

BİLGİSAYAR DESTEKLİ KAM TASARIMI
VE ÖRNEK BİR KAM MEKANİZMASININ
İMALATI
Murat KOYUNBAKAN
Yüksek Lisans
Makine Eğitimi Anabilim Dalı
Nisan-2006

BİLGİSAYAR DESTEKLİ KAM TASARIMI
VE ÖRNEK BİR KAM MEKANİZMASININ
İMALATI

Murat KOYUNBAKAN

Dumlupınar Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Makina Eğitimi Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman: Yrd. Doç. Dr. Muammer GAVAS

Nisan-2006

KABUL VE ONAY SAYFASI

Murat KOYUNBAKAN 'ın YÜKSEK LİSANS tezi olarak hazırladığı BİLGİSAYAR DESTEKLİ KAM TASARIMI VE ÖRNEK BİR KAM MEKANİZMASININ İMALATI başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir

31/03/2006

Üye : Doç. Dr. Rasim İPEK

Üye : Yrd. Doç. Dr. Muammer GAVAS (Danışman)

Üye : Yrd. Doç. Dr. İsmet ÇELİK

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. M. Sabri ÖZYURT
Fen Bilimleri Enstitüsü Müdürü

BİLGİSAYAR DESTEKLİ KAM TASARIMI VE ÖRNEK BİR KAM MEKANİZMASININ İMALATI

Murat KOYUNBAKAN

Makine Eğitimi, Yüksek Lisans Tezi, 2005,

Tez Danışmanı: Yrd. Doç. Dr. Muammer GAVAS

ÖZET

Bir çok kullanma alanına sahip olan kam mekanizmalarının geleneksel yöntemlerle tasarımları yapılırken birtakım sorunlarla karşılaşmaktadır. Geleneksel tasarım yöntemleri tamamen tasarımcının el becerisine bağlıdır. Son yıllarda bilgisayar destekli tasarım yöntemleri ile bir çok makine parçasının tasarımı daha kolay ve daha kısa zamanda yapılabilmektedir.

Bu çalışmada, çıkış uzvu öteleme hareketi yapan kam mekanizmaları için bilgisayar destekli kam tasarım programı geliştirilmiştir. Program Borland Delphi 7 programlama dili kullanılarak yazılmıştır. Yazılan bu program ile kam'a ait yol, hız ve ivme grafikleri ile kam profilinin çizdirilmesi hedeflenmiştir. Ayrıca örnek bir kam mekanizmasının tasarımı ve imalatı yapılarak ege sapının ahşap malzemeden üretimi gerçekleştirilmiştir.

Bilgisayar destekli tasarım programının çıkış uzvu öteleme hareketi yapan kamların tasarımı hız ve doğruluk bakımından tasarımcıya önemli katkıları olduğu görülmüştür.

Anahtar Kelimeler: Bilgisayar destekli tasarım, basınç açısı, kamların imalatı, kam mekanizmaları.

COMPUTER AIDED OF CAM DESIGN AND A SAMPLE OF CAM MECHANISM MANUFACTURING

Murat KOYUNBAKAN

Mechanical Education, M.S.Thesis, 2005

Thesis Supervisor: Assist. Prof. Muammer GAVAS

SUMMARY

Cam mechanisms which have lots of using fields are faced with some problems in the design of traditional methods. Traditional design methods completely depend on the skill of the designer. Nowadays, the design of many items of the machines can be done more easily and in a shorter time by computer aided design methods.

In this work, computer aided cam design program was developed for cam mechanisms whose exit unit does push movement. This program was written by using 'Borland Delphi 7' program language. Getting the profile of cam and the graphics of the way, speed and acceleration that belong to cam drawn was aimed by this program. Also the design and manufacture of a sample cam mechanism was done and the production of the handle of the file from wooden material was realized.

It was seen that computer aided design program has important assistances to designer in terms of speed and uprightness in the design of cams whose exit unit does push movement.

Key Words: Computer Aided Design (CAD), pressure angle, cams manufacturing, cam mechanisms.

TEŞEKKÜR

Çalışmalarım süresince yardım ve fedakarlıklarını esirgemeyen, bilgi ve katkılarıyla beni yönlendiren değerli danışman hocam Yrd. Doç. Dr. Muammer GAVAS' a, bilgisayar programının hazırlanmasındaki yardımlarından dolayı Anadolu Üniversitesi Bilgi İşlem Dairesi şeflerinden Hüryaşa ASLAN Bey'e ve Simav Teknik Eğitim Fakültesi'nde görevli Arş. Gör. Mustafa Erginli' ye teşekkür ederim.

Ayrıca imalat aşamasındaki yardımlarından dolayı Simav Teknik Eğitim Fakültesi Makine Eğitimi Bölümü Teknisyenlerinden İbrahim ÇAKIN, Ramazan CANER ve mesai arkadaşlarıma teşekkür ederim.

Maddi ve manevi tüm destekleriyle daima yanımda olan annem, babam, eşim ve oğlum Ömer Faruk'a ...

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	iv
SUMMARY	v
ŞEKİLLER DİZİNİ.....	x
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
1.1. Mekanizmanın Tanımı Ve Mekanizmaların Sınıflandırılması.....	1
1.1. Mekanizmanın tanımı.....	1
1.2. Mekanizmaların sınıflandırılması	2
2. KAM TANIMI VE KAM MEKANİZMALARININ SINIFLANDIRILMASI	5
2.1. Kam Tanımı	5
2.2. Kam Mekanizmalarının Sınıflandırılması	5
2.2.1. Biçimlerine göre kam mekanizmaları	6
2.2.1.1. Düzlemsel kam mekanizmaları	6
2.2.1.2. Uzaysal kam mekanizmaları	7
2.2.2. İzleyicinin şekline göre kam mekanizmaları.....	8
2.3. Kam Mekanizmalarının Kullanıldığı Yerler, Avantaj Ve Dezavantajları.....	11
2.3.1. Kam mekanizmalarının kullanıldığı yerler	11
2.3.2. Kam mekanizmalarının avantajları	11
2.3.3. Kam mekanizmalarının dezavantajları.....	12
2.4. Kamların İmalatı	12
3. KAM TASARIMI	14
3.1. Kam Terimleri.....	14
3.2. İzleyicinin Hareketleri.....	15

İÇİNDEKİLER (devam)

	<u>Sayfa</u>
3.3. Kam Profilinin Grafik Olarak Belirlenmesi	16
3.4. Kinematik Parametrelerin Belirlenmesi.....	17
3.5. Kam Mekanizmalarının Çalışmasının Kinematik Şartı	21
3.6. Kam Mekanizmalarının Çalışmasının Dinamik Şart	22
3.6.1. Çıkış uzvu öteleme hareketi yapan kam mekanizmalarında dinamik şart	22
3.6.2. Çıkış uzvu sarkaç hareketi yapan kam mekanizmalarında dinamik şart.....	26
3.7. Çıkış Uzvu Öteleme Hareketi Yapan Kam Mekanizmasının Sentezi Yapılarak Ani Dönme Merkezlerinin Bulunması Ve Kam Profilinin Çizilmesi	27
4. KAM TASARIM PROGRAMININ HAZIRLANMASI.....	32
4.1. Kam Mekanizmaları Üzerine Yapılan Çalışmalar	32
4.2. Programın Hazırlanması.....	33
4.2.1. Hareket Grafiğinin Çizilmesi	33
4.2.2. En Çok Kullanılan Hareket Denklemleri ve Farklı Hareket Denklemlerinin Kam Profiline Olan Etkisi.....	34
4.2.3. Ani Dönme Merkezlerinin ve kam dönme merkezinin belirlenmesi ile kam profilinin çizilmesi	35
4.3. Kam Tasarım Programının Tanıtılması.....	38
5. ÖRNEK BİR KAM MEKANİZMASININ TASARIMI VE İMALATI	45
5.1. Mekanizmada Kullanılacak Kamın Tasarımı ve İmalatı.....	46
5.2. Mekanizmada Kullanılan Diğer Parçalar Ve Mekanizmanın Tanıtımı	48
6. SONUÇ VE ÖNERİLER	50
KAYNAKLAR DİZİNİ	51

İÇİNDEKİLER (devam)

EKLER

1. Kam Mekanizmalarında Kullanılan Hareket Denklemleri.
2. Kam Tasarım Programının Kodları.
3. Parçaların Yapım Resimleri ve Mekanizmanın Montaj Resmi.
4. Geleneksel Yöntemlerle Ve Tasarım Programıyla Tasarlanan Kamın Karşılaştırılması.
- 5- Örnek Kam Mekanizmasına Ait Resimler.

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
1.1. Mekanizmaların sınıfları	3
2.1. Kam ve izleyici	5
2.2. Giriş parçası dönme hareketi yapan kam mekanizmaları.....	6
2.3. Giriş parçası öteleme hareketi yapan kam mekanizmaları.....	7
2.4. Uzaysal kam mekanizmaları	7
2.5. Kam çifti yüzeyine etkiyen normal kuvvet	8
2.6. Öteleme yapan izleyiciler.....	9
2.7. Salınım yapan izleyiciler.....	10
2.8. Örnek kam mekanizmaları	10
2.9. Otomat torna tezgahları.....	11
3.1. Kam terimleri	15
3.2. İzleyici hareketleri a) BKB, b) BKİ, c) Kİ.....	16
3.3. Kam profilinin belirlenmesi	17
3.4. İzleyiciye ait hareket diyagramı	18
3.5. İzleyiciye ait hız ve ivme grafikleri	21
3.6. Kam mekanizmalarının çalışmasındaki kinematik şartın gösterimi.....	21
3.7. Çıkış uzvu öteleme hareketi yapan kam mekanizmasında dinamik şartın incelenmesi .	23
3.8. Çıkış uzvu sarkaç hareketi yapan kam mekanizmasında dinamik şartın incelenmesi ...	26
3.9. Yol-zaman ve hız-zaman diyagramları	28
3.10. Ani dönme merkezleri yörüngesinin çizimi	30
3.11. Dönme merkezinin bulunması	30
3.12. Kam profilinin çizimi	31
4.1. Sabit hızlı hareket grafiği	34
4.2. Hareket denklemlerinin kam profiline etkisi.....	35
4.3. Ani dönme merkezlerine teğet doğruların çizimi.....	37
4.4. Mecburi eksanrisitenin gösterimi	38

ŞEKİLLER DİZİNİ (Devam)

<u>Sekil</u>	<u>Sayfa</u>
4.5. Kam tasarım programının kullanıcı arayüz görüntüsü	39
4.6. Giriş parametrelerinin girilip işlemi başlata basıldıktan sonraki arayüz görüntüsü	40
4.7. Yol, hız ve ivme grafikleri ile kam profili görüntüleri (a: Yol Grafiği, b: Hız Grafiği, c: İvme Grafiği, d: Kam Profili)	41
5.1. Eğe sapının teknik resmi	45
5.2. Torna kalemlerine eğe sapı profilinin verilmesi	45
5.3. İmalatı yapılacak kama ait yol-zaman grafiği	46
5.4. Ani dönme merkezlerinin çizimi ve kam dönme merkezinin bulunması.....	47
5.5. İmalatı yapılacak kam profilinin çizimi	48

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1. İzleyiciye ait konum değerleri.....	18
3.2. Verilenler çizelgesi	27

1.GİRİŞ

Makinelerin ve onları oluşturan mekanizmaların tarihi insanlığın var oluşuna kadar uzanır. İnsanlar zorunlu ihtiyaçlarını gidermek için önceleri basit makineler icat etmişler ve kullanmışlardır. Ancak öyle bir zaman gelmiştir ki makineler zorunlu ihtiyaçlardan ziyade hayatı kolaylaştıran ihtiyaçları karşılamak üzere üretilmeye başlanmıştır.

19. yüzyılın başlarında makine imalatı hızla yükselişe geçmiş ve bununla birlikte makineyi oluşturan sistemlerde de bir takım yenilenmelere ve yeni mekanizmaların tasarımına ihtiyaç giderek artmıştır.

Makinelerin temelini oluşturan mekanizmalar çok çeşitlidir ve makinelerin tasarımlarında yaygın olarak kullanılmaktadırlar. Bu mekanizmalardan biri olan kam mekanizmaları kuvvet ve hareket iletiminde yaygın olarak kullanılmaktadır.

Kam mekanizmalarının tasarımı ve imalatı diğer mekanizmalara göre daha zor olmakla birlikte karmaşık hareketlerin elde edilmesinde en çok tercih edilen mekanizmalardır. Kam mekanizmalarının tasarımında grafiksel metotlar kullanılmaktadır. Grafiksel metotların kullanılmasında tasarımcının el becerisi ne kadar iyi olursa olsun bir takım hassas kısımlar gözden kaçabilmekte ve bu yüzden imalat sırasında bazı zorluklarla karşılaşabilmektedir.

Bilgisayar teknolojisinin gelişmesine bağlı olarak bir çok makine parçası bilgisayar destekli olarak tasarlanmaktadır. Bu şekilde tasarımda karşılaşılan bir çok zorluklar aşılabilmektedir. Kam mekanizmalarının tasarımı için de böyle bir bilgisayar destekli tasarım programına ihtiyaç vardır.

Bu çalışmada kam mekanizmaları tanıtılacak ve kam mekanizmalarının tasarımları açıklanacaktır. Ayrıca kam mekanizmaları için geliştirilmiş olan bilgisayar programının işlevleri hakkında bilgi verilecek ve çalışmanın son bölümünde örnek olarak hazırlanmış kam mekanizmasının tasarımı ve imalatı açıklanacaktır.

1.1.Mekanizmanın Tanımı Ve Mekanizmaların Sınıflandırılması

1.1.1. Mekanizmanın tanımı

Mekanizma, kuvvet ve hareket iletimi için kullanılabilen rijit cisimlerin rijit mafsallarla birleştirildiği sistem olarak tanımlanabilmektedir. Makine ise; tabiatta mevcut mekanik

kuvvetlerin belirli bir hareket ile birlikte iş yapmasını sağlayabilen, kuvvete karşı direnç gösterebilen cisimlerin birleştirilmesi ile oluşturulan bir sistemdir.

