*hp[1]*hp[1]+24*ap[1]*bp[2]*bp[2]*cp[2]*hp[1]*hp[1]+8*ap[2]*ap[2]* \\
 &cp[2]*dp[1]*hp[1]*hp[1]+16*ap[2]*ap[2]*cp[1]*dp[2]*hp[1]*hp[1]- \\
 &24*ap[1]*ap[2]*cp[2]*dp[2]*hp[1]*hp[1]+16*ap[2]*ap[2]*bp[2]*ep[1]*hp[1]*hp[1]+32*ap[2]*ap[2] \\
 &*bp[1]*ep[2]*hp[1]*hp[1]- \\
 &48*ap[1]*ap[2]*bp[2]*ep[2]*hp[1]*hp[1]+8*ap[2]*ap[2]*cp[2]*fp[1]*hp[1]*hp[1]+16*ap[2]*ap[2]* \\
 &cp[1]*fp[2]*hp[1]*hp[1]-24*ap[1]*ap[2]*cp[2]*fp[2]*hp[1]*hp[1]- \\
 &8*ap[2]*ap[2]*ap[2]*hp[1]*hp[1]*hp[1]- \\
 &4*ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*hp[2]+4*ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*hp[2]- \\
 &8*ap[2]*cp[1]*cp[2]*dp[1]*dp[1]*hp[2]+8*ap[1]*cp[2]*cp[2]*dp[1]*dp[1]*hp[2]+12*ap[2]*ap[2]*d \\
 &p[1]*dp[1]*dp[1]*hp[2]+4*ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*hp[2]- \\
 &4*ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*hp[2]+8*ap[2]*cp[1]*cp[1]*dp[1]*dp[2]*hp[2]- \\
 &8*ap[1]*cp[1]*cp[2]*dp[1]*dp[2]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &24*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*hp[2]+12*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*hp[2]- \\
 &32*ap[2]*bp[2]*cp[1]*dp[1]*ep[1]*hp[2]- \\
 &32*ap[2]*bp[1]*cp[2]*dp[1]*ep[1]*hp[2]+64*ap[1]*bp[2]*cp[2]*dp[1]*ep[1]*hp[2]+32*ap[2]*bp[1] \\
 &*cp[1]*dp[2]*ep[1]*hp[2]-16*ap[1]*bp[2]*cp[1]*dp[2]*ep[1]*hp[2]- \\
 &16*ap[1]*bp[1]*cp[2]*dp[2]*ep[1]*hp[2]- \\
 &32*ap[2]*bp[1]*bp[2]*ep[1]*ep[1]*hp[2]+32*ap[1]*bp[2]*bp[2]*cp[1]*ep[1]*hp[2]+96*ap[2]*ap[2] \\
 &*dp[1]*ep[1]*ep[1]*hp[2]- \\
 &64*ap[1]*ap[2]*dp[2]*ep[1]*ep[1]*hp[2]+32*ap[2]*bp[1]*cp[1]*dp[1]*ep[2]*hp[2]- \\
 &16*ap[1]*bp[2]*cp[1]*dp[1]*ep[2]*hp[2]- \\
 &16*ap[1]*bp[1]*cp[2]*dp[1]*ep[2]*hp[2]+32*ap[2]*bp[1]*bp[1]*ep[1]*ep[2]*hp[2]- \\
 &32*ap[1]*bp[1]*bp[2]*ep[1]*ep[2]*hp[2]- \\
 &128*ap[1]*ap[2]*dp[1]*ep[1]*ep[2]*hp[2]+64*ap[1]*ap[1]*dp[2]*ep[1]*ep[2]*hp[2]+32*ap[1]*ap[1] \\
 &]*dp[1]*ep[2]*ep[2]*hp[2]+24*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*hp[2]- \\
 &24*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*hp[2]- \\
 &16*ap[2]*cp[1]*cp[2]*dp[1]*fp[1]*hp[2]+16*ap[1]*cp[2]*cp[2]*dp[1]*fp[1]*hp[2]- \\
 &12*ap[2]*ap[2]*dp[1]*dp[1]*fp[1]*hp[2]- \\
 &12*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*hp[2]+12*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*hp[2]+8*ap[2]*cp[1]* \\
 &cp[1]*dp[2]*fp[1]*hp[2]- \\
 &8*ap[1]*cp[1]*cp[2]*dp[2]*fp[1]*hp[2]+16*ap[1]*ap[2]*dp[1]*dp[2]*fp[1]*hp[2]- \\
 &4*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*hp[2]-32*ap[2]*bp[2]*cp[1]*ep[1]*fp[1]*hp[2]- \\
 &32*ap[2]*bp[1]*cp[2]*ep[1]*fp[1]*hp[2]+64*ap[1]*bp[2]*cp[2]*ep[1]*fp[1]*hp[2]+96*ap[2]*ap[2]* \\
 &ep[1]*ep[1]*fp[1]*hp[2]+32*ap[2]*bp[1]*cp[1]*ep[2]*fp[1]*hp[2]- \\
 &16*ap[1]*bp[2]*cp[1]*ep[2]*fp[1]*hp[2]-16*ap[1]*bp[1]*cp[2]*ep[2]*fp[1]*hp[2]- \\
 &128*ap[1]*ap[2]*ep[1]*ep[2]*fp[1]*hp[2]+32*ap[1]*ap[1]*ep[2]*ep[2]*fp[1]*hp[2]+28*ap[2]*bp[1] \\
 &*bp[2]*fp[1]*fp[1]*hp[2]-28*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*hp[2]- \\
 &8*ap[2]*cp[1]*cp[2]*fp[1]*fp[1]*hp[2]+8*ap[1]*cp[2]*cp[2]*fp[1]*fp[1]*hp[2]- \\
 &60*ap[2]*ap[2]*dp[1]*fp[1]*fp[1]*hp[2]+40*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*hp[2]- \\
 &36*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*hp[2]- \\
 &12*ap[2]*bp[1]*bp[1]*dp[1]*fp[2]*hp[2]+12*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*hp[2]+8*ap[2]*cp[1]* \\
 &cp[1]*dp[1]*fp[2]*hp[2]- \\
 &8*ap[1]*cp[1]*cp[2]*dp[1]*fp[2]*hp[2]+8*ap[1]*ap[2]*dp[1]*dp[1]*fp[2]*hp[2]- \\
 &8*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*hp[2]+32*ap[2]*bp[1]*cp[1]*ep[1]*fp[2]*hp[2]- \\
 &16*ap[1]*bp[2]*cp[1]*ep[1]*fp[2]*hp[2]-16*ap[1]*bp[1]*cp[2]*ep[1]*fp[2]*hp[2]- \\
 &64*ap[1]*ap[2]*ep[1]*ep[1]*fp[2]*hp[2]+64*ap[1]*ap[1]*ep[1]*ep[2]*fp[2]*hp[2]- \\
 &28*ap[2]*bp[1]*bp[1]*fp[1]*fp[2]*hp[2]+28*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*hp[2]+8*ap[2]*cp[1]*c \\
 &p[1]*fp[1]*fp[2]*hp[2]- \\
 &8*ap[1]*cp[1]*cp[2]*fp[1]*fp[2]*hp[2]+80*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &40*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*hp[2]+72*ap[1]*ap[2]*fp[1]*fp[1]*fp[2]*hp[2]- \\
 &20*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*hp[2]-36*ap[1]*ap[1]*fp[1]*fp[2]*fp[2]*hp[2]- \\
 &8*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*hp[2]+8*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*hp[2]+16*ap[2]*bp[2]*c \\
 &p[1]*cp[1]*gp[1]*hp[2]+32*ap[2]*bp[1]*cp[1]*cp[2]*gp[1]*hp[2]- \\
 &32*ap[1]*bp[2]*cp[1]*cp[2]*gp[1]*hp[2]-16*ap[1]*bp[1]*cp[2]*cp[2]*gp[1]*hp[2]- \\
 &12*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*hp[2]+4*ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*hp[2]- \\
 &4*ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*hp[2]+12*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*hp[2]- \\
 &48*ap[2]*ap[2]*cp[1]*ep[1]*gp[1]*hp[2]+16*ap[1]*ap[2]*cp[2]*ep[1]*gp[1]*hp[2]- \\
 &16*ap[1]*ap[2]*cp[1]*ep[2]*gp[1]*hp[2]+48*ap[1]*ap[1]*cp[2]*ep[2]*gp[1]*hp[2]+36*ap[2]*ap[2] \\
 &*bp[1]*fp[1]*gp[1]*hp[2]- \\
 &12*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*hp[2]+12*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*hp[2]- \\
 &36*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*hp[2]+12*ap[1]*ap[2]*ap[2]*gp[1]*gp[1]*hp[2]+8*ap[2]*bp[1]* \\
 &bp[1]*bp[1]*gp[2]*hp[2]-8*ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*hp[2]- \\
 &48*ap[2]*bp[1]*cp[1]*cp[1]*gp[2]*hp[2]+16*ap[1]*bp[2]*cp[1]*cp[1]*gp[2]*hp[2]+32*ap[1]*bp[1] \\
 &*cp[1]*cp[2]*gp[2]*hp[2]+24*ap[1]*ap[2]*bp[1]*dp[1]*gp[2]*hp[2]- \\
 &16*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*hp[2]- \\
 &8*ap[1]*ap[1]*bp[1]*dp[2]*gp[2]*hp[2]+96*ap[1]*ap[2]*cp[1]*ep[1]*gp[2]*hp[2]- \\
 &64*ap[1]*ap[1]*cp[2]*ep[1]*gp[2]*hp[2]-32*ap[1]*ap[1]*cp[1]*ep[2]*gp[2]*hp[2]- \\
 &72*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*hp[2]+48*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*hp[2]+24*ap[1]*ap[1] \\
 &*bp[1]*fp[2]*gp[2]*hp[2]- \\
 &24*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*hp[2]+12*ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*hp[2]+32*ap[2]*bp[1] \\
 &*bp[2]*cp[1]*hp[1]*hp[2]- \\
 &16*ap[1]*bp[2]*bp[2]*cp[1]*hp[1]*hp[2]+16*ap[2]*bp[1]*bp[1]*cp[2]*hp[1]*hp[2]- \\
 &32*ap[1]*bp[1]*bp[2]*cp[2]*hp[1]*hp[2]- \\
 &24*ap[2]*ap[2]*cp[1]*dp[1]*hp[1]*hp[2]+8*ap[1]*ap[2]*cp[2]*dp[1]*hp[1]*hp[2]- \\
 &8*ap[1]*ap[2]*cp[1]*dp[2]*hp[1]*hp[2]+24*ap[1]*ap[1]*cp[2]*dp[2]*hp[1]*hp[2]- \\
 &48*ap[2]*ap[2]*bp[1]*ep[1]*hp[1]*hp[2]+16*ap[1]*ap[2]*bp[2]*ep[1]*hp[1]*hp[2]- \\
 &16*ap[1]*ap[2]*bp[1]*ep[2]*hp[1]*hp[2]+48*ap[1]*ap[1]*bp[2]*ep[2]*hp[1]*hp[2]- \\
 &24*ap[2]*ap[2]*cp[1]*fp[1]*hp[1]*hp[2]+8*ap[1]*ap[2]*cp[2]*fp[1]*hp[1]*hp[2]- \\
 &8*ap[1]*ap[2]*cp[1]*fp[2]*hp[1]*hp[2]+24*ap[1]*ap[1]*cp[2]*fp[2]*hp[1]*hp[2]+24*ap[1]*ap[2]*a \\
 &p[2]*hp[1]*hp[1]*hp[2]- \\
 &24*ap[2]*bp[1]*bp[1]*cp[1]*hp[2]*hp[2]+16*ap[1]*bp[1]*bp[2]*cp[1]*hp[2]*hp[2]+8*ap[1]*bp[1]* \\
 &bp[1]*cp[2]*hp[2]*hp[2]+24*ap[1]*ap[2]*cp[1]*dp[1]*hp[2]*hp[2]- \\
 &16*ap[1]*ap[1]*cp[2]*dp[1]*hp[2]*hp[2]- \\
 &8*ap[1]*ap[1]*cp[1]*dp[2]*hp[2]*hp[2]+48*ap[1]*ap[2]*bp[1]*ep[1]*hp[2]*hp[2]- \\
 &32*ap[1]*ap[1]*bp[2]*ep[1]*hp[2]*hp[2]- \\
 &16*ap[1]*ap[1]*bp[1]*ep[2]*hp[2]*hp[2]+24*ap[1]*ap[2]*cp[1]*fp[1]*hp[2]*hp[2]-
 \end{aligned}$$

$$16*ap[1]*ap[1]*cp[2]*fp[1]*hp[2]*hp[2]-8*ap[1]*ap[1]*cp[1]*fp[2]*hp[2]*hp[2]-$$

$$24*ap[1]*ap[1]*ap[2]*hp[1]*hp[2]*hp[2]+8*ap[1]*ap[1]*ap[1]*hp[2]*hp[2]*hp[2]$$

$w_4=-$

$$3*ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*gp[1]+3*ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*gp[1]+12*ap[2]*cp[1]*c$$

$$p[2]*dp[1]*dp[2]*gp[1]-12*ap[1]*cp[2]*cp[2]*dp[1]*dp[2]*gp[1]-$$

$$9*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*gp[1]+3*ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*gp[1]-$$

$$3*ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*gp[1]-$$

$$12*ap[2]*cp[1]*cp[1]*dp[2]*dp[2]*gp[1]+12*ap[1]*cp[1]*cp[2]*dp[2]*dp[2]*gp[1]+18*ap[1]*ap[2]$$

$$*dp[1]*dp[2]*dp[2]*gp[1]-9*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*gp[1]-$$

$$8*ap[2]*bp[2]*cp[1]*dp[2]*ep[1]*gp[1]-$$

$$8*ap[2]*bp[1]*cp[2]*dp[2]*ep[1]*gp[1]+16*ap[1]*bp[2]*cp[2]*dp[2]*ep[1]*gp[1]+16*ap[2]*ap[2]*$$

$$dp[2]*ep[1]*ep[1]*gp[1]-8*ap[2]*bp[2]*cp[1]*dp[1]*cp[2]*gp[1]-$$

$$8*ap[2]*bp[1]*cp[2]*dp[1]*ep[2]*gp[1]+16*ap[1]*bp[2]*cp[2]*dp[1]*ep[2]*gp[1]+32*ap[2]*bp[1]*$$

$$cp[1]*dp[2]*ep[2]*gp[1]-16*ap[1]*bp[2]*cp[1]*dp[2]*ep[2]*gp[1]-$$

$$16*ap[1]*bp[1]*cp[2]*dp[2]*ep[2]*gp[1]-$$

$$80*ap[2]*bp[1]*bp[2]*ep[1]*ep[2]*gp[1]+80*ap[1]*bp[2]*bp[2]*ep[1]*ep[2]*gp[1]+64*ap[2]*cp[1]$$

$$*cp[2]*ep[1]*ep[2]*gp[1]-$$

$$64*ap[1]*cp[2]*cp[2]*ep[1]*ep[2]*gp[1]+32*ap[2]*ap[2]*dp[1]*ep[1]*ep[2]*gp[1]-$$

$$64*ap[1]*ap[2]*dp[2]*ep[1]*ep[2]*gp[1]+80*ap[2]*bp[1]*bp[1]*ep[2]*ep[2]*gp[1]-$$

$$80*ap[1]*bp[1]*bp[2]*ep[2]*ep[2]*gp[1]-$$

$$64*ap[2]*cp[1]*cp[1]*ep[2]*ep[2]*gp[1]+64*ap[1]*cp[1]*cp[2]*ep[2]*ep[2]*gp[1]-$$

$$32*ap[1]*ap[2]*dp[1]*ep[2]*ep[2]*gp[1]+48*ap[1]*ap[1]*dp[2]*ep[2]*ep[2]*gp[1]+5*ap[2]*bp[1]*$$

$$bp[2]*dp[2]*fp[1]*gp[1]-5*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*gp[1]-$$

$$20*ap[2]*cp[1]*cp[2]*dp[2]*fp[1]*gp[1]+20*ap[1]*cp[2]*cp[2]*dp[2]*fp[1]*gp[1]+30*ap[2]*ap[2]*$$

$$dp[1]*dp[2]*fp[1]*gp[1]-30*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*gp[1]-$$

$$72*ap[2]*bp[2]*cp[1]*ep[2]*fp[1]*gp[1]-$$

$$72*ap[2]*bp[1]*cp[2]*ep[2]*fp[1]*gp[1]+144*ap[1]*bp[2]*cp[2]*ep[2]*fp[1]*gp[1]+288*ap[2]*ap[$$

$$2]*ep[1]*ep[2]*fp[1]*gp[1]-288*ap[1]*ap[2]*cp[2]*ep[2]*fp[1]*gp[1]-$$

$$25*ap[2]*ap[2]*dp[2]*fp[1]*fp[1]*gp[1]+5*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*gp[1]-$$

$$5*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*gp[1]-$$

$$20*ap[2]*cp[1]*cp[2]*dp[1]*fp[2]*gp[1]+20*ap[1]*cp[2]*cp[2]*dp[1]*fp[2]*gp[1]+15*ap[2]*ap[2]*$$

$$dp[1]*dp[1]*fp[2]*gp[1]-$$

$$10*ap[2]*bp[1]*bp[1]*dp[2]*fp[2]*gp[1]+10*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*gp[1]+40*ap[2]*cp[1]$$

$$*cp[1]*dp[2]*fp[2]*gp[1]-40*ap[1]*cp[1]*cp[2]*dp[2]*fp[2]*gp[1]-$$

$$60*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*gp[1]+45*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*gp[1]-$$

$$72*ap[2]*bp[2]*cp[1]*ep[1]*fp[2]*gp[1]-$$

$$\begin{aligned}
 &72*ap[2]*bp[1]*cp[2]*ep[1]*fp[2]*gp[1]+144*ap[1]*bp[2]*cp[2]*ep[1]*fp[2]*gp[1]+144*ap[2]*ap[2]*ep[1]*ep[1]*fp[2]*gp[1]+288*ap[2]*bp[1]*cp[1]*ep[2]*fp[2]*gp[1]- \\
 &144*ap[1]*bp[2]*cp[1]*ep[2]*fp[2]*gp[1]-144*ap[1]*bp[1]*cp[2]*ep[2]*fp[2]*gp[1]- \\
 &576*ap[1]*ap[2]*ep[1]*ep[2]*fp[2]*gp[1]+432*ap[1]*ap[1]*ep[2]*ep[2]*fp[2]*gp[1]+77*ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*gp[1]-77*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*gp[1]- \\
 &52*ap[2]*cp[1]*cp[2]*fp[1]*fp[2]*gp[1]+52*ap[1]*cp[2]*cp[2]*fp[1]*fp[2]*gp[1]- \\
 &50*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*gp[1]+100*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*gp[1]- \\
 &129*ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*gp[1]- \\
 &77*ap[2]*bp[1]*bp[1]*fp[2]*fp[2]*gp[1]+77*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*gp[1]+52*ap[2]*cp[1]*cp[1]*fp[2]*fp[2]*gp[1]- \\
 &52*ap[1]*cp[1]*cp[2]*fp[2]*fp[2]*gp[1]+50*ap[1]*ap[2]*dp[1]*fp[2]*fp[2]*gp[1]- \\
 &75*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*gp[1]+258*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*gp[1]- \\
 &129*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*gp[1]- \\
 &ap[2]*bp[1]*bp[2]*bp[2]*gp[1]*gp[1]+ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*gp[1]+8*ap[2]*bp[2]*cp[1]*cp[2]*gp[1]*gp[1]+4*ap[2]*bp[1]*cp[2]*cp[2]*gp[1]*gp[1]- \\
 &12*ap[1]*bp[2]*cp[2]*cp[2]*gp[1]*gp[1]-3*ap[2]*ap[2]*bp[2]*dp[1]*gp[1]*gp[1]- \\
 &6*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*gp[1]+9*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*gp[1]- \\
 &8*ap[2]*ap[2]*cp[2]*ep[1]*gp[1]*gp[1]- \\
 &16*ap[2]*ap[2]*cp[1]*ep[2]*gp[1]*gp[1]+24*ap[1]*ap[2]*cp[2]*ep[2]*gp[1]*gp[1]+5*ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*gp[1]+10*ap[2]*ap[2]*bp[1]*fp[2]*gp[1]*gp[1]- \\
 &15*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*gp[1]+3*ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*gp[1]+3*ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*gp[2]-3*ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*gp[2]- \\
 &12*ap[2]*cp[1]*cp[2]*dp[1]*dp[1]*gp[2]+12*ap[1]*cp[2]*cp[2]*dp[1]*dp[1]*gp[2]+9*ap[2]*ap[2]*dp[1]*dp[1]*dp[1]*gp[2]- \\
 &3*ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*gp[2]+3*ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*gp[2]+12*ap[2]*cp[1]*cp[1]*dp[1]*dp[2]*gp[2]-12*ap[1]*cp[1]*cp[2]*dp[1]*dp[2]*gp[2]- \\
 &18*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*gp[2]+9*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*gp[2]+16*ap[2]*bp[2]*cp[1]*dp[1]*ep[1]*gp[2]+16*ap[2]*bp[1]*cp[2]*dp[1]*ep[1]*gp[2]- \\
 &32*ap[1]*bp[2]*cp[2]*dp[1]*ep[1]*gp[2]- \\
 &16*ap[2]*bp[1]*cp[1]*dp[2]*ep[1]*gp[2]+8*ap[1]*bp[2]*cp[1]*dp[2]*ep[1]*gp[2]+8*ap[1]*bp[1]*cp[2]*dp[2]*ep[1]*gp[2]+80*ap[2]*bp[1]*bp[2]*ep[1]*ep[1]*gp[2]- \\
 &80*ap[1]*bp[2]*bp[2]*ep[1]*ep[1]*gp[2]- \\
 &64*ap[2]*cp[1]*cp[2]*ep[1]*ep[1]*gp[2]+64*ap[1]*cp[2]*cp[2]*ep[1]*ep[1]*gp[2]- \\
 &48*ap[2]*ap[2]*dp[1]*ep[1]*ep[1]*gp[2]+32*ap[1]*ap[2]*dp[2]*ep[1]*ep[1]*gp[2]- \\
 &16*ap[2]*bp[1]*cp[1]*dp[1]*ep[2]*gp[2]+8*ap[1]*bp[2]*cp[1]*dp[1]*ep[2]*gp[2]+8*ap[1]*bp[1]*cp[2]*dp[1]*ep[2]*gp[2]- \\
 &80*ap[2]*bp[1]*bp[1]*ep[1]*ep[2]*gp[2]+80*ap[1]*bp[1]*bp[2]*ep[1]*ep[2]*gp[2]+64*ap[2]*cp[1]
 \end{aligned}$$