Mekanizma ve makine birbiri ile karıştırılan iki kavramdır. Mekanizma ve makine arasındaki en belirgin fark makinenin belli bir amaca yönelik üretilmiş olmasıdır. Mekanizma ise daha geneldir ve farklı makinelerde kullanılabilir. Örneğin bir içten yanmalı motor ve bir hidrolik pompa farklı makinelerdir. Ancak her ikisinde de krank-biyel mekanizması kullanılmaktadır.

Mekanizma, makine içinde belli bir hareketi belirleyen cisimlerin oluşturduğu sistemin bir modelidir [1].

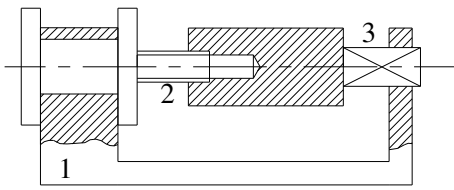
1.1.2. Mekanizmaların sınıflandırılması

Mekanizma tekniği konularını ilk defa bir bilim dalı olarak ele alan Reuleaux'ya göre mekanizmalar altı temel grupta sınıflandırılabilirler. Şekil 1.1'de mekanizmaların örnek resimleri verilmiştir.

Mekanizmalar aşağıdaki şekilde sınıflandırılabilir.

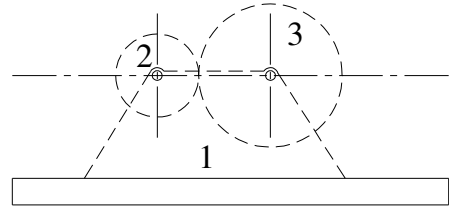
1. Vida mekanizmaları,
2. Çark mekanizmaları (dişli çarklar, sürtünme çarkları),
3. Kam mekanizmaları,
4. Kol mekanizmaları,
5. Kayış-kasnak mekanizmaları,
6. Cırcır veya mandal mekanizmaları [1].

Bu sınıflandırma basit mekanizmalar için uygulanabilmekte ancak teknolojinin gelişmesiyle karmaşık mekanizmalar da kullanılmaktadır.



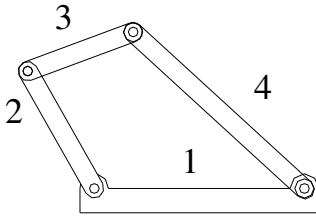
(a)

Vida Mekanizması



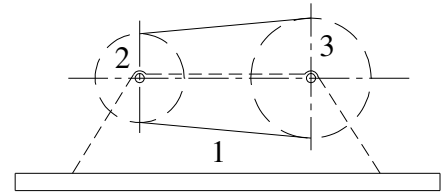
(b)

Çark Mekanizması



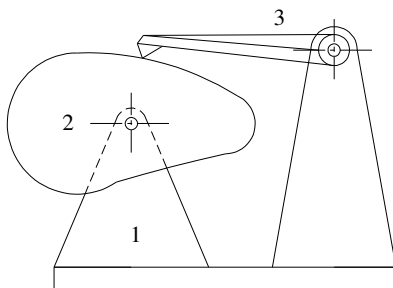
(c)

Kol Mekanizması



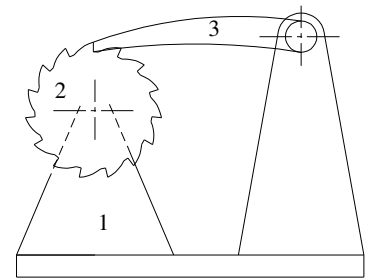
(d)

Kayış-Kasnak Mekanizması



(e)

Kam Mekanizması



(f)

Çırcır Mekanizması

Şekil 1.1. Mekanizmaların sınıfları [1].

Mekanizmaların sınıflandırılmasına ilave olarak bir mekanizmanın tipini belirleyen tüm özelliklerin sıralanması daha önemlidir. Bu özellikler;

1. Mekanizmanın çalıştığı uzay serbestlik derecesi,
2. Mekanizma serbestlik derecesi,

3. Mekanizma para sayısı,
4. Mekanizmada mafsalsayısı,
5. Mekanizmada bulunan mafsalsayıları.

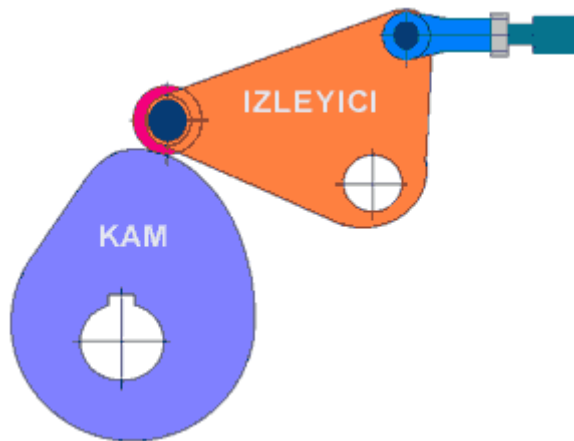
Bir mekanizmanın tanımı için yukarıda verilen tüm parametreler geçerlidir ve mekanizmaları sınıflandırmak için kullanılabilir. Reuleaux'nun sınıflandırması ise kısmen mafsalsayılarına göre yapılan bir sınıflandırmadır [1].

2.KAM TANIMI VE KAM MEKANİZMALARININ SINIFLANDIRILMASI

2.1. Kam Tanımı

Kam, başka bir elemana (izleyiciye) yüzey teması yoluyla istenen bir hareketi yaptırmak için kullanılan bir makine elemanıdır [2, 3]. Kamlar, tasarımcının öngördüğü hareketi basit bir yolla ve tam istendiği şekilde sağlayabilen, makine elemanlarının en etkin mekanizmalarından biridir. Kam-izleyici mekanizmaları basit, ucuz, az yer tutan, az parça içeren elemanlardır. Herhangi bir izleyici hareketini elde etmek zor değildir. Bu sebeplerle kam mekanizmaları makine tasarımında çok yoğun kullanılırlar. Şekil 2.1’de örnek bir kam mekanizması gösterilmiştir [2].

Gelişen teknoloji sayesinde servo motorlar bir pinyon ve kremayer yardımıyla dijital kontrol araçları ile bir kamın hareketini tamamen taklit edebildiğinden bazı uygulamalarda kamlar yerine kullanılırlar. Bu uygulamaya elektronik kam denir [2]. Birleşik mekanizmaların kullanımı, mekanizmaların giriş hızlarının kontrol edilebilirliği fikrini ortaya çıkarmıştır. Geçen on yıl boyunca servo motor ve kontrol sistemleri hızla gelişmiştir. Yüksek hassasiyetli makinelerde mikro-komputür kontrollü servo motorların mekanizmaya hareket verici olarak sıradan motorlar yerine kullanımı doğru olandır [4].



Şekil 2.1. Kam ve izleyici [2].

2.2. Kam Mekanizmalarının Sınıflandırılması

Kam mekanizmaları biçimlerine ve izleyicilerine göre iki şekilde sınıflandırılırlar.

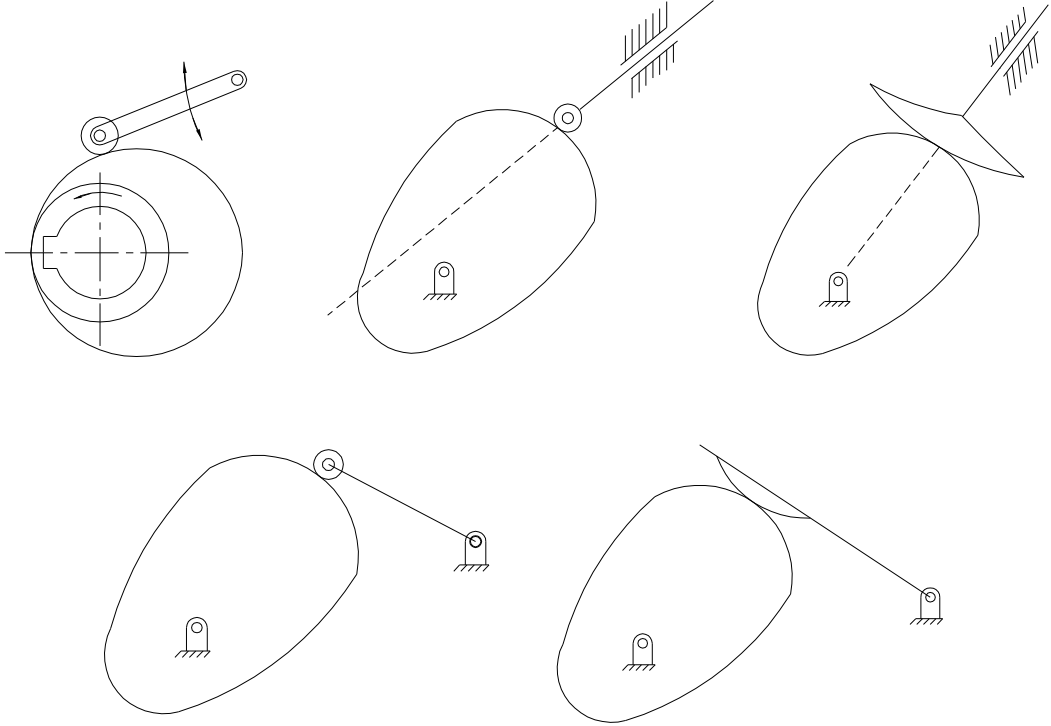
2.2.1. Biçimlerine göre kam mekanizmaları

Kam mekanizmaları biçimlerine göre düzlemsel ve uzaysal olmak üzere iki grupta incelenir [2].

2.2.1.1 Düzlemsel kam mekanizmaları

Giriş parçası dönme hareketi yapan kam mekanizmaları;

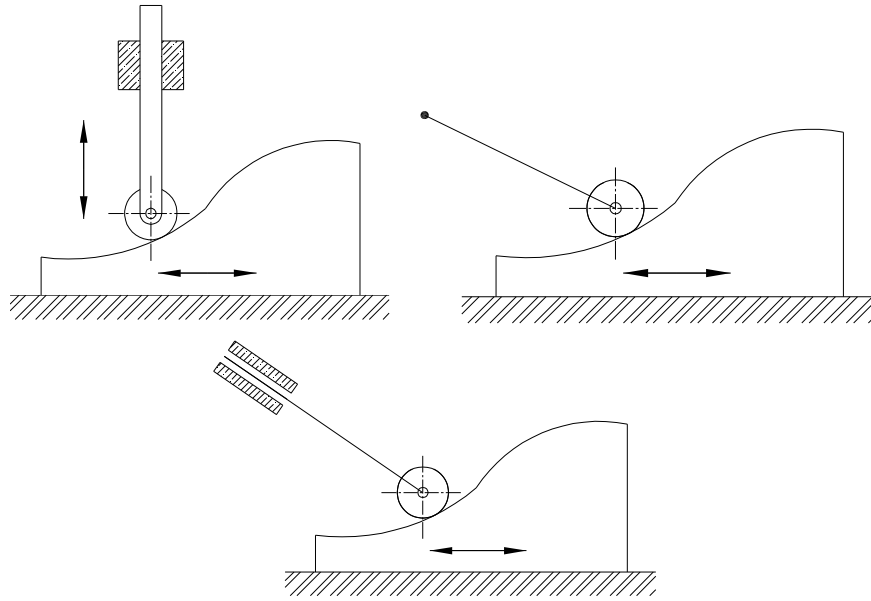
Bu tip kam mekanizmalarında giriş parçası dönme hareketi yaparken, çıkış parçası öteleme, dönme veya karmaşık hareket yapabilir [5, 6, 7]. Şekil 2.2' de giriş parçası dönme hareketi yapan kam mekanizmaları gösterilmiştir [2].



Şekil 2.2. Giriş parçası dönme hareketi yapan kam mekanizmaları [2]

Giriş parçası öteleme hareketi yapan kam mekanizmaları;

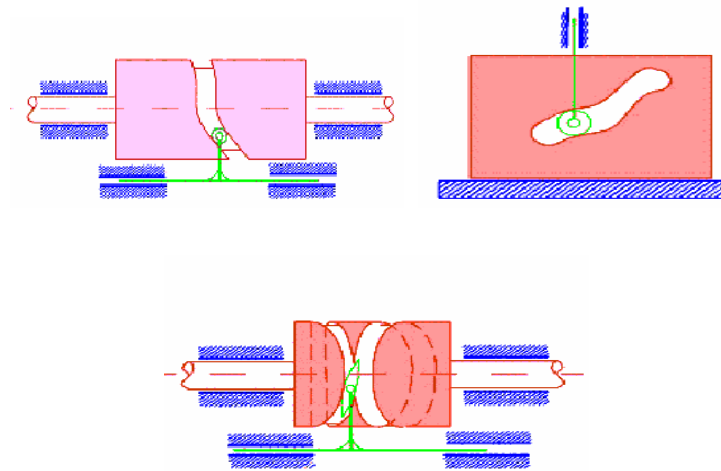
Giriş parçası öteleme hareketi yapan kam mekanizmalarında çıkış uzvu öteleme, sarkaç veya karmaşık hareket yapabilir. Şekil 2.3'de giriş parçası öteleme hareketi yapan kam mekanizmaları gösterilmektedir [5, 6].



Şekil 2.3. Giriş parçası öteleme hareketi yapan kam mekanizmaları [8].

2.2.1.2 Uzaysal kam mekanizmaları

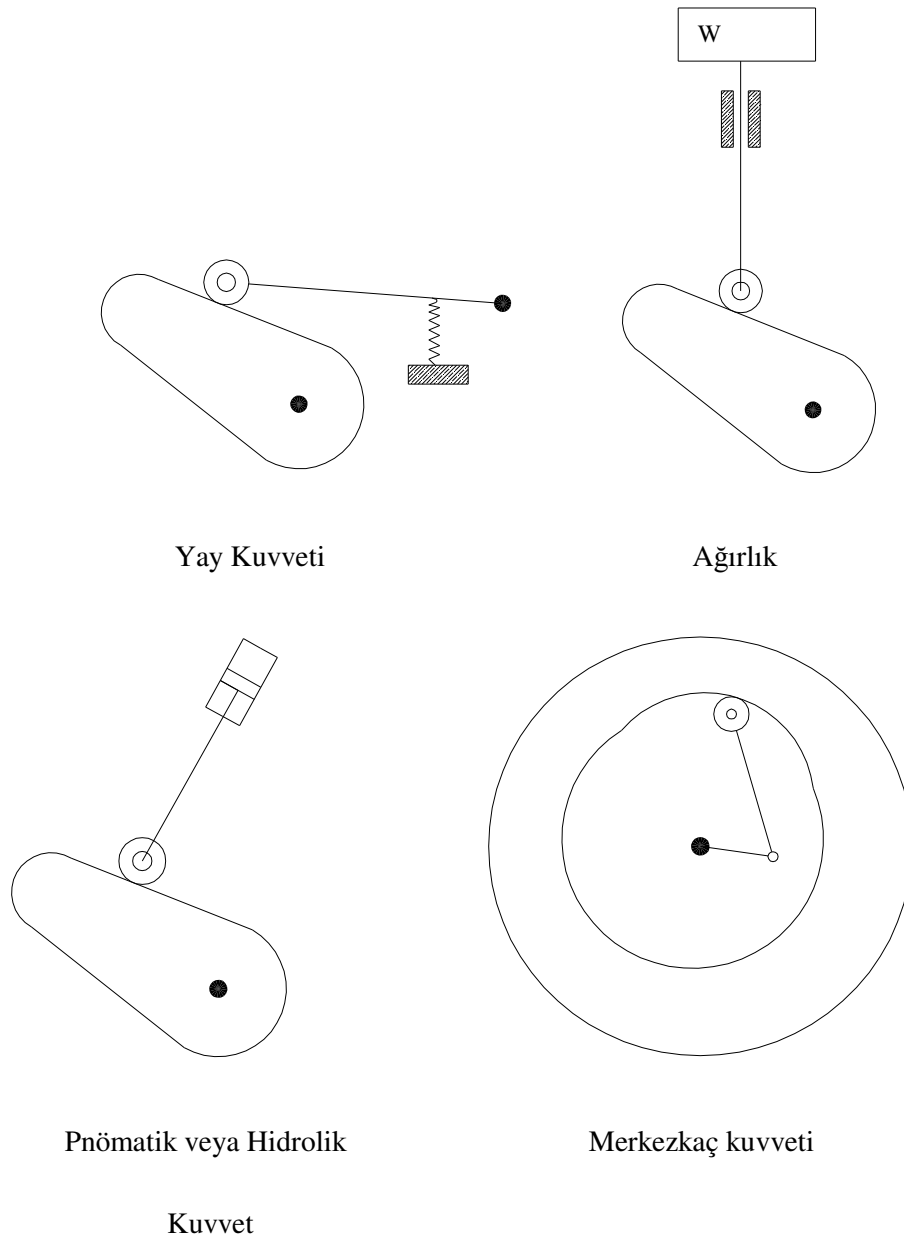
Uzaysal tipteki kam mekanizmalarında cam ile tahrik edilen izleyici aynı düzlemde olmayabilir. Bu tür kam mekanizmalarının bir çok çeşitleri vardır. Bunların tasarımı ve imalatı düzlemsel kam mekanizmalarına göre daha zordur. Şekil 2.4’de uzaysal kam mekanizmalarına örnekler verilmiştir [5, 6].



Şekil 2.4. Uzaysal kam mekanizmaları [8].

2.2.2. İzleyicinin şekline göre kam mekanizmaları

Kam ile izleyicinin temas şekline göre kam çiftleri kuvvet kapalı veya şekil kapalı olarak sınıflandırılabilirler. Kuvvet kapalı kam çiftleri daha yaygın olup kam çifti yüzeyine etkiyen normal kuvvetin ne şekilde oluştuğuna göre sınıflandırılırlar. (yay, ağırlık, pnömatik vb.) Şekil 2.5’de bu sınıflandırma gösterilmiştir [8].

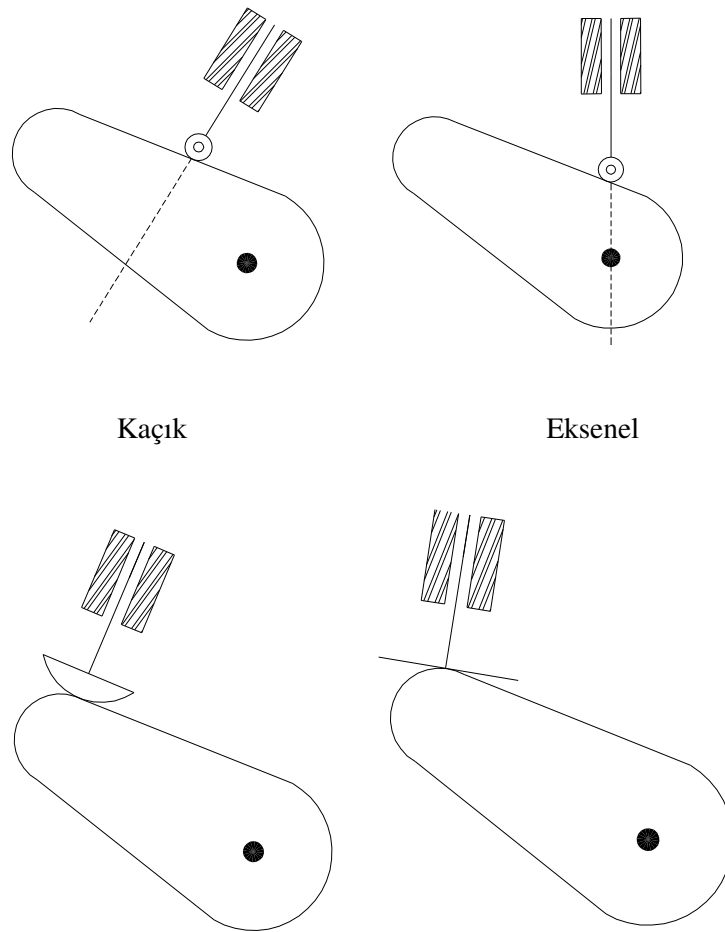


Şekil 2.5. Kam çifti yüzeyine etkiyen normal kuvvet [2].

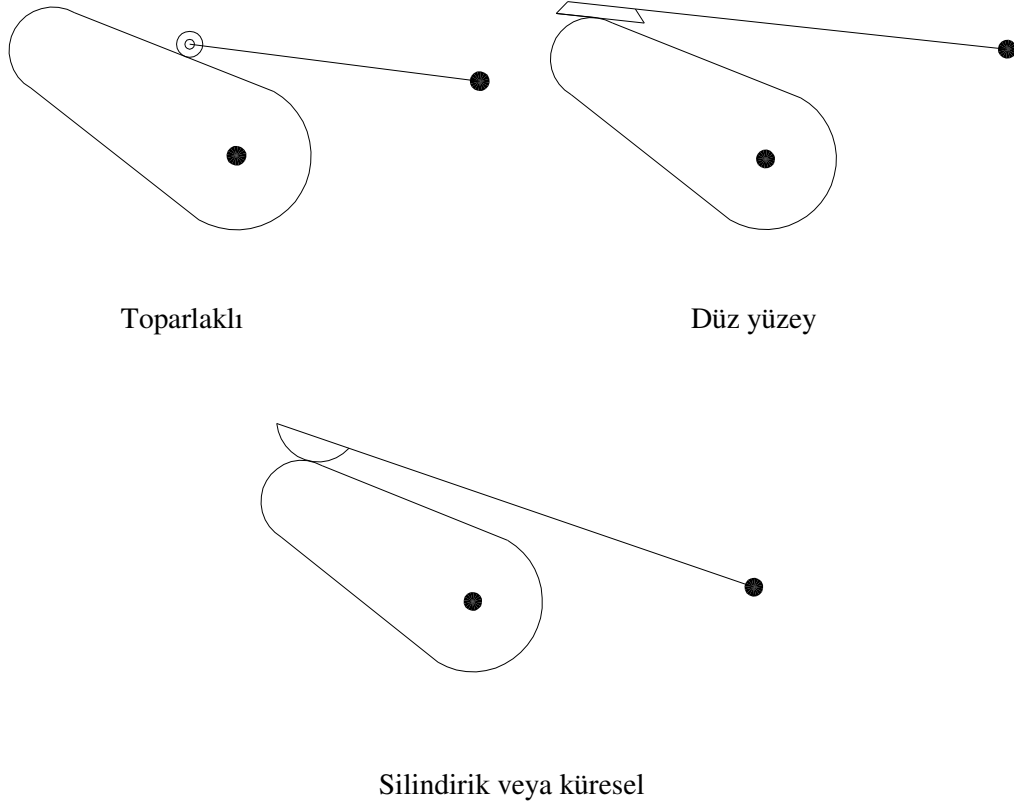
Şekil kapalı kamlarda kam ve izleyici iki noktadan temas ederler ve kinematik çiftlerin teması için ek bir kuvvete ihtiyaç yoktur. İzleyici parça olarak adlandırılan, genellikle basit geometrik yapıya sahip kinematik elemanı olan parça ise iki değişik şekilde sınıflandırılır [7, 8].

1. İzleyicinin geometrik şekline göre düz yüzey, toparlaklı, küresel izleyiciler.
2. İzleyici parçanın hareket şekline bağlı olarak öteleyen veya salınan izleyiciler.

Şekil 2.6'da ve Şekil 2.7'de izleyicilerin çeşitleri gösterilmiştir.

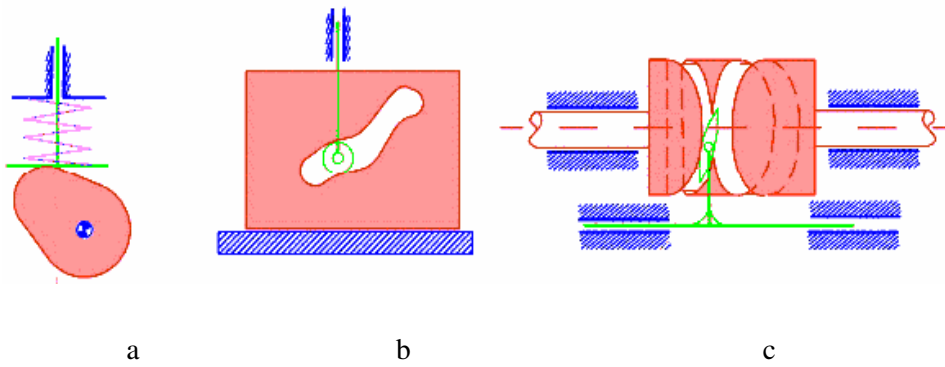


Şekil 2.6. Öteleme yapan izleyiciler [8].



Şekil 2.7. Salınım yapan izleyiciler [8].

Bir kam mekanizmasını tanımlarken Şekil 2.8’de olduğu gibi, verilmiş olan sınıflandırmalardan mümkün olduğunca fazlası verilmeye çalışılır. Örneğin; aksenal öteleme yapan, düz-yüzeyle izleyicili, kuvvet kapalı radyal kam (a) , toparlaklı öteleme yapan izleyicili, şekil kapalı, kamalı kam (b) veya öteleme yapan silindirik izleyicili silindirik kam (c) gibidir [8].



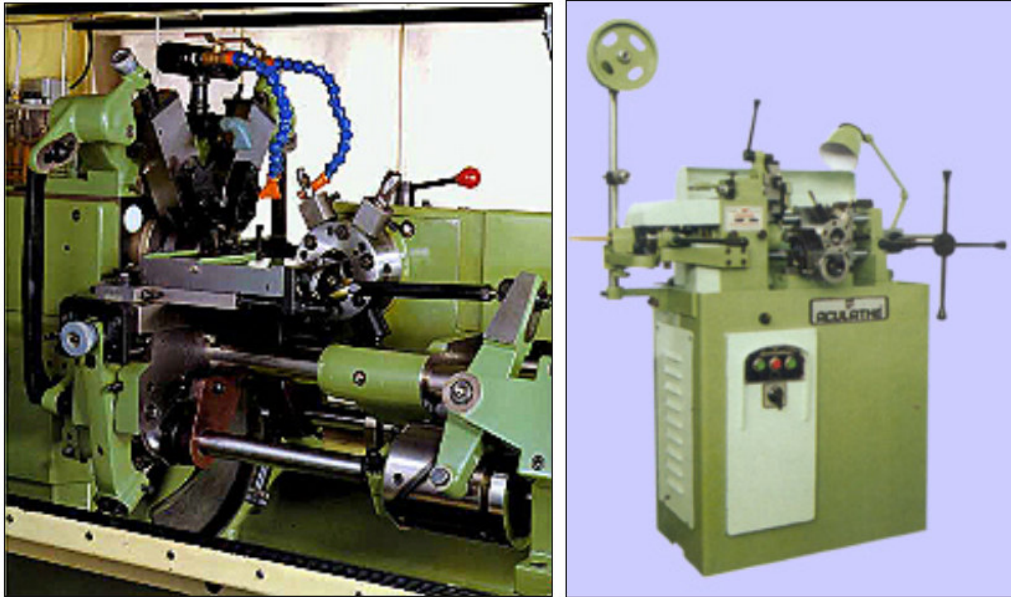
Şekil 2.8. Örnek kam mekanizmaları [8].

2.3. Kam Mekanizmalarının Kullanıldığı Yerler, Avantaj Ve Dezavantajları

2.3.1. Kam mekanizmalarının kullanıldığı yerler

Kam mekanizmaları düzgün dairesel bir hareketi, bir takipçi aracılığı ile doğrusal harekete çeviren ve mekanizmaya bir iş yaptırmaya yarayan makine elemanları olduğu için kullanım alanları çok geniştir. Kamlar, torna tezgahlarında zamanlama otomatik ayarı kumanda tertibatlarında, iş tezgahlarında, otomatik makinelerde, patlamalı ve yanmalı motorlarda emme ve eksoz supaplarının açılıp kapatılmasında, matbaa makinelerinde kağıtların alınmasında, dikiş makinelerinde zigzag işlerinde, bazı çamaşır makinelerinde yarım devir ve otomatik işlemlerinde, dokuma tezgahlarında desenlerin elde edilmesinde ve buna benzeyen bir çok alanda kullanılmaktadırlar [9].

Kam mekanizmalarının kullanıldığı otomat torna tezgahları Şekil 2.9'da verilmiştir.



Şekil 2.9. Otomat torna tezgahları [11, 12].

2.3.2. Kam mekanizmalarının avantajları

- Basit olmaları ve az yer kaplamaları sebebiyle yaygın olarak kullanılırlar.
- Başka tip mekanizmalarda yapılması güç olan, düzensiz, alışılmamış hareketlerin gerçekleştirilmesine olanak sağlarlar.