$$\begin{aligned}
 & *cp[1]*ep[1]*ep[2]*gp[2]- \\
 & 64*ap[1]*cp[1]*cp[2]*ep[1]*ep[2]*gp[2]+64*ap[1]*ap[2]*dp[1]*ep[1]*ep[2]*gp[2]- \\
 & 32*ap[1]*ap[1]*dp[2]*ep[1]*ep[2]*gp[2]-16*ap[1]*ap[1]*dp[1]*ep[2]*ep[2]*gp[2]- \\
 & 10*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*gp[2]+10*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*gp[2]+40*ap[2]*cp[1] \\
 & *cp[2]*dp[1]*fp[1]*gp[2]-40*ap[1]*cp[2]*cp[2]*dp[1]*fp[1]*gp[2]- \\
 & 45*ap[2]*ap[2]*dp[1]*dp[1]*fp[1]*gp[2]+5*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*gp[2]- \\
 & 5*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*gp[2]- \\
 & 20*ap[2]*cp[1]*cp[1]*dp[2]*fp[1]*gp[2]+20*ap[1]*cp[1]*cp[2]*dp[2]*fp[1]*gp[2]+60*ap[1]*ap[2]* \\
 & dp[1]*dp[2]*fp[1]*gp[2]- \\
 & 15*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*gp[2]+144*ap[2]*bp[2]*cp[1]*ep[1]*fp[1]*gp[2]+144*ap[2]*bp[\\
 & 1]*cp[2]*ep[1]*fp[1]*gp[2]-288*ap[1]*bp[2]*cp[2]*ep[1]*fp[1]*gp[2]- \\
 & 432*ap[2]*ap[2]*ep[1]*ep[1]*fp[1]*gp[2]- \\
 & 144*ap[2]*bp[1]*cp[1]*ep[2]*fp[1]*gp[2]+72*ap[1]*bp[2]*cp[1]*ep[2]*fp[1]*gp[2]+72*ap[1]*bp[1] \\
 & *cp[2]*ep[2]*fp[1]*gp[2]+576*ap[1]*ap[2]*ep[1]*ep[2]*fp[1]*gp[2]- \\
 & 144*ap[1]*ap[1]*ep[2]*ep[2]*fp[1]*gp[2]- \\
 & 77*ap[2]*bp[1]*bp[2]*fp[1]*fp[1]*gp[2]+77*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*gp[2]+52*ap[2]*cp[1]* \\
 & cp[2]*fp[1]*fp[1]*gp[2]- \\
 & 52*ap[1]*cp[2]*cp[2]*fp[1]*fp[1]*gp[2]+75*ap[2]*ap[2]*dp[1]*fp[1]*fp[1]*gp[2]- \\
 & 50*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*gp[2]+129*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*gp[2]+5*ap[2]*bp[1]* \\
 & bp[1]*dp[1]*fp[2]*gp[2]-5*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*gp[2]- \\
 & 20*ap[2]*cp[1]*cp[1]*dp[1]*fp[2]*gp[2]+20*ap[1]*cp[1]*cp[2]*dp[1]*fp[2]*gp[2]+30*ap[1]*ap[2]* \\
 & dp[1]*dp[1]*fp[2]*gp[2]-30*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*gp[2]- \\
 & 144*ap[2]*bp[1]*cp[1]*ep[1]*fp[2]*gp[2]+72*ap[1]*bp[2]*cp[1]*ep[1]*fp[2]*gp[2]+72*ap[1]*bp[1] \\
 & *cp[2]*ep[1]*fp[2]*gp[2]+288*ap[1]*ap[2]*ep[1]*ep[1]*fp[2]*gp[2]- \\
 & 288*ap[1]*ap[1]*ep[1]*ep[2]*fp[2]*gp[2]+77*ap[2]*bp[1]*bp[1]*fp[1]*fp[2]*gp[2]- \\
 & 77*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*gp[2]- \\
 & 52*ap[2]*cp[1]*cp[1]*fp[1]*fp[2]*gp[2]+52*ap[1]*cp[1]*cp[2]*fp[1]*fp[2]*gp[2]- \\
 & 100*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*gp[2]+50*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*gp[2]- \\
 & 258*ap[1]*ap[2]*fp[1]*fp[1]*fp[2]*gp[2]+25*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*gp[2]+129*ap[1]*ap[1] \\
 & *fp[1]*fp[2]*fp[2]*gp[2]+2*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*gp[2]- \\
 & 2*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*gp[2]-8*ap[2]*bp[2]*cp[1]*cp[1]*gp[1]*gp[2]- \\
 & 16*ap[2]*bp[1]*cp[1]*cp[2]*gp[1]*gp[2]+16*ap[1]*bp[2]*cp[1]*cp[2]*gp[1]*gp[2]+8*ap[1]*bp[1]* \\
 & cp[2]*cp[2]*gp[1]*gp[2]+9*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*gp[2]- \\
 & 3*ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*gp[2]+3*ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*gp[2]- \\
 & 9*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*gp[2]+24*ap[2]*ap[2]*cp[1]*ep[1]*gp[1]*gp[2]- \\
 & 8*ap[1]*ap[2]*cp[2]*ep[1]*gp[1]*gp[2]+8*ap[1]*ap[2]*cp[1]*ep[2]*gp[1]*gp[2]- \\
 & 24*ap[1]*ap[1]*cp[2]*ep[2]*gp[1]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &15*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*gp[2]+5*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*gp[2]- \\
 &5*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*gp[2]+15*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*gp[2]- \\
 &9*ap[1]*ap[2]*ap[2]*gp[1]*gp[1]*gp[2]- \\
 &ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*gp[2]+ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*gp[2]+12*ap[2]*bp[1]*cp[1] \\
 &*cp[1]*gp[2]*gp[2]-4*ap[1]*bp[2]*cp[1]*cp[1]*gp[2]*gp[2]- \\
 &8*ap[1]*bp[1]*cp[1]*cp[2]*gp[2]*gp[2]- \\
 &9*ap[1]*ap[2]*bp[1]*dp[1]*gp[2]*gp[2]+6*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*gp[2]+3*ap[1]*ap[1]*bp \\
 &[1]*dp[2]*gp[2]*gp[2]- \\
 &24*ap[1]*ap[2]*cp[1]*ep[1]*gp[2]*gp[2]+16*ap[1]*ap[1]*cp[2]*ep[1]*gp[2]*gp[2]+8*ap[1]*ap[1]* \\
 &cp[1]*ep[2]*gp[2]*gp[2]+15*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*gp[2]- \\
 &10*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*gp[2]- \\
 &5*ap[1]*ap[1]*bp[1]*fp[2]*gp[2]*gp[2]+9*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*gp[2]- \\
 &3*ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*gp[2]+12*ap[2]*bp[2]*cp[1]*dp[1]*dp[2]*hp[1]+12*ap[2]*bp[1]* \\
 &cp[2]*dp[1]*dp[2]*hp[1]-24*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 &24*ap[2]*bp[1]*cp[1]*dp[2]*dp[2]*hp[1]+12*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*hp[1]+12*ap[1]*bp[1] \\
 &*cp[2]*dp[2]*dp[2]*hp[1]- \\
 &8*ap[2]*bp[1]*bp[2]*dp[2]*ep[1]*hp[1]+8*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*hp[1]+32*ap[2]*cp[1]*c \\
 &p[2]*dp[2]*ep[1]*hp[1]-32*ap[1]*cp[2]*cp[2]*dp[2]*ep[1]*hp[1]- \\
 &48*ap[2]*ap[2]*dp[1]*dp[2]*ep[1]*hp[1]+48*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*hp[1]- \\
 &8*ap[2]*bp[1]*bp[2]*dp[1]*ep[2]*hp[1]+8*ap[1]*bp[2]*bp[2]*dp[1]*ep[2]*hp[1]+32*ap[2]*cp[1]*c \\
 &p[2]*dp[1]*ep[2]*hp[1]-32*ap[1]*cp[2]*cp[2]*dp[1]*ep[2]*hp[1]- \\
 &24*ap[2]*ap[2]*dp[1]*dp[1]*ep[2]*hp[1]+16*ap[2]*bp[1]*bp[1]*dp[2]*ep[2]*hp[1]- \\
 &16*ap[1]*bp[1]*bp[2]*dp[2]*ep[2]*hp[1]- \\
 &64*ap[2]*cp[1]*cp[1]*dp[2]*ep[2]*hp[1]+64*ap[1]*cp[1]*cp[2]*dp[2]*ep[2]*hp[1]+96*ap[1]*ap[2] \\
 &*dp[1]*dp[2]*ep[2]*hp[1]- \\
 &72*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*hp[1]+64*ap[2]*bp[2]*cp[1]*ep[1]*ep[2]*hp[1]+64*ap[2]*bp[1] \\
 &*cp[2]*ep[1]*ep[2]*hp[1]-128*ap[1]*bp[2]*cp[2]*ep[1]*ep[2]*hp[1]- \\
 &128*ap[2]*ap[2]*ep[1]*ep[1]*ep[2]*hp[1]- \\
 &128*ap[2]*bp[1]*cp[1]*ep[2]*ep[2]*hp[1]+64*ap[1]*bp[2]*cp[1]*ep[2]*ep[2]*hp[1]+64*ap[1]*bp[1] \\
 &]*cp[2]*ep[2]*ep[2]*hp[1]+256*ap[1]*ap[2]*ep[1]*ep[2]*ep[2]*hp[1]- \\
 &128*ap[1]*ap[1]*ep[2]*ep[2]*ep[2]*hp[1]-20*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*hp[1]- \\
 &20*ap[2]*bp[1]*cp[2]*dp[2]*fp[1]*hp[1]+40*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*hp[1]+80*ap[2]*ap[2] \\
 &*dp[2]*ep[1]*fp[1]*hp[1]- \\
 &72*ap[2]*bp[1]*bp[2]*ep[2]*fp[1]*hp[1]+72*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*hp[1]+32*ap[2]*cp[1] \\
 &*cp[2]*ep[2]*fp[1]*hp[1]- \\
 &32*ap[1]*cp[2]*cp[2]*ep[2]*fp[1]*hp[1]+80*ap[2]*ap[2]*dp[1]*ep[2]*fp[1]*hp[1]- \\
 &160*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*hp[1]+104*ap[2]*ap[2]*ep[2]*fp[1]*fp[1]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 20*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*hp[1]- \\
 & 20*ap[2]*bp[1]*cp[2]*dp[1]*fp[2]*hp[1]+40*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*hp[1]+80*ap[2]*bp[1] \\
 & *cp[1]*dp[2]*fp[2]*hp[1]-40*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*hp[1]- \\
 & 40*ap[1]*bp[1]*cp[2]*dp[2]*fp[2]*hp[1]- \\
 & 72*ap[2]*bp[1]*bp[2]*ep[1]*fp[2]*hp[1]+72*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*hp[1]+32*ap[2]*cp[1] \\
 & *cp[2]*ep[1]*fp[2]*hp[1]- \\
 & 32*ap[1]*cp[2]*cp[2]*ep[1]*fp[2]*hp[1]+80*ap[2]*ap[2]*dp[1]*ep[1]*fp[2]*hp[1]- \\
 & 160*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*hp[1]+144*ap[2]*bp[1]*bp[1]*ep[2]*fp[2]*hp[1]- \\
 & 144*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*hp[1]- \\
 & 64*ap[2]*cp[1]*cp[1]*ep[2]*fp[2]*hp[1]+64*ap[1]*cp[1]*cp[2]*ep[2]*fp[2]*hp[1]- \\
 & 160*ap[1]*ap[2]*dp[1]*ep[2]*fp[2]*hp[1]+240*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*hp[1]- \\
 & 52*ap[2]*bp[2]*cp[1]*fp[1]*fp[2]*hp[1]- \\
 & 52*ap[2]*bp[1]*cp[2]*fp[1]*fp[2]*hp[1]+104*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*hp[1]+208*ap[2]*ap[2] \\
 &]*ep[1]*fp[1]*fp[2]*hp[1]- \\
 & 416*ap[1]*ap[2]*ep[2]*fp[1]*fp[2]*hp[1]+104*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*hp[1]- \\
 & 52*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*hp[1]-52*ap[1]*bp[1]*cp[2]*fp[2]*fp[2]*hp[1]- \\
 & 208*ap[1]*ap[2]*ep[1]*fp[2]*fp[2]*hp[1]+312*ap[1]*ap[1]*ep[2]*fp[2]*fp[2]*hp[1]+8*ap[2]*bp[2] \\
 & *bp[2]*cp[1]*gp[1]*hp[1]+16*ap[2]*bp[1]*bp[2]*cp[2]*gp[1]*hp[1]- \\
 & 24*ap[1]*bp[2]*bp[2]*cp[2]*gp[1]*hp[1]- \\
 & 32*ap[2]*cp[1]*cp[2]*cp[2]*gp[1]*hp[1]+32*ap[1]*cp[2]*cp[2]*cp[2]*gp[1]*hp[1]+24*ap[2]*ap[2] \\
 & *cp[2]*dp[1]*gp[1]*hp[1]+48*ap[2]*ap[2]*cp[1]*dp[2]*gp[1]*hp[1]- \\
 & 72*ap[1]*ap[2]*cp[2]*dp[2]*gp[1]*hp[1]-16*ap[2]*ap[2]*bp[2]*ep[1]*gp[1]*hp[1]- \\
 & 32*ap[2]*ap[2]*bp[1]*ep[2]*gp[1]*hp[1]+48*ap[1]*ap[2]*bp[2]*ep[2]*gp[1]*hp[1]- \\
 & 40*ap[2]*ap[2]*cp[2]*fp[1]*gp[1]*hp[1]- \\
 & 80*ap[2]*ap[2]*cp[1]*fp[2]*gp[1]*hp[1]+120*ap[1]*ap[2]*cp[2]*fp[2]*gp[1]*hp[1]- \\
 & 16*ap[2]*bp[1]*bp[2]*cp[1]*gp[2]*hp[1]+8*ap[1]*bp[2]*bp[2]*cp[1]*gp[2]*hp[1]- \\
 & 8*ap[2]*bp[1]*bp[1]*cp[2]*gp[2]*hp[1]+16*ap[1]*bp[1]*bp[2]*cp[2]*gp[2]*hp[1]+32*ap[2]*cp[1]* \\
 & cp[1]*cp[2]*gp[2]*hp[1]-32*ap[1]*cp[1]*cp[2]*cp[2]*gp[2]*hp[1]- \\
 & 36*ap[2]*ap[2]*cp[1]*dp[1]*gp[2]*hp[1]+12*ap[1]*ap[2]*cp[2]*dp[1]*gp[2]*hp[1]- \\
 & 12*ap[1]*ap[2]*cp[1]*dp[2]*gp[2]*hp[1]+36*ap[1]*ap[1]*cp[2]*dp[2]*gp[2]*hp[1]+24*ap[2]*ap[2] \\
 & *bp[1]*ep[1]*gp[2]*hp[1]- \\
 & 8*ap[1]*ap[2]*bp[2]*ep[1]*gp[2]*hp[1]+8*ap[1]*ap[2]*bp[1]*cp[2]*gp[2]*hp[1]- \\
 & 24*ap[1]*ap[1]*bp[2]*ep[2]*gp[2]*hp[1]+60*ap[2]*ap[2]*cp[1]*fp[1]*gp[2]*hp[1]- \\
 & 20*ap[1]*ap[2]*cp[2]*fp[1]*gp[2]*hp[1]+20*ap[1]*ap[2]*cp[1]*fp[2]*gp[2]*hp[1]- \\
 & 60*ap[1]*ap[1]*cp[2]*fp[2]*gp[2]*hp[1]+4*ap[2]*bp[1]*bp[2]*bp[2]*hp[1]*hp[1]- \\
 & 4*ap[1]*bp[2]*bp[2]*bp[2]*hp[1]*hp[1]-32*ap[2]*bp[2]*cp[1]*cp[2]*hp[1]*hp[1]- \\
 & 16*ap[2]*bp[1]*cp[2]*cp[2]*hp[1]*hp[1]+48*ap[1]*bp[2]*cp[2]*cp[2]*hp[1]*hp[1]+12*ap[2]*ap[2]
 \end{aligned}$$

$$\begin{aligned}
 & *bp[2]*dp[1]*hp[1]*hp[1]+24*ap[2]*ap[2]*bp[1]*dp[2]*hp[1]*hp[1]- \\
 & 36*ap[1]*ap[2]*bp[2]*dp[2]*hp[1]*hp[1]+32*ap[2]*ap[2]*cp[2]*ep[1]*hp[1]*hp[1]+64*ap[2]*ap[2] \\
 & *cp[1]*ep[2]*hp[1]*hp[1]-96*ap[1]*ap[2]*cp[2]*ep[2]*hp[1]*hp[1]- \\
 & 20*ap[2]*ap[2]*bp[2]*fp[1]*hp[1]*hp[1]- \\
 & 40*ap[2]*ap[2]*bp[1]*fp[2]*hp[1]*hp[1]+60*ap[1]*ap[2]*bp[2]*fp[2]*hp[1]*hp[1]- \\
 & 36*ap[2]*ap[2]*ap[2]*gp[1]*hp[1]*hp[1]+36*ap[1]*ap[2]*ap[2]*gp[2]*hp[1]*hp[1]- \\
 & 12*ap[2]*bp[2]*cp[1]*dp[1]*dp[1]*hp[2]- \\
 & 12*ap[2]*bp[1]*cp[2]*dp[1]*dp[1]*hp[2]+24*ap[1]*bp[2]*cp[2]*dp[1]*dp[1]*hp[2]+24*ap[2]*bp[1] \\
 & *cp[1]*dp[1]*dp[2]*hp[2]-12*ap[1]*bp[2]*cp[1]*dp[1]*dp[2]*hp[2]- \\
 & 12*ap[1]*bp[1]*cp[2]*dp[1]*dp[2]*hp[2]+16*ap[2]*bp[1]*bp[2]*dp[1]*ep[1]*hp[2]- \\
 & 16*ap[1]*bp[2]*bp[2]*dp[1]*ep[1]*hp[2]- \\
 & 64*ap[2]*cp[1]*cp[2]*dp[1]*ep[1]*hp[2]+64*ap[1]*cp[2]*cp[2]*dp[1]*ep[1]*hp[2]+72*ap[2]*ap[2] \\
 & *dp[1]*dp[1]*ep[1]*hp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*dp[2]*ep[1]*hp[2]+8*ap[1]*bp[1]*bp[2]*dp[2]*ep[1]*hp[2]+32*ap[2]*cp[1]*c \\
 & p[1]*dp[2]*ep[1]*hp[2]-32*ap[1]*cp[1]*cp[2]*dp[2]*ep[1]*hp[2]- \\
 & 96*ap[1]*ap[2]*dp[1]*dp[2]*ep[1]*hp[2]+24*ap[1]*ap[1]*dp[2]*dp[2]*ep[1]*hp[2]- \\
 & 64*ap[2]*bp[2]*cp[1]*ep[1]*ep[1]*hp[2]- \\
 & 64*ap[2]*bp[1]*cp[2]*ep[1]*ep[1]*hp[2]+128*ap[1]*bp[2]*cp[2]*ep[1]*ep[1]*hp[2]+128*ap[2]*ap[\\
 & 2]*ep[1]*ep[1]*ep[1]*hp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*dp[1]*ep[2]*hp[2]+8*ap[1]*bp[1]*bp[2]*dp[1]*ep[2]*hp[2]+32*ap[2]*cp[1]*c \\
 & p[1]*dp[1]*ep[2]*hp[2]-32*ap[1]*cp[1]*cp[2]*dp[1]*ep[2]*hp[2]- \\
 & 48*ap[1]*ap[2]*dp[1]*dp[1]*ep[2]*hp[2]+48*ap[1]*ap[1]*dp[1]*dp[2]*ep[2]*hp[2]+128*ap[2]*bp[1] \\
 &]*cp[1]*ep[1]*ep[2]*hp[2]-64*ap[1]*bp[2]*cp[1]*ep[1]*ep[2]*hp[2]- \\
 & 64*ap[1]*bp[1]*cp[2]*ep[1]*ep[2]*hp[2]- \\
 & 256*ap[1]*ap[2]*ep[1]*ep[1]*ep[2]*hp[2]+128*ap[1]*ap[1]*ep[1]*ep[2]*ep[2]*hp[2]+40*ap[2]*bp[\\
 & 2]*cp[1]*dp[1]*fp[1]*hp[2]+40*ap[2]*bp[1]*cp[2]*dp[1]*fp[1]*hp[2]- \\
 & 80*ap[1]*bp[2]*cp[2]*dp[1]*fp[1]*hp[2]- \\
 & 40*ap[2]*bp[1]*cp[1]*dp[2]*fp[1]*hp[2]+20*ap[1]*bp[2]*cp[1]*dp[2]*fp[1]*hp[2]+20*ap[1]*bp[1] \\
 & *cp[2]*dp[2]*fp[1]*hp[2]+144*ap[2]*bp[1]*bp[2]*ep[1]*fp[1]*hp[2]- \\
 & 144*ap[1]*bp[2]*bp[2]*ep[1]*fp[1]*hp[2]- \\
 & 64*ap[2]*cp[1]*cp[2]*ep[1]*fp[1]*hp[2]+64*ap[1]*cp[2]*cp[2]*ep[1]*fp[1]*hp[2]- \\
 & 240*ap[2]*ap[2]*dp[1]*ep[1]*fp[1]*hp[2]+160*ap[1]*ap[2]*dp[2]*ep[1]*fp[1]*hp[2]- \\
 & 72*ap[2]*bp[1]*bp[1]*ep[2]*fp[1]*hp[2]+72*ap[1]*bp[1]*bp[2]*ep[2]*fp[1]*hp[2]+32*ap[2]*cp[1] \\
 & *cp[1]*ep[2]*fp[1]*hp[2]- \\
 & 32*ap[1]*cp[1]*cp[2]*ep[2]*fp[1]*hp[2]+160*ap[1]*ap[2]*dp[1]*ep[2]*fp[1]*hp[2]- \\
 & 80*ap[1]*ap[1]*dp[2]*ep[2]*fp[1]*hp[2]+52*ap[2]*bp[2]*cp[1]*fp[1]*fp[1]*hp[2]+52*ap[2]*bp[1]* \\
 & cp[2]*fp[1]*fp[1]*hp[2]-104*ap[1]*bp[2]*cp[2]*fp[1]*fp[1]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 312*ap[2]*ap[2]*cp[1]*fp[1]*fp[1]*hp[2]+208*ap[1]*ap[2]*ep[2]*fp[1]*fp[1]*hp[2]- \\
 & 40*ap[2]*bp[1]*cp[1]*dp[1]*fp[2]*hp[2]+20*ap[1]*bp[2]*cp[1]*dp[1]*fp[2]*hp[2]+20*ap[1]*bp[1] \\
 & *cp[2]*dp[1]*fp[2]*hp[2]- \\
 & 72*ap[2]*bp[1]*bp[1]*ep[1]*fp[2]*hp[2]+72*ap[1]*bp[1]*bp[2]*ep[1]*fp[2]*hp[2]+32*ap[2]*cp[1] \\
 & *cp[1]*ep[1]*fp[2]*hp[2]- \\
 & 32*ap[1]*cp[1]*cp[2]*ep[1]*fp[2]*hp[2]+160*ap[1]*ap[2]*dp[1]*ep[1]*fp[2]*hp[2]- \\
 & 80*ap[1]*ap[1]*dp[2]*ep[1]*fp[2]*hp[2]-80*ap[1]*ap[1]*dp[1]*ep[2]*fp[2]*hp[2]- \\
 & 104*ap[2]*bp[1]*cp[1]*fp[1]*fp[2]*hp[2]+52*ap[1]*bp[2]*cp[1]*fp[1]*fp[2]*hp[2]+52*ap[1]*bp[1] \\
 & *cp[2]*fp[1]*fp[2]*hp[2]+416*ap[1]*ap[2]*ep[1]*fp[1]*fp[2]*hp[2]- \\
 & 208*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*hp[2]-104*ap[1]*ap[1]*ep[1]*fp[2]*fp[2]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*hp[2]+8*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*hp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*hp[2]+16*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*hp[2]+32*ap[2]*cp[1]* \\
 & cp[1]*cp[2]*gp[1]*hp[2]-32*ap[1]*cp[1]*cp[2]*cp[2]*gp[1]*hp[2]- \\
 & 36*ap[2]*ap[2]*cp[1]*dp[1]*gp[1]*hp[2]+12*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*hp[2]- \\
 & 12*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*hp[2]+36*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*hp[2]+24*ap[2]*ap[2] \\
 & *bp[1]*ep[1]*gp[1]*hp[2]- \\
 & 8*ap[1]*ap[2]*bp[2]*ep[1]*gp[1]*hp[2]+8*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*hp[2]- \\
 & 24*ap[1]*ap[1]*bp[2]*ep[2]*gp[1]*hp[2]+60*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*hp[2]- \\
 & 20*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*hp[2]+20*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*hp[2]- \\
 & 60*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*hp[2]+24*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*hp[2]- \\
 & 16*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*hp[2]-32*ap[2]*cp[1]*cp[1]*cp[1]*gp[2]*hp[2]- \\
 & 8*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*hp[2]+32*ap[1]*cp[1]*cp[1]*cp[2]*gp[2]*hp[2]+72*ap[1]*ap[2]* \\
 & cp[1]*dp[1]*gp[2]*hp[2]-48*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*hp[2]- \\
 & 24*ap[1]*ap[1]*cp[1]*dp[2]*gp[2]*hp[2]- \\
 & 48*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*hp[2]+32*ap[1]*ap[1]*bp[2]*ep[1]*gp[2]*hp[2]+16*ap[1]*ap[1] \\
 & *bp[1]*ep[2]*gp[2]*hp[2]- \\
 & 120*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*hp[2]+80*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*hp[2]+40*ap[1]*ap[1] \\
 & *cp[1]*fp[2]*gp[2]*hp[2]- \\
 & 8*ap[2]*bp[1]*bp[1]*bp[2]*hp[1]*hp[2]+8*ap[1]*bp[1]*bp[2]*bp[2]*hp[1]*hp[2]+32*ap[2]*bp[2]*c \\
 & p[1]*cp[1]*hp[1]*hp[2]+64*ap[2]*bp[1]*cp[1]*cp[2]*hp[1]*hp[2]- \\
 & 64*ap[1]*bp[2]*cp[1]*cp[2]*hp[1]*hp[2]-32*ap[1]*bp[1]*cp[2]*cp[2]*hp[1]*hp[2]- \\
 & 36*ap[2]*ap[2]*bp[1]*dp[1]*hp[1]*hp[2]+12*ap[1]*ap[2]*bp[2]*dp[1]*hp[1]*hp[2]- \\
 & 12*ap[1]*ap[2]*bp[1]*dp[2]*hp[1]*hp[2]+36*ap[1]*ap[1]*bp[2]*dp[2]*hp[1]*hp[2]- \\
 & 96*ap[2]*ap[2]*cp[1]*ep[1]*hp[1]*hp[2]+32*ap[1]*ap[2]*cp[2]*ep[1]*hp[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*cp[1]*ep[2]*hp[1]*hp[2]+96*ap[1]*ap[1]*cp[2]*ep[2]*hp[1]*hp[2]+60*ap[2]*ap[2] \\
 & *bp[1]*fp[1]*hp[1]*hp[2]- \\
 & 20*ap[1]*ap[2]*bp[2]*fp[1]*hp[1]*hp[2]+20*ap[1]*ap[2]*bp[1]*fp[2]*hp[1]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &60*ap[1]*ap[1]*bp[2]*fp[2]*hp[1]*hp[2]+72*ap[1]*ap[2]*ap[2]*gp[1]*hp[1]*hp[2]- \\
 &72*ap[1]*ap[1]*ap[2]*gp[2]*hp[1]*hp[2]+4*ap[2]*bp[1]*bp[1]*bp[1]*hp[2]*hp[2]- \\
 &4*ap[1]*bp[1]*bp[1]*bp[2]*hp[2]*hp[2]- \\
 &48*ap[2]*bp[1]*cp[1]*cp[1]*hp[2]*hp[2]+16*ap[1]*bp[2]*cp[1]*cp[1]*hp[2]*hp[2]+32*ap[1]*bp[1] \\
 &*cp[1]*cp[2]*hp[2]*hp[2]+36*ap[1]*ap[2]*bp[1]*dp[1]*hp[2]*hp[2]- \\
 &24*ap[1]*ap[1]*bp[2]*dp[1]*hp[2]*hp[2]- \\
 &12*ap[1]*ap[1]*bp[1]*dp[2]*hp[2]*hp[2]+96*ap[1]*ap[2]*cp[1]*cp[1]*hp[2]*hp[2]- \\
 &64*ap[1]*ap[1]*cp[2]*ep[1]*hp[2]*hp[2]-32*ap[1]*ap[1]*cp[1]*ep[2]*hp[2]*hp[2]- \\
 &60*ap[1]*ap[2]*bp[1]*fp[1]*hp[2]*hp[2]+40*ap[1]*ap[1]*bp[2]*fp[1]*hp[2]*hp[2]+20*ap[1]*ap[1] \\
 &*bp[1]*fp[2]*hp[2]*hp[2]- \\
 &36*ap[1]*ap[1]*ap[2]*gp[1]*hp[2]*hp[2]+36*ap[1]*ap[1]*ap[1]*gp[2]*hp[2]*hp[2]
 \end{aligned}$$

$$\begin{aligned}
 w_5 = &-2*ap[2]*bp[2]*cp[1]*dp[1]*dp[2]*gp[1]- \\
 &2*ap[2]*bp[1]*cp[2]*dp[1]*dp[2]*gp[1]+4*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*gp[1]+4*ap[2]*bp[1]*cp \\
 &[1]*dp[2]*dp[2]*gp[1]-2*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*gp[1]- \\
 &2*ap[1]*bp[1]*cp[2]*dp[2]*dp[2]*gp[1]- \\
 &4*ap[2]*bp[1]*bp[2]*dp[2]*ep[1]*gp[1]+4*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*gp[1]+8*ap[2]*ap[2]*dp \\
 &[1]*dp[2]*ep[1]*gp[1]-8*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*gp[1]- \\
 &4*ap[2]*bp[1]*bp[2]*dp[1]*ep[2]*gp[1]+4*ap[1]*bp[2]*bp[2]*dp[1]*ep[2]*gp[1]+4*ap[2]*ap[2]*dp \\
 &[1]*dp[1]*ep[2]*gp[1]+8*ap[2]*bp[1]*bp[1]*dp[2]*ep[2]*gp[1]- \\
 &8*ap[1]*bp[1]*bp[2]*dp[2]*ep[2]*gp[1]- \\
 &16*ap[1]*ap[2]*dp[1]*dp[2]*ep[2]*gp[1]+12*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*gp[1]- \\
 &128*ap[2]*bp[2]*cp[1]*ep[1]*ep[2]*gp[1]- \\
 &128*ap[2]*bp[1]*cp[2]*ep[1]*ep[2]*gp[1]+256*ap[1]*bp[2]*cp[2]*ep[1]*ep[2]*gp[1]+256*ap[2]*ap \\
 &[2]*ep[1]*ep[1]*ep[2]*gp[1]+256*ap[2]*bp[1]*cp[1]*ep[2]*ep[2]*gp[1]- \\
 &128*ap[1]*bp[2]*cp[1]*ep[2]*ep[2]*gp[1]-128*ap[1]*bp[1]*cp[2]*ep[2]*ep[2]*gp[1]- \\
 &512*ap[1]*ap[2]*ep[1]*ep[2]*ep[2]*gp[1]+256*ap[1]*ap[1]*ep[2]*ep[2]*ep[2]*gp[1]- \\
 &2*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*gp[1]- \\
 &2*ap[2]*bp[1]*cp[2]*dp[2]*fp[1]*gp[1]+4*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*gp[1]+8*ap[2]*ap[2]*dp[\\
 &2]*ep[1]*fp[1]*gp[1]+124*ap[2]*bp[1]*bp[2]*ep[2]*fp[1]*gp[1]- \\
 &124*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*gp[1]- \\
 &128*ap[2]*cp[1]*cp[2]*ep[2]*fp[1]*gp[1]+128*ap[1]*cp[2]*cp[2]*ep[2]*fp[1]*gp[1]+8*ap[2]*ap[2] \\
 &*dp[1]*ep[2]*fp[1]*gp[1]-16*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*gp[1]- \\
 &252*ap[2]*ap[2]*ep[2]*fp[1]*fp[1]*gp[1]-2*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*gp[1]- \\
 &2*ap[2]*bp[1]*cp[2]*dp[1]*fp[2]*gp[1]+4*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*gp[1]+8*ap[2]*bp[1]*cp[\\
 &1]*dp[2]*fp[2]*gp[1]-4*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*gp[1]- \\
 &4*ap[1]*bp[1]*cp[2]*dp[2]*fp[2]*gp[1]+124*ap[2]*bp[1]*bp[2]*ep[1]*fp[2]*gp[1]-
 \end{aligned}$$

$$\begin{aligned}
 &124*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*gp[1]- \\
 &128*ap[2]*cp[1]*cp[2]*ep[1]*fp[2]*gp[1]+128*ap[1]*cp[2]*cp[2]*ep[1]*fp[2]*gp[1]+8*ap[2]*ap[2] \\
 &*dp[1]*ep[1]*fp[2]*gp[1]-16*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*gp[1]- \\
 &248*ap[2]*bp[1]*bp[1]*ep[2]*fp[2]*gp[1]+248*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*gp[1]+256*ap[2]*cp \\
 &[1]*cp[1]*ep[2]*fp[2]*gp[1]-256*ap[1]*cp[1]*cp[2]*ep[2]*fp[2]*gp[1]- \\
 &16*ap[1]*ap[2]*dp[1]*ep[2]*fp[2]*gp[1]+24*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*gp[1]+126*ap[2]*bp[2] \\
 &*cp[1]*fp[1]*fp[2]*gp[1]+126*ap[2]*bp[1]*cp[2]*fp[1]*fp[2]*gp[1]- \\
 &252*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*gp[1]- \\
 &504*ap[2]*ap[2]*ep[1]*fp[1]*fp[2]*gp[1]+1008*ap[1]*ap[2]*ep[2]*fp[1]*fp[2]*gp[1]- \\
 &252*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*gp[1]+126*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*gp[1]+126*ap[1]*bp[\\
 &1]*cp[2]*fp[2]*fp[2]*gp[1]+504*ap[1]*ap[2]*ep[1]*fp[2]*fp[2]*gp[1]- \\
 &756*ap[1]*ap[1]*ep[2]*fp[2]*fp[2]*gp[1]+2*ap[2]*bp[2]*bp[2]*cp[1]*gp[1]*gp[1]+4*ap[2]*bp[1]* \\
 &bp[2]*cp[2]*gp[1]*gp[1]-6*ap[1]*bp[2]*bp[2]*cp[2]*gp[1]*gp[1]- \\
 &2*ap[2]*ap[2]*cp[2]*dp[1]*gp[1]*gp[1]- \\
 &4*ap[2]*ap[2]*cp[1]*dp[2]*gp[1]*gp[1]+6*ap[1]*ap[2]*cp[2]*dp[2]*gp[1]*gp[1]- \\
 &4*ap[2]*ap[2]*bp[2]*ep[1]*gp[1]*gp[1]- \\
 &8*ap[2]*ap[2]*bp[1]*ep[2]*gp[1]*gp[1]+12*ap[1]*ap[2]*bp[2]*ep[2]*gp[1]*gp[1]- \\
 &2*ap[2]*ap[2]*cp[2]*fp[1]*gp[1]*gp[1]- \\
 &4*ap[2]*ap[2]*cp[1]*fp[2]*gp[1]*gp[1]+6*ap[1]*ap[2]*cp[2]*fp[2]*gp[1]*gp[1]+2*ap[2]*bp[2]*cp[\\
 &1]*dp[1]*dp[1]*gp[2]+2*ap[2]*bp[1]*cp[2]*dp[1]*dp[1]*gp[2]- \\
 &4*ap[1]*bp[2]*cp[2]*dp[1]*dp[1]*gp[2]- \\
 &4*ap[2]*bp[1]*cp[1]*dp[1]*dp[2]*gp[2]+2*ap[1]*bp[2]*cp[1]*dp[1]*dp[2]*gp[2]+2*ap[1]*bp[1]*cp \\
 &[2]*dp[1]*dp[2]*gp[2]+8*ap[2]*bp[1]*bp[2]*dp[1]*ep[1]*gp[2]- \\
 &8*ap[1]*bp[2]*bp[2]*dp[1]*ep[1]*gp[2]-12*ap[2]*ap[2]*dp[1]*dp[1]*ep[1]*gp[2]- \\
 &4*ap[2]*bp[1]*bp[1]*dp[2]*ep[1]*gp[2]+4*ap[1]*bp[1]*bp[2]*dp[2]*ep[1]*gp[2]+16*ap[1]*ap[2]*d \\
 &p[1]*dp[2]*ep[1]*gp[2]- \\
 &4*ap[1]*ap[1]*dp[2]*dp[2]*ep[1]*gp[2]+128*ap[2]*bp[2]*cp[1]*ep[1]*ep[1]*gp[2]+128*ap[2]*bp[1] \\
 &]*cp[2]*ep[1]*ep[1]*gp[2]-256*ap[1]*bp[2]*cp[2]*ep[1]*ep[1]*gp[2]- \\
 &256*ap[2]*ap[2]*ep[1]*ep[1]*ep[1]*gp[2]- \\
 &4*ap[2]*bp[1]*bp[1]*dp[1]*ep[2]*gp[2]+4*ap[1]*bp[1]*bp[2]*dp[1]*ep[2]*gp[2]+8*ap[1]*ap[2]*dp \\
 &[1]*dp[1]*ep[2]*gp[2]-8*ap[1]*ap[1]*dp[1]*dp[2]*ep[2]*gp[2]- \\
 &256*ap[2]*bp[1]*cp[1]*ep[1]*ep[2]*gp[2]+128*ap[1]*bp[2]*cp[1]*ep[1]*ep[2]*gp[2]+128*ap[1]*b \\
 &p[1]*cp[2]*ep[1]*ep[2]*gp[2]+512*ap[1]*ap[2]*ep[1]*ep[1]*ep[2]*gp[2]- \\
 &256*ap[1]*ap[1]*ep[1]*ep[2]*ep[2]*gp[2]+4*ap[2]*bp[2]*cp[1]*dp[1]*fp[1]*gp[2]+4*ap[2]*bp[1]* \\
 &cp[2]*dp[1]*fp[1]*gp[2]-8*ap[1]*bp[2]*cp[2]*dp[1]*fp[1]*gp[2]- \\
 &4*ap[2]*bp[1]*cp[1]*dp[2]*fp[1]*gp[2]+2*ap[1]*bp[2]*cp[1]*dp[2]*fp[1]*gp[2]+2*ap[1]*bp[1]*cp[\\
 &2]*dp[2]*fp[1]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 248*ap[2]*bp[1]*bp[2]*cp[1]*fp[1]*gp[2]+248*ap[1]*bp[2]*bp[2]*cp[1]*fp[1]*gp[2]+256*ap[2]*cp[1]*cp[2]*ep[1]*fp[1]*gp[2]-256*ap[1]*cp[2]*cp[2]*ep[1]*fp[1]*gp[2]- \\
 & 24*ap[2]*ap[2]*dp[1]*ep[1]*fp[1]*gp[2]+16*ap[1]*ap[2]*dp[2]*ep[1]*fp[1]*gp[2]+124*ap[2]*bp[1]*bp[1]*ep[2]*fp[1]*gp[2]-124*ap[1]*bp[1]*bp[2]*ep[2]*fp[1]*gp[2]- \\
 & 128*ap[2]*cp[1]*cp[1]*ep[2]*fp[1]*gp[2]+128*ap[1]*cp[1]*cp[2]*ep[2]*fp[1]*gp[2]+16*ap[1]*ap[2]*dp[1]*ep[2]*fp[1]*gp[2]-8*ap[1]*ap[1]*dp[2]*ep[2]*fp[1]*gp[2]- \\
 & 126*ap[2]*bp[2]*cp[1]*fp[1]*fp[1]*gp[2]- \\
 & 126*ap[2]*bp[1]*cp[2]*fp[1]*fp[1]*gp[2]+252*ap[1]*bp[2]*cp[2]*fp[1]*fp[1]*gp[2]+756*ap[2]*ap[2]*ep[1]*fp[1]*fp[1]*gp[2]-504*ap[1]*ap[2]*ep[2]*fp[1]*fp[1]*gp[2]- \\
 & 4*ap[2]*bp[1]*cp[1]*dp[1]*fp[2]*gp[2]+2*ap[1]*bp[2]*cp[1]*dp[1]*fp[2]*gp[2]+2*ap[1]*bp[1]*cp[2]*dp[1]*fp[2]*gp[2]+124*ap[2]*bp[1]*bp[1]*ep[1]*fp[2]*gp[2]- \\
 & 124*ap[1]*bp[1]*bp[2]*ep[1]*fp[2]*gp[2]- \\
 & 128*ap[2]*cp[1]*cp[1]*ep[1]*fp[2]*gp[2]+128*ap[1]*cp[1]*cp[2]*ep[1]*fp[2]*gp[2]+16*ap[1]*ap[2]*dp[1]*cp[1]*fp[2]*gp[2]-8*ap[1]*ap[1]*dp[2]*ep[1]*fp[2]*gp[2]- \\
 & 8*ap[1]*ap[1]*dp[1]*ep[2]*fp[2]*gp[2]+252*ap[2]*bp[1]*cp[1]*fp[1]*fp[2]*gp[2]- \\
 & 126*ap[1]*bp[2]*cp[1]*fp[1]*fp[2]*gp[2]-126*ap[1]*bp[1]*cp[2]*fp[1]*fp[2]*gp[2]- \\
 & 1008*ap[1]*ap[2]*ep[1]*fp[1]*fp[2]*gp[2]+504*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*gp[2]+252*ap[1]*ap[1]*ep[1]*fp[2]*fp[2]*gp[2]- \\
 & 8*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*gp[2]+4*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*gp[2]- \\
 & 4*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*gp[2]+8*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*gp[2]+6*ap[2]*ap[2]*cp[1]*dp[1]*gp[1]*gp[2]- \\
 & 2*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*gp[2]+2*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*gp[2]- \\
 & 6*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*gp[2]+12*ap[2]*ap[2]*bp[1]*ep[1]*gp[1]*gp[2]- \\
 & 4*ap[1]*ap[2]*bp[2]*ep[1]*gp[1]*gp[2]+4*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*gp[2]- \\
 & 12*ap[1]*ap[1]*bp[2]*ep[2]*gp[1]*gp[2]+6*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*gp[2]- \\
 & 2*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*gp[2]+2*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*gp[2]- \\
 & 6*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*gp[2]+6*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*gp[2]- \\
 & 4*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*gp[2]-2*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*gp[2]- \\
 & 6*ap[1]*ap[2]*cp[1]*dp[1]*gp[2]*gp[2]+4*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*gp[2]+2*ap[1]*ap[1]*cp[1]*dp[2]*gp[2]*gp[2]- \\
 & 12*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*gp[2]+8*ap[1]*ap[1]*bp[2]*cp[1]*gp[2]*gp[2]+4*ap[1]*ap[1]*bp[1]*ep[2]*gp[2]*gp[2]- \\
 & 6*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*gp[2]+4*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*gp[2]+2*ap[1]*ap[1]*cp[1]*fp[2]*gp[2]*gp[2]- \\
 & 2*ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*hp[1]+2*ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*hp[1]+32*ap[2]*cp[1]*cp[2]*dp[1]*dp[2]*hp[1]-32*ap[1]*cp[2]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 & 30*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*hp[1]+2*ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 2*ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*hp[1]- \\
 & 32*ap[2]*cp[1]*cp[1]*dp[2]*dp[2]*hp[1]+32*ap[1]*cp[1]*cp[2]*dp[2]*dp[2]*hp[1]+60*ap[1]*ap[2] \\
 & *dp[1]*dp[2]*dp[2]*hp[1]-30*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*hp[1]- \\
 & 128*ap[2]*bp[1]*bp[2]*ep[1]*ep[2]*hp[1]+128*ap[1]*bp[2]*bp[2]*ep[1]*ep[2]*hp[1]+128*ap[2]*c \\
 & p[1]*cp[2]*ep[1]*ep[2]*hp[1]- \\
 & 128*ap[1]*cp[2]*cp[2]*ep[1]*ep[2]*hp[1]+128*ap[2]*bp[1]*bp[1]*ep[2]*ep[2]*hp[1]- \\
 & 128*ap[1]*bp[1]*bp[2]*ep[2]*ep[2]*hp[1]- \\
 & 128*ap[2]*cp[1]*cp[1]*ep[2]*ep[2]*hp[1]+128*ap[1]*cp[1]*cp[2]*ep[2]*ep[2]*hp[1]- \\
 & 2*ap[2]*bp[1]*bp[2]*dp[2]*fp[1]*hp[1]+2*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*hp[1]- \\
 & 32*ap[2]*cp[1]*cp[2]*dp[2]*fp[1]*hp[1]+32*ap[1]*cp[2]*cp[2]*dp[2]*fp[1]*hp[1]+68*ap[2]*ap[2]* \\
 & dp[1]*dp[2]*fp[1]*hp[1]-68*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*hp[1]- \\
 & 128*ap[2]*bp[2]*cp[1]*ep[2]*fp[1]*hp[1]- \\
 & 128*ap[2]*bp[1]*cp[2]*ep[2]*fp[1]*hp[1]+256*ap[1]*bp[2]*cp[2]*ep[2]*fp[1]*hp[1]+512*ap[2]*ap \\
 & [2]*ep[1]*ep[2]*fp[1]*hp[1]-512*ap[1]*ap[2]*ep[2]*ep[2]*fp[1]*hp[1]- \\
 & 30*ap[2]*ap[2]*dp[2]*fp[1]*fp[1]*hp[1]- \\
 & 2*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*hp[1]+2*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*hp[1]- \\
 & 32*ap[2]*cp[1]*cp[2]*dp[1]*fp[2]*hp[1]+32*ap[1]*cp[2]*cp[2]*dp[1]*fp[2]*hp[1]+34*ap[2]*ap[2]* \\
 & dp[1]*dp[1]*fp[2]*hp[1]+4*ap[2]*bp[1]*bp[1]*dp[2]*fp[2]*hp[1]- \\
 & 4*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*hp[1]+64*ap[2]*cp[1]*cp[1]*dp[2]*fp[2]*hp[1]- \\
 & 64*ap[1]*cp[1]*cp[2]*dp[2]*fp[2]*hp[1]- \\
 & 136*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*hp[1]+102*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*hp[1]- \\
 & 128*ap[2]*bp[2]*cp[1]*ep[1]*fp[2]*hp[1]- \\
 & 128*ap[2]*bp[1]*cp[2]*ep[1]*fp[2]*hp[1]+256*ap[1]*bp[2]*cp[2]*ep[1]*fp[2]*hp[1]+256*ap[2]*ap \\
 & [2]*ep[1]*ep[1]*fp[2]*hp[1]+512*ap[2]*bp[1]*cp[1]*cp[2]*fp[2]*hp[1]- \\
 & 256*ap[1]*bp[2]*cp[1]*ep[2]*fp[2]*hp[1]-256*ap[1]*bp[1]*cp[2]*ep[2]*fp[2]*hp[1]- \\
 & 1024*ap[1]*ap[2]*ep[1]*ep[2]*fp[2]*hp[1]+768*ap[1]*ap[1]*ep[2]*ep[2]*fp[2]*hp[1]+126*ap[2]*b \\
 & p[1]*bp[2]*fp[1]*fp[2]*hp[1]-126*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*hp[1]- \\
 & 96*ap[2]*cp[1]*cp[2]*fp[1]*fp[2]*hp[1]+96*ap[1]*cp[2]*cp[2]*fp[1]*fp[2]*hp[1]- \\
 & 60*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*hp[1]+120*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*hp[1]- \\
 & 222*ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*hp[1]- \\
 & 126*ap[2]*bp[1]*bp[1]*fp[2]*fp[2]*hp[1]+126*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*hp[1]+96*ap[2]*cp[1] \\
 &]*cp[1]*fp[2]*fp[2]*hp[1]- \\
 & 96*ap[1]*cp[1]*cp[2]*fp[2]*fp[2]*hp[1]+60*ap[1]*ap[2]*dp[1]*fp[2]*fp[2]*hp[1]- \\
 & 90*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*hp[1]+444*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*hp[1]- \\
 & 222*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*hp[1]+4*ap[2]*bp[1]*bp[2]*bp[2]*gp[1]*hp[1]- \\
 & 4*ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*hp[1]-4*ap[2]*ap[2]*bp[2]*dp[1]*gp[1]*hp[1]- \\
 & 8*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*hp[1]+12*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 4*ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*hp[1]- \\
 & 8*ap[2]*ap[2]*bp[1]*fp[2]*gp[1]*hp[1]+12*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*hp[1]+6*ap[2]*ap[2]*ap \\
 & [2]*gp[1]*gp[1]*hp[1]- \\
 & 4*ap[2]*bp[1]*bp[1]*bp[2]*gp[2]*hp[1]+4*ap[1]*bp[1]*bp[2]*bp[2]*gp[2]*hp[1]+6*ap[2]*ap[2]*bp \\
 & [1]*dp[1]*gp[2]*hp[1]- \\
 & 2*ap[1]*ap[2]*bp[2]*dp[1]*gp[2]*hp[1]+2*ap[1]*ap[2]*bp[1]*dp[2]*gp[2]*hp[1]- \\
 & 6*ap[1]*ap[1]*bp[2]*dp[2]*gp[2]*hp[1]+6*ap[2]*ap[2]*bp[1]*fp[1]*gp[2]*hp[1]- \\
 & 2*ap[1]*ap[2]*bp[2]*fp[1]*gp[2]*hp[1]+2*ap[1]*ap[2]*bp[1]*fp[2]*gp[2]*hp[1]- \\
 & 6*ap[1]*ap[1]*bp[2]*fp[2]*gp[2]*hp[1]- \\
 & 12*ap[1]*ap[2]*ap[2]*gp[1]*gp[2]*hp[1]+6*ap[1]*ap[1]*ap[2]*gp[2]*gp[2]*hp[1]- \\
 & 32*ap[2]*cp[1]*cp[2]*cp[2]*hp[1]*hp[1]+32*ap[1]*cp[2]*cp[2]*cp[2]*hp[1]*hp[1]+32*ap[2]*ap[2] \\
 & *cp[2]*dp[1]*hp[1]*hp[1]+64*ap[2]*ap[2]*cp[1]*dp[2]*hp[1]*hp[1]- \\
 & 96*ap[1]*ap[2]*cp[2]*dp[2]*hp[1]*hp[1]-32*ap[2]*ap[2]*cp[2]*fp[1]*hp[1]*hp[1]- \\
 & 64*ap[2]*ap[2]*cp[1]*fp[2]*hp[1]*hp[1]+96*ap[1]*ap[2]*cp[2]*fp[2]*hp[1]*hp[1]- \\
 & 32*ap[2]*ap[2]*ap[2]*hp[1]*hp[1]*hp[1]+2*ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*hp[2]- \\
 & 2*ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*hp[2]- \\
 & 32*ap[2]*cp[1]*cp[2]*dp[1]*dp[1]*hp[2]+32*ap[1]*cp[2]*cp[2]*dp[1]*dp[1]*hp[2]+30*ap[2]*ap[2] \\
 & *dp[1]*dp[1]*dp[1]*hp[2]- \\
 & 2*ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*hp[2]+2*ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*hp[2]+32*ap[2]*cp[1]*c \\
 & p[1]*dp[1]*dp[2]*hp[2]-32*ap[1]*cp[1]*cp[2]*dp[1]*dp[2]*hp[2]- \\
 & 60*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*hp[2]+30*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*hp[2]+128*ap[2]*bp[\\
 & 1]*bp[2]*ep[1]*ep[1]*hp[2]-128*ap[1]*bp[2]*bp[2]*ep[1]*ep[1]*hp[2]- \\
 & 128*ap[2]*cp[1]*cp[2]*ep[1]*ep[1]*hp[2]+128*ap[1]*cp[2]*cp[2]*ep[1]*ep[1]*hp[2]- \\
 & 128*ap[2]*bp[1]*bp[1]*ep[1]*ep[2]*hp[2]+128*ap[1]*bp[1]*bp[2]*ep[1]*ep[2]*hp[2]+128*ap[2]*c \\
 & p[1]*cp[1]*ep[1]*ep[2]*hp[2]- \\
 & 128*ap[1]*cp[1]*cp[2]*ep[1]*ep[2]*hp[2]+4*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*hp[2]- \\
 & 4*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*hp[2]+64*ap[2]*cp[1]*cp[2]*dp[1]*fp[1]*hp[2]- \\
 & 64*ap[1]*cp[2]*cp[2]*dp[1]*fp[1]*hp[2]-102*ap[2]*ap[2]*dp[1]*dp[1]*fp[1]*hp[2]- \\
 & 2*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*hp[2]+2*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*hp[2]- \\
 & 32*ap[2]*cp[1]*cp[1]*dp[2]*fp[1]*hp[2]+32*ap[1]*cp[1]*cp[2]*dp[2]*fp[1]*hp[2]+136*ap[1]*ap[2] \\
 & *dp[1]*dp[2]*fp[1]*hp[2]- \\
 & 34*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*hp[2]+256*ap[2]*bp[2]*cp[1]*ep[1]*fp[1]*hp[2]+256*ap[2]*bp[\\
 & 1]*cp[2]*ep[1]*fp[1]*hp[2]-512*ap[1]*bp[2]*cp[2]*ep[1]*fp[1]*hp[2]- \\
 & 768*ap[2]*ap[2]*ep[1]*ep[1]*fp[1]*hp[2]- \\
 & 256*ap[2]*bp[1]*cp[1]*ep[2]*fp[1]*hp[2]+128*ap[1]*bp[2]*cp[1]*ep[2]*fp[1]*hp[2]+128*ap[1]*bp \\
 & [1]*cp[2]*ep[2]*fp[1]*hp[2]+1024*ap[1]*ap[2]*cp[1]*ep[2]*fp[1]*hp[2]- \\
 & 256*ap[1]*ap[1]*ep[2]*ep[2]*fp[1]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &126*ap[2]*bp[1]*bp[2]*fp[1]*fp[1]*hp[2]+126*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*hp[2]+96*ap[2]*cp[1] \\
 &]*cp[2]*fp[1]*fp[1]*hp[2]- \\
 &96*ap[1]*cp[2]*cp[2]*fp[1]*fp[1]*hp[2]+90*ap[2]*ap[2]*dp[1]*fp[1]*fp[1]*hp[2]- \\
 &60*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*hp[2]+222*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*hp[2]- \\
 &2*ap[2]*bp[1]*bp[1]*dp[1]*fp[2]*hp[2]+2*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*hp[2]- \\
 &32*ap[2]*cp[1]*cp[1]*dp[1]*fp[2]*hp[2]+32*ap[1]*cp[1]*cp[2]*dp[1]*fp[2]*hp[2]+68*ap[1]*ap[2]* \\
 &dp[1]*dp[1]*fp[2]*hp[2]-68*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*hp[2]- \\
 &256*ap[2]*bp[1]*cp[1]*ep[1]*fp[2]*hp[2]+128*ap[1]*bp[2]*cp[1]*ep[1]*fp[2]*hp[2]+128*ap[1]*bp \\
 &[1]*cp[2]*ep[1]*fp[2]*hp[2]+512*ap[1]*ap[2]*ep[1]*ep[1]*fp[2]*hp[2]- \\
 &512*ap[1]*ap[1]*ep[1]*ep[2]*fp[2]*hp[2]+126*ap[2]*bp[1]*bp[1]*fp[1]*fp[2]*hp[2]- \\
 &126*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*hp[2]- \\
 &96*ap[2]*cp[1]*cp[1]*fp[1]*fp[2]*hp[2]+96*ap[1]*cp[1]*cp[2]*fp[1]*fp[2]*hp[2]- \\
 &120*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*hp[2]+60*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*hp[2]- \\
 &444*ap[1]*ap[2]*fp[1]*fp[1]*fp[2]*hp[2]+30*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*hp[2]+222*ap[1]*ap[1] \\
 &]*fp[1]*fp[2]*fp[2]*hp[2]- \\
 &4*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*hp[2]+4*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*hp[2]+6*ap[2]*ap[2]*bp \\
 &[1]*dp[1]*gp[1]*hp[2]- \\
 &2*ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*hp[2]+2*ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*hp[2]- \\
 &6*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*hp[2]+6*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*hp[2]- \\
 &2*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*hp[2]+2*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*hp[2]- \\
 &6*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*hp[2]- \\
 &6*ap[1]*ap[2]*ap[2]*gp[1]*gp[1]*hp[2]+4*ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*hp[2]- \\
 &4*ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*hp[2]- \\
 &12*ap[1]*ap[2]*bp[1]*dp[1]*gp[2]*hp[2]+8*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*hp[2]+4*ap[1]*ap[1]*b \\
 &p[1]*dp[2]*gp[2]*hp[2]- \\
 &12*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*hp[2]+8*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*hp[2]+4*ap[1]*ap[1]*b \\
 &p[1]*fp[2]*gp[2]*hp[2]+12*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*hp[2]- \\
 &6*ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*hp[2]+64*ap[2]*cp[1]*cp[1]*cp[2]*hp[1]*hp[2]- \\
 &64*ap[1]*cp[1]*cp[2]*cp[2]*hp[1]*hp[2]- \\
 &96*ap[2]*ap[2]*cp[1]*dp[1]*hp[1]*hp[2]+32*ap[1]*ap[2]*cp[2]*dp[1]*hp[1]*hp[2]- \\
 &32*ap[1]*ap[2]*cp[1]*dp[2]*hp[1]*hp[2]+96*ap[1]*ap[1]*cp[2]*dp[2]*hp[1]*hp[2]+96*ap[2]*ap[2] \\
 &)*cp[1]*fp[1]*hp[1]*hp[2]- \\
 &32*ap[1]*ap[2]*cp[2]*fp[1]*hp[1]*hp[2]+32*ap[1]*ap[2]*cp[1]*fp[2]*hp[1]*hp[2]- \\
 &96*ap[1]*ap[1]*cp[2]*fp[2]*hp[1]*hp[2]+96*ap[1]*ap[2]*ap[2]*hp[1]*hp[1]*hp[2]- \\
 &32*ap[2]*cp[1]*cp[1]*cp[1]*hp[2]*hp[2]+32*ap[1]*cp[1]*cp[1]*cp[2]*hp[2]*hp[2]+96*ap[1]*ap[2] \\
 &)*cp[1]*dp[1]*hp[2]*hp[2]-64*ap[1]*ap[1]*cp[2]*dp[1]*hp[2]*hp[2]- \\
 &32*ap[1]*ap[1]*cp[1]*dp[2]*hp[2]*hp[2]-
 \end{aligned}$$

$$96*ap[1]*ap[2]*cp[1]*fp[1]*hp[2]*hp[2]+64*ap[1]*ap[1]*cp[2]*fp[1]*hp[2]*hp[2]+32*ap[1]*ap[1]*cp[1]*fp[2]*hp[2]*hp[2]-$$

$$96*ap[1]*ap[1]*ap[2]*hp[1]*hp[2]*hp[2]+32*ap[1]*ap[1]*ap[1]*hp[2]*hp[2]*hp[2]$$