- Ayarlanabilir ve kolayca değiştirilebilirler [5, 6].
- Yüzeysel kam mekanizmaları uzun darbe gereksinimlerine ihtiyaç duyulan uygulamalar için çok uygundur [13].
- İstenilen izleyici hareketini basit yolla temin edebilir olmalarından sıklıkla kullanılırlar [14].

2.3.3. Kam mekanizmalarının dezavantajları

- İmalatı güç ve pahalıdır.
- Yüksek hız ve büyük yüklerde kritik durumlar ortaya çıkabilir.
- Temas noktalarında kolayca aşınma olur.
- Bu aşınma sayesinde profil toleransını kaybeder ve hareket kanunu istenilen limitler içerisinde yapılamaz [5, 6].
- Düşük derecede esneklik sunarlar [14].

2.4. Kamların İmalatı

Kullanılacak kamın hassasiyeti, kam mekanizmasını ilgilendiren en önemli hususlardan biridir. Sıhatsız yapılacak bir imalat mekanizmanın işlevsel kabiliyetini ciddi olarak engeller. Bu nedenle kam imalatı konusunda takım tezgahlarından faydalanmak veya özel üretim uygulamalarına yönelmek konusunda çalışmalar yapılmıştır.

Belirli bir kamın üretiminde kullanılan metot kam uygulamasının ve üretim miktarının bir fonksiyonudur. Basit bir bağlama kalıbı için üretilen kam ile yüksek hızlı bir tekstil makinesinin kamı doğruluk ve aşınma karakteristikleri açısından aynı özellikleri taşımayacaktır. Bu da üretim metodunu kesin şekilde belirleyen bir farktır. Biçim ve uygulamaya bağlı olarak kamın çevre profili, takım tezgahlarından biriyle veya özel dizayn edilmiş tezgahlardan biriyle imal edilebilmektedir. Genel olarak kamın çevre profilini imal etme yolları aşağıdaki gibidir.

1. Tesviye Usulü İmalat

- a) Profilin kısmen tezgah kullanımıyla ve el işçiliği ile elde edilmesidir.
- b) Profilin klasik takım tezgahlarıyla elde edilmesi.

2. Artışlı Kesme Yoluyla İmalat

- a) Çevresel işlemeyle profile yaklaşma.
- b) Kam yüzeyine teğet işlemeyle profile yaklaşma.

3. Kopya Kontrollü İmalat

- a) Şablon kamların kullanıldığı özel dizayn edilmiş mekanik sistemli tezgahlarda imalat.
 - b) Delikli bant kontrollü NC tezgahlarda imalat.
 - c) Bilgisayar kontrollü CNC tezgahlarda imalat.
4. Alevle kesme
 5. Dökme
 6. Dövme kalıplarıyla dövme
 7. Kalıpta basma [15].

3.KAM TASARIMI

3.1. Kam Terimleri

Bir kam profilini oluşturan çeşitli parametreleri tanımlayan kam terimleri Şekil 3.1’de verilmiştir. Bu terimlerin açıklamaları ise aşağıdaki gibidir.

- İz noktası : Bir düz yüzeyli izleyicide referans noktası veya makaralı izleyicinin merkezi.

- Kam profili : İzleyici ile temasta bulunan kamın çalışma yüzeyi.

- Temel daire : Dönme merkezinden kam yüzeyine teğet çizilebilen en küçük daire temel dairedir. Temel daire çapı, kam büyüklüğünü tespit eden yarıçap r_B ile gösterilir.

- İzleme eğrisi : Bu eğri, izleme noktasının yörüngesidir. Radyal nokta iticisi için, bu eğri kam yüzeyi ile çakışır.

- Birinci daire : Dönme merkezinden izleme eğrisine teğet olarak çizilebilen en küçük dairedir. Birinci daire yarıçapı r_A ile gösterilir.

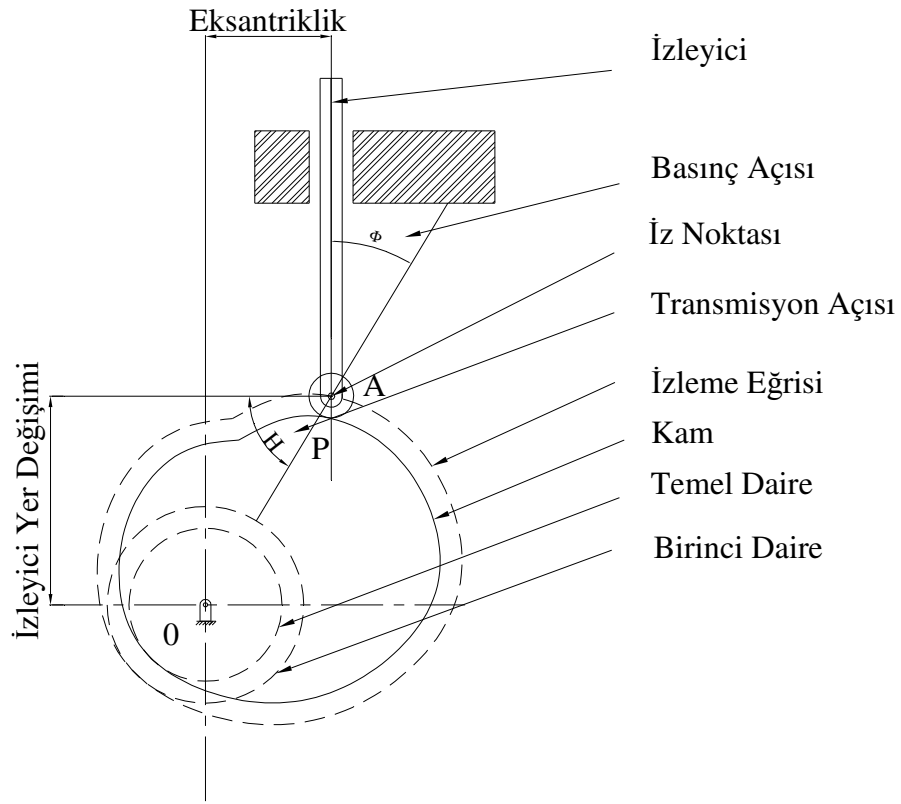
- İzleyici yer değişimi : Durma pozisyonundan itibaren, zamana veya kamın dönme açısına bağlı olarak, izleyicinin hareket miktarı.

- Basınç açısı : İzleme eğrisinin normal doğru ile aynı yerde, iticinin radyal doğrultusu arasındaki açıdır. Bu makaralı itici için normal eksen, makara merkezinden ve kam yüzeyinde temas noktasından geçer. Basınç açısı çok büyük olursa kamın yükselme kısmında, iticinin sıkışmasına neden olabilir.

- Eğrilik yarıçapı : Kam profilinin herhangi bir noktasında, kam eğrisine çizilen tanjant dairesinin yarıçapıdır.

- Eksantriklik : Kam ve izleyici merkezleri arasındaki uzaklık.

- Transmisyon açısı : Herhangi bir noktadaki izleme eğrisinin normali ile izleyici arasındaki açı [16].



Şekil 3.1. Kam terimleri [16].

3.2. İzleyicinin Hareketleri

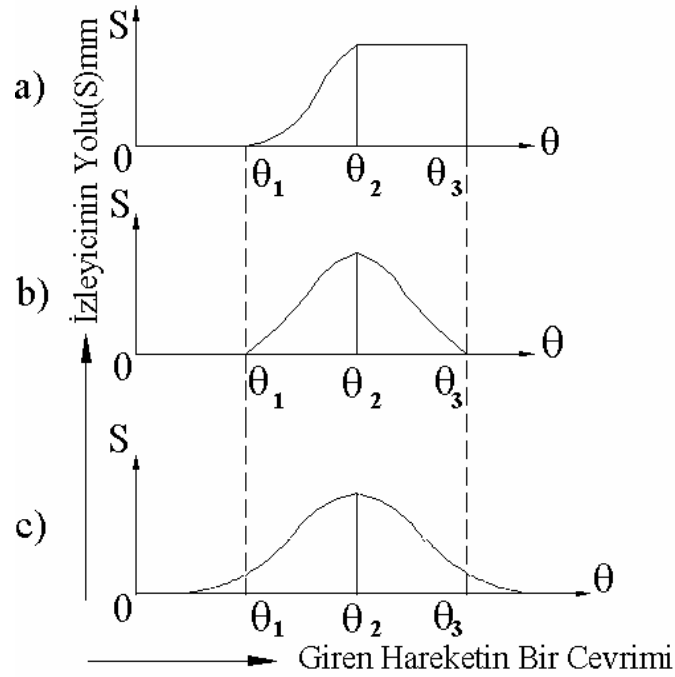
Sabit hızla dönen kam mili ile hareket alan kamın, bir çevrim boyunca dönmesi sırasında izleyici bazı hareketleri yapar. Bu hareketler:

- Bekleme : İzleyicinin hareketsiz kalması,
- Kalkış : İzleyicinin kam merkezinden uzaklaşması,
- İniş : İzleyicinin kam merkezine yaklaşması.

Pratikte en çok kullanılan izleyici hareketleri üç şekildedir. Bunlar ;

1. Bekleme-Kalkış-Bekleme (BKB)
2. Bekleme-Kalkış-İniş (BKİ)
3. Kalkış-İniş (Kİ)

Şekil 3.2’de izleyici hareketleri verilmiştir [16].

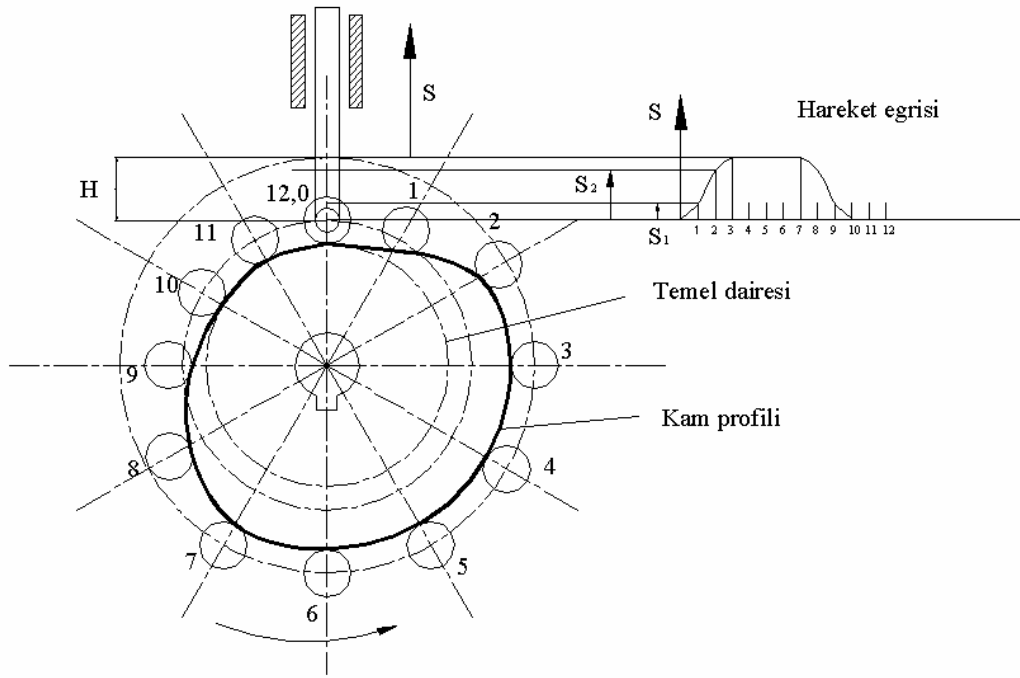


Şekil 3.2. İzleyici hareketleri a) BKB, b) BKİ, c) Kİ [16].

3.3. Kam Profilinin Grafik Olarak Belirlenmesi

İstenilen hareket eğrisini verecek olan kam profilinin grafik olarak belirlenmesi Şekil 3.3'de gösterilmekte olan örnek kam için açıklanacaktır. Kam profilini belirlemek için ilk olarak belirli bir toparlak çapı ve temel dairesi çapı belirlenmelidir. Temel dairesi yarı çapı (r_t) değeri kinematik olarak bağlama açısına göre bulunur. Toparlak çapı (r_r) ise genel olarak kama gelen yükler belirlendikten sonra belirlenir. Burada her iki değer bilindiği var sayılacaktır. Yarı çapı r_t+r_r olan bölüm dairesi çizilir. Hareket eğrisi ve bölüm dairesi aynı sayıda eşit aralıklarda bölünür. (Şekilde 30° aralıklar ele alınmıştır. Uygulamada, bilhassa yükseliş ve geri dönüş kısımlarında, bu aralığın istenilen hassasiyette elde edilebilmesi için çok küçük seçilmesi gereklidir). Kam profilinin belirlenmesinde kinematik yer değişim uygulanır. Bunun için kam sabit olarak kabul edilecek ve sabit parça kamın dönme yönünün tersine bağlı konumlar aynı kalacak şekilde döndürülecektir. Örneğin şekilde, kamın 30° saat yelkovanına ters yönde dönmesi bu kinematik yer değişim ile, sabit parça 30° saat yelkovanı yönünde dönmesidir ve izleyici eksen bu durumda dikey ile 30° açı yapmaktadır. Kam ile izleyici arasında aynı bağlı konumun korunabilmesi için bu arada izleyicinin bu yeni eksen yönünde hareket eğrisinde gösterilen s_1 kadar yukarıya öteleme yapması gerekir (İlk konumda toparlak merkezi kam merkezinden r_t+r_r kadar uzakta olduğundan 1 numaralı konumda toparlak merkezi kam

merkezinden $r_1+r_2+s_1$ kadar uzakta olacaktır). Bu durumda, kinematik yer deęiřime gre, izleyicinin yeni konumu belirlenmiř olur (kesik izgi ile izleyicinin konumu gsterilmiřtir). Benzer iřlem dięer konumlar iin yapıldıęında, kamın her dnme aısı iin toparlaęın kama gre baęlı konumu belirlenir. Kam profili toparlaęın tm baęlı konumlarına teęet olan dzgn eęridir. Őekil 3.3’de kam profilinin belirlenmesi gsterilmiřtir [8].