$w_6 = -$

$$3*ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*gp[1]+3*ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*gp[1]+8*ap[2]*cp[1]*cp[2]*dp[1]*dp[2]*gp[1]-8*ap[1]*cp[2]*cp[2]*dp[1]*dp[2]*gp[1]-$$

$$5*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*gp[1]+3*ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*gp[1]-$$

$$3*ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*gp[1]-$$

$$8*ap[2]*cp[1]*cp[1]*dp[2]*dp[2]*gp[1]+8*ap[1]*cp[1]*cp[2]*dp[2]*dp[2]*gp[1]+10*ap[1]*ap[2]*dp[1]*dp[2]*dp[2]*gp[1]-5*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*gp[1]-$$

$$16*ap[2]*bp[2]*cp[1]*dp[2]*ep[1]*gp[1]-$$

$$16*ap[2]*bp[1]*cp[2]*dp[2]*ep[1]*gp[1]+32*ap[1]*bp[2]*cp[2]*dp[2]*ep[1]*gp[1]+32*ap[2]*ap[2]*dp[2]*ep[1]*ep[1]*gp[1]-16*ap[2]*bp[2]*cp[1]*dp[1]*ep[2]*gp[1]-$$

$$16*ap[2]*bp[1]*cp[2]*dp[1]*ep[2]*gp[1]+32*ap[1]*bp[2]*cp[2]*dp[1]*ep[2]*gp[1]+64*ap[2]*bp[1]*cp[1]*dp[2]*ep[2]*gp[1]-32*ap[1]*bp[2]*cp[1]*dp[2]*ep[2]*gp[1]-$$

$$32*ap[1]*bp[1]*cp[2]*dp[2]*ep[2]*gp[1]+160*ap[2]*bp[1]*bp[2]*ep[1]*ep[2]*gp[1]-$$

$$160*ap[1]*bp[2]*bp[2]*ep[1]*ep[2]*gp[1]-$$

$$192*ap[2]*cp[1]*cp[2]*ep[1]*ep[2]*gp[1]+192*ap[1]*cp[2]*cp[2]*ep[1]*ep[2]*gp[1]+64*ap[2]*ap[2]*dp[1]*ep[1]*ep[2]*gp[1]-128*ap[1]*ap[2]*dp[2]*ep[1]*ep[2]*gp[1]-$$

$$160*ap[2]*bp[1]*bp[1]*ep[2]*ep[2]*gp[1]+160*ap[1]*bp[1]*bp[2]*ep[2]*ep[2]*gp[1]+192*ap[2]*cp[1]*cp[1]*ep[2]*ep[2]*gp[1]-192*ap[1]*cp[1]*cp[2]*ep[2]*ep[2]*gp[1]-$$

$$64*ap[1]*ap[2]*dp[1]*ep[2]*ep[2]*gp[1]+96*ap[1]*ap[1]*dp[2]*ep[2]*ep[2]*gp[1]+13*ap[2]*bp[1]*bp[2]*dp[2]*fp[1]*gp[1]-13*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*gp[1]-$$

$$24*ap[2]*cp[1]*cp[2]*dp[2]*fp[1]*gp[1]+24*ap[1]*cp[2]*cp[2]*dp[2]*fp[1]*gp[1]+22*ap[2]*ap[2]*dp[1]*dp[2]*fp[1]*gp[1]-$$

$$22*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*gp[1]+176*ap[2]*bp[2]*cp[1]*ep[2]*fp[1]*gp[1]+176*ap[2]*bp[1]*cp[2]*ep[2]*fp[1]*gp[1]-352*ap[1]*bp[2]*cp[2]*ep[2]*fp[1]*gp[1]-$$

$$704*ap[2]*ap[2]*ep[1]*ep[2]*fp[1]*gp[1]+704*ap[1]*ap[2]*ep[2]*ep[2]*fp[1]*gp[1]-$$

$$37*ap[2]*ap[2]*dp[2]*fp[1]*fp[1]*gp[1]+13*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*gp[1]-$$

$$13*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*gp[1]-$$

$$24*ap[2]*cp[1]*cp[2]*dp[1]*fp[2]*gp[1]+24*ap[1]*cp[2]*cp[2]*dp[1]*fp[2]*gp[1]+11*ap[2]*ap[2]*dp[1]*dp[1]*fp[2]*gp[1]-$$

$$26*ap[2]*bp[1]*bp[1]*dp[2]*fp[2]*gp[1]+26*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*gp[1]+48*ap[2]*cp[1]*cp[1]*dp[2]*fp[2]*gp[1]-48*ap[1]*cp[1]*cp[2]*dp[2]*fp[2]*gp[1]-$$

$$44*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*gp[1]+33*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*gp[1]+176*ap[2]*bp[2]*cp[1]*ep[1]*fp[2]*gp[1]+176*ap[2]*bp[1]*cp[2]*ep[1]*fp[2]*gp[1]-$$

$$\begin{aligned}
 & 352*ap[1]*bp[2]*cp[2]*ep[1]*fp[2]*gp[1]-352*ap[2]*ap[2]*ep[1]*ep[1]*fp[2]*gp[1]- \\
 & 704*ap[2]*bp[1]*cp[1]*ep[2]*fp[2]*gp[1]+352*ap[1]*bp[2]*cp[1]*ep[2]*fp[2]*gp[1]+352*ap[1]*bp \\
 & [1]*cp[2]*ep[2]*fp[2]*gp[1]+1408*ap[1]*ap[2]*ep[1]*ep[2]*fp[2]*gp[1]- \\
 & 1056*ap[1]*ap[1]*ep[2]*ep[2]*fp[2]*gp[1]- \\
 & 163*ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*gp[1]+163*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*gp[1]+200*ap[2]*cp[\\
 & 1]*cp[2]*fp[1]*fp[2]*gp[1]-200*ap[1]*cp[2]*cp[2]*fp[1]*fp[2]*gp[1]- \\
 & 74*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*gp[1]+148*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*gp[1]+363*ap[2]*ap[2] \\
 &]*fp[1]*fp[1]*fp[2]*gp[1]+163*ap[2]*bp[1]*bp[1]*fp[2]*fp[2]*gp[1]- \\
 & 163*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*gp[1]- \\
 & 200*ap[2]*cp[1]*cp[1]*fp[2]*fp[2]*gp[1]+200*ap[1]*cp[1]*cp[2]*fp[2]*fp[2]*gp[1]+74*ap[1]*ap[2] \\
 &]*dp[1]*fp[2]*fp[2]*gp[1]-111*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*gp[1]- \\
 & 726*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*gp[1]+363*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*gp[1]- \\
 & 5*ap[2]*bp[1]*bp[2]*bp[2]*gp[1]*gp[1]+5*ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*gp[1]+16*ap[2]*bp[2]*c \\
 & p[1]*cp[2]*gp[1]*gp[1]+8*ap[2]*bp[1]*cp[2]*cp[2]*gp[1]*gp[1]- \\
 & 24*ap[1]*bp[2]*cp[2]*cp[2]*gp[1]*gp[1]-3*ap[2]*ap[2]*bp[2]*dp[1]*gp[1]*gp[1]- \\
 & 6*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*gp[1]+9*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*gp[1]- \\
 & 16*ap[2]*ap[2]*cp[2]*ep[1]*gp[1]*gp[1]- \\
 & 32*ap[2]*ap[2]*cp[1]*ep[2]*gp[1]*gp[1]+48*ap[1]*ap[2]*cp[2]*ep[2]*gp[1]*gp[1]+13*ap[2]*ap[2] \\
 & *bp[2]*fp[1]*gp[1]*gp[1]+26*ap[2]*ap[2]*bp[1]*fp[2]*gp[1]*gp[1]- \\
 & 39*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*gp[1]+3*ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*gp[1]+3*ap[2]*bp[1]*b \\
 & p[2]*dp[1]*dp[1]*gp[2]-3*ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*gp[2]- \\
 & 8*ap[2]*cp[1]*cp[2]*dp[1]*dp[1]*gp[2]+8*ap[1]*cp[2]*cp[2]*dp[1]*dp[1]*gp[2]+5*ap[2]*ap[2]*dp \\
 & [1]*dp[1]*dp[1]*gp[2]- \\
 & 3*ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*gp[2]+3*ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*gp[2]+8*ap[2]*cp[1]*cp \\
 & [1]*dp[1]*dp[2]*gp[2]-8*ap[1]*cp[1]*cp[2]*dp[1]*dp[2]*gp[2]- \\
 & 10*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*gp[2]+5*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*gp[2]+32*ap[2]*bp[2]* \\
 & cp[1]*dp[1]*ep[1]*gp[2]+32*ap[2]*bp[1]*cp[2]*dp[1]*ep[1]*gp[2]- \\
 & 64*ap[1]*bp[2]*cp[2]*dp[1]*ep[1]*gp[2]- \\
 & 32*ap[2]*bp[1]*cp[1]*dp[2]*ep[1]*gp[2]+16*ap[1]*bp[2]*cp[1]*dp[2]*ep[1]*gp[2]+16*ap[1]*bp[1] \\
 & *cp[2]*dp[2]*ep[1]*gp[2]- \\
 & 160*ap[2]*bp[1]*bp[2]*ep[1]*ep[1]*gp[2]+160*ap[1]*bp[2]*bp[2]*ep[1]*ep[1]*gp[2]+192*ap[2]*c \\
 & p[1]*cp[2]*ep[1]*cp[1]*gp[2]-192*ap[1]*cp[2]*cp[2]*ep[1]*ep[1]*gp[2]- \\
 & 96*ap[2]*ap[2]*dp[1]*ep[1]*ep[1]*gp[2]+64*ap[1]*ap[2]*dp[2]*ep[1]*ep[1]*gp[2]- \\
 & 32*ap[2]*bp[1]*cp[1]*dp[1]*ep[2]*gp[2]+16*ap[1]*bp[2]*cp[1]*dp[1]*ep[2]*gp[2]+16*ap[1]*bp[1] \\
 & *cp[2]*dp[1]*ep[2]*gp[2]+160*ap[2]*bp[1]*bp[1]*ep[1]*ep[2]*gp[2]- \\
 & 160*ap[1]*bp[1]*bp[2]*ep[1]*ep[2]*gp[2]- \\
 & 192*ap[2]*cp[1]*cp[1]*ep[1]*ep[2]*gp[2]+192*ap[1]*cp[1]*cp[2]*ep[1]*ep[2]*gp[2]+128*ap[1]*ap
 \end{aligned}$$