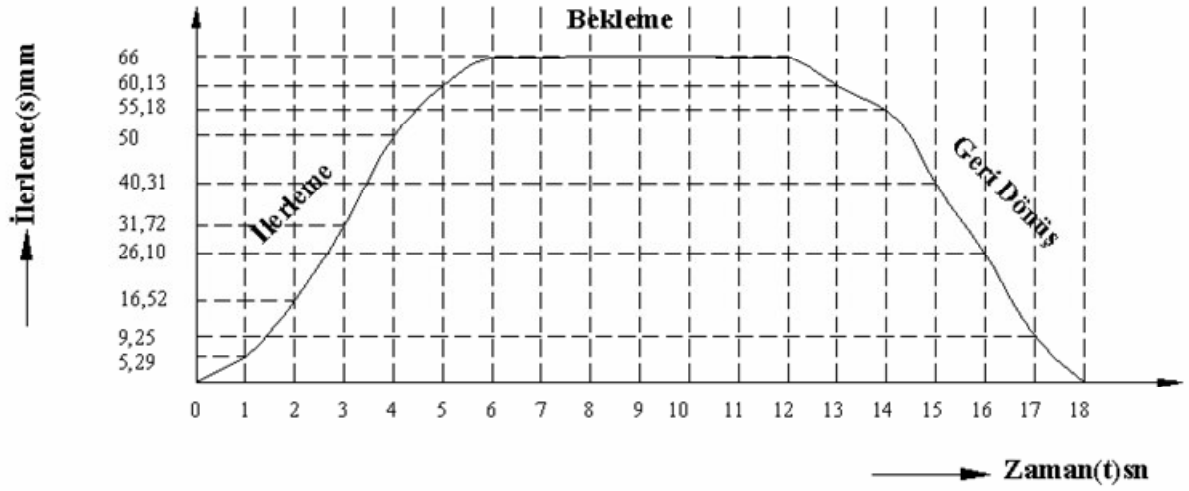


Őekil 3.3. Kam profilinin belirlenmesi [8].

3.4. Kinematik Parametrelerin Belirlenmesi

Kam tasarımında hareket diyagramının elde edilmesinden sonra bu diyagramdan yola ıkarak hız ve ivme diyagramları izilir ve kinematik parametreler tespit edilir. Ařaęıdaki rnekte adım adım gidilerek sırası ile tm diyagramlar izilmiř ve parametreler hesaplanmıřtır.

İzleyiciye (ikinci uzuv) ait hareket diyagramı ařaęıdaki gibi verilmiřtir.



Şekil 3.4. İzleyiciye ait hareket diyagramı

İzleyiciye ait konum değerleri aşağıdaki çizelgede belirtilmiştir.

Çizelge 3.1. İzleyiciye ait konum değerleri

Konum	İlerleme (Si) mm	Zaman (sn)
0	0	0
1	5,29	1
2	16,52	2
3	31,72	3
4	50	4
5	60,13	5
6	66,06	6
7	66,06	7
8	66,06	8
9	66,06	9
10	66,06	10
11	66,06	11
12	66,06	12
13	60,13	13
14	55,18	14
15	40,31	15
16	26,10	16
17	9,25	17
18	0	18

Kama ait diğer sabitler aşağıda verilmiştir.

Devir sayısı $n = 100$ dev/dk

Açısal hız $\omega = 10,472$ rad/sn

1 periyot için gerekli olan zaman

$$P = \frac{60sn}{100dev/dk} = 0.6sn \quad \Delta t = \frac{P}{18} = 0.0\bar{3}sn$$

Herhangi bir konum için hız değerleri aşağıdaki biçimde hesaplanabilir.

$$V_i = \frac{\Delta S}{\Delta t} mm / sn$$

$$V_{0.5} = \frac{S_1 - S_0}{\Delta t} = \frac{(5.29 - 0)}{0,0\bar{3}} = 158.701 mm / sn$$

$$V_{1.5} = \frac{S_2 - S_1}{\Delta t} = \frac{(16.52 - 5.29)}{0,0\bar{3}} = 336.9 mm / sn$$

$$V_{2.5} = \frac{S_3 - S_2}{\Delta t} = \frac{(31.72 - 16.52)}{0,0\bar{3}} = 456.004 mm / sn$$

İvme değerlerinin hesaplanması aşağıda verilmiştir.

$$a_i = \frac{\Delta V}{\Delta t} mm / sn^2$$

$$a_1 = \frac{V_{1.5} - V_{0.5}}{\Delta t} = \frac{(336.9 - 158.701)}{0,0\bar{3}} = 5345.975 mm / sn^2$$

$$a_2 = \frac{V_{2.5} - V_{1.5}}{\Delta t} = \frac{(456.004 - 336.9)}{0,0\bar{3}} = 3573.477 mm / sn^2$$

Yukarıdaki işlemler tüm zamanlar için uygulanabilir ve bu şekilde hız ve ivme değerleri hesaplanabilir.

Kinematik parametrelerin hesaplanmasında grafiksel türev alma metodu kullanılmaktadır. Bu metot uygulanırken; yol zaman grafiğinde her bir konum için düşey doğrultuda işaretlenmiş noktalar bir önceki ile birleştirilerek, bu noktalara ait farklı eğimlerdeki doğrular elde edilir. Türev 0-1,1-2,2-3 ... konumları arasında alındığından, hızlar 0.5, 1.5, 2.5, noktalarında bulunur [5, 6].

Hız zaman grafiği çizilirken zaman koordinatlarının negatif bölgesinde ve aynı doğrultusunda H mesafesinde bir O_1 noktası belirlenir. Yol zaman grafiğinde çizilen eğimlerin paralelleri O_1 noktasına taşınır ve hız eksenine ile kesiştirilir. Daha sonra zaman eksenine paralel çizgiler çizilir ve zaman ekseninden çizilen dik doğrularla kesiştirilir. Bulunan noktalar o konumlardaki hız değerleridir.

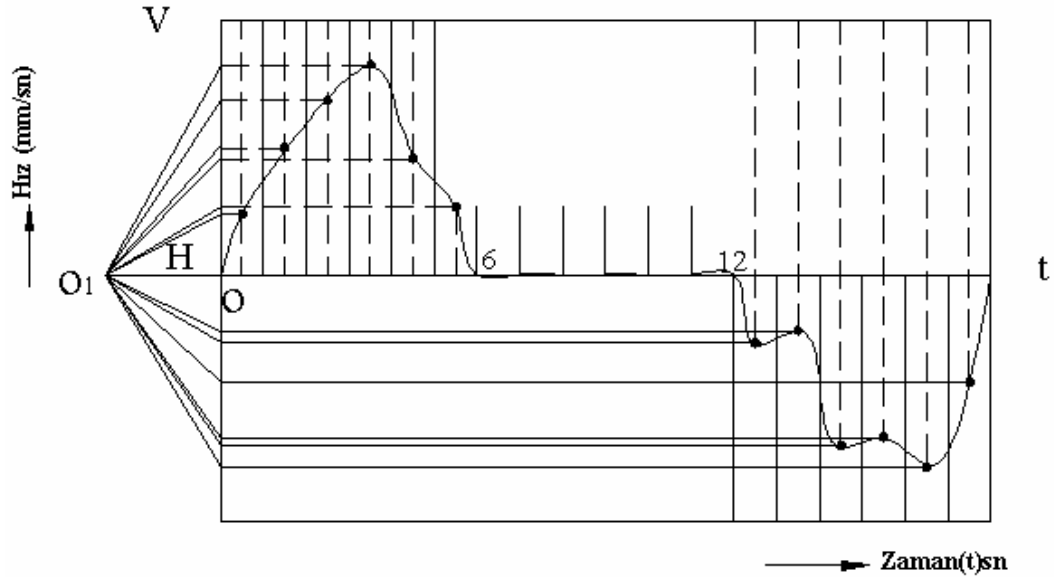
Başlangıç ve bitiş noktalarındaki hız değerleri sıfır olduğu kabul edilir ve bütün noktalar birleştirilerek hız grafiği elde edilir. Hız grafiğindeki değerler K_v hız ölçeği ile çarpılarak bütün konumlardaki gerçek hız değerleri bulunur. K_v hız ölçeği aşağıdaki formül ile hesaplanır.

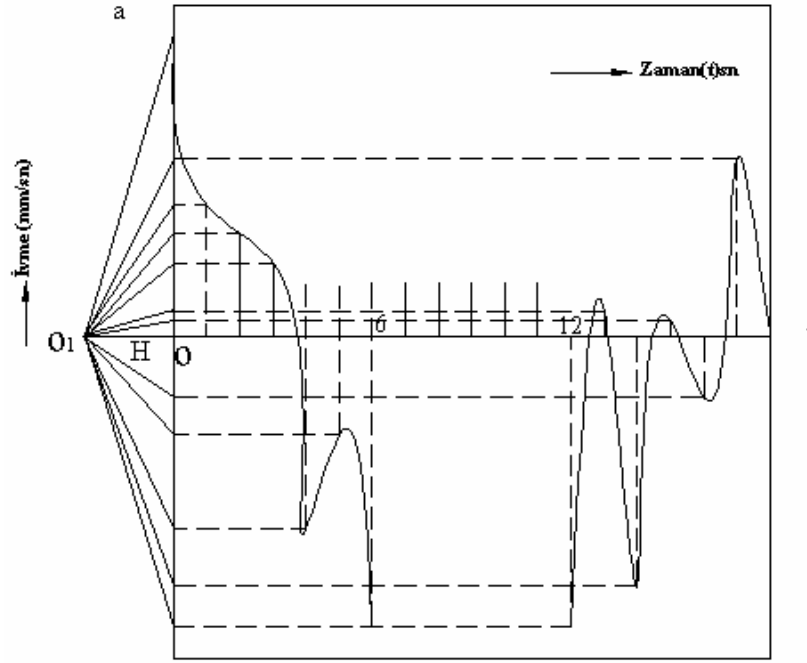
$$K_v = \frac{K_s}{K_t.H} \dots\dots\dots (1)$$

Hız zaman grafiğindeki $V_{0.5}$, $V_{1.5}$, $V_{2.5}$ noktaları birer doğru ile birleştirilir. Hız zaman grafiğindeki gibi keyfi bir O_2 noktası seçilir. Hız zaman grafiğinde çizilen doğruların paralelleri O_2 noktasına taşınır ve ivme eksenine ile kesiştirilir. Daha sonra zaman eksenine paralelleri çizilir ve zaman eksenindeki noktalardan dik doğrular çıkılarak kesiştirilir. Bulunan noktalar ivme değerlerini göstermektedir. Grafikten okunan değerler K_a ivme ölçeği ile çarpılarak gerçek ivme değerleri bulunur. K_a ivme ölçeği aşağıdaki formül ile hesaplanır.

$$K_a = \frac{K_v}{K_t.H} [5, 6] \dots\dots\dots (2)$$

Hız -zaman ve ivme -zaman grafikleri aşağıda Şekil 3.5’de görülmektedir.

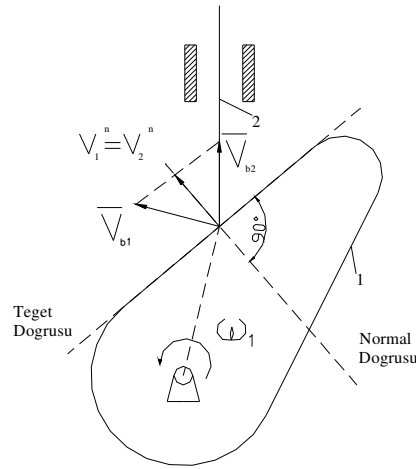




Şekil 3.5. İzleyiciye ait hız ve ivme grafikleri

3.5. Kam Mekanizmalarının Çalışmasının Kinematik Şartı

Kam mekanizmaları kam ve iticinin birlikte temas halinde olmasıyla meydana gelmektedir. Kam mekanizmalarında amaç önceden belirlenmiş hareketin sürekli olarak hatasız bir şekilde sağlanmasıdır. Şekil 3.6'da kam mekanizmasının çalışmasının dinamik şartı gösterilmiştir [5, 6].



Şekil 3.6. Kam mekanizmalarının çalışmasındaki kinematik şartın gösterimi [5,6].

- V_1 : 1. uzvun temas noktasındaki hız vektörü.
 V_2 : 2. uzvun temas noktasındaki hız vektörü.
 V_1^n : V_1 hızının temas noktasındaki normal bileşeni.
 V_2^n : V_2 hızının temas noktasındaki normal bileşeni.
 ω_1 : Kamın açısal hızı

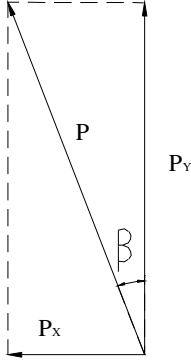
Kam mekanizmasının görevini tam olarak yerine getirebilmesi için, çalışmasının her anında kam ile çıkış uzvu sürekli olarak temas halinde bulunmalıdır. Sürekli temasın kesilmesi çıkış uzvunun hareket diyagramının değişmesine neden olur ve kam mekanizması kendisinden istenilen görevi yerine getirememeye başlar. Bunun önüne geçilebilmesi için her iki uzuv arasında temasın sürekliliğini sağlayacak konstrüktif tedbirlere başvurulur. Kam mekanizmalarında kam ile, çıkış uzvunun sürekli temasını sağlayabilmek için bu uzuvların temas noktalarındaki hızlarının normal bileşen vektörlerinin eşit olması gerekmektedir. Şekil 3.6'da bu hızların normal bileşenlerinin eşit olduğu görülmektedir. Bu şartı sağlayabilmek için bazen geometrik şekillerde değişiklik yapılır bazen de kuvvet kullanılır [5,6].