$$\begin{aligned}
 & [2]*dp[1]*ep[1]*ep[2]*gp[2]-64*ap[1]*ap[1]*dp[2]*ep[1]*ep[2]*gp[2]- \\
 & 32*ap[1]*ap[1]*dp[1]*ep[2]*ep[2]*gp[2]- \\
 & 26*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*gp[2]+26*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*gp[2]+48*ap[2]*cp[1] \\
 & *cp[2]*dp[1]*fp[1]*gp[2]-48*ap[1]*cp[2]*cp[2]*dp[1]*fp[1]*gp[2]- \\
 & 33*ap[2]*ap[2]*dp[1]*dp[1]*fp[1]*gp[2]+13*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*gp[2]- \\
 & 13*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*gp[2]- \\
 & 24*ap[2]*cp[1]*cp[1]*dp[2]*fp[1]*gp[2]+24*ap[1]*cp[1]*cp[2]*dp[2]*fp[1]*gp[2]+44*ap[1]*ap[2]* \\
 & dp[1]*dp[2]*fp[1]*gp[2]-11*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*gp[2]- \\
 & 352*ap[2]*bp[2]*cp[1]*ep[1]*fp[1]*gp[2]- \\
 & 352*ap[2]*bp[1]*cp[2]*ep[1]*fp[1]*gp[2]+704*ap[1]*bp[2]*cp[2]*ep[1]*fp[1]*gp[2]+1056*ap[2]*a \\
 & p[2]*ep[1]*ep[1]*fp[1]*gp[2]+352*ap[2]*bp[1]*cp[1]*ep[2]*fp[1]*gp[2]- \\
 & 176*ap[1]*bp[2]*cp[1]*ep[2]*fp[1]*gp[2]-176*ap[1]*bp[1]*cp[2]*ep[2]*fp[1]*gp[2]- \\
 & 1408*ap[1]*ap[2]*ep[1]*ep[2]*fp[1]*gp[2]+352*ap[1]*ap[1]*ep[2]*ep[2]*fp[1]*gp[2]+163*ap[2]*b \\
 & p[1]*bp[2]*fp[1]*fp[1]*gp[2]-163*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*gp[2]- \\
 & 200*ap[2]*cp[1]*cp[2]*fp[1]*fp[1]*gp[2]+200*ap[1]*cp[2]*cp[2]*fp[1]*fp[1]*gp[2]+111*ap[2]*ap[\\
 & 2]*dp[1]*fp[1]*fp[1]*gp[2]-74*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*gp[2]- \\
 & 363*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*gp[2]+13*ap[2]*bp[1]*bp[1]*dp[1]*fp[2]*gp[2]- \\
 & 13*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*gp[2]- \\
 & 24*ap[2]*cp[1]*cp[1]*dp[1]*fp[2]*gp[2]+24*ap[1]*cp[1]*cp[2]*dp[1]*fp[2]*gp[2]+22*ap[1]*ap[2]* \\
 & dp[1]*dp[1]*fp[2]*gp[2]- \\
 & 22*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*gp[2]+352*ap[2]*bp[1]*cp[1]*ep[1]*fp[2]*gp[2]- \\
 & 176*ap[1]*bp[2]*cp[1]*ep[1]*fp[2]*gp[2]-176*ap[1]*bp[1]*cp[2]*ep[1]*fp[2]*gp[2]- \\
 & 704*ap[1]*ap[2]*ep[1]*ep[1]*fp[2]*gp[2]+704*ap[1]*ap[1]*ep[1]*ep[2]*fp[2]*gp[2]- \\
 & 163*ap[2]*bp[1]*bp[1]*fp[1]*fp[2]*gp[2]+163*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*gp[2]+200*ap[2]*cp[\\
 & 1]*cp[1]*fp[1]*fp[2]*gp[2]-200*ap[1]*cp[1]*cp[2]*fp[1]*fp[2]*gp[2]- \\
 & 148*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*gp[2]+74*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*gp[2]+726*ap[1]*ap[2] \\
 & *fp[1]*fp[1]*fp[2]*gp[2]+37*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*gp[2]- \\
 & 363*ap[1]*ap[1]*fp[1]*fp[2]*fp[2]*gp[2]+10*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*gp[2]- \\
 & 10*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*gp[2]-16*ap[2]*bp[2]*cp[1]*cp[1]*gp[1]*gp[2]- \\
 & 32*ap[2]*bp[1]*cp[1]*cp[2]*gp[1]*gp[2]+32*ap[1]*bp[2]*cp[1]*cp[2]*gp[1]*gp[2]+16*ap[1]*bp[1] \\
 & *cp[2]*cp[2]*gp[1]*gp[2]+9*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*gp[2]- \\
 & 3*ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*gp[2]+3*ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*gp[2]- \\
 & 9*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*gp[2]+48*ap[2]*ap[2]*cp[1]*ep[1]*gp[1]*gp[2]- \\
 & 16*ap[1]*ap[2]*cp[2]*ep[1]*gp[1]*gp[2]+16*ap[1]*ap[2]*cp[1]*ep[2]*gp[1]*gp[2]- \\
 & 48*ap[1]*ap[1]*cp[2]*ep[2]*gp[1]*gp[2]- \\
 & 39*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*gp[2]+13*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*gp[2]- \\
 & 13*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*gp[2]+39*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 9*ap[1]*ap[2]*ap[2]*gp[1]*gp[1]*gp[2]- \\
 & 5*ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*gp[2]+5*ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*gp[2]+24*ap[2]*bp[1]*c \\
 & p[1]*cp[1]*gp[2]*gp[2]-8*ap[1]*bp[2]*cp[1]*cp[1]*gp[2]*gp[2]- \\
 & 16*ap[1]*bp[1]*cp[1]*cp[2]*gp[2]*gp[2]- \\
 & 9*ap[1]*ap[2]*bp[1]*dp[1]*gp[2]*gp[2]+6*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*gp[2]+3*ap[1]*ap[1]*bp \\
 & [1]*dp[2]*gp[2]*gp[2]- \\
 & 48*ap[1]*ap[2]*cp[1]*ep[1]*gp[2]*gp[2]+32*ap[1]*ap[1]*cp[2]*ep[1]*gp[2]*gp[2]+16*ap[1]*ap[1] \\
 & *cp[1]*ep[2]*gp[2]*gp[2]+39*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*gp[2]- \\
 & 26*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*gp[2]- \\
 & 13*ap[1]*ap[1]*bp[1]*fp[2]*gp[2]*gp[2]+9*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*gp[2]- \\
 & 3*ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*gp[2]+8*ap[2]*bp[2]*cp[1]*dp[1]*dp[2]*hp[1]+8*ap[2]*bp[1]*cp \\
 & [2]*dp[1]*dp[2]*hp[1]-16*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 & 16*ap[2]*bp[1]*cp[1]*dp[2]*dp[2]*hp[1]+8*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*hp[1]+8*ap[1]*bp[1]*c \\
 & p[2]*dp[2]*dp[2]*hp[1]- \\
 & 16*ap[2]*bp[1]*bp[2]*dp[2]*ep[1]*hp[1]+16*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*hp[1]+32*ap[2]*cp[1] \\
 & *cp[2]*dp[2]*ep[1]*hp[1]-32*ap[1]*cp[2]*cp[2]*dp[2]*ep[1]*hp[1]- \\
 & 32*ap[2]*ap[2]*dp[1]*dp[2]*ep[1]*hp[1]+32*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*hp[1]- \\
 & 16*ap[2]*bp[1]*bp[2]*dp[1]*ep[2]*hp[1]+16*ap[1]*bp[2]*bp[2]*dp[1]*cp[2]*hp[1]+32*ap[2]*cp[1] \\
 & *cp[2]*dp[1]*cp[2]*hp[1]-32*ap[1]*cp[2]*cp[2]*dp[1]*ep[2]*hp[1]- \\
 & 16*ap[2]*ap[2]*dp[1]*dp[1]*ep[2]*hp[1]+32*ap[2]*bp[1]*bp[1]*dp[2]*ep[2]*hp[1]- \\
 & 32*ap[1]*bp[1]*bp[2]*dp[2]*ep[2]*hp[1]- \\
 & 64*ap[2]*cp[1]*cp[1]*dp[2]*ep[2]*hp[1]+64*ap[1]*cp[1]*cp[2]*dp[2]*ep[2]*hp[1]+64*ap[1]*ap[2] \\
 & *dp[1]*dp[2]*ep[2]*hp[1]-48*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*hp[1]- \\
 & 192*ap[2]*bp[2]*cp[1]*ep[1]*ep[2]*hp[1]- \\
 & 192*ap[2]*bp[1]*cp[2]*ep[1]*ep[2]*hp[1]+384*ap[1]*bp[2]*cp[2]*ep[1]*ep[2]*hp[1]+384*ap[2]*ap \\
 & [2]*ep[1]*ep[1]*ep[2]*hp[1]+384*ap[2]*bp[1]*cp[1]*ep[2]*ep[2]*hp[1]- \\
 & 192*ap[1]*bp[2]*cp[1]*ep[2]*ep[2]*hp[1]-192*ap[1]*bp[1]*cp[2]*ep[2]*ep[2]*hp[1]- \\
 & 768*ap[1]*ap[2]*ep[1]*ep[2]*ep[2]*hp[1]+384*ap[1]*ap[1]*ep[2]*ep[2]*ep[2]*hp[1]- \\
 & 24*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*hp[1]- \\
 & 24*ap[2]*bp[1]*cp[2]*dp[2]*fp[1]*hp[1]+48*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*hp[1]+96*ap[2]*ap[2] \\
 & *dp[2]*ep[1]*fp[1]*hp[1]+176*ap[2]*bp[1]*bp[2]*cp[2]*fp[1]*hp[1]- \\
 & 176*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*hp[1]- \\
 & 224*ap[2]*cp[1]*cp[2]*ep[2]*fp[1]*hp[1]+224*ap[1]*cp[2]*cp[2]*ep[2]*fp[1]*hp[1]+96*ap[2]*ap[2] \\
 & *dp[1]*ep[2]*fp[1]*hp[1]-192*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*hp[1]- \\
 & 400*ap[2]*ap[2]*ep[2]*fp[1]*fp[1]*hp[1]-24*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*hp[1]- \\
 & 24*ap[2]*bp[1]*cp[2]*dp[1]*fp[2]*hp[1]+48*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*hp[1]+96*ap[2]*bp[1] \\
 & *cp[1]*dp[2]*fp[2]*hp[1]-48*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 &48*ap[1]*bp[1]*cp[2]*dp[2]*fp[2]*hp[1]+176*ap[2]*bp[1]*bp[2]*cp[1]*fp[2]*hp[1]- \\
 &176*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*hp[1]- \\
 &224*ap[2]*cp[1]*cp[2]*ep[1]*fp[2]*hp[1]+224*ap[1]*cp[2]*cp[2]*ep[1]*fp[2]*hp[1]+96*ap[2]*ap[2] \\
 &]*dp[1]*ep[1]*fp[2]*hp[1]-192*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*hp[1]- \\
 &352*ap[2]*bp[1]*bp[1]*ep[2]*fp[2]*hp[1]+352*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*hp[1]+448*ap[2]*cp \\
 &[1]*cp[1]*ep[2]*fp[2]*hp[1]-448*ap[1]*cp[1]*cp[2]*ep[2]*fp[2]*hp[1]- \\
 &192*ap[1]*ap[2]*dp[1]*ep[2]*fp[2]*hp[1]+288*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*hp[1]+200*ap[2]*bp \\
 &[2]*cp[1]*fp[1]*fp[2]*hp[1]+200*ap[2]*bp[1]*cp[2]*fp[1]*fp[2]*hp[1]- \\
 &400*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*hp[1]- \\
 &800*ap[2]*ap[2]*ep[1]*fp[1]*fp[2]*hp[1]+1600*ap[1]*ap[2]*cp[2]*fp[1]*fp[2]*hp[1]- \\
 &400*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*hp[1]+200*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*hp[1]+200*ap[1]*bp[\\
 &1]*cp[2]*fp[2]*fp[2]*hp[1]+800*ap[1]*ap[2]*ep[1]*fp[2]*fp[2]*hp[1]- \\
 &1200*ap[1]*ap[1]*ep[2]*fp[2]*fp[2]*hp[1]+16*ap[2]*bp[2]*bp[2]*cp[1]*gp[1]*hp[1]+32*ap[2]*bp[\\
 &1]*bp[2]*cp[2]*gp[1]*hp[1]-48*ap[1]*bp[2]*bp[2]*cp[2]*gp[1]*hp[1]- \\
 &32*ap[2]*cp[1]*cp[2]*cp[2]*gp[1]*hp[1]+32*ap[1]*cp[2]*cp[2]*cp[2]*gp[1]*hp[1]+16*ap[2]*ap[2] \\
 &)*cp[2]*dp[1]*gp[1]*hp[1]+32*ap[2]*ap[2]*cp[1]*dp[2]*gp[1]*hp[1]- \\
 &48*ap[1]*ap[2]*cp[2]*dp[2]*gp[1]*hp[1]-32*ap[2]*ap[2]*bp[2]*ep[1]*gp[1]*hp[1]- \\
 &64*ap[2]*ap[2]*bp[1]*ep[2]*gp[1]*hp[1]+96*ap[1]*ap[2]*bp[2]*ep[2]*gp[1]*hp[1]- \\
 &48*ap[2]*ap[2]*cp[2]*fp[1]*gp[1]*hp[1]- \\
 &96*ap[2]*ap[2]*cp[1]*fp[2]*gp[1]*hp[1]+144*ap[1]*ap[2]*cp[2]*fp[2]*gp[1]*hp[1]- \\
 &32*ap[2]*bp[1]*bp[2]*cp[1]*gp[2]*hp[1]+16*ap[1]*bp[2]*bp[2]*cp[1]*gp[2]*hp[1]- \\
 &16*ap[2]*bp[1]*bp[1]*cp[2]*gp[2]*hp[1]+32*ap[1]*bp[1]*bp[2]*cp[2]*gp[2]*hp[1]+32*ap[2]*cp[1] \\
 &)*cp[1]*cp[2]*gp[2]*hp[1]-32*ap[1]*cp[1]*cp[2]*cp[2]*gp[2]*hp[1]- \\
 &24*ap[2]*ap[2]*cp[1]*dp[1]*gp[2]*hp[1]+8*ap[1]*ap[2]*cp[2]*dp[1]*gp[2]*hp[1]- \\
 &8*ap[1]*ap[2]*cp[1]*dp[2]*gp[2]*hp[1]+24*ap[1]*ap[1]*cp[2]*dp[2]*gp[2]*hp[1]+48*ap[2]*ap[2]* \\
 &bp[1]*ep[1]*gp[2]*hp[1]- \\
 &16*ap[1]*ap[2]*bp[2]*ep[1]*gp[2]*hp[1]+16*ap[1]*ap[2]*bp[1]*ep[2]*gp[2]*hp[1]- \\
 &48*ap[1]*ap[1]*bp[2]*ep[2]*gp[2]*hp[1]+72*ap[2]*ap[2]*cp[1]*fp[1]*gp[2]*hp[1]- \\
 &24*ap[1]*ap[2]*cp[2]*fp[1]*gp[2]*hp[1]+24*ap[1]*ap[2]*cp[1]*fp[2]*gp[2]*hp[1]- \\
 &72*ap[1]*ap[1]*cp[2]*fp[2]*gp[2]*hp[1]+8*ap[2]*bp[1]*bp[2]*bp[2]*hp[1]*hp[1]- \\
 &8*ap[1]*bp[2]*bp[2]*bp[2]*hp[1]*hp[1]-32*ap[2]*bp[2]*cp[1]*cp[2]*hp[1]*hp[1]- \\
 &16*ap[2]*bp[1]*cp[2]*cp[2]*hp[1]*hp[1]+48*ap[1]*bp[2]*cp[2]*cp[2]*hp[1]*hp[1]+8*ap[2]*ap[2]* \\
 &bp[2]*dp[1]*hp[1]*hp[1]+16*ap[2]*ap[2]*bp[1]*dp[2]*hp[1]*hp[1]- \\
 &24*ap[1]*ap[2]*bp[2]*dp[2]*hp[1]*hp[1]+32*ap[2]*ap[2]*cp[2]*ep[1]*hp[1]*hp[1]+64*ap[2]*ap[2] \\
 &)*cp[1]*ep[2]*hp[1]*hp[1]-96*ap[1]*ap[2]*cp[2]*ep[2]*hp[1]*hp[1]- \\
 &24*ap[2]*ap[2]*bp[2]*fp[1]*hp[1]*hp[1]- \\
 &48*ap[2]*ap[2]*bp[1]*fp[2]*hp[1]*hp[1]+72*ap[1]*ap[2]*bp[2]*fp[2]*hp[1]*hp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 24*ap[2]*ap[2]*ap[2]*gp[1]*hp[1]*hp[1]+24*ap[1]*ap[2]*ap[2]*gp[2]*hp[1]*hp[1]- \\
 & 8*ap[2]*bp[2]*cp[1]*dp[1]*dp[1]*hp[2]- \\
 & 8*ap[2]*bp[1]*cp[2]*dp[1]*dp[1]*hp[2]+16*ap[1]*bp[2]*cp[2]*dp[1]*dp[1]*hp[2]+16*ap[2]*bp[1]* \\
 & cp[1]*dp[1]*dp[2]*hp[2]-8*ap[1]*bp[2]*cp[1]*dp[1]*dp[2]*hp[2]- \\
 & 8*ap[1]*bp[1]*cp[2]*dp[1]*dp[2]*hp[2]+32*ap[2]*bp[1]*bp[2]*dp[1]*cp[1]*hp[2]- \\
 & 32*ap[1]*bp[2]*bp[2]*dp[1]*ep[1]*hp[2]- \\
 & 64*ap[2]*cp[1]*cp[2]*dp[1]*ep[1]*hp[2]+64*ap[1]*cp[2]*cp[2]*dp[1]*ep[1]*hp[2]+48*ap[2]*ap[2] \\
 & *dp[1]*dp[1]*ep[1]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[1]*dp[2]*ep[1]*hp[2]+16*ap[1]*bp[1]*bp[2]*dp[2]*ep[1]*hp[2]+32*ap[2]*cp[1] \\
 & *cp[1]*dp[2]*ep[1]*hp[2]-32*ap[1]*cp[1]*cp[2]*dp[2]*ep[1]*hp[2]- \\
 & 64*ap[1]*ap[2]*dp[1]*dp[2]*ep[1]*hp[2]+16*ap[1]*ap[1]*dp[2]*dp[2]*ep[1]*hp[2]+192*ap[2]*bp[2] \\
 & *cp[1]*ep[1]*ep[1]*hp[2]+192*ap[2]*bp[1]*cp[2]*ep[1]*ep[1]*hp[2]- \\
 & 384*ap[1]*bp[2]*cp[2]*ep[1]*ep[1]*hp[2]-384*ap[2]*ap[2]*ep[1]*ep[1]*ep[1]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[1]*dp[1]*ep[2]*hp[2]+16*ap[1]*bp[1]*bp[2]*dp[1]*ep[2]*hp[2]+32*ap[2]*cp[1] \\
 & *cp[1]*dp[1]*ep[2]*hp[2]-32*ap[1]*cp[1]*cp[2]*dp[1]*ep[2]*hp[2]- \\
 & 32*ap[1]*ap[2]*dp[1]*dp[1]*ep[2]*hp[2]+32*ap[1]*ap[1]*dp[1]*dp[2]*ep[2]*hp[2]- \\
 & 384*ap[2]*bp[1]*cp[1]*ep[1]*ep[2]*hp[2]+192*ap[1]*bp[2]*cp[1]*ep[1]*ep[2]*hp[2]+192*ap[1]*b \\
 & p[1]*cp[2]*ep[1]*cp[2]*hp[2]+768*ap[1]*ap[2]*ep[1]*ep[1]*ep[2]*hp[2]- \\
 & 384*ap[1]*ap[1]*ep[1]*ep[2]*ep[2]*hp[2]+48*ap[2]*bp[2]*cp[1]*dp[1]*fp[1]*hp[2]+48*ap[2]*bp[1] \\
 & *cp[2]*dp[1]*fp[1]*hp[2]-96*ap[1]*bp[2]*cp[2]*dp[1]*fp[1]*hp[2]- \\
 & 48*ap[2]*bp[1]*cp[1]*dp[2]*fp[1]*hp[2]+24*ap[1]*bp[2]*cp[1]*dp[2]*fp[1]*hp[2]+24*ap[1]*bp[1] \\
 & *cp[2]*dp[2]*fp[1]*hp[2]- \\
 & 352*ap[2]*bp[1]*bp[2]*ep[1]*fp[1]*hp[2]+352*ap[1]*bp[2]*bp[2]*ep[1]*fp[1]*hp[2]+448*ap[2]*cp \\
 & [1]*cp[2]*ep[1]*fp[1]*hp[2]-448*ap[1]*cp[2]*cp[2]*ep[1]*fp[1]*hp[2]- \\
 & 288*ap[2]*ap[2]*dp[1]*ep[1]*fp[1]*hp[2]+192*ap[1]*ap[2]*dp[2]*ep[1]*fp[1]*hp[2]+176*ap[2]*bp \\
 & [1]*bp[1]*ep[2]*fp[1]*hp[2]-176*ap[1]*bp[1]*bp[2]*ep[2]*fp[1]*hp[2]- \\
 & 224*ap[2]*cp[1]*cp[1]*ep[2]*fp[1]*hp[2]+224*ap[1]*cp[1]*cp[2]*ep[2]*fp[1]*hp[2]+192*ap[1]*ap[\\
 & 2]*dp[1]*ep[2]*fp[1]*hp[2]-96*ap[1]*ap[1]*dp[2]*ep[2]*fp[1]*hp[2]- \\
 & 200*ap[2]*bp[2]*cp[1]*fp[1]*fp[1]*hp[2]- \\
 & 200*ap[2]*bp[1]*cp[2]*fp[1]*fp[1]*hp[2]+400*ap[1]*bp[2]*cp[2]*fp[1]*fp[1]*hp[2]+1200*ap[2]*a \\
 & p[2]*ep[1]*fp[1]*fp[1]*hp[2]-800*ap[1]*ap[2]*ep[2]*fp[1]*fp[1]*hp[2]- \\
 & 48*ap[2]*bp[1]*cp[1]*dp[1]*fp[2]*hp[2]+24*ap[1]*bp[2]*cp[1]*dp[1]*fp[2]*hp[2]+24*ap[1]*bp[1] \\
 & *cp[2]*dp[1]*fp[2]*hp[2]+176*ap[2]*bp[1]*bp[1]*ep[1]*fp[2]*hp[2]- \\
 & 176*ap[1]*bp[1]*bp[2]*ep[1]*fp[2]*hp[2]- \\
 & 224*ap[2]*cp[1]*cp[1]*ep[1]*fp[2]*hp[2]+224*ap[1]*cp[1]*cp[2]*ep[1]*fp[2]*hp[2]+192*ap[1]*ap[\\
 & 2]*dp[1]*ep[1]*fp[2]*hp[2]-96*ap[1]*ap[1]*dp[2]*ep[1]*fp[2]*hp[2]- \\
 & 96*ap[1]*ap[1]*dp[1]*ep[2]*fp[2]*hp[2]+400*ap[2]*bp[1]*cp[1]*fp[1]*fp[2]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 200*ap[1]*bp[2]*cp[1]*fp[1]*fp[2]*hp[2]-200*ap[1]*bp[1]*cp[2]*fp[1]*fp[2]*hp[2]- \\
 & 1600*ap[1]*ap[2]*ep[1]*fp[1]*fp[2]*hp[2]+800*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*hp[2]+400*ap[1]*ap \\
 & [1]*ep[1]*fp[2]*fp[2]*hp[2]- \\
 & 32*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*hp[2]+16*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*hp[2]+32*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*hp[2]+32*ap[2]*cp[1] \\
 & *cp[1]*cp[2]*gp[1]*hp[2]-32*ap[1]*cp[1]*cp[2]*cp[2]*gp[1]*hp[2]- \\
 & 24*ap[2]*ap[2]*cp[1]*dp[1]*gp[1]*hp[2]+8*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*hp[2]- \\
 & 8*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*hp[2]+24*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*hp[2]+48*ap[2]*ap[2]* \\
 & bp[1]*ep[1]*gp[1]*hp[2]- \\
 & 16*ap[1]*ap[2]*bp[2]*ep[1]*gp[1]*hp[2]+16*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*hp[2]- \\
 & 48*ap[1]*ap[1]*bp[2]*ep[2]*gp[1]*hp[2]+72*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*hp[2]- \\
 & 24*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*hp[2]+24*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*hp[2]- \\
 & 72*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*hp[2]+48*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*hp[2]- \\
 & 32*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*hp[2]-32*ap[2]*cp[1]*cp[1]*cp[1]*gp[2]*hp[2]- \\
 & 16*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*hp[2]+32*ap[1]*cp[1]*cp[1]*cp[2]*gp[2]*hp[2]+48*ap[1]*ap[2] \\
 & *cp[1]*dp[1]*gp[2]*hp[2]-32*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*hp[2]- \\
 & 16*ap[1]*ap[1]*cp[1]*dp[2]*gp[2]*hp[2]- \\
 & 96*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*hp[2]+64*ap[1]*ap[1]*bp[2]*ep[1]*gp[2]*hp[2]+32*ap[1]*ap[1] \\
 & *bp[1]*ep[2]*gp[2]*hp[2]- \\
 & 144*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*hp[2]+96*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*hp[2]+48*ap[1]*ap[1] \\
 & *cp[1]*fp[2]*gp[2]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[1]*bp[2]*hp[1]*hp[2]+16*ap[1]*bp[1]*bp[2]*bp[2]*hp[1]*hp[2]+32*ap[2]*bp[2] \\
 & *cp[1]*cp[1]*hp[1]*hp[2]+64*ap[2]*bp[1]*cp[1]*cp[2]*hp[1]*hp[2]- \\
 & 64*ap[1]*bp[2]*cp[1]*cp[2]*hp[1]*hp[2]-32*ap[1]*bp[1]*cp[2]*cp[2]*hp[1]*hp[2]- \\
 & 24*ap[2]*ap[2]*bp[1]*dp[1]*hp[1]*hp[2]+8*ap[1]*ap[2]*bp[2]*dp[1]*hp[1]*hp[2]- \\
 & 8*ap[1]*ap[2]*bp[1]*dp[2]*hp[1]*hp[2]+24*ap[1]*ap[1]*bp[2]*dp[2]*hp[1]*hp[2]- \\
 & 96*ap[2]*ap[2]*cp[1]*ep[1]*hp[1]*hp[2]+32*ap[1]*ap[2]*cp[2]*ep[1]*hp[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*cp[1]*ep[2]*hp[1]*hp[2]+96*ap[1]*ap[1]*cp[2]*ep[2]*hp[1]*hp[2]+72*ap[2]*ap[2] \\
 & *bp[1]*fp[1]*hp[1]*hp[2]- \\
 & 24*ap[1]*ap[2]*bp[2]*fp[1]*hp[1]*hp[2]+24*ap[1]*ap[2]*bp[1]*fp[2]*hp[1]*hp[2]- \\
 & 72*ap[1]*ap[1]*bp[2]*fp[2]*hp[1]*hp[2]+48*ap[1]*ap[2]*ap[2]*gp[1]*hp[1]*hp[2]- \\
 & 48*ap[1]*ap[1]*ap[2]*gp[2]*hp[1]*hp[2]+8*ap[2]*bp[1]*bp[1]*bp[1]*hp[2]*hp[2]- \\
 & 8*ap[1]*bp[1]*bp[1]*bp[2]*hp[2]*hp[2]- \\
 & 48*ap[2]*bp[1]*cp[1]*cp[1]*hp[2]*hp[2]+16*ap[1]*bp[2]*cp[1]*cp[1]*hp[2]*hp[2]+32*ap[1]*bp[1] \\
 & *cp[1]*cp[2]*hp[2]*hp[2]+24*ap[1]*ap[2]*bp[1]*dp[1]*hp[2]*hp[2]- \\
 & 16*ap[1]*ap[1]*bp[2]*dp[1]*hp[2]*hp[2]- \\
 & 8*ap[1]*ap[1]*bp[1]*dp[2]*hp[2]*hp[2]+96*ap[1]*ap[2]*cp[1]*ep[1]*hp[2]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &64*ap[1]*ap[1]*cp[2]*ep[1]*hp[2]*hp[2]-32*ap[1]*ap[1]*cp[1]*ep[2]*hp[2]*hp[2]- \\
 &72*ap[1]*ap[2]*bp[1]*fp[1]*hp[2]*hp[2]+48*ap[1]*ap[1]*bp[2]*fp[1]*hp[2]*hp[2]+24*ap[1]*ap[1] \\
 &*bp[1]*fp[2]*hp[2]*hp[2]- \\
 &24*ap[1]*ap[1]*ap[2]*gp[1]*hp[2]*hp[2]+24*ap[1]*ap[1]*ap[1]*gp[2]*hp[2]*hp[2] \\
 \\
 &w_7=-8*ap[2]*bp[2]*cp[1]*dp[1]*dp[2]*gp[1]- \\
 &8*ap[2]*bp[1]*cp[2]*dp[1]*dp[2]*gp[1]+16*ap[1]*bp[2]*cp[2]*dp[1]*dp[2]*gp[1]+16*ap[2]*bp[1]* \\
 &cp[1]*dp[2]*dp[2]*gp[1]-8*ap[1]*bp[2]*cp[1]*dp[2]*dp[2]*gp[1]- \\
 &8*ap[1]*bp[1]*cp[2]*dp[2]*dp[2]*gp[1]+16*ap[2]*bp[1]*bp[2]*dp[2]*ep[1]*gp[1]- \\
 &16*ap[1]*bp[2]*bp[2]*dp[2]*ep[1]*gp[1]- \\
 &32*ap[2]*cp[1]*cp[2]*dp[2]*ep[1]*gp[1]+32*ap[1]*cp[2]*cp[2]*dp[2]*ep[1]*gp[1]+32*ap[2]*ap[2] \\
 &*dp[1]*dp[2]*ep[1]*gp[1]- \\
 &32*ap[1]*ap[2]*dp[2]*dp[2]*ep[1]*gp[1]+16*ap[2]*bp[1]*bp[2]*dp[1]*cp[2]*gp[1]- \\
 &16*ap[1]*bp[2]*bp[2]*dp[1]*cp[2]*gp[1]- \\
 &32*ap[2]*cp[1]*cp[2]*dp[1]*ep[2]*gp[1]+32*ap[1]*cp[2]*cp[2]*dp[1]*ep[2]*gp[1]+16*ap[2]*ap[2] \\
 &*dp[1]*dp[1]*ep[2]*gp[1]- \\
 &32*ap[2]*bp[1]*bp[1]*dp[2]*ep[2]*gp[1]+32*ap[1]*bp[1]*bp[2]*dp[2]*ep[2]*gp[1]+64*ap[2]*cp[1] \\
 &*cp[1]*dp[2]*cp[2]*gp[1]-64*ap[1]*cp[1]*cp[2]*dp[2]*ep[2]*gp[1]- \\
 &64*ap[1]*ap[2]*dp[1]*dp[2]*ep[2]*gp[1]+48*ap[1]*ap[1]*dp[2]*dp[2]*ep[2]*gp[1]+192*ap[2]*bp[2] \\
 &]*cp[1]*ep[1]*ep[2]*gp[1]+192*ap[2]*bp[1]*cp[2]*ep[1]*ep[2]*gp[1]- \\
 &384*ap[1]*bp[2]*cp[2]*ep[1]*ep[2]*gp[1]-384*ap[2]*ap[2]*ep[1]*ep[1]*ep[2]*gp[1]- \\
 &384*ap[2]*bp[1]*cp[1]*ep[2]*ep[2]*gp[1]+192*ap[1]*bp[2]*cp[1]*ep[2]*ep[2]*gp[1]+192*ap[1]*b \\
 &p[1]*cp[2]*ep[2]*ep[2]*gp[1]+768*ap[1]*ap[2]*ep[1]*ep[2]*ep[2]*gp[1]- \\
 &384*ap[1]*ap[1]*ep[2]*ep[2]*ep[2]*gp[1]+24*ap[2]*bp[2]*cp[1]*dp[2]*fp[1]*gp[1]+24*ap[2]*bp[1] \\
 &]*cp[2]*dp[2]*fp[1]*gp[1]-48*ap[1]*bp[2]*cp[2]*dp[2]*fp[1]*gp[1]- \\
 &96*ap[2]*ap[2]*dp[2]*ep[1]*fp[1]*gp[1]- \\
 &176*ap[2]*bp[1]*bp[2]*ep[2]*fp[1]*gp[1]+176*ap[1]*bp[2]*bp[2]*ep[2]*fp[1]*gp[1]+224*ap[2]*cp \\
 &[1]*cp[2]*ep[2]*fp[1]*gp[1]-224*ap[1]*cp[2]*cp[2]*ep[2]*fp[1]*gp[1]- \\
 &96*ap[2]*ap[2]*dp[1]*ep[2]*fp[1]*gp[1]+192*ap[1]*ap[2]*dp[2]*ep[2]*fp[1]*gp[1]+400*ap[2]*ap[\\
 &2]*ep[2]*fp[1]*fp[1]*gp[1]+24*ap[2]*bp[2]*cp[1]*dp[1]*fp[2]*gp[1]+24*ap[2]*bp[1]*cp[2]*dp[1]* \\
 &fp[2]*gp[1]-48*ap[1]*bp[2]*cp[2]*dp[1]*fp[2]*gp[1]- \\
 &96*ap[2]*bp[1]*cp[1]*dp[2]*fp[2]*gp[1]+48*ap[1]*bp[2]*cp[1]*dp[2]*fp[2]*gp[1]+48*ap[1]*bp[1] \\
 &*cp[2]*dp[2]*fp[2]*gp[1]- \\
 &176*ap[2]*bp[1]*bp[2]*ep[1]*fp[2]*gp[1]+176*ap[1]*bp[2]*bp[2]*ep[1]*fp[2]*gp[1]+224*ap[2]*cp \\
 &[1]*cp[2]*ep[1]*fp[2]*gp[1]-224*ap[1]*cp[2]*cp[2]*ep[1]*fp[2]*gp[1]- \\
 &96*ap[2]*ap[2]*dp[1]*ep[1]*fp[2]*gp[1]+192*ap[1]*ap[2]*dp[2]*ep[1]*fp[2]*gp[1]+352*ap[2]*bp[\\
 &1]*bp[1]*ep[2]*fp[2]*gp[1]-352*ap[1]*bp[1]*bp[2]*ep[2]*fp[2]*gp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 448*ap[2]*cp[1]*cp[1]*ep[2]*fp[2]*gp[1]+448*ap[1]*cp[1]*cp[2]*ep[2]*fp[2]*gp[1]+192*ap[1]*ap[2]*dp[1]*ep[2]*fp[2]*gp[1]-288*ap[1]*ap[1]*dp[2]*ep[2]*fp[2]*gp[1]- \\
 & 200*ap[2]*bp[2]*cp[1]*fp[1]*fp[2]*gp[1]- \\
 & 200*ap[2]*bp[1]*cp[2]*fp[1]*fp[2]*gp[1]+400*ap[1]*bp[2]*cp[2]*fp[1]*fp[2]*gp[1]+800*ap[2]*ap[2]*ep[1]*fp[1]*fp[2]*gp[1]- \\
 & 1600*ap[1]*ap[2]*ep[2]*fp[1]*fp[2]*gp[1]+400*ap[2]*bp[1]*cp[1]*fp[2]*fp[2]*gp[1]- \\
 & 200*ap[1]*bp[2]*cp[1]*fp[2]*fp[2]*gp[1]-200*ap[1]*bp[1]*cp[2]*fp[2]*fp[2]*gp[1]- \\
 & 800*ap[1]*ap[2]*ep[1]*fp[2]*fp[2]*gp[1]+1200*ap[1]*ap[1]*ep[2]*fp[2]*fp[2]*gp[1]- \\
 & 8*ap[2]*bp[2]*bp[2]*cp[1]*gp[1]*gp[1]- \\
 & 16*ap[2]*bp[1]*bp[2]*cp[2]*gp[1]*gp[1]+24*ap[1]*bp[2]*bp[2]*cp[2]*gp[1]*gp[1]+16*ap[2]*cp[1]*cp[2]*cp[2]*gp[1]*gp[1]-16*ap[1]*cp[2]*cp[2]*cp[2]*gp[1]*gp[1]- \\
 & 8*ap[2]*ap[2]*cp[2]*dp[1]*gp[1]*gp[1]- \\
 & 16*ap[2]*ap[2]*cp[1]*dp[2]*gp[1]*gp[1]+24*ap[1]*ap[2]*cp[2]*dp[2]*gp[1]*gp[1]+16*ap[2]*ap[2]*bp[2]*ep[1]*gp[1]*gp[1]+32*ap[2]*ap[2]*bp[1]*ep[2]*gp[1]*gp[1]- \\
 & 48*ap[1]*ap[2]*bp[2]*ep[2]*gp[1]*gp[1]+24*ap[2]*ap[2]*cp[2]*fp[1]*gp[1]*gp[1]+48*ap[2]*ap[2]*cp[1]*fp[2]*gp[1]*gp[1]- \\
 & 72*ap[1]*ap[2]*cp[2]*fp[2]*gp[1]*gp[1]+8*ap[2]*bp[2]*cp[1]*dp[1]*dp[1]*gp[2]+8*ap[2]*bp[1]*cp[2]*dp[1]*dp[1]*gp[2]-16*ap[1]*bp[2]*cp[2]*dp[1]*dp[1]*gp[2]- \\
 & 16*ap[2]*bp[1]*cp[1]*dp[1]*dp[2]*gp[2]+8*ap[1]*bp[2]*cp[1]*dp[1]*dp[2]*gp[2]+8*ap[1]*bp[1]*cp[2]*dp[1]*dp[2]*gp[2]- \\
 & 32*ap[2]*bp[1]*bp[2]*dp[1]*ep[1]*gp[2]+32*ap[1]*bp[2]*bp[2]*dp[1]*ep[1]*gp[2]+64*ap[2]*cp[1]*cp[2]*dp[1]*ep[1]*gp[2]-64*ap[1]*cp[2]*cp[2]*dp[1]*ep[1]*gp[2]- \\
 & 48*ap[2]*ap[2]*dp[1]*dp[1]*ep[1]*gp[2]+16*ap[2]*bp[1]*bp[1]*dp[2]*ep[1]*gp[2]- \\
 & 16*ap[1]*bp[1]*bp[2]*dp[2]*ep[1]*gp[2]- \\
 & 32*ap[2]*cp[1]*cp[1]*dp[2]*ep[1]*gp[2]+32*ap[1]*cp[1]*cp[2]*dp[2]*ep[1]*gp[2]+64*ap[1]*ap[2]*dp[1]*dp[2]*ep[1]*gp[2]-16*ap[1]*ap[1]*dp[2]*dp[2]*ep[1]*gp[2]- \\
 & 192*ap[2]*bp[2]*cp[1]*ep[1]*ep[1]*gp[2]- \\
 & 192*ap[2]*bp[1]*cp[2]*ep[1]*ep[1]*gp[2]+384*ap[1]*bp[2]*cp[2]*ep[1]*ep[1]*gp[2]+384*ap[2]*ap[2]*ep[1]*ep[1]*ep[1]*gp[2]+16*ap[2]*bp[1]*bp[1]*dp[1]*ep[2]*gp[2]- \\
 & 16*ap[1]*bp[1]*bp[2]*dp[1]*ep[2]*gp[2]- \\
 & 32*ap[2]*cp[1]*cp[1]*dp[1]*ep[2]*gp[2]+32*ap[1]*cp[1]*cp[2]*dp[1]*ep[2]*gp[2]+32*ap[1]*ap[2]*dp[1]*dp[1]*ep[2]*gp[2]- \\
 & 32*ap[1]*ap[1]*dp[1]*dp[2]*ep[2]*gp[2]+384*ap[2]*bp[1]*cp[1]*ep[1]*ep[2]*gp[2]- \\
 & 192*ap[1]*bp[2]*cp[1]*ep[1]*ep[2]*gp[2]-192*ap[1]*bp[1]*cp[2]*ep[1]*ep[2]*gp[2]- \\
 & 768*ap[1]*ap[2]*ep[1]*ep[1]*ep[2]*gp[2]+384*ap[1]*ap[1]*ep[1]*ep[2]*ep[2]*gp[2]- \\
 & 48*ap[2]*bp[2]*cp[1]*dp[1]*fp[1]*gp[2]- \\
 & 48*ap[2]*bp[1]*cp[2]*dp[1]*fp[1]*gp[2]+96*ap[1]*bp[2]*cp[2]*dp[1]*fp[1]*gp[2]+48*ap[2]*bp[1]
 \end{aligned}$$