3.6. Kam Mekanizmalarının Çalışmasının Dinamik Şartı

Kam mekanizmalarında çalışma esnasında uzuvlar arasında bir temas söz konusu olduğundan uzuvlar arasında sürekli olarak bir kuvvet aktarımı gerçekleşmektedir. Bu kuvvet aktarımı sürekli olduğundan dolayı çalışma esnasında dinamik kuvvetlerin sebep olduğu sorunlar çıkabilmektedir. Bu nedenle dinamik sorunların daha tasarım aşamasında iken göz önüne alınması ve asgari seviyede tutulabilmesi için kam mekanizmalarındaki dinamik olayların incelenmesi gerekmektedir. Dinamik şart incelenirken kam mekanizmaları çıkış uzvu öteleme ve sarkaç hareketi yapan kam mekanizmaları olmak üzere iki grupta ele alınacaktır [5, 6]

3.6.1. Çıkış uzvu öteleme hareketi yapan kam mekanizmalarında dinamik şart

Kam mekanizmalarının sentezinde dinamik olarak kam mekanizmalarının özelliklerinin belirlenmesi önemli yer tutar. Dinamik özelliklerin tasarım aşamasında kontrol altında tutulabilmesi için basınç açısı parametresinin belirli sınırları aşmaması gerekir. Dinamik etkiler daha tasarım aşamasında ele alındığı takdirde kam mekanizmasının gerçek çalışması esnasında olacak olaylar daha sağlıklı bir biçimde incelenecektir. Şekil 3.7' de çıkış uzvu öteleme hareketi yapan kam mekanizmasında dinamik şartın gösterimi verilmiştir.



$\beta = 0^\circ$ olduğunda $P_Y = P$ olur.

B açısı büyüdükçe P_X kuvvetinin değeri artacağından kayıp kuvvet miktarı artacak ve bu yüzden mekanizmanın verimi düşecektir.

$$\text{Verim; } \eta = \frac{\Delta h}{\Delta f} = \frac{\bar{q} \cdot \Delta s}{P_Y \cdot \Delta s} \text{ biçiminde ifade edilebilir.} \dots\dots\dots (3)$$

Δh = hareket işi

Δf = faydalı iş

Bilinmeyen kuvvetleri bulmak ve P ile Q arasındaki ilişkiyi görebilmek için statik denge denklemleri yazılır.

$$\Sigma F_x = 0 \Rightarrow N_1 - N_2 + P \cdot \sin \beta = 0 \dots\dots\dots (4)$$

$$\Sigma F_y = 0 \Rightarrow P \cdot \cos \beta - T_1 - T_2 - Q = 0 \dots\dots\dots (5)$$

$$\Sigma M_b = 0 \Rightarrow N_2 \cdot y - P \cdot \sin \beta \cdot L = 0 \dots\dots\dots (6)$$

$$3. \text{ denklemden } N_2 \text{ çekilirse; } N_2 = \frac{P \cdot \sin \beta \cdot L}{y} \dots\dots\dots (7)$$

N_2 1. denklemde yerine konulursa;

$$N_1 = P \cdot \sin \beta \left(\frac{L}{y} + 1 \right) \dots\dots\dots (8)$$

$$T_1 = \mu \cdot N_1 \quad T_2 = \mu \cdot N_2$$

μ = sürtünme katsayısı

$$T_1 = \mu \cdot P \cdot \sin \beta \left(\frac{L}{y} + 1 \right) \dots\dots\dots (9)$$

$$T_2 = \mu \cdot P \cdot \sin \beta \frac{L}{y} \dots\dots\dots (10)$$

T_1 ve T_2 2. denklemde yerine yazılırsa;

$$P \cdot \cos \beta - \left[\mu \cdot P \cdot \sin \beta \cdot \left(\frac{L}{y} + 1 \right) \right] - \left[\mu \cdot P \cdot \sin \beta \cdot \frac{L}{y} \right] - Q = 0$$

$$P = \frac{Q}{\cos \beta - \left[\mu \cdot \sin \beta \cdot \left(\frac{L}{y} + 1 \right) \right] - \left[\mu \cdot \sin \beta \cdot \frac{L}{y} \right]}$$

$$P = \frac{Q}{\cos \beta - \mu \cdot \sin \beta \cdot \left(\frac{2 \cdot L}{y} + 1 \right)} \dots \dots \dots (11)$$

P hareket kuvvetinin ifadesinde paydanın sıfır olması hali mekanizmanın çalışmasındaki kritik bir durumu gösterir. Payda sıfır olursa P hareket kuvveti sonsuza gideceği için kam mekanizmasının sağlam kalması mümkün değildir.

$$\cos \beta_{kr} - \mu \cdot \sin \beta_{kr} \cdot \left(\frac{2 \cdot L}{y} + 1 \right) = 0 \quad \beta_{kr} = \text{kritik basınç açısı} \dots \dots \dots (12)$$

$$\cos \beta_{kr} = \mu \cdot \sin \beta_{kr} \cdot \left(\frac{2 \cdot L}{y} + 1 \right)$$

$$\tan \beta_{kr} = \frac{1}{\mu \cdot \left(\frac{2 \cdot L}{y} + 1 \right)} \quad \mu = 0,1 \quad \text{ve} \quad L = 2 \cdot y \quad \text{olduğu kabul edilirse;}$$

$$\tan \beta_{kr} = 2$$

$$\beta_{kr} = 63,30^\circ \quad \text{olarak hesaplanır.}$$

Basınç açısının kritik basınç açısı değerine ulaştığı durumda Q iş kuvvetini karşılayabilmek için $P = \infty$ olması gereklidir. Bu nedenle Q'un değerinin çok küçük olduğu durumlarda bile kam mekanizmasının çalışması mümkün olmaz. Bu durumun önüne geçebilmek için β açısının değeri tasarım aşamasındayken kritik açı değerinden küçük seçilmesi gereklidir.

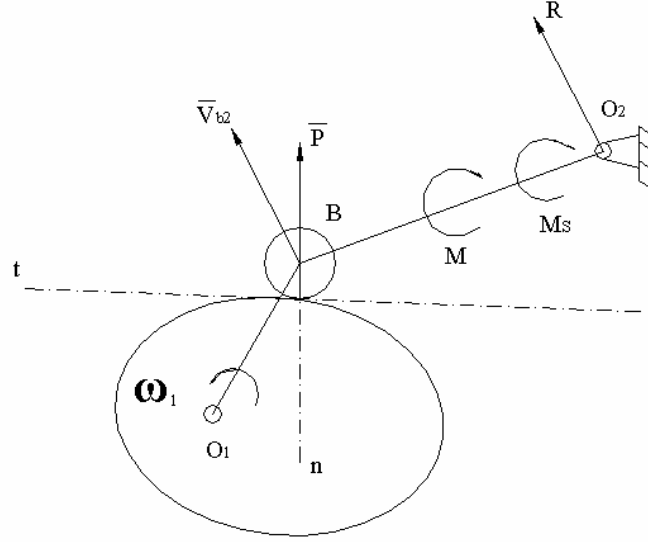
$$B \leq \beta_o < \beta_{kr} = 63,30^\circ$$

$\beta_o =$ olabilecek basınç açısı

Çıkış uzvu öteleme hareketi yapan kam mekanizmalarının tasarımında tavsiye edilen basınç açısı değeri yaklaşık olarak $\beta_o = 45^\circ$ kabul edilebilir [5, 6].

3.6.2. Çıkış uzvu sarkaç hareketi yapan kam mekanizmalarında dinamik şart

Çıkış uzvu sarkaç hareketi yapan kam mekanizmalarının dinamik şartı, öteleme hareketi yapan kam mekanizmalarıninkine benzer şekilde bulunmaktadır. Şekil 3.8’ de çıkış uzvu sarkaç hareketi yapan kam mekanizmasında dinamik şartın gösterimi verilmiştir.



Şekil 3.8. Çıkış uzvu sarkaç hareketi yapan kam mekanizmasında dinamik şartın incelenmesi [5,6].

Şekildeki harflerin açılımı aşağıdaki gibidir:

M: İş momenti

M_s : Mafsaldaki sürtünme momenti

P: Kamın çıkış uzvuna etki ettirdiği hareket kuvveti

R: Mafsal kuvveti

V_{b2} : Çıkış uzvunun temas noktasındaki hızı

β : basınç açısı

$L = |O_2B|$

Bilinmeyen kuvvetlerin bulunması , P ve Q kuvvetleri arasındaki ilişkiyi görmek için Şekil 3.8’den statik denge denklemleri yazılır.

$$\Sigma F_x = 0 \Rightarrow P + R = 0 \Rightarrow P = -R \dots\dots\dots (13)$$

$$\Sigma M(O_2) = 0 \Rightarrow P.L \cos \beta - M_s - M = 0 \dots\dots\dots (14)$$

$$M_s = P \cdot \mu \cdot \frac{d}{2}$$

μ : Sürtünme katsayısı

d : Mafsal çapı

$$P = \frac{M}{L \left(\cos \beta - \mu \cdot \frac{d}{2L} \right)} \dots \dots \dots (15)$$

Kritik açığı bulmak için payda sıfıra eşitlenirse ;

$$L \left(\cos \beta - \mu \cdot \frac{d}{2L} \right) = 0$$

$$\cos \beta = \mu \cdot \frac{d}{2L} \dots \dots \dots (16)$$

Eğer;

$$\mu = 0.1 \quad \text{ve} \quad \frac{d}{2L} = 0.1 \quad \text{ise}$$

$$\cos \beta_{kr} = 0.005$$

$\beta_{kr} = 89.7^\circ$ olarak hesaplanır.

$\beta < \beta_o < \beta_{kr} = 89.7^\circ$ olmalıdır.

Buna göre, çıkış uzvu sarkaç hareketi yapan kam mekanizmaları için olabilecek basınç açısı β_o in 60° alınması tavsiye edilebilir. Bu şekilde tasarımı yapılan kam mekanizmalarının dinamik sorunları daha az ve verimi daha yüksek olacaktır [5, 6]

3.7. Çıkış Uzvu Öteleme Hareketi Yapan Kam Mekanizmasının Sentezi Yapılarak Ani Dönme Merkezlerinin Bulunması Ve Kam Profilinin Çizilmesi

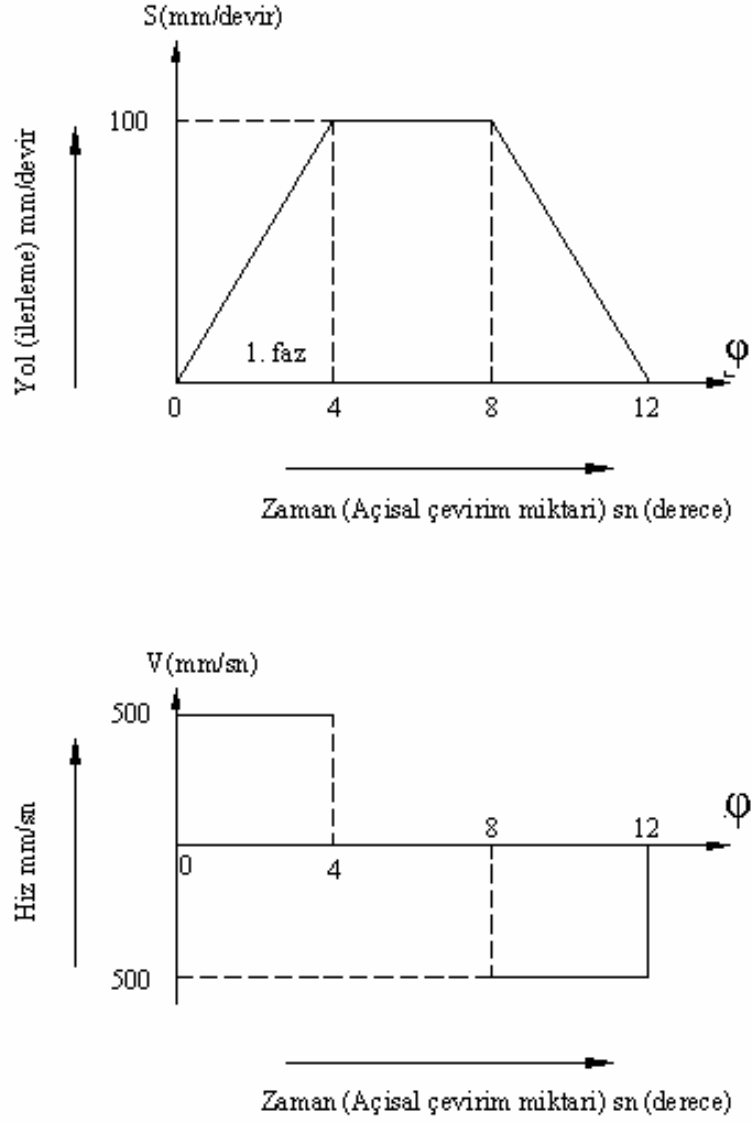
Bu konu aşağıdaki örnekte açıklanmıştır. Gerekli olan değerler aşağıdaki Çizelge 3.2’de verilmiştir.

Çizelge 3.2. Verilenler çizelgesi

Sembol	Değeri	Birimi	Açıklama
n_1	100	dev/dak	Kama ait devir sayısı
β_o	45	($^\circ$) derece	Olabilecek basınç açısı
E	30	Mm	Eksantrisite miktarı (çıkış uzvunun sağında)

$V_{b_2}^n = V_{b_1}^n$ uzuvlarının temas noktasındaki hızlarının normal bileşenleri eşittir.

Çıkış uzvuna ait yol – zaman ve hız – zaman diyagramları Şekil 3.9’ da verilmiştir.



Şekil 3.9. Yol-zaman ve hız-zaman diyagramları

1. faz : Sabit hızlı çıkış fazı

2. faz : Bekleme fazı

3. faz : Sabit hızlı iniş fazı

Kamın bir devri için geçen süre $T = 60/100 = 0.6$ sn

Fazların tamamı eşit uzunlukta olduğundan dolayı $\Delta t = 0.6/3 = 0.2$ sn 'dir.

$$\omega = \frac{n_1 \cdot 2\pi}{60} = 10.472 \text{ rad / sn}$$

$$V_0 = V_4 = \frac{S_4 - S_0}{t_4 - t_0} = \frac{\Delta S}{\Delta t} = \frac{100}{0.2} = 500 \text{ mm/sn}$$

$$V_{12} = V_8 = \frac{S_{12} - S_8}{t_{12} - t_8} = \frac{\Delta S}{\Delta t} = \frac{-100}{0.2} = -500 \text{ mm/sn}$$

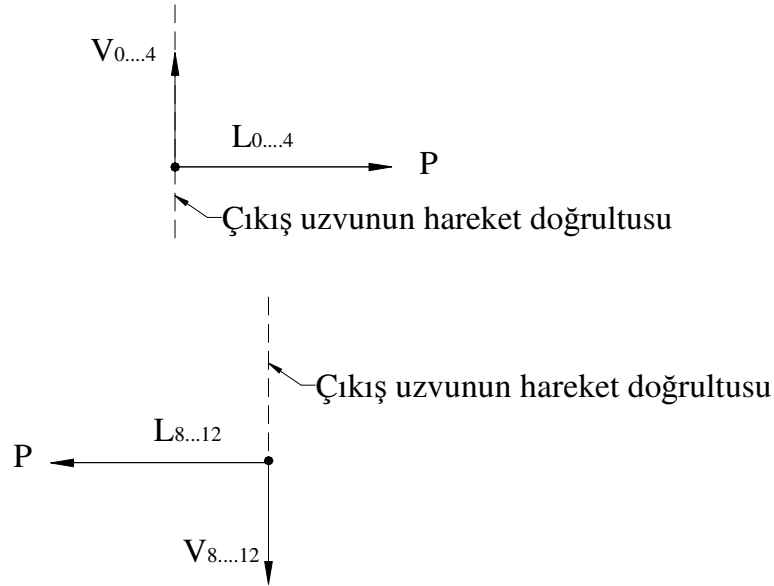
çıkış ve iniş fazlarındaki sabit hız değerleri hesaplanır.