$$\begin{aligned}
 & *cp[1]*dp[2]*fp[1]*gp[2]-24*ap[1]*bp[2]*cp[1]*dp[2]*fp[1]*gp[2]- \\
 & 24*ap[1]*bp[1]*cp[2]*dp[2]*fp[1]*gp[2]+352*ap[2]*bp[1]*bp[2]*ep[1]*fp[1]*gp[2]- \\
 & 352*ap[1]*bp[2]*bp[2]*ep[1]*fp[1]*gp[2]- \\
 & 448*ap[2]*cp[1]*cp[2]*ep[1]*fp[1]*gp[2]+448*ap[1]*cp[2]*cp[2]*ep[1]*fp[1]*gp[2]+288*ap[2]*ap[\\
 & 2]*dp[1]*ep[1]*fp[1]*gp[2]-192*ap[1]*ap[2]*dp[2]*ep[1]*fp[1]*gp[2]- \\
 & 176*ap[2]*bp[1]*bp[1]*ep[2]*fp[1]*gp[2]+176*ap[1]*bp[1]*bp[2]*ep[2]*fp[1]*gp[2]+224*ap[2]*cp \\
 & [1]*cp[1]*ep[2]*fp[1]*gp[2]-224*ap[1]*cp[1]*cp[2]*ep[2]*fp[1]*gp[2]- \\
 & 192*ap[1]*ap[2]*dp[1]*ep[2]*fp[1]*gp[2]+96*ap[1]*ap[1]*dp[2]*ep[2]*fp[1]*gp[2]+200*ap[2]*bp[\\
 & 2]*cp[1]*fp[1]*fp[1]*gp[2]+200*ap[2]*bp[1]*cp[2]*fp[1]*fp[1]*gp[2]- \\
 & 400*ap[1]*bp[2]*cp[2]*fp[1]*fp[1]*gp[2]- \\
 & 1200*ap[2]*ap[2]*ep[1]*fp[1]*fp[1]*gp[2]+800*ap[1]*ap[2]*ep[2]*fp[1]*fp[1]*gp[2]+48*ap[2]*bp[\\
 & 1]*cp[1]*dp[1]*fp[2]*gp[2]-24*ap[1]*bp[2]*cp[1]*dp[1]*fp[2]*gp[2]- \\
 & 24*ap[1]*bp[1]*cp[2]*dp[1]*fp[2]*gp[2]- \\
 & 176*ap[2]*bp[1]*bp[1]*ep[1]*fp[2]*gp[2]+176*ap[1]*bp[1]*bp[2]*ep[1]*fp[2]*gp[2]+224*ap[2]*cp \\
 & [1]*cp[1]*ep[1]*fp[2]*gp[2]-224*ap[1]*cp[1]*cp[2]*ep[1]*fp[2]*gp[2]- \\
 & 192*ap[1]*ap[2]*dp[1]*ep[1]*fp[2]*gp[2]+96*ap[1]*ap[1]*dp[2]*ep[1]*fp[2]*gp[2]+96*ap[1]*ap[1] \\
 & *dp[1]*ep[2]*fp[2]*gp[2]- \\
 & 400*ap[2]*bp[1]*cp[1]*fp[1]*fp[2]*gp[2]+200*ap[1]*bp[2]*cp[1]*fp[1]*fp[2]*gp[2]+200*ap[1]*bp[\\
 & 1]*cp[2]*fp[1]*fp[2]*gp[2]+1600*ap[1]*ap[2]*ep[1]*fp[1]*fp[2]*gp[2]- \\
 & 800*ap[1]*ap[1]*ep[2]*fp[1]*fp[2]*gp[2]- \\
 & 400*ap[1]*ap[1]*ep[1]*fp[2]*fp[2]*gp[2]+32*ap[2]*bp[1]*bp[2]*cp[1]*gp[1]*gp[2]- \\
 & 16*ap[1]*bp[2]*bp[2]*cp[1]*gp[1]*gp[2]+16*ap[2]*bp[1]*bp[1]*cp[2]*gp[1]*gp[2]- \\
 & 32*ap[1]*bp[1]*bp[2]*cp[2]*gp[1]*gp[2]- \\
 & 32*ap[2]*cp[1]*cp[1]*cp[2]*gp[1]*gp[2]+32*ap[1]*cp[1]*cp[2]*cp[2]*gp[1]*gp[2]+24*ap[2]*ap[2] \\
 & *cp[1]*dp[1]*gp[1]*gp[2]- \\
 & 8*ap[1]*ap[2]*cp[2]*dp[1]*gp[1]*gp[2]+8*ap[1]*ap[2]*cp[1]*dp[2]*gp[1]*gp[2]- \\
 & 24*ap[1]*ap[1]*cp[2]*dp[2]*gp[1]*gp[2]- \\
 & 48*ap[2]*ap[2]*bp[1]*ep[1]*gp[1]*gp[2]+16*ap[1]*ap[2]*bp[2]*ep[1]*gp[1]*gp[2]- \\
 & 16*ap[1]*ap[2]*bp[1]*ep[2]*gp[1]*gp[2]+48*ap[1]*ap[1]*bp[2]*ep[2]*gp[1]*gp[2]- \\
 & 72*ap[2]*ap[2]*cp[1]*fp[1]*gp[1]*gp[2]+24*ap[1]*ap[2]*cp[2]*fp[1]*gp[1]*gp[2]- \\
 & 24*ap[1]*ap[2]*cp[1]*fp[2]*gp[1]*gp[2]+72*ap[1]*ap[1]*cp[2]*fp[2]*gp[1]*gp[2]- \\
 & 24*ap[2]*bp[1]*bp[1]*cp[1]*gp[2]*gp[2]+16*ap[1]*bp[1]*bp[2]*cp[1]*gp[2]*gp[2]+16*ap[2]*cp[1] \\
 & *cp[1]*cp[1]*gp[2]*gp[2]+8*ap[1]*bp[1]*bp[1]*cp[2]*gp[2]*gp[2]- \\
 & 16*ap[1]*cp[1]*cp[1]*cp[2]*gp[2]*gp[2]- \\
 & 24*ap[1]*ap[2]*cp[1]*dp[1]*gp[2]*gp[2]+16*ap[1]*ap[1]*cp[2]*dp[1]*gp[2]*gp[2]+8*ap[1]*ap[1]* \\
 & cp[1]*dp[2]*gp[2]*gp[2]+48*ap[1]*ap[2]*bp[1]*ep[1]*gp[2]*gp[2]- \\
 & 32*ap[1]*ap[1]*bp[2]*ep[1]*gp[2]*gp[2]-
 \end{aligned}$$

$$\begin{aligned}
 &16*ap[1]*ap[1]*bp[1]*ep[2]*gp[2]*gp[2]+72*ap[1]*ap[2]*cp[1]*fp[1]*gp[2]*gp[2]- \\
 &48*ap[1]*ap[1]*cp[2]*fp[1]*gp[2]*gp[2]-24*ap[1]*ap[1]*cp[1]*fp[2]*gp[2]*gp[2]- \\
 &8*ap[2]*bp[1]*bp[2]*dp[1]*dp[2]*hp[1]+8*ap[1]*bp[2]*bp[2]*dp[1]*dp[2]*hp[1]+48*ap[2]*cp[1]*c \\
 &p[2]*dp[1]*dp[2]*hp[1]-48*ap[1]*cp[2]*cp[2]*dp[1]*dp[2]*hp[1]- \\
 &40*ap[2]*ap[2]*dp[1]*dp[1]*dp[2]*hp[1]+8*ap[2]*bp[1]*bp[1]*dp[2]*dp[2]*hp[1]- \\
 &8*ap[1]*bp[1]*bp[2]*dp[2]*dp[2]*hp[1]- \\
 &48*ap[2]*cp[1]*cp[1]*dp[2]*dp[2]*hp[1]+48*ap[1]*cp[1]*cp[2]*dp[2]*dp[2]*hp[1]+80*ap[1]*ap[2] \\
 &*dp[1]*dp[2]*dp[2]*hp[1]-40*ap[1]*ap[1]*dp[2]*dp[2]*dp[2]*hp[1]- \\
 &32*ap[2]*bp[2]*cp[1]*dp[2]*ep[1]*hp[1]- \\
 &32*ap[2]*bp[1]*cp[2]*dp[2]*ep[1]*hp[1]+64*ap[1]*bp[2]*cp[2]*dp[2]*ep[1]*hp[1]+64*ap[2]*ap[2] \\
 &*dp[2]*ep[1]*cp[1]*hp[1]-32*ap[2]*bp[2]*cp[1]*dp[1]*ep[2]*hp[1]- \\
 &32*ap[2]*bp[1]*cp[2]*dp[1]*ep[2]*hp[1]+64*ap[1]*bp[2]*cp[2]*dp[1]*ep[2]*hp[1]+128*ap[2]*bp[1] \\
 &]*cp[1]*dp[2]*ep[2]*hp[1]-64*ap[1]*bp[2]*cp[1]*dp[2]*ep[2]*hp[1]- \\
 &64*ap[1]*bp[1]*cp[2]*dp[2]*ep[2]*hp[1]+192*ap[2]*bp[1]*bp[2]*cp[1]*ep[2]*hp[1]- \\
 &192*ap[1]*bp[2]*bp[2]*ep[1]*ep[2]*hp[1]- \\
 &256*ap[2]*cp[1]*cp[2]*ep[1]*ep[2]*hp[1]+256*ap[1]*cp[2]*cp[2]*ep[1]*ep[2]*hp[1]+128*ap[2]*ap \\
 &[2]*dp[1]*ep[1]*ep[2]*hp[1]-256*ap[1]*ap[2]*dp[2]*ep[1]*ep[2]*hp[1]- \\
 &192*ap[2]*bp[1]*bp[1]*ep[2]*ep[2]*hp[1]+192*ap[1]*bp[1]*bp[2]*ep[2]*ep[2]*hp[1]+256*ap[2]*c \\
 &p[1]*cp[1]*ep[2]*ep[2]*hp[1]-256*ap[1]*cp[1]*cp[2]*ep[2]*ep[2]*hp[1]- \\
 &128*ap[1]*ap[2]*dp[1]*ep[2]*ep[2]*hp[1]+192*ap[1]*ap[1]*dp[2]*ep[2]*ep[2]*hp[1]+24*ap[2]*bp[\\
 &1]*bp[2]*dp[2]*fp[1]*hp[1]-24*ap[1]*bp[2]*bp[2]*dp[2]*fp[1]*hp[1]- \\
 &80*ap[2]*cp[1]*cp[2]*dp[2]*fp[1]*hp[1]+80*ap[1]*cp[2]*cp[2]*dp[2]*fp[1]*hp[1]+112*ap[2]*ap[2] \\
 &*dp[1]*dp[2]*fp[1]*hp[1]- \\
 &112*ap[1]*ap[2]*dp[2]*dp[2]*fp[1]*hp[1]+224*ap[2]*bp[2]*cp[1]*ep[2]*fp[1]*hp[1]+224*ap[2]*bp \\
 &[1]*cp[2]*ep[2]*fp[1]*hp[1]-448*ap[1]*bp[2]*cp[2]*ep[2]*fp[1]*hp[1]- \\
 &896*ap[2]*ap[2]*ep[1]*ep[2]*fp[1]*hp[1]+896*ap[1]*ap[2]*ep[2]*ep[2]*fp[1]*hp[1]- \\
 &104*ap[2]*ap[2]*dp[2]*fp[1]*fp[1]*hp[1]+24*ap[2]*bp[1]*bp[2]*dp[1]*fp[2]*hp[1]- \\
 &24*ap[1]*bp[2]*bp[2]*dp[1]*fp[2]*hp[1]- \\
 &80*ap[2]*cp[1]*cp[2]*dp[1]*fp[2]*hp[1]+80*ap[1]*cp[2]*cp[2]*dp[1]*fp[2]*hp[1]+56*ap[2]*ap[2]* \\
 &dp[1]*dp[1]*fp[2]*hp[1]- \\
 &48*ap[2]*bp[1]*bp[1]*dp[2]*fp[2]*hp[1]+48*ap[1]*bp[1]*bp[2]*dp[2]*fp[2]*hp[1]+160*ap[2]*cp[1] \\
 &]*cp[1]*dp[2]*fp[2]*hp[1]-160*ap[1]*cp[1]*cp[2]*dp[2]*fp[2]*hp[1]- \\
 &224*ap[1]*ap[2]*dp[1]*dp[2]*fp[2]*hp[1]+168*ap[1]*ap[1]*dp[2]*dp[2]*fp[2]*hp[1]+224*ap[2]*bp \\
 &[2]*cp[1]*ep[1]*fp[2]*hp[1]+224*ap[2]*bp[1]*cp[2]*ep[1]*fp[2]*hp[1]- \\
 &448*ap[1]*bp[2]*cp[2]*ep[1]*fp[2]*hp[1]-448*ap[2]*ap[2]*ep[1]*ep[1]*fp[2]*hp[1]- \\
 &896*ap[2]*bp[1]*cp[1]*ep[2]*fp[2]*hp[1]+448*ap[1]*bp[2]*cp[1]*ep[2]*fp[2]*hp[1]+448*ap[1]*bp \\
 &[1]*cp[2]*ep[2]*fp[2]*hp[1]+1792*ap[1]*ap[2]*cp[1]*ep[2]*fp[2]*hp[1]-
 \end{aligned}$$

$1344*ap[1]*ap[1]*ep[2]*ep[2]*fp[2]*hp[1]-$
 $200*ap[2]*bp[1]*bp[2]*fp[1]*fp[2]*hp[1]+200*ap[1]*bp[2]*bp[2]*fp[1]*fp[2]*hp[1]+304*ap[2]*cp[1]*cp[2]*fp[1]*fp[2]*hp[1]-304*ap[1]*cp[2]*cp[2]*fp[1]*fp[2]*hp[1]-$
 $208*ap[2]*ap[2]*dp[1]*fp[1]*fp[2]*hp[1]+416*ap[1]*ap[2]*dp[2]*fp[1]*fp[2]*hp[1]+504*ap[2]*ap[2]*fp[1]*fp[1]*fp[2]*hp[1]+200*ap[2]*bp[1]*bp[1]*fp[2]*fp[2]*hp[1]-$
 $200*ap[1]*bp[1]*bp[2]*fp[2]*fp[2]*hp[1]-$
 $304*ap[2]*cp[1]*cp[1]*fp[2]*fp[2]*hp[1]+304*ap[1]*cp[1]*cp[2]*fp[2]*fp[2]*hp[1]+208*ap[1]*ap[2]*dp[1]*fp[2]*fp[2]*hp[1]-312*ap[1]*ap[1]*dp[2]*fp[2]*fp[2]*hp[1]-$
 $1008*ap[1]*ap[2]*fp[1]*fp[2]*fp[2]*hp[1]+504*ap[1]*ap[1]*fp[2]*fp[2]*fp[2]*hp[1]-$
 $16*ap[2]*bp[1]*bp[2]*bp[2]*gp[1]*hp[1]+16*ap[1]*bp[2]*bp[2]*bp[2]*gp[1]*hp[1]+64*ap[2]*bp[2]*cp[1]*cp[2]*gp[1]*hp[1]+32*ap[2]*bp[1]*cp[2]*cp[2]*gp[1]*hp[1]-$
 $96*ap[1]*bp[2]*cp[2]*cp[2]*gp[1]*hp[1]-16*ap[2]*ap[2]*bp[2]*dp[1]*gp[1]*hp[1]-$
 $32*ap[2]*ap[2]*bp[1]*dp[2]*gp[1]*hp[1]+48*ap[1]*ap[2]*bp[2]*dp[2]*gp[1]*hp[1]-$
 $64*ap[2]*ap[2]*cp[2]*ep[1]*gp[1]*hp[1]-$
 $128*ap[2]*ap[2]*cp[1]*ep[2]*gp[1]*hp[1]+192*ap[1]*ap[2]*cp[2]*ep[2]*gp[1]*hp[1]+48*ap[2]*ap[2]*bp[2]*fp[1]*gp[1]*hp[1]+96*ap[2]*ap[2]*bp[1]*fp[2]*gp[1]*hp[1]-$
 $144*ap[1]*ap[2]*bp[2]*fp[2]*gp[1]*hp[1]+24*ap[2]*ap[2]*ap[2]*gp[1]*gp[1]*hp[1]+16*ap[2]*bp[1]*bp[1]*bp[2]*gp[2]*hp[1]-16*ap[1]*bp[1]*bp[2]*bp[2]*gp[2]*hp[1]-$
 $32*ap[2]*bp[2]*cp[1]*cp[1]*gp[2]*hp[1]-$
 $64*ap[2]*bp[1]*cp[1]*cp[2]*gp[2]*hp[1]+64*ap[1]*bp[2]*cp[1]*cp[2]*gp[2]*hp[1]+32*ap[1]*bp[1]*cp[2]*cp[2]*gp[2]*hp[1]+24*ap[2]*ap[2]*bp[1]*dp[1]*gp[2]*hp[1]-$
 $8*ap[1]*ap[2]*bp[2]*dp[1]*gp[2]*hp[1]+8*ap[1]*ap[2]*bp[1]*dp[2]*gp[2]*hp[1]-$
 $24*ap[1]*ap[1]*bp[2]*dp[2]*gp[2]*hp[1]+96*ap[2]*ap[2]*cp[1]*ep[1]*gp[2]*hp[1]-$
 $32*ap[1]*ap[2]*cp[2]*ep[1]*gp[2]*hp[1]+32*ap[1]*ap[2]*cp[1]*ep[2]*gp[2]*hp[1]-$
 $96*ap[1]*ap[1]*cp[2]*ep[2]*gp[2]*hp[1]-$
 $72*ap[2]*ap[2]*bp[1]*fp[1]*gp[2]*hp[1]+24*ap[1]*ap[2]*bp[2]*fp[1]*gp[2]*hp[1]-$
 $24*ap[1]*ap[2]*bp[1]*fp[2]*gp[2]*hp[1]+72*ap[1]*ap[1]*bp[2]*fp[2]*gp[2]*hp[1]-$
 $48*ap[1]*ap[2]*ap[2]*gp[1]*gp[2]*hp[1]+24*ap[1]*ap[1]*ap[2]*gp[2]*gp[2]*hp[1]+16*ap[2]*bp[2]*bp[2]*cp[1]*hp[1]*hp[1]+32*ap[2]*bp[1]*bp[2]*cp[2]*hp[1]*hp[1]-$
 $48*ap[1]*bp[2]*bp[2]*cp[2]*hp[1]*hp[1]-$
 $64*ap[2]*cp[1]*cp[2]*cp[2]*hp[1]*hp[1]+64*ap[1]*cp[2]*cp[2]*cp[2]*hp[1]*hp[1]+48*ap[2]*ap[2]*cp[2]*dp[1]*hp[1]*hp[1]+96*ap[2]*ap[2]*cp[1]*dp[2]*hp[1]*hp[1]-$
 $144*ap[1]*ap[2]*cp[2]*dp[2]*hp[1]*hp[1]-32*ap[2]*ap[2]*bp[2]*ep[1]*hp[1]*hp[1]-$
 $64*ap[2]*ap[2]*bp[1]*ep[2]*hp[1]*hp[1]+96*ap[1]*ap[2]*bp[2]*ep[2]*hp[1]*hp[1]-$
 $80*ap[2]*ap[2]*cp[2]*fp[1]*hp[1]*hp[1]-$
 $160*ap[2]*ap[2]*cp[1]*fp[2]*hp[1]*hp[1]+240*ap[1]*ap[2]*cp[2]*fp[2]*hp[1]*hp[1]-$
 $48*ap[2]*ap[2]*ap[2]*hp[1]*hp[1]*hp[1]+8*ap[2]*bp[1]*bp[2]*dp[1]*dp[1]*hp[2]-$

$$\begin{aligned}
 &8*ap[1]*bp[2]*bp[2]*dp[1]*dp[1]*hp[2]- \\
 &48*ap[2]*cp[1]*cp[2]*dp[1]*dp[1]*hp[2]+48*ap[1]*cp[2]*cp[2]*dp[1]*dp[1]*hp[2]+40*ap[2]*ap[2] \\
 &*dp[1]*dp[1]*dp[1]*hp[2]- \\
 &8*ap[2]*bp[1]*bp[1]*dp[1]*dp[2]*hp[2]+8*ap[1]*bp[1]*bp[2]*dp[1]*dp[2]*hp[2]+48*ap[2]*cp[1]*c \\
 &p[1]*dp[1]*dp[2]*hp[2]-48*ap[1]*cp[1]*cp[2]*dp[1]*dp[2]*hp[2]- \\
 &80*ap[1]*ap[2]*dp[1]*dp[1]*dp[2]*hp[2]+40*ap[1]*ap[1]*dp[1]*dp[2]*dp[2]*hp[2]+64*ap[2]*bp[2] \\
 &*cp[1]*dp[1]*cp[1]*hp[2]+64*ap[2]*bp[1]*cp[2]*dp[1]*cp[1]*hp[2]- \\
 &128*ap[1]*bp[2]*cp[2]*dp[1]*ep[1]*hp[2]- \\
 &64*ap[2]*bp[1]*cp[1]*dp[2]*ep[1]*hp[2]+32*ap[1]*bp[2]*cp[1]*dp[2]*ep[1]*hp[2]+32*ap[1]*bp[1] \\
 &*cp[2]*dp[2]*cp[1]*hp[2]- \\
 &192*ap[2]*bp[1]*bp[2]*ep[1]*ep[1]*hp[2]+192*ap[1]*bp[2]*bp[2]*ep[1]*ep[1]*hp[2]+256*ap[2]*c \\
 &p[1]*cp[2]*ep[1]*cp[1]*hp[2]-256*ap[1]*cp[2]*cp[2]*ep[1]*ep[1]*hp[2]- \\
 &192*ap[2]*ap[2]*dp[1]*ep[1]*ep[1]*hp[2]+128*ap[1]*ap[2]*dp[2]*ep[1]*ep[1]*hp[2]- \\
 &64*ap[2]*bp[1]*cp[1]*dp[1]*ep[2]*hp[2]+32*ap[1]*bp[2]*cp[1]*dp[1]*cp[2]*hp[2]+32*ap[1]*bp[1] \\
 &*cp[2]*dp[1]*ep[2]*hp[2]+192*ap[2]*bp[1]*bp[1]*ep[1]*ep[2]*hp[2]- \\
 &192*ap[1]*bp[1]*bp[2]*ep[1]*ep[2]*hp[2]- \\
 &256*ap[2]*cp[1]*cp[1]*ep[1]*ep[2]*hp[2]+256*ap[1]*cp[1]*cp[2]*ep[1]*ep[2]*hp[2]+256*ap[1]*ap \\
 &[2]*dp[1]*ep[1]*ep[2]*hp[2]-128*ap[1]*ap[1]*dp[2]*ep[1]*ep[2]*hp[2]- \\
 &64*ap[1]*ap[1]*dp[1]*ep[2]*ep[2]*hp[2]- \\
 &48*ap[2]*bp[1]*bp[2]*dp[1]*fp[1]*hp[2]+48*ap[1]*bp[2]*bp[2]*dp[1]*fp[1]*hp[2]+160*ap[2]*cp[1] \\
 &]*cp[2]*dp[1]*fp[1]*hp[2]-160*ap[1]*cp[2]*cp[2]*dp[1]*fp[1]*hp[2]- \\
 &168*ap[2]*ap[2]*dp[1]*dp[1]*fp[1]*hp[2]+24*ap[2]*bp[1]*bp[1]*dp[2]*fp[1]*hp[2]- \\
 &24*ap[1]*bp[1]*bp[2]*dp[2]*fp[1]*hp[2]- \\
 &80*ap[2]*cp[1]*cp[1]*dp[2]*fp[1]*hp[2]+80*ap[1]*cp[1]*cp[2]*dp[2]*fp[1]*hp[2]+224*ap[1]*ap[2] \\
 &*dp[1]*dp[2]*fp[1]*hp[2]-56*ap[1]*ap[1]*dp[2]*dp[2]*fp[1]*hp[2]- \\
 &448*ap[2]*bp[2]*cp[1]*ep[1]*fp[1]*hp[2]- \\
 &448*ap[2]*bp[1]*cp[2]*ep[1]*fp[1]*hp[2]+896*ap[1]*bp[2]*cp[2]*ep[1]*fp[1]*hp[2]+1344*ap[2]*a \\
 &p[2]*ep[1]*ep[1]*fp[1]*hp[2]+448*ap[2]*bp[1]*cp[1]*ep[2]*fp[1]*hp[2]- \\
 &224*ap[1]*bp[2]*cp[1]*ep[2]*fp[1]*hp[2]-224*ap[1]*bp[1]*cp[2]*ep[2]*fp[1]*hp[2]- \\
 &1792*ap[1]*ap[2]*ep[1]*ep[2]*fp[1]*hp[2]+448*ap[1]*ap[1]*ep[2]*ep[2]*fp[1]*hp[2]+200*ap[2]*b \\
 &p[1]*bp[2]*fp[1]*fp[1]*hp[2]-200*ap[1]*bp[2]*bp[2]*fp[1]*fp[1]*hp[2]- \\
 &304*ap[2]*cp[1]*cp[2]*fp[1]*fp[1]*hp[2]+304*ap[1]*cp[2]*cp[2]*fp[1]*fp[1]*hp[2]+312*ap[2]*ap[\\
 &2]*dp[1]*fp[1]*fp[1]*hp[2]-208*ap[1]*ap[2]*dp[2]*fp[1]*fp[1]*hp[2]- \\
 &504*ap[2]*ap[2]*fp[1]*fp[1]*fp[1]*hp[2]+24*ap[2]*bp[1]*bp[1]*dp[1]*fp[2]*hp[2]- \\
 &24*ap[1]*bp[1]*bp[2]*dp[1]*fp[2]*hp[2]- \\
 &80*ap[2]*cp[1]*cp[1]*dp[1]*fp[2]*hp[2]+80*ap[1]*cp[1]*cp[2]*dp[1]*fp[2]*hp[2]+112*ap[1]*ap[2] \\
 &*dp[1]*dp[1]*fp[2]*hp[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 112*ap[1]*ap[1]*dp[1]*dp[2]*fp[2]*hp[2]+448*ap[2]*bp[1]*cp[1]*ep[1]*fp[2]*hp[2]- \\
 & 224*ap[1]*bp[2]*cp[1]*ep[1]*fp[2]*hp[2]-224*ap[1]*bp[1]*cp[2]*ep[1]*fp[2]*hp[2]- \\
 & 896*ap[1]*ap[2]*ep[1]*ep[1]*fp[2]*hp[2]+896*ap[1]*ap[1]*ep[1]*ep[2]*fp[2]*hp[2]- \\
 & 200*ap[2]*bp[1]*bp[1]*fp[1]*fp[2]*hp[2]+200*ap[1]*bp[1]*bp[2]*fp[1]*fp[2]*hp[2]+304*ap[2]*cp[\\
 & 1]*cp[1]*fp[1]*fp[2]*hp[2]-304*ap[1]*cp[1]*cp[2]*fp[1]*fp[2]*hp[2]- \\
 & 416*ap[1]*ap[2]*dp[1]*fp[1]*fp[2]*hp[2]+208*ap[1]*ap[1]*dp[2]*fp[1]*fp[2]*hp[2]+1008*ap[1]*a \\
 & p[2]*fp[1]*fp[1]*fp[2]*hp[2]+104*ap[1]*ap[1]*dp[1]*fp[2]*fp[2]*hp[2]- \\
 & 504*ap[1]*ap[1]*fp[1]*fp[2]*fp[2]*hp[2]+16*ap[2]*bp[1]*bp[1]*bp[2]*gp[1]*hp[2]- \\
 & 16*ap[1]*bp[1]*bp[2]*bp[2]*gp[1]*hp[2]-32*ap[2]*bp[2]*cp[1]*cp[1]*gp[1]*hp[2]- \\
 & 64*ap[2]*bp[1]*cp[1]*cp[2]*gp[1]*hp[2]+64*ap[1]*bp[2]*cp[1]*cp[2]*gp[1]*hp[2]+32*ap[1]*bp[1] \\
 & *cp[2]*cp[2]*gp[1]*hp[2]+24*ap[2]*ap[2]*bp[1]*dp[1]*gp[1]*hp[2]- \\
 & 8*ap[1]*ap[2]*bp[2]*dp[1]*gp[1]*hp[2]+8*ap[1]*ap[2]*bp[1]*dp[2]*gp[1]*hp[2]- \\
 & 24*ap[1]*ap[1]*bp[2]*dp[2]*gp[1]*hp[2]+96*ap[2]*ap[2]*cp[1]*ep[1]*gp[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*cp[2]*ep[1]*gp[1]*hp[2]+32*ap[1]*ap[2]*cp[1]*ep[2]*gp[1]*hp[2]- \\
 & 96*ap[1]*ap[1]*cp[2]*ep[2]*gp[1]*hp[2]- \\
 & 72*ap[2]*ap[2]*bp[1]*fp[1]*gp[1]*hp[2]+24*ap[1]*ap[2]*bp[2]*fp[1]*gp[1]*hp[2]- \\
 & 24*ap[1]*ap[2]*bp[1]*fp[2]*gp[1]*hp[2]+72*ap[1]*ap[1]*bp[2]*fp[2]*gp[1]*hp[2]- \\
 & 24*ap[1]*ap[2]*ap[2]*gp[1]*gp[1]*hp[2]- \\
 & 16*ap[2]*bp[1]*bp[1]*bp[1]*gp[2]*hp[2]+16*ap[1]*bp[1]*bp[1]*bp[2]*gp[2]*hp[2]+96*ap[2]*bp[1] \\
 &]*cp[1]*cp[1]*gp[2]*hp[2]-32*ap[1]*bp[2]*cp[1]*cp[1]*gp[2]*hp[2]- \\
 & 64*ap[1]*bp[1]*cp[1]*cp[2]*gp[2]*hp[2]- \\
 & 48*ap[1]*ap[2]*bp[1]*dp[1]*gp[2]*hp[2]+32*ap[1]*ap[1]*bp[2]*dp[1]*gp[2]*hp[2]+16*ap[1]*ap[1] \\
 & *bp[1]*dp[2]*gp[2]*hp[2]- \\
 & 192*ap[1]*ap[2]*cp[1]*ep[1]*gp[2]*hp[2]+128*ap[1]*ap[1]*cp[2]*ep[1]*gp[2]*hp[2]+64*ap[1]*ap[\\
 & 1]*cp[1]*ep[2]*gp[2]*hp[2]+144*ap[1]*ap[2]*bp[1]*fp[1]*gp[2]*hp[2]- \\
 & 96*ap[1]*ap[1]*bp[2]*fp[1]*gp[2]*hp[2]- \\
 & 48*ap[1]*ap[1]*bp[1]*fp[2]*gp[2]*hp[2]+48*ap[1]*ap[1]*ap[2]*gp[1]*gp[2]*hp[2]- \\
 & 24*ap[1]*ap[1]*ap[1]*gp[2]*gp[2]*hp[2]- \\
 & 64*ap[2]*bp[1]*bp[2]*cp[1]*hp[1]*hp[2]+32*ap[1]*bp[2]*bp[2]*cp[1]*hp[1]*hp[2]- \\
 & 32*ap[2]*bp[1]*bp[1]*cp[2]*hp[1]*hp[2]+64*ap[1]*bp[1]*bp[2]*cp[2]*hp[1]*hp[2]+128*ap[2]*cp[\\
 & 1]*cp[1]*cp[2]*hp[1]*hp[2]-128*ap[1]*cp[1]*cp[2]*cp[2]*hp[1]*hp[2]- \\
 & 144*ap[2]*ap[2]*cp[1]*dp[1]*hp[1]*hp[2]+48*ap[1]*ap[2]*cp[2]*dp[1]*hp[1]*hp[2]- \\
 & 48*ap[1]*ap[2]*cp[1]*dp[2]*hp[1]*hp[2]+144*ap[1]*ap[1]*cp[2]*dp[2]*hp[1]*hp[2]+96*ap[2]*ap[2] \\
 &]*bp[1]*ep[1]*hp[1]*hp[2]- \\
 & 32*ap[1]*ap[2]*bp[2]*ep[1]*hp[1]*hp[2]+32*ap[1]*ap[2]*bp[1]*ep[2]*hp[1]*hp[2]- \\
 & 96*ap[1]*ap[1]*bp[2]*ep[2]*hp[1]*hp[2]+240*ap[2]*ap[2]*cp[1]*fp[1]*hp[1]*hp[2]- \\
 & 80*ap[1]*ap[2]*cp[2]*fp[1]*hp[1]*hp[2]+80*ap[1]*ap[2]*cp[1]*fp[2]*hp[1]*hp[2]-
 \end{aligned}$$

$$\begin{aligned} & 240*ap[1]*ap[1]*cp[2]*fp[2]*hp[1]*hp[2]+144*ap[1]*ap[2]*ap[2]*hp[1]*hp[1]*hp[2]+48*ap[2]*bp[1]*bp[1]*cp[1]*hp[2]*hp[2]-32*ap[1]*bp[1]*bp[2]*cp[1]*hp[2]*hp[2]- \\ & 64*ap[2]*cp[1]*cp[1]*cp[1]*hp[2]*hp[2]- \\ & 16*ap[1]*bp[1]*bp[1]*cp[2]*hp[2]*hp[2]+64*ap[1]*cp[1]*cp[1]*cp[2]*hp[2]*hp[2]+144*ap[1]*ap[2]*cp[1]*dp[1]*hp[2]*hp[2]-96*ap[1]*ap[1]*cp[2]*dp[1]*hp[2]*hp[2]- \\ & 48*ap[1]*ap[1]*cp[1]*dp[2]*hp[2]*hp[2]- \\ & 96*ap[1]*ap[2]*bp[1]*ep[1]*hp[2]*hp[2]+64*ap[1]*ap[1]*bp[2]*ep[1]*hp[2]*hp[2]+32*ap[1]*ap[1]*bp[1]*ep[2]*hp[2]*hp[2]- \\ & 240*ap[1]*ap[2]*cp[1]*fp[1]*hp[2]*hp[2]+160*ap[1]*ap[1]*cp[2]*fp[1]*hp[2]*hp[2]+80*ap[1]*ap[1]*cp[1]*fp[2]*hp[2]*hp[2]- \\ & 144*ap[1]*ap[1]*ap[2]*hp[1]*hp[2]*hp[2]+48*ap[1]*ap[1]*ap[1]*hp[2]*hp[2]*hp[2] \end{aligned}$$



EK 5. Yörünge Sentezi Modülü Denklem (3.7) Katsayıları

$$\begin{aligned} pa_{pk} = & -2*a[k]*c[3]*d[2]*f[1]+2*a[3]*c[k]*d[2]*f[1]+2*a[k]*c[2]*d[3]*f[1]-2*a[2]*c[k]*d[3]*f[1]- \\ & 2*a[3]*c[2]*d[k]*f[1]+2*a[2]*c[3]*d[k]*f[1]+2*a[k]*c[3]*d[1]*f[2]-2*a[3]*c[k]*d[1]*f[2]- \\ & 2*a[k]*c[1]*d[3]*f[2]+2*a[1]*c[k]*d[3]*f[2]+2*a[3]*c[1]*d[k]*f[2]-2*a[1]*c[3]*d[k]*f[2]- \\ & 2*a[k]*c[2]*d[1]*f[3]+2*a[2]*c[k]*d[1]*f[3]+2*a[k]*c[1]*d[2]*f[3]-2*a[1]*c[k]*d[2]*f[3]- \\ & 2*a[2]*c[1]*d[k]*f[3]+2*a[1]*c[2]*d[k]*f[3]+2*a[3]*c[2]*d[1]*f[k]-2*a[2]*c[3]*d[1]*f[k]- \\ & 2*a[3]*c[1]*d[2]*f[k]+2*a[1]*c[3]*d[2]*f[k]+2*a[2]*c[1]*d[3]*f[k]-2*a[1]*c[2]*d[3]*f[k] \end{aligned}$$

$$\begin{aligned} pb_{pk} = & -2*a[k]*d[3]*f[2]*g[1]+2*a[3]*d[k]*f[2]*g[1]+2*a[k]*d[2]*f[3]*g[1]-2*a[2]*d[k]*f[3]*g[1]- \\ & 2*a[3]*d[2]*f[k]*g[1]+2*a[2]*d[3]*f[k]*g[1]+2*a[k]*d[3]*f[1]*g[2]-2*a[3]*d[k]*f[1]*g[2]- \\ & 2*a[k]*d[1]*f[3]*g[2]+2*a[1]*d[k]*f[3]*g[2]+2*a[3]*d[1]*f[k]*g[2]-2*a[1]*d[3]*f[k]*g[2]- \\ & 2*a[k]*d[2]*f[1]*g[3]+2*a[2]*d[k]*f[1]*g[3]+2*a[k]*d[1]*f[2]*g[3]-2*a[1]*d[k]*f[2]*g[3]- \\ & 2*a[2]*d[1]*f[k]*g[3]+2*a[1]*d[2]*f[k]*g[3]+2*a[3]*d[2]*f[1]*g[k]-2*a[2]*d[3]*f[1]*g[k]- \\ & 2*a[3]*d[1]*f[2]*g[k]+2*a[1]*d[3]*f[2]*g[k]+2*a[2]*d[1]*f[3]*g[k]-2*a[1]*d[2]*f[3]*g[k] \end{aligned}$$

$$\begin{aligned} pc_{pk} = & 2*a[k]*b[3]*c[2]*e[1]-2*a[3]*b[k]*c[2]*e[1]- \\ & 2*a[k]*b[2]*c[3]*e[1]+2*a[2]*b[k]*c[3]*e[1]+2*a[3]*b[2]*c[k]*e[1]-2*a[2]*b[3]*c[k]*e[1]- \\ & 2*a[k]*b[3]*c[1]*e[2]+2*a[3]*b[k]*c[1]*e[2]+2*a[k]*b[1]*c[3]*e[2]-2*a[1]*b[k]*c[3]*e[2]- \\ & 2*a[3]*b[1]*c[k]*e[2]+2*a[1]*b[3]*c[k]*e[2]+2*a[k]*b[2]*c[1]*e[3]-2*a[2]*b[k]*c[1]*e[3]- \\ & 2*a[k]*b[1]*c[2]*e[3]+2*a[1]*b[k]*c[2]*e[3]+2*a[2]*b[1]*c[k]*e[3]-2*a[1]*b[2]*c[k]*e[3]- \\ & 2*a[3]*b[2]*c[1]*e[k]+2*a[2]*b[3]*c[1]*e[k]+2*a[3]*b[1]*c[2]*e[k]-2*a[1]*b[3]*c[2]*e[k]- \\ & 2*a[2]*b[1]*c[3]*e[k]+2*a[1]*b[2]*c[3]*e[k] \end{aligned}$$

$$\begin{aligned} pd_{pk} = & -2*a[k]*b[3]*e[2]*g[1]+2*a[3]*b[k]*e[2]*g[1]+2*a[k]*b[2]*e[3]*g[1]-2*a[2]*b[k]*e[3]*g[1]- \\ & 2*a[3]*b[2]*e[k]*g[1]+2*a[2]*b[3]*e[k]*g[1]+2*a[k]*b[3]*e[1]*g[2]-2*a[3]*b[k]*e[1]*g[2]- \\ & 2*a[k]*b[1]*e[3]*g[2]+2*a[1]*b[k]*e[3]*g[2]+2*a[3]*b[1]*e[k]*g[2]-2*a[1]*b[3]*e[k]*g[2]- \\ & 2*a[k]*b[2]*e[1]*g[3]+2*a[2]*b[k]*e[1]*g[3]+2*a[k]*b[1]*e[2]*g[3]-2*a[1]*b[k]*e[2]*g[3]- \\ & 2*a[2]*b[1]*e[k]*g[3]+2*a[1]*b[2]*e[k]*g[3]+2*a[3]*b[2]*e[1]*g[k]-2*a[2]*b[3]*e[1]*g[k]- \\ & 2*a[3]*b[1]*e[2]*g[k]+2*a[1]*b[3]*e[2]*g[k]+2*a[2]*b[1]*e[3]*g[k]-2*a[1]*b[2]*e[3]*g[k] \end{aligned}$$

$$\begin{aligned} pe_{pk} = & -(a[k]*c[3]*d[2]*e[1]+a[3]*c[k]*d[2]*e[1]+a[k]*c[2]*d[3]*e[1]-a[2]*c[k]*d[3]*e[1]- \\ & a[3]*c[2]*d[k]*e[1]+a[2]*c[3]*d[k]*e[1]+a[k]*c[3]*d[1]*e[2]-a[3]*c[k]*d[1]*e[2]- \\ & a[k]*c[1]*d[3]*e[2]+a[1]*c[k]*d[3]*e[2]+a[3]*c[1]*d[k]*e[2]-a[1]*c[3]*d[k]*e[2]- \\ & a[k]*c[2]*d[1]*e[3]+a[2]*c[k]*d[1]*e[3]+a[k]*c[1]*d[2]*e[3]-a[1]*c[k]*d[2]*e[3]- \\ & a[2]*c[1]*d[k]*e[3]+a[1]*c[2]*d[k]*e[3]+a[3]*c[2]*d[1]*e[k]-a[2]*c[3]*d[1]*e[k]- \\ & a[3]*c[1]*d[2]*e[k]+a[1]*c[3]*d[2]*e[k]+a[2]*c[1]*d[3]*e[k]- \\ & a[1]*c[2]*d[3]*e[k]+a[k]*b[3]*c[2]*f[1]-a[3]*b[k]*c[2]*f[1]- \end{aligned}$$

$$\begin{aligned}
 & a[k]*b[2]*c[3]*f[1]+a[2]*b[k]*c[3]*f[1]+a[3]*b[2]*c[k]*f[1]-a[2]*b[3]*c[k]*f[1]- \\
 & a[k]*b[3]*c[1]*f[2]+a[3]*b[k]*c[1]*f[2]+a[k]*b[1]*c[3]*f[2]-a[1]*b[k]*c[3]*f[2]- \\
 & a[3]*b[1]*c[k]*f[2]+a[1]*b[3]*c[k]*f[2]+a[k]*b[2]*c[1]*f[3]-a[2]*b[k]*c[1]*f[3]- \\
 & a[k]*b[1]*c[2]*f[3]+a[1]*b[k]*c[2]*f[3]+a[2]*b[1]*c[k]*f[3]-a[1]*b[2]*c[k]*f[3]- \\
 & a[3]*b[2]*c[1]*f[k]+a[2]*b[3]*c[1]*f[k]+a[3]*b[1]*c[2]*f[k]-a[1]*b[3]*c[2]*f[k]- \\
 & a[2]*b[1]*c[3]*f[k]+a[1]*b[2]*c[3]*f[k]- \\
 & a[k]*b[3]*d[2]*g[1]+a[3]*b[k]*d[2]*g[1]+a[k]*b[2]*d[3]*g[1]-a[2]*b[k]*d[3]*g[1]- \\
 & a[3]*b[2]*d[k]*g[1]+a[2]*b[3]*d[k]*g[1]- \\
 & a[k]*e[3]*f[2]*g[1]+a[3]*e[k]*f[2]*g[1]+a[k]*e[2]*f[3]*g[1]-a[2]*e[k]*f[3]*g[1]- \\
 & a[3]*e[2]*f[k]*g[1]+a[2]*e[3]*f[k]*g[1]+a[k]*b[3]*d[1]*g[2]-a[3]*b[k]*d[1]*g[2]- \\
 & a[k]*b[1]*d[3]*g[2]+a[1]*b[k]*d[3]*g[2]+a[3]*b[1]*d[k]*g[2]- \\
 & a[1]*b[3]*d[k]*g[2]+a[k]*e[3]*f[1]*g[2]-a[3]*e[k]*f[1]*g[2]- \\
 & a[k]*e[1]*f[3]*g[2]+a[1]*e[k]*f[3]*g[2]+a[3]*e[1]*f[k]*g[2]-a[1]*e[3]*f[k]*g[2]- \\
 & a[k]*b[2]*d[1]*g[3]+a[2]*b[k]*d[1]*g[3]+a[k]*b[1]*d[2]*g[3]-a[1]*b[k]*d[2]*g[3]- \\
 & a[2]*b[1]*d[k]*g[3]+a[1]*b[2]*d[k]*g[3]- \\
 & a[k]*e[2]*f[1]*g[3]+a[2]*e[k]*f[1]*g[3]+a[k]*e[1]*f[2]*g[3]-a[1]*e[k]*f[2]*g[3]- \\
 & a[2]*e[1]*f[k]*g[3]+a[1]*e[2]*f[k]*g[3]+a[3]*b[2]*d[1]*g[k]-a[2]*b[3]*d[1]*g[k]- \\
 & a[3]*b[1]*d[2]*g[k]+a[1]*b[3]*d[2]*g[k]+a[2]*b[1]*d[3]*g[k]- \\
 & a[1]*b[2]*d[3]*g[k]+a[3]*e[2]*f[1]*g[k]-a[2]*e[3]*f[1]*g[k]- \\
 & a[3]*e[1]*f[2]*g[k]+a[1]*e[3]*f[2]*g[k]+a[2]*e[1]*f[3]*g[k]-a[1]*e[2]*f[3]*g[k]
 \end{aligned}$$

$$\begin{aligned}
 pf_{pk} &= a[k]*b[3]*c[2]*d[1]-a[3]*b[k]*c[2]*d[1]- \\
 & a[k]*b[2]*c[3]*d[1]+a[2]*b[k]*c[3]*d[1]+a[3]*b[2]*c[k]*d[1]-a[2]*b[3]*c[k]*d[1]- \\
 & a[k]*b[3]*c[1]*d[2]+a[3]*b[k]*c[1]*d[2]+a[k]*b[1]*c[3]*d[2]-a[1]*b[k]*c[3]*d[2]- \\
 & a[3]*b[1]*c[k]*d[2]+a[1]*b[3]*c[k]*d[2]+a[k]*b[2]*c[1]*d[3]-a[2]*b[k]*c[1]*d[3]- \\
 & a[k]*b[1]*c[2]*d[3]+a[1]*b[k]*c[2]*d[3]+a[2]*b[1]*c[k]*d[3]-a[1]*b[2]*c[k]*d[3]- \\
 & a[3]*b[2]*c[1]*d[k]+a[2]*b[3]*c[1]*d[k]+a[3]*b[1]*c[2]*d[k]-a[1]*b[3]*c[2]*d[k]- \\
 & a[2]*b[1]*c[3]*d[k]+a[1]*b[2]*c[3]*d[k]- \\
 & a[k]*c[3]*e[2]*f[1]+a[3]*c[k]*e[2]*f[1]+a[k]*c[2]*e[3]*f[1]-a[2]*c[k]*e[3]*f[1]- \\
 & a[3]*c[2]*e[k]*f[1]+a[2]*c[3]*e[k]*f[1]+a[k]*c[3]*e[1]*f[2]-a[3]*c[k]*e[1]*f[2]- \\
 & a[k]*c[1]*e[3]*f[2]+a[1]*c[k]*e[3]*f[2]+a[3]*c[1]*e[k]*f[2]-a[1]*c[3]*e[k]*f[2]- \\
 & a[k]*c[2]*e[1]*f[3]+a[2]*c[k]*e[1]*f[3]+a[k]*c[1]*e[2]*f[3]-a[1]*c[k]*e[2]*f[3]- \\
 & a[2]*c[1]*e[k]*f[3]+a[1]*c[2]*e[k]*f[3]+a[3]*c[2]*e[1]*f[k]-a[2]*c[3]*e[1]*f[k]- \\
 & a[3]*c[1]*e[2]*f[k]+a[1]*c[3]*e[2]*f[k]+a[2]*c[1]*e[3]*f[k]-a[1]*c[2]*e[3]*f[k]- \\
 & a[k]*d[3]*e[2]*g[1]+a[3]*d[k]*e[2]*g[1]+a[k]*d[2]*e[3]*g[1]-a[2]*d[k]*e[3]*g[1]- \\
 & a[3]*d[2]*e[k]*g[1]+a[2]*d[3]*e[k]*g[1]- \\
 & a[k]*b[3]*f[2]*g[1]+a[3]*b[k]*f[2]*g[1]+a[k]*b[2]*f[3]*g[1]-a[2]*b[k]*f[3]*g[1]-
 \end{aligned}$$

$$\begin{aligned}
 & a[3]*b[2]*f[k]*g[1]+a[2]*b[3]*f[k]*g[1]+a[k]*d[3]*e[1]*g[2]-a[3]*d[k]*e[1]*g[2]- \\
 & a[k]*d[1]*e[3]*g[2]+a[1]*d[k]*e[3]*g[2]+a[3]*d[1]*e[k]*g[2]- \\
 & a[1]*d[3]*e[k]*g[2]+a[k]*b[3]*f[1]*g[2]-a[3]*b[k]*f[1]*g[2]- \\
 & a[k]*b[1]*f[3]*g[2]+a[1]*b[k]*f[3]*g[2]+a[3]*b[1]*f[k]*g[2]-a[1]*b[3]*f[k]*g[2]- \\
 & a[k]*d[2]*e[1]*g[3]+a[2]*d[k]*e[1]*g[3]+a[k]*d[1]*e[2]*g[3]-a[1]*d[k]*e[2]*g[3]- \\
 & a[2]*d[1]*e[k]*g[3]+a[1]*d[2]*e[k]*g[3]- \\
 & a[k]*b[2]*f[1]*g[3]+a[2]*b[k]*f[1]*g[3]+a[k]*b[1]*f[2]*g[3]-a[1]*b[k]*f[2]*g[3]- \\
 & a[2]*b[1]*f[k]*g[3]+a[1]*b[2]*f[k]*g[3]+a[3]*d[2]*e[1]*g[k]-a[2]*d[3]*e[1]*g[k]- \\
 & a[3]*d[1]*e[2]*g[k]+a[1]*d[3]*e[2]*g[k]+a[2]*d[1]*e[3]*g[k]- \\
 & a[1]*d[2]*e[3]*g[k]+a[3]*b[2]*f[1]*g[k]-a[2]*b[3]*f[1]*g[k]- \\
 & a[3]*b[1]*f[2]*g[k]+a[1]*b[3]*f[2]*g[k]+a[2]*b[1]*f[3]*g[k]-a[1]*b[2]*f[3]*g[k]
 \end{aligned}$$

$$\begin{aligned}
 pg_{pk} = & -(a[k]*c[3]*d[2]*e[1]+a[3]*c[k]*d[2]*e[1]+a[k]*c[2]*d[3]*e[1]-a[2]*c[k]*d[3]*e[1]- \\
 & a[3]*c[2]*d[k]*e[1]+a[2]*c[3]*d[k]*e[1]+a[k]*c[3]*d[1]*e[2]-a[3]*c[k]*d[1]*e[2]- \\
 & a[k]*c[1]*d[3]*e[2]+a[1]*c[k]*d[3]*e[2]+a[3]*c[1]*d[k]*e[2]-a[1]*c[3]*d[k]*e[2]- \\
 & a[k]*c[2]*d[1]*e[3]+a[2]*c[k]*d[1]*e[3]+a[k]*c[1]*d[2]*e[3]-a[1]*c[k]*d[2]*e[3]- \\
 & a[2]*c[1]*d[k]*e[3]+a[1]*c[2]*d[k]*e[3]+a[3]*c[2]*d[1]*e[k]-a[2]*c[3]*d[1]*e[k]- \\
 & a[3]*c[1]*d[2]*e[k]+a[1]*c[3]*d[2]*e[k]+a[2]*c[1]*d[3]*e[k]- \\
 & a[1]*c[2]*d[3]*e[k]+a[k]*b[3]*c[2]*f[1]-a[3]*b[k]*c[2]*f[1]- \\
 & a[k]*b[2]*c[3]*f[1]+a[2]*b[k]*c[3]*f[1]+a[3]*b[2]*c[k]*f[1]-a[2]*b[3]*c[k]*f[1]- \\
 & a[k]*b[3]*c[1]*f[2]+a[3]*b[k]*c[1]*f[2]+a[k]*b[1]*c[3]*f[2]-a[1]*b[k]*c[3]*f[2]- \\
 & a[3]*b[1]*c[k]*f[2]+a[1]*b[3]*c[k]*f[2]+a[k]*b[2]*c[1]*f[3]-a[2]*b[k]*c[1]*f[3]- \\
 & a[k]*b[1]*c[2]*f[3]+a[1]*b[k]*c[2]*f[3]+a[2]*b[1]*c[k]*f[3]-a[1]*b[2]*c[k]*f[3]- \\
 & a[3]*b[2]*c[1]*f[k]+a[2]*b[3]*c[1]*f[k]+a[3]*b[1]*c[2]*f[k]-a[1]*b[3]*c[2]*f[k]- \\
 & a[2]*b[1]*c[3]*f[k]+a[1]*b[2]*c[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]*e[3]*f[2]*g[1]-a[3]*e[k]*f[2]*g[1]- \\
 & a[k]*e[2]*f[3]*g[1]+a[2]*e[k]*f[3]*g[1]+a[3]*e[2]*f[k]*g[1]-a[2]*e[3]*f[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]*e[3]*f[1]*g[2]+a[3]*e[k]*f[1]*g[2]+a[k]*e[1]*f[3]*g[2]-a[1]*e[k]*f[3]*g[2]- \\
 & a[3]*e[1]*f[k]*g[2]+a[1]*e[3]*f[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]*e[2]*f[1]*g[3]-a[2]*e[k]*f[1]*g[3]- \\
 & a[k]*e[1]*f[2]*g[3]+a[1]*e[k]*f[2]*g[3]+a[2]*e[1]*f[k]*g[3]-a[1]*e[2]*f[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]-
 \end{aligned}$$

$$a[3]*e[2]*f[1]*g[k]+a[2]*e[3]*f[1]*g[k]+a[3]*e[1]*f[2]*g[k]-a[1]*e[3]*f[2]*g[k]-$$
$$a[2]*e[1]*f[3]*g[k]+a[1]*e[2]*f[3]*g[k]$$

$$i=1, 2$$

$$k=i+3$$



EK 6. Yörünge Sentezi Modülü Denklem (3.7) Katsayıları

$$q_0 = ap[2]*ap[2]*cp[1]*cp[1] -$$

$$2*ap[1]*ap[2]*cp[1]*cp[2] + ap[1]*ap[1]*cp[2]*cp[2] + ap[2]*cp[2]*ep[1]*ep[1] -$$

$$ap[2]*cp[1]*ep[1]*ep[2] -$$

$$ap[1]*cp[2]*ep[1]*ep[2] + ap[1]*cp[1]*ep[2]*ep[2] + 2*ap[2]*cp[2]*ep[1]*gp[1] -$$

$$ap[2]*cp[1]*ep[2]*gp[1] - ap[1]*cp[2]*ep[2]*gp[1] + ap[2]*cp[2]*gp[1]*gp[1] -$$

$$ap[2]*cp[1]*ep[1]*gp[2] - ap[1]*cp[2]*ep[1]*gp[2] + 2*ap[1]*cp[1]*ep[2]*gp[2] -$$

$$ap[2]*cp[1]*gp[1]*gp[2] - ap[1]*cp[2]*gp[1]*gp[2] + ap[1]*cp[1]*gp[2]*gp[2]$$

$$q_1 = 4*ap[2]*bp[2]*cp[1]*cp[1] - 4*ap[2]*bp[1]*cp[1]*cp[2] -$$

$$4*ap[1]*bp[2]*cp[1]*cp[2] + 4*ap[1]*bp[1]*cp[2]*cp[2] + 4*ap[2]*ap[2]*cp[1]*dp[1] -$$

$$4*ap[1]*ap[2]*cp[2]*dp[1] -$$

$$4*ap[1]*ap[2]*cp[1]*dp[2] + 4*ap[1]*ap[1]*cp[2]*dp[2] + 2*bp[2]*cp[2]*ep[1]*ep[1] + 2*ap[2]*dp[2]*$$