Ani dönme merkezleri yörüngesinin çizilebilmesi için gerekli olan L değerleri hesaplanır.

$$L = \frac{V_i}{\omega} \dots\dots\dots (17)$$

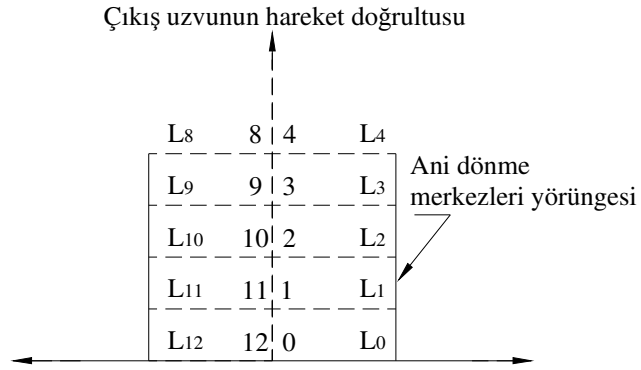
$$L_{0...4} = \frac{500}{10.472} = 47.75 \text{ mm} \quad \text{Çıkış uzvunun sağ tarafında.}$$

$$L_{0...4} = \frac{-500}{10.472} = -47.75 \text{ mm} \quad \text{Çıkış uzvunun sol tarafında.}$$



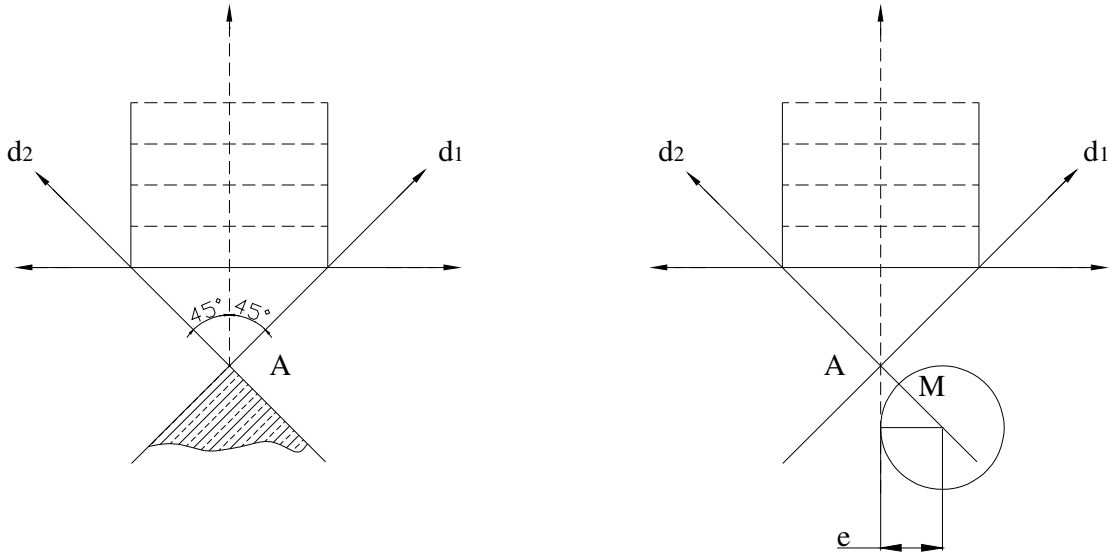
L değerleri hesaplandıktan sonra ani dönme merkezleri yörüngesi Şekil 3.10' daki gibi çizilir. Çıkış uzvunun hareket doğrultusuyla $\beta_0 = 45^\circ$ açı yapacak bir doğru çizilir ve ani dönme merkezleri yörüngesiyle temas ettirilir ve bu işlem iniş ve çıkış hareketleri için tekrarlanır. Çizilen doğruların kesim noktaları bulunur. Bu noktanın altında kalan alanda kam dönme

merkezi vardır. Bu alanın dışında alınacak bir nokta olabilecek basınç açısından büyük olacaktır ve bu istenmeyen bir durumdur.



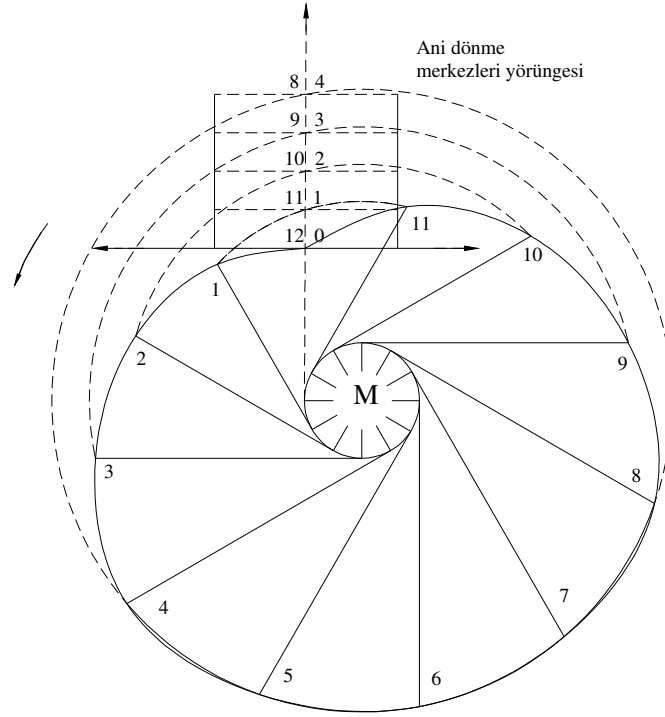
Şekil 3.10. Ani dönme merkezleri yörüngesinin çizimi [5, 6].

Kamın dönme merkezi olarak seçilecek noktanın yeri kam profilinin boyutlarını direkt olarak etkilediğinden eksantrisitenin değeri de göz önüne alınarak dönme merkezi doğruların kesim noktasına en yakın biçimde seçilmelidir. Şekil 3.11' de doğruların çizimi ve dönme merkezinin seçimi gösterilmektedir.



Şekil 3.11. Dönme merkezinin bulunması [5, 6].

Kamın dönme merkezi bulunduktan sonra izafi geri döndürme metodu kullanılarak kam profili çizilir. Kam profilinin çizimi Şekil 3.12' de gösterilmiştir [5, 6].



Şekil 3.12. Kam profilinin çizimi [5, 6].

4.KAM TASARIM PROGRAMININ HAZIRLANMASI

4.1. Kam Mekanizmaları Üzerine Yapılan Çalışmalar

Q. Yu ve H. P. Lee: Yapmış oldukları çalışmada toparlaklı izleyicili bir kam mekanizmasının boyutlarının optimizasyonu probleminin analizini yapmışlardır. Bu analizin yapımında hareket eğrisinin tanımlanmasında parametrik polinomların bir ailesi kullanılmıştır. Kullanmış oldukları metot kinematik özellikler, ivme, hız, hareket eğriliğinin simetrikliği, kam profilinin eğriliği gibi çeşitli tasarım gerekliliklerini içermektedir. Hazırladıkları bu tasarım metodunun kamın sentezi için mevcut olan tasarım metotları için bir alternatif olduğunu savunmaktadırlar [17].

M. Nishioka ve T. Nishimura:Yapmış oldukları çalışmada benzer kam mekanizmalarının her biçimini kapsayabilen parametrik bir formülasyon türetmişlerdir. Parametrik analizin sonucu olarak ta yeni bir mekanizma bulmuşlardır. Buldukları bu kam aslında dahili bir kam mekanizmasıdır. Mekanizmanın temel şekli hem basınç açısı hem de alttan kesilme kısıtlamaları için, toparlaklı izleyicinin eşit olarak dağıtılma tahminine dayanarak türetmişlerdir. Sonuç olarak benzer geleneksel mekanizmalardan daha avantajlı bir mekanizma elde ettiklerini ileri sürmektedirler [13].

O. Navaro ve diğerleri: Düzlemsel kam mekanizmalarının optimizasyonu üzerine bir çalışma yapmışlardır. Yapmış oldukları bu çalışmada düzlemsel kam mekanizmalarının optimizasyonu için basınç açısı sınırlamaları altında prosedürler ileri sürmüşlerdir. Ayrıca kullanıcıların optimizasyon yapabilecekleri bir grafik kullanıcı arayüzünü tanıtmışlardır. Öteleme ve salınımlı izleyicilerle düzlemsel kam mekanizmalarının boyutlarının minimize edilmesi için prosedürleri tanıtmışlardır. Bu prosedürler güvenli basınç açısı sınırlıkları ve makul en küçük temel daire yarıçapıyla bir kamın tasarımına izin vermektedir [3].

Tarık Küçük: Yapmış olduğu çalışmada kam mekanizmalarının teorisinden yola çıkarak tasarım zorlukları ve bu zorlukları aşmada bilgisayar kullanımının gerekliliklerini anlatmıştır. Ayrıca kam tasarımı için Delphi 3 programlama dili kullanarak bir tasarım programı hazırlamıştır.[6]

Günay Emirdağ: yapmış olduğu çalışmada yüksek hızlı kam mekanizmalarının kinematik karakteristiklerini irdelemiş ve Fortran-IV programlama dili kullanarak bir tasarım programı hazırlamıştır.[16]

4.2. Programın Hazırlanması

Tasarım programı, çıkış uzvu öteleme hareketi yapan kam mekanizmaları için yazılmıştır. Bu nedenle çıkış uzvu öteleme hareketi yapan kam mekanizmalarının geleneksel tasarımlarında kullanılan grafiksel metottaki matematiksel işlemler bilgisayar ortamına aktarılmıştır. Normalde tasarımcının yapması gereken bu işlemlerin bilgisayar ortamına aktarılması ile tasarımın hatasız ve hızlı olması hedeflenmiştir.

Kam tasarım programının hazırlanmasında Borland Delphi 7 programlama dili kullanılmıştır. Bu programlama dilinin tercih edilmesinin nedeni görsellik olarak kullanıcıya daha fazla imkân vermesidir.

Programın yazılmasında kam mekanizmalarının sentezinde uygulanan geleneksel işlemlere bağlı kalmıştır. Bu işlemler aşağıdaki gibi sıralanabilir:

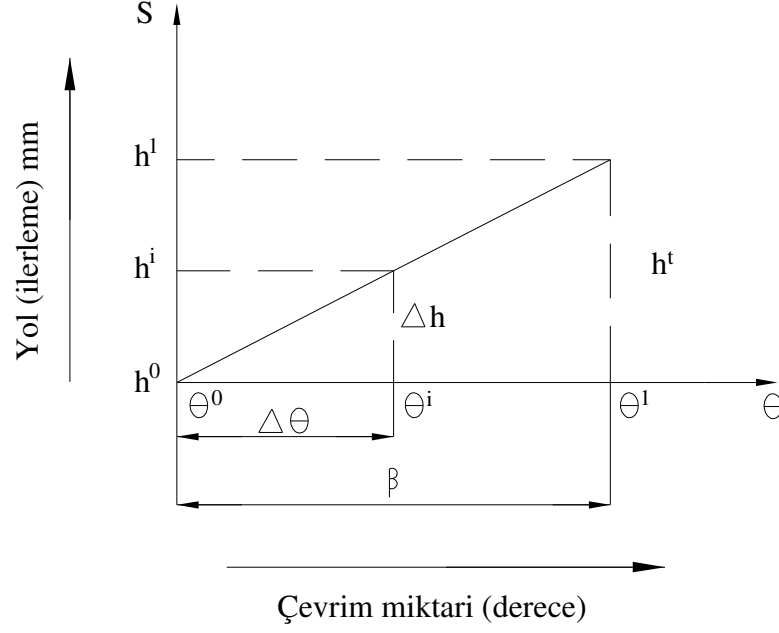
1. Hareket (yol-zaman) grafiğinin çizilebilmesi için gerekli olan verilerin belirlenmesi ve grafiğin çizilmesi,
2. Hareket grafiğine göre hız ve ivme grafiklerinin çizilmesi,
3. Giriş parametrelerinin belirlenmesi ve bunlara bağlı olarak kam profilinin çizilmesi.

4.2.1 Hareket Grafiğinin Çizilmesi

Hareket grafiğinin belirlenmesi işlemi kam tasarımının ilk ve en önemli adımudur. Bu nedenle hatasız olması gerekmektedir. Bu işlemin hatasız olması için aşağıdaki verilerin belirlenmesi gerekmektedir.

1. Faz sayısı,
2. Her bir fazın hareket tipi ve hareket kanunu,
3. Her bir fazın toplam ilerleme ve dönme miktarları.

Hareket grafiğinin bilgisayar ortamında çizilebilmesi için her bir faza ait hareket denkleminin belirlenmesi gerekmektedir. Örneğin bir fazda çıkış uzvunun sabit hızlı hareket etmesi istenilirse bu faz için sabit hızlı hareket denklemi kullanılır. Şekil 4.1.'de sabit hızlı hareket grafiği verilmiştir.



Şekil 4.1. Sabit hızlı hareket grafiği

Şekil 4.1.'deki sabit hızlı hareket grafiğinin denklemi aşağıdaki gibidir.

$$S = h_0 + h_t \left(\frac{\Delta\theta}{\beta} \right) \dots\dots\dots (18)$$

Her bir faz için seçilen hareket denkleminin 1. türevi hız denklemini, 2. türevi ise ivme denklemini verir. Hız ve ivme değerleri önceki adım ile aradaki yol farkından hesaplanabilir. Türevleri hesaplamak çokda gerekli değildir. Programda yol denklemlerinin çözümü yapılmış ve daha sonra belirtilen yöntemle hız ve ivme değerleri hesaplanarak grafikleri çizilmiştir.

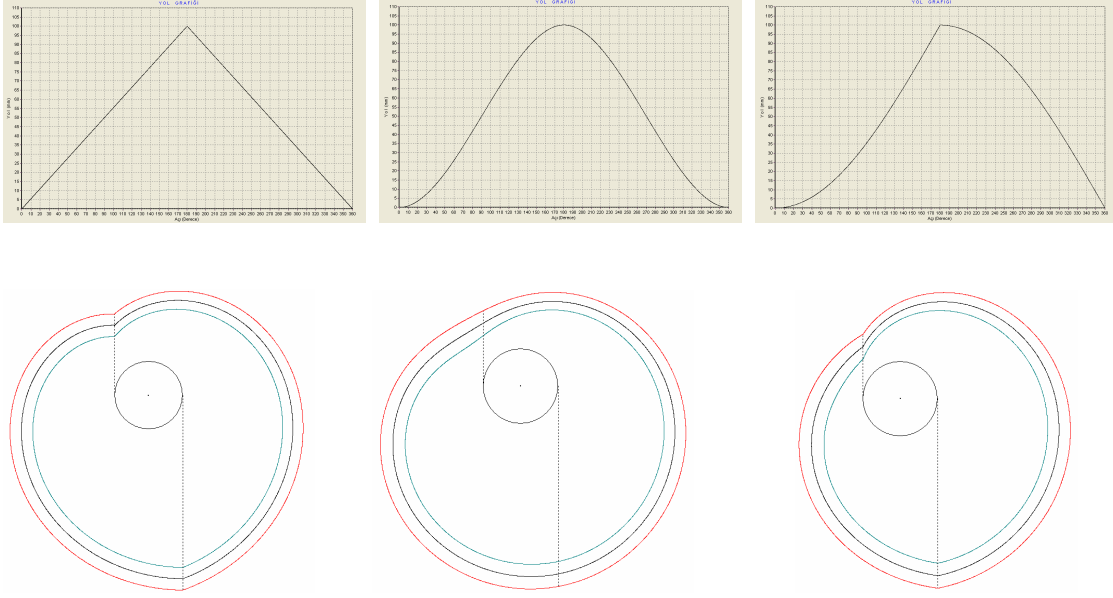
4.2.2. En Çok Kullanılan Hareket Denklemleri ve Farklı Hareket Denklemlerinin Kam Profiline Olan Etkisi

Kam mekanizmalarının tasarımlarında en çok tercih edilen hareket denklemleri:

1. Sabit hızlı hareket,
2. Sabit ivmeli hareket,
3. Basit harmonik hareket,
4. Sikloidal hareket denklemleridir.

Kam mekanizmalarının tasarımında kullanılan hareket denklemleri Ek 1'de verilmiştir.

Farklı hareket denklemleri ile tasarımları yapılmış aynı istenilen hareketi üretebilecek değişik kam profilleri aşağıda Şekil 4.2'de gösterilmiştir. Şekiller incelendiğinde hareket denklemlerinin yol grafiklerine ve yol grafiklerinin de kam profillerine etkileri görülmektedir.



Şekil 4.2. Hareket denklemlerinin kam profiline etkisi.

4.2.3. Ani Dönme Merkezlerinin ve Kam Dönme Merkezinin Belirlenmesi İle Kam Profiline Çizilmesi

Ani dönme merkezleri (ADM) yörüngesine ait noktalar ADM [i] dizisi olarak gösterilirse bu dizinin her bir elemanı kartezyen koordinat sisteminde bir noktaya tekabül eder. Bu noktaların bulunmasında kama ait hız değerleri kullanılır. Aşağıdaki formülde hız değerlerinin nasıl kullanıldığı gösterilmiştir [6].

$$ADM[i].x = \frac{Vg[i]}{\omega_1} \text{ mm} \quad \text{ya da} \quad ADM[i].x = \frac{Va[i]}{\omega_i} \dots\dots\dots (19)$$

$$ADM[i].y = S[i] \text{ mm} \dots\dots\dots (20)$$

Ani dönme merkezleri yörüngesinin belirlenmesinden sonra bu yörüngelere teğet doğrular çizilmelidir. Bu doğruların çizilebilmesi için ADM [i] dizisi ADM1 [i] ve ADM2 [i] diye iki alt diziyeye ayrılır. Bu alt diziler aşağıdaki gibi gösterilebilir.

ADM [i].x > 0 ise

ADM1 [i].x = ADM [i].x ve ADM1 [i].y = ADM [i].y olur.

ADM [i].x < 0 ise

ADM2 [i].x = ADM [i].x ve ADM2 [i].y = ADM [i].y olur. Bu şekilde alt diziler elde edilmiş olur [5,6].

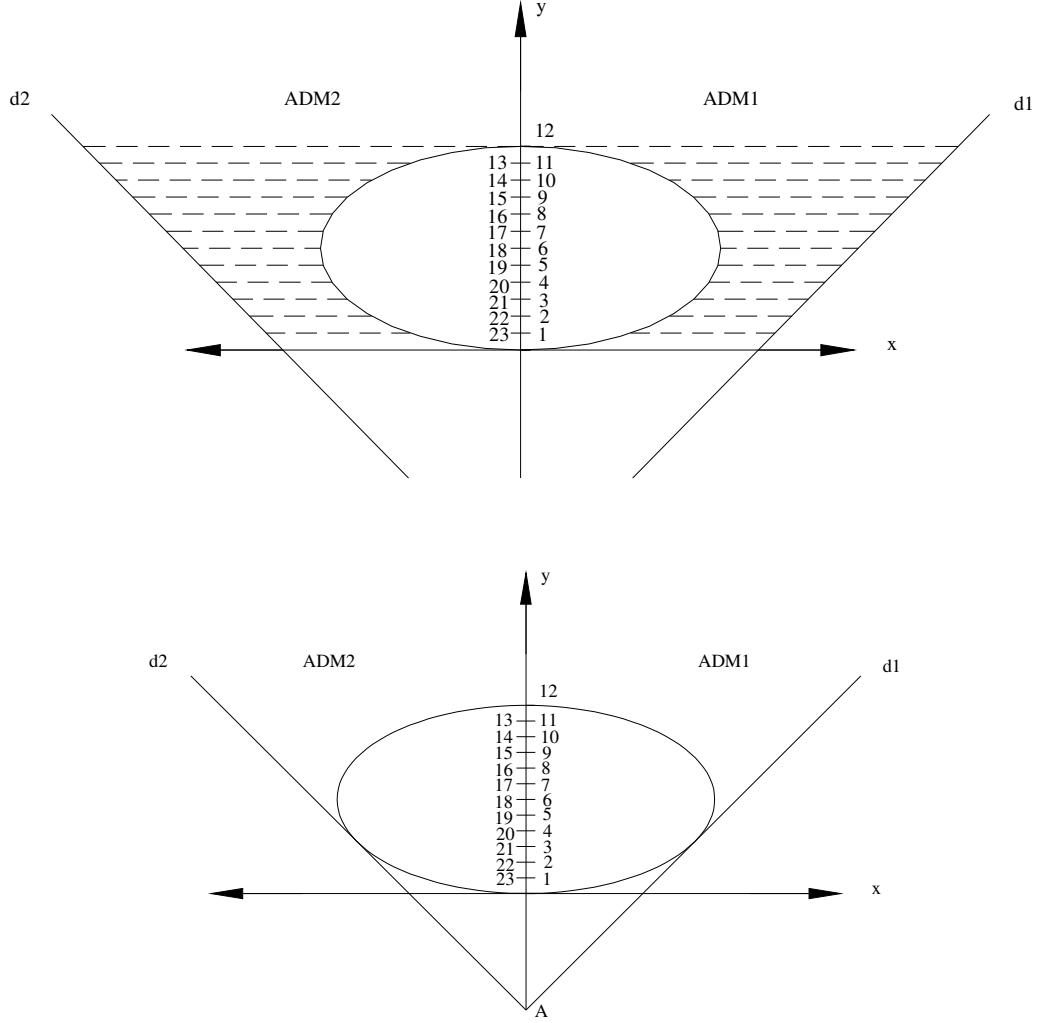
Ani dönme merkezlerine teğet olan doğruların eğimleri bellidir ve basınç açısına bağlı olarak değişmektedir. Teğet doğruları aşağıdaki gibi ifade edilebilir.

$$d_1 = m_1 \cdot x + e_1 \quad m_1 = \tan(90 - \beta)$$

$$d_2 = m_1 \cdot x + e_1 \quad m_1 = \tan(90 - \beta)$$

d_1 ve d_2 doğruları önce ADM ye temas etmeyecek biçimde çizilir. Bu doğrular çizilirken e_1 ve e_2 sabitleri seçilir. ADM1 ve ADM2 dizilerindeki noktaların sırasıyla d_1 ve d_2 doğrularına olan uzaklıkları hesaplanır ve bulunan en kısa uzaklık hangi noktada ise teğet doğruların sabitleri değiştirilerek doğrular ani dönme merkezleri yörüngelerine teğet ettirilmiş olur. Teğet doğrularının eğimleri ve sabitlerinin bulunmasından sonra bu iki doğrunun kesim noktasının (A) x ve y bileşenleri doğruların ortak çözümünden bulunur.

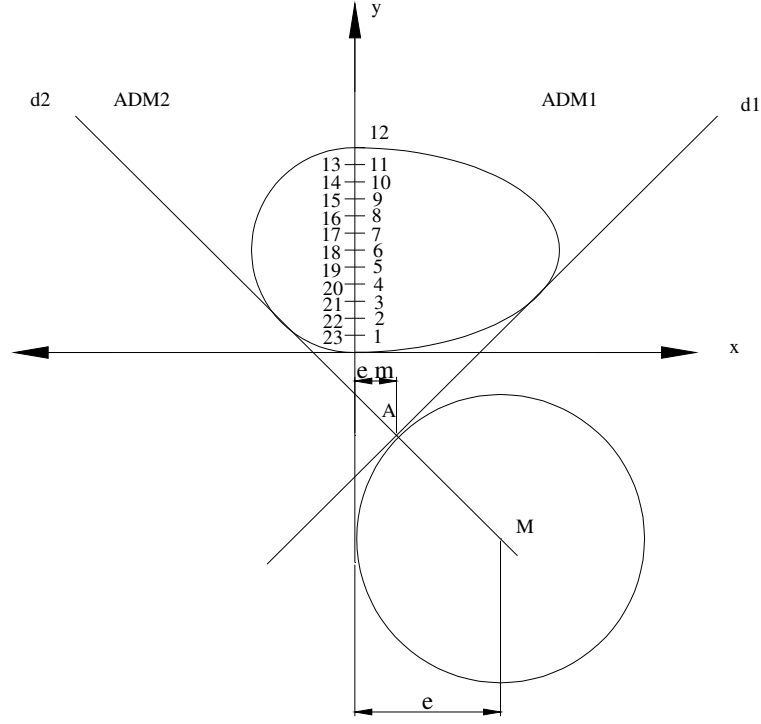
Kesim noktasının x değeri $m_1 \cdot x + e_1 = m_2 \cdot x + e_2$ eşitliğinden, y değeri ise $A_y = m_1 \cdot x + e_1$ ya da $A_y = m_2 \cdot x + e_2$ ifadelerinden bulunur. Şekil 4.3'de teğet doğrularının gerçek konumlarının ve kesim noktalarının nasıl bulunduğu gösterilmiştir.



Şekil 4.3. Ani dönme merkezlerine teğet doğruların çizimi [6].

Teğet doğrularının kesim noktasının bulunmasından sonra kam dönme merkezinin tespiti yapılabilir. Dönme merkezi belirlenirken teğet doğrularının kesim noktasının altında kalan alan içinde ve minimum kam boyutları için ise dönme merkezinin teğet doğrular üzerinde seçilmesi gerekmektedir. Dönme merkezi olarak belirlenecek nokta eksantrite değerinin pozitif (+) olması durumunda d_2 teğeti üzerinde, negatif (-) olması durumunda ise d_1 teğeti üzerinde seçilmelidir.

Teğetlerin kesim noktaları bazen hareket doğrultusu üzerinde olmayabilir. Böyle durumlarda kesim noktasıyla hareket doğrultusu arasında kalan mesafeye mecburi eksantrisite denir. Şekil 4.4'de Mecburi eksantrisite gösterilmiştir.



Şekil 4.4. Mecburi eksanrisitenin gösterimi [6].

Kamın dönme merkezi bulunduktan sonra kam profilinin çizimine geçilebilir. Öncelikle teorik bir kam profili çizilir. Kam mekanizmalarında genellikle bir yuvarlanma elemanı kullanılır. Gerçek kam profili elde edilirken teorik kam profili üzerinde yarıçapı yuvarlanma elemanının yarıçapı olan daireler çizilir. Bu dairelere ait daire denklemleri hesaplandıktan sonra dairelere alttan ve üstten teğet doğrular çizilir. Bu doğruların kesim noktalarının birleşiminden oluşan kapalı eğri ise gerçek kam profilidir. Matematikte bu tür eğrilere zarf eğrisi denir.

4.3. Kam Tasarım Programının Tanıtılması

Çıkış uzvu öteleme hareketi yapan kam mekanizmaları için hazırlanmış bu program Borland Delphi 7.0 program dili kullanılarak yazılmıştır. Program Windows uyumludur. Klasik kam tasarımında tasarımcının yapması gereken hesaplamalar ve çizimler program ile hatasız yapılmaktadır. Programın kullanıcı ara yüzü Şekil 4.5’de verilmiştir.

Kam Tasarımı

Lütfen parametreleri girip İŞLEMİ BAŞLAT butonunu tıklayınız

Parametreler

Proje adı: Örnek kam tasarımı

Faz sayısı: 2

Dönüş hızı: 100 devir/dak

Dönüş yönü
 Saat yönünde
 Saatin tersi yönde

Eksantrisite: 30 mm

Eksantrisite yönü
 Sağda
 