$$ep[1]*ep[1] - 2*bp[2]*cp[1]*ep[1]*ep[2] - 2*bp[1]*cp[2]*ep[1]*ep[2] - 2*ap[2]*dp[1]*ep[1]*ep[2] -$$

$$2*ap[1]*dp[2]*ep[1]*ep[2] + 2*bp[1]*cp[1]*ep[2]*ep[2] + 2*ap[1]*dp[1]*ep[2]*ep[2] + 8*ap[2]*cp[2]*$$

$$ep[1]*fp[1] - 4*ap[2]*cp[1]*ep[2]*fp[1] - 4*ap[1]*cp[2]*ep[2]*fp[1] - 4*ap[2]*cp[1]*ep[1]*fp[2] -$$

$$4*ap[1]*cp[2]*ep[1]*fp[2] + 8*ap[1]*cp[1]*ep[2]*fp[2] + 4*bp[2]*cp[2]*ep[1]*gp[1] + 4*ap[2]*dp[2]*e$$

$$p[1]*gp[1] - 2*bp[2]*cp[1]*ep[2]*gp[1] - 2*bp[1]*cp[2]*ep[2]*gp[1] - 2*ap[2]*dp[1]*ep[2]*gp[1] -$$

$$2*ap[1]*dp[2]*ep[2]*gp[1] + 8*ap[2]*cp[2]*fp[1]*gp[1] - 4*ap[2]*cp[1]*fp[2]*gp[1] -$$

$$4*ap[1]*cp[2]*fp[2]*gp[1] + 2*bp[2]*cp[2]*gp[1]*gp[1] + 2*ap[2]*dp[2]*gp[1]*gp[1] -$$

$$2*bp[2]*cp[1]*ep[1]*gp[2] - 2*bp[1]*cp[2]*ep[1]*gp[2] - 2*ap[2]*dp[1]*ep[1]*gp[2] -$$

$$2*ap[1]*dp[2]*ep[1]*gp[2] + 4*bp[1]*cp[1]*ep[2]*gp[2] + 4*ap[1]*dp[1]*ep[2]*gp[2] -$$

$$4*ap[2]*cp[1]*fp[1]*gp[2] - 4*ap[1]*cp[2]*fp[1]*gp[2] + 8*ap[1]*cp[1]*fp[2]*gp[2] -$$

$$2*bp[2]*cp[1]*gp[1]*gp[2] - 2*bp[1]*cp[2]*gp[1]*gp[2] - 2*ap[2]*dp[1]*gp[1]*gp[2] -$$

$$2*ap[1]*dp[2]*gp[1]*gp[2] + 2*bp[1]*cp[1]*gp[2]*gp[2] + 2*ap[1]*dp[1]*gp[2]*gp[2]$$

$$q_2 = -2*ap[2]*ap[2]*cp[1]*cp[1] + 4*bp[2]*bp[2]*cp[1]*cp[1] + 4*ap[1]*ap[2]*cp[1]*cp[2] -$$

$$8*bp[1]*bp[2]*cp[1]*cp[2] -$$

$$2*ap[1]*ap[1]*cp[2]*cp[2] + 4*bp[1]*bp[1]*cp[2]*cp[2] + 16*ap[2]*bp[2]*cp[1]*dp[1] -$$

$$8*ap[2]*bp[1]*cp[2]*dp[1] - 8*ap[1]*bp[2]*cp[2]*dp[1] + 4*ap[2]*ap[2]*dp[1]*dp[1] -$$

$$8*ap[2]*bp[1]*cp[1]*dp[2] - 8*ap[1]*bp[2]*cp[1]*dp[2] + 16*ap[1]*bp[1]*cp[2]*dp[2] -$$

$$8*ap[1]*ap[2]*dp[1]*dp[2] + 4*ap[1]*ap[1]*dp[2]*dp[2] + 2*ap[2]*cp[2]*ep[1]*ep[1] + 4*bp[2]*dp[2]*$$

$$ep[1]*ep[1] - 2*ap[2]*cp[1]*ep[1]*ep[2] - 2*ap[1]*cp[2]*ep[1]*ep[2] - 4*bp[2]*dp[1]*ep[1]*ep[2] -$$

$$4*bp[1]*dp[2]*ep[1]*ep[2] + 2*ap[1]*cp[1]*ep[2]*ep[2] + 4*bp[1]*dp[1]*ep[2]*ep[2] + 16*bp[2]*cp[2]$$

$$*ep[1]*fp[1] + 16*ap[2]*dp[2]*ep[1]*fp[1] - 8*bp[2]*cp[1]*ep[2]*fp[1] - 8*bp[1]*cp[2]*ep[2]*fp[1] -$$

$$8*ap[2]*dp[1]*cp[2]*fp[1] - 8*ap[1]*dp[2]*ep[2]*fp[1] + 16*ap[2]*cp[2]*fp[1]*fp[1] -$$

$$8*bp[2]*cp[1]*ep[1]*fp[2] - 8*bp[1]*cp[2]*ep[1]*fp[2] - 8*ap[2]*dp[1]*ep[1]*fp[2] -$$

$$\begin{aligned}
 &8*ap[1]*dp[2]*ep[1]*fp[2]+16*bp[1]*cp[1]*ep[2]*fp[2]+16*ap[1]*dp[1]*ep[2]*fp[2]- \\
 &16*ap[2]*cp[1]*fp[1]*fp[2]-16*ap[1]*cp[2]*fp[1]*fp[2]+16*ap[1]*cp[1]*fp[2]*fp[2]- \\
 &12*ap[2]*cp[2]*ep[1]*gp[1]+8*bp[2]*dp[2]*ep[1]*gp[1]+6*ap[2]*cp[1]*ep[2]*gp[1]+6*ap[1]*cp[2] \\
 &*ep[2]*gp[1]-4*bp[2]*dp[1]*ep[2]*gp[1]- \\
 &4*bp[1]*dp[2]*ep[2]*gp[1]+16*bp[2]*cp[2]*fp[1]*gp[1]+16*ap[2]*dp[2]*fp[1]*gp[1]- \\
 &8*bp[2]*cp[1]*fp[2]*gp[1]-8*bp[1]*cp[2]*fp[2]*gp[1]-8*ap[2]*dp[1]*fp[2]*gp[1]- \\
 &8*ap[1]*dp[2]*fp[2]*gp[1]- \\
 &14*ap[2]*cp[2]*gp[1]*gp[1]+4*bp[2]*dp[2]*gp[1]*gp[1]+6*ap[2]*cp[1]*ep[1]*gp[2]+6*ap[1]*cp[2] \\
 &*ep[1]*gp[2]-4*bp[2]*dp[1]*ep[1]*gp[2]-4*bp[1]*dp[2]*ep[1]*gp[2]- \\
 &12*ap[1]*cp[1]*ep[2]*gp[2]+8*bp[1]*dp[1]*ep[2]*gp[2]-8*bp[2]*cp[1]*fp[1]*gp[2]- \\
 &8*bp[1]*cp[2]*fp[1]*gp[2]-8*ap[2]*dp[1]*fp[1]*gp[2]- \\
 &8*ap[1]*dp[2]*fp[1]*gp[2]+16*bp[1]*cp[1]*fp[2]*gp[2]+16*ap[1]*dp[1]*fp[2]*gp[2]+14*ap[2]*cp[\\
 &1]*gp[1]*gp[2]+14*ap[1]*cp[2]*gp[1]*gp[2]-4*bp[2]*dp[1]*gp[1]*gp[2]- \\
 &4*bp[1]*dp[2]*gp[1]*gp[2]-14*ap[1]*cp[1]*gp[2]*gp[2]+4*bp[1]*dp[1]*gp[2]*gp[2] \\
 \\
 &q_3=-4*ap[2]*bp[2]*cp[1]*cp[1]+4*ap[2]*bp[1]*cp[1]*cp[2]+4*ap[1]*bp[2]*cp[1]*cp[2]- \\
 &4*ap[1]*bp[1]*cp[2]*cp[2]- \\
 &4*ap[2]*ap[2]*cp[1]*dp[1]+16*bp[2]*bp[2]*cp[1]*dp[1]+4*ap[1]*ap[2]*cp[2]*dp[1]- \\
 &16*bp[1]*bp[2]*cp[2]*dp[1]+16*ap[2]*bp[2]*dp[1]*dp[1]+4*ap[1]*ap[2]*cp[1]*dp[2]- \\
 &16*bp[1]*bp[2]*cp[1]*dp[2]-4*ap[1]*ap[1]*cp[2]*dp[2]+16*bp[1]*bp[1]*cp[2]*dp[2]- \\
 &16*ap[2]*bp[1]*dp[1]*dp[2]- \\
 &16*ap[1]*bp[2]*dp[1]*dp[2]+16*ap[1]*bp[1]*dp[2]*dp[2]+6*bp[2]*cp[2]*ep[1]*ep[1]+6*ap[2]*dp[\\
 &2]*ep[1]*ep[1]-6*bp[2]*cp[1]*ep[1]*ep[2]-6*bp[1]*cp[2]*ep[1]*ep[2]-6*ap[2]*dp[1]*ep[1]*ep[2]- \\
 &6*ap[1]*dp[2]*ep[1]*ep[2]+6*bp[1]*cp[1]*ep[2]*ep[2]+6*ap[1]*dp[1]*ep[2]*ep[2]- \\
 &8*ap[2]*cp[2]*ep[1]*fp[1]+32*bp[2]*dp[2]*ep[1]*fp[1]+4*ap[2]*cp[1]*ep[2]*fp[1]+4*ap[1]*cp[2]* \\
 &ep[2]*fp[1]-16*bp[2]*dp[1]*ep[2]*fp[1]- \\
 &16*bp[1]*dp[2]*ep[2]*fp[1]+32*bp[2]*cp[2]*fp[1]*fp[1]+32*ap[2]*dp[2]*fp[1]*fp[1]+4*ap[2]*cp[1] \\
 &]*ep[1]*fp[2]+4*ap[1]*cp[2]*ep[1]*fp[2]-16*bp[2]*dp[1]*ep[1]*fp[2]-16*bp[1]*dp[2]*ep[1]*fp[2]- \\
 &8*ap[1]*cp[1]*ep[2]*fp[2]+32*bp[1]*dp[1]*ep[2]*fp[2]-32*bp[2]*cp[1]*fp[1]*fp[2]- \\
 &32*bp[1]*cp[2]*fp[1]*fp[2]-32*ap[2]*dp[1]*fp[1]*fp[2]- \\
 &32*ap[1]*dp[2]*fp[1]*fp[2]+32*bp[1]*cp[1]*fp[2]*fp[2]+32*ap[1]*dp[1]*fp[2]*fp[2]- \\
 &20*bp[2]*cp[2]*ep[1]*gp[1]- \\
 &20*ap[2]*dp[2]*ep[1]*gp[1]+10*bp[2]*cp[1]*ep[2]*gp[1]+10*bp[1]*cp[2]*ep[2]*gp[1]+10*ap[2]*d \\
 &p[1]*ep[2]*gp[1]+10*ap[1]*dp[2]*ep[2]*gp[1]- \\
 &72*ap[2]*cp[2]*fp[1]*gp[1]+32*bp[2]*dp[2]*fp[1]*gp[1]+36*ap[2]*cp[1]*fp[2]*gp[1]+36*ap[1]*cp \\
 &[2]*fp[2]*gp[1]-16*bp[2]*dp[1]*fp[2]*gp[1]-16*bp[1]*dp[2]*fp[2]*gp[1]- \\
 &26*bp[2]*cp[2]*gp[1]*gp[1]-
 \end{aligned}$$

$$\begin{aligned}
 & 26*ap[2]*dp[2]*gp[1]*gp[1]+10*bp[2]*cp[1]*ep[1]*gp[2]+10*bp[1]*cp[2]*ep[1]*gp[2]+10*ap[2]*d \\
 & p[1]*ep[1]*gp[2]+10*ap[1]*dp[2]*ep[1]*gp[2]-20*bp[1]*cp[1]*ep[2]*gp[2]- \\
 & 20*ap[1]*dp[1]*ep[2]*gp[2]+36*ap[2]*cp[1]*fp[1]*gp[2]+36*ap[1]*cp[2]*fp[1]*gp[2]- \\
 & 16*bp[2]*dp[1]*fp[1]*gp[2]-16*bp[1]*dp[2]*fp[1]*gp[2]- \\
 & 72*ap[1]*cp[1]*fp[2]*gp[2]+32*bp[1]*dp[1]*fp[2]*gp[2]+26*bp[2]*cp[1]*gp[1]*gp[2]+26*bp[1]*c \\
 & p[2]*gp[1]*gp[2]+26*ap[2]*dp[1]*gp[1]*gp[2]+26*ap[1]*dp[2]*gp[1]*gp[2]- \\
 & 26*bp[1]*cp[1]*gp[2]*gp[2]-26*ap[1]*dp[1]*gp[2]*gp[2]
 \end{aligned}$$

$$\begin{aligned}
 q_4 = & -(ap[2]*ap[2]*cp[1]*cp[1])+2*ap[1]*ap[2]*cp[1]*cp[2]- \\
 & ap[1]*ap[1]*cp[2]*cp[2]+16*bp[2]*bp[2]*dp[1]*dp[1]- \\
 & 32*bp[1]*bp[2]*dp[1]*dp[2]+16*bp[1]*bp[1]*dp[2]*dp[2]- \\
 & ap[2]*cp[2]*ep[1]*ep[1]+16*bp[2]*dp[2]*ep[1]*ep[1]+ap[2]*cp[1]*ep[1]*ep[2]+ap[1]*cp[2]*ep[1]* \\
 & ep[2]-16*bp[2]*dp[1]*ep[1]*ep[2]-16*bp[1]*dp[2]*ep[1]*ep[2]- \\
 & ap[1]*cp[1]*ep[2]*ep[2]+16*bp[1]*dp[1]*ep[2]*ep[2]- \\
 & 64*ap[2]*cp[2]*fp[1]*fp[1]+64*bp[2]*dp[2]*fp[1]*fp[1]+64*ap[2]*cp[1]*fp[1]*fp[2]+64*ap[1]*cp[\\
 & 2]*fp[1]*fp[2]-64*bp[2]*dp[1]*fp[1]*fp[2]-64*bp[1]*dp[2]*fp[1]*fp[2]- \\
 & 64*ap[1]*cp[1]*fp[2]*fp[2]+64*bp[1]*dp[1]*fp[2]*fp[2]-2*ap[2]*cp[2]*ep[1]*gp[1]- \\
 & 32*bp[2]*dp[2]*cp[1]*gp[1]+ap[2]*cp[1]*ep[2]*gp[1]+ap[1]*cp[2]*ep[2]*gp[1]+16*bp[2]*dp[1]*ep \\
 & [2]*gp[1]+16*bp[1]*dp[2]*ep[2]*gp[1]-128*bp[2]*cp[2]*fp[1]*gp[1]- \\
 & 128*ap[2]*dp[2]*fp[1]*gp[1]+64*bp[2]*cp[1]*fp[2]*gp[1]+64*bp[1]*cp[2]*fp[2]*gp[1]+64*ap[2]*d \\
 & p[1]*fp[2]*gp[1]+64*ap[1]*dp[2]*fp[2]*gp[1]+63*ap[2]*cp[2]*gp[1]*gp[1]- \\
 & 48*bp[2]*dp[2]*gp[1]*gp[1]+ap[2]*cp[1]*ep[1]*gp[2]+ap[1]*cp[2]*ep[1]*gp[2]+16*bp[2]*dp[1]*ep \\
 & [1]*gp[2]+16*bp[1]*dp[2]*ep[1]*gp[2]-2*ap[1]*cp[1]*ep[2]*gp[2]- \\
 & 32*bp[1]*dp[1]*ep[2]*gp[2]+64*bp[2]*cp[1]*fp[1]*gp[2]+64*bp[1]*cp[2]*fp[1]*gp[2]+64*ap[2]*d \\
 & p[1]*fp[1]*gp[2]+64*ap[1]*dp[2]*fp[1]*gp[2]-128*bp[1]*cp[1]*fp[2]*gp[2]- \\
 & 128*ap[1]*dp[1]*fp[2]*gp[2]-63*ap[2]*cp[1]*gp[1]*gp[2]- \\
 & 63*ap[1]*cp[2]*gp[1]*gp[2]+48*bp[2]*dp[1]*gp[1]*gp[2]+48*bp[1]*dp[2]*gp[1]*gp[2]+63*ap[1]*c \\
 & p[1]*gp[2]*gp[2]-48*bp[1]*dp[1]*gp[2]*gp[2]
 \end{aligned}$$

$$\begin{aligned}
 q_5 = & -8*ap[2]*bp[2]*cp[1]*cp[1]+8*ap[2]*bp[1]*cp[1]*cp[2]+8*ap[1]*bp[2]*cp[1]*cp[2]- \\
 & 8*ap[1]*bp[1]*cp[2]*cp[2]- \\
 & 8*ap[2]*ap[2]*cp[1]*dp[1]+16*bp[2]*bp[2]*cp[1]*dp[1]+8*ap[1]*ap[2]*cp[2]*dp[1]- \\
 & 16*bp[1]*bp[2]*cp[2]*dp[1]+16*ap[2]*bp[2]*dp[1]*dp[1]+8*ap[1]*ap[2]*cp[1]*dp[2]- \\
 & 16*bp[1]*bp[2]*cp[1]*dp[2]-8*ap[1]*ap[1]*cp[2]*dp[2]+16*bp[1]*bp[1]*cp[2]*dp[2]- \\
 & 16*ap[2]*bp[1]*dp[1]*dp[2]- \\
 & 16*ap[1]*bp[2]*dp[1]*dp[2]+16*ap[1]*bp[1]*dp[2]*dp[2]+4*bp[2]*cp[2]*ep[1]*ep[1]+4*ap[2]*dp[\\
 & 2]*ep[1]*ep[1]-4*bp[2]*cp[1]*ep[1]*ep[2]-4*bp[1]*cp[2]*ep[1]*ep[2]-4*ap[2]*dp[1]*ep[1]*ep[2]-
 \end{aligned}$$

$$\begin{aligned}
 & 4*ap[1]*dp[2]*cp[1]*ep[2]+4*bp[1]*cp[1]*ep[2]*ep[2]+4*ap[1]*dp[1]*ep[2]*ep[2]- \\
 & 16*ap[2]*cp[2]*ep[1]*fp[1]+32*bp[2]*dp[2]*ep[1]*fp[1]+8*ap[2]*cp[1]*ep[2]*fp[1]+8*ap[1]*cp[2] \\
 & *ep[2]*fp[1]-16*bp[2]*dp[1]*ep[2]*fp[1]-16*bp[1]*dp[2]*ep[2]*fp[1]-96*bp[2]*cp[2]*fp[1]*fp[1]- \\
 & 96*ap[2]*dp[2]*fp[1]*fp[1]+8*ap[2]*cp[1]*ep[1]*fp[2]+8*ap[1]*cp[2]*ep[1]*fp[2]- \\
 & 16*bp[2]*dp[1]*cp[1]*fp[2]-16*bp[1]*dp[2]*ep[1]*fp[2]- \\
 & 16*ap[1]*cp[1]*ep[2]*fp[2]+32*bp[1]*dp[1]*cp[2]*fp[2]+96*bp[2]*cp[1]*fp[1]*fp[2]+96*bp[1]*cp[\\
 & 2]*fp[1]*fp[2]+96*ap[2]*dp[1]*fp[1]*fp[2]+96*ap[1]*dp[2]*fp[1]*fp[2]- \\
 & 96*bp[1]*cp[1]*fp[2]*fp[2]-96*ap[1]*dp[1]*fp[2]*fp[2]-24*bp[2]*cp[2]*ep[1]*gp[1]- \\
 & 24*ap[2]*dp[2]*ep[1]*gp[1]+12*bp[2]*cp[1]*ep[2]*gp[1]+12*bp[1]*cp[2]*ep[2]*gp[1]+12*ap[2]*d \\
 & p[1]*ep[2]*gp[1]+12*ap[1]*dp[2]*ep[2]*gp[1]+176*ap[2]*cp[2]*fp[1]*gp[1]- \\
 & 224*bp[2]*dp[2]*fp[1]*gp[1]-88*ap[2]*cp[1]*fp[2]*gp[1]- \\
 & 88*ap[1]*cp[2]*fp[2]*gp[1]+112*bp[2]*dp[1]*fp[2]*gp[1]+112*bp[1]*dp[2]*fp[2]*gp[1]+100*bp[2] \\
 &]*cp[2]*gp[1]*gp[1]+100*ap[2]*dp[2]*gp[1]*gp[1]+12*bp[2]*cp[1]*ep[1]*gp[2]+12*bp[1]*cp[2]*e \\
 & p[1]*gp[2]+12*ap[2]*dp[1]*ep[1]*gp[2]+12*ap[1]*dp[2]*ep[1]*gp[2]-24*bp[1]*cp[1]*ep[2]*gp[2]- \\
 & 24*ap[1]*dp[1]*ep[2]*gp[2]-88*ap[2]*cp[1]*fp[1]*gp[2]- \\
 & 88*ap[1]*cp[2]*fp[1]*gp[2]+112*bp[2]*dp[1]*fp[1]*gp[2]+112*bp[1]*dp[2]*fp[1]*gp[2]+176*ap[1] \\
 & *cp[1]*fp[2]*gp[2]-224*bp[1]*dp[1]*fp[2]*gp[2]-100*bp[2]*cp[1]*gp[1]*gp[2]- \\
 & 100*bp[1]*cp[2]*gp[1]*gp[2]-100*ap[2]*dp[1]*gp[1]*gp[2]- \\
 & 100*ap[1]*dp[2]*gp[1]*gp[2]+100*bp[1]*cp[1]*gp[2]*gp[2]+100*ap[1]*dp[1]*gp[2]*gp[2]
 \end{aligned}$$

$$\begin{aligned}
 q_6 &= 4*ap[2]*ap[2]*cp[1]*cp[1]-8*bp[2]*bp[2]*cp[1]*cp[1]- \\
 & 8*ap[1]*ap[2]*cp[1]*cp[2]+16*bp[1]*bp[2]*cp[1]*cp[2]+4*ap[1]*ap[1]*cp[2]*cp[2]- \\
 & 8*bp[1]*bp[1]*cp[2]*cp[2]- \\
 & 32*ap[2]*bp[2]*cp[1]*dp[1]+16*ap[2]*bp[1]*cp[2]*dp[1]+16*ap[1]*bp[2]*cp[2]*dp[1]- \\
 & 8*ap[2]*ap[2]*dp[1]*dp[1]+32*bp[2]*bp[2]*dp[1]*dp[1]+16*ap[2]*bp[1]*cp[1]*dp[2]+16*ap[1]*bp \\
 & [2]*cp[1]*dp[2]-32*ap[1]*bp[1]*cp[2]*dp[2]+16*ap[1]*ap[2]*dp[1]*dp[2]- \\
 & 64*bp[1]*bp[2]*dp[1]*dp[2]-8*ap[1]*ap[1]*dp[2]*dp[2]+32*bp[1]*bp[1]*dp[2]*dp[2]- \\
 & 4*ap[2]*cp[2]*ep[1]*ep[1]+24*bp[2]*dp[2]*ep[1]*ep[1]+4*ap[2]*cp[1]*ep[1]*ep[2]+4*ap[1]*cp[2] \\
 & *ep[1]*ep[2]-24*bp[2]*dp[1]*ep[1]*ep[2]-24*bp[1]*dp[2]*ep[1]*ep[2]- \\
 & 4*ap[1]*cp[1]*ep[2]*ep[2]+24*bp[1]*dp[1]*ep[2]*ep[2]-32*bp[2]*cp[2]*ep[1]*fp[1]- \\
 & 32*ap[2]*dp[2]*ep[1]*fp[1]+16*bp[2]*cp[1]*cp[2]*fp[1]+16*bp[1]*cp[2]*ep[2]*fp[1]+16*ap[2]*dp \\
 & [1]*ep[2]*fp[1]+16*ap[1]*dp[2]*ep[2]*fp[1]+96*ap[2]*cp[2]*fp[1]*fp[1]- \\
 & 128*bp[2]*dp[2]*fp[1]*fp[1]+16*bp[2]*cp[1]*ep[1]*fp[2]+16*bp[1]*cp[2]*ep[1]*fp[2]+16*ap[2]*d \\
 & p[1]*ep[1]*fp[2]+16*ap[1]*dp[2]*ep[1]*fp[2]-32*bp[1]*cp[1]*ep[2]*fp[2]- \\
 & 32*ap[1]*dp[1]*ep[2]*fp[2]-96*ap[2]*cp[1]*fp[1]*fp[2]- \\
 & 96*ap[1]*cp[2]*fp[1]*fp[2]+128*bp[2]*dp[1]*fp[1]*fp[2]+128*bp[1]*dp[2]*fp[1]*fp[2]+96*ap[1]*c \\
 & p[1]*fp[2]*fp[2]-128*bp[1]*dp[1]*fp[2]*fp[2]+24*ap[2]*cp[2]*ep[1]*gp[1]-
 \end{aligned}$$

$$\begin{aligned} &80*bp[2]*dp[2]*ep[1]*gp[1]-12*ap[2]*cp[1]*ep[2]*gp[1]- \\ &12*ap[1]*cp[2]*ep[2]*gp[1]+40*bp[2]*dp[1]*ep[2]*gp[1]+40*bp[1]*dp[2]*ep[2]*gp[1]+224*bp[2]* \\ &cp[2]*fp[1]*gp[1]+224*ap[2]*dp[2]*fp[1]*gp[1]-112*bp[2]*cp[1]*fp[2]*gp[1]- \\ &112*bp[1]*cp[2]*fp[2]*gp[1]-112*ap[2]*dp[1]*fp[2]*gp[1]-112*ap[1]*dp[2]*fp[2]*gp[1]- \\ &100*ap[2]*cp[2]*gp[1]*gp[1]+152*bp[2]*dp[2]*gp[1]*gp[1]-12*ap[2]*cp[1]*ep[1]*gp[2]- \\ &12*ap[1]*cp[2]*ep[1]*gp[2]+40*bp[2]*dp[1]*ep[1]*gp[2]+40*bp[1]*dp[2]*ep[1]*gp[2]+24*ap[1]*c \\ &p[1]*ep[2]*gp[2]-80*bp[1]*dp[1]*ep[2]*gp[2]-112*bp[2]*cp[1]*fp[1]*gp[2]- \\ &112*bp[1]*cp[2]*fp[1]*gp[2]-112*ap[2]*dp[1]*fp[1]*gp[2]- \\ &112*ap[1]*dp[2]*fp[1]*gp[2]+224*bp[1]*cp[1]*fp[2]*gp[2]+224*ap[1]*dp[1]*fp[2]*gp[2]+100*ap[\\ &2]*cp[1]*gp[1]*gp[2]+100*ap[1]*cp[2]*gp[1]*gp[2]-152*bp[2]*dp[1]*gp[1]*gp[2]- \\ &152*bp[1]*dp[2]*gp[1]*gp[2]-100*ap[1]*cp[1]*gp[2]*gp[2]+152*bp[1]*dp[1]*gp[2]*gp[2] \end{aligned}$$



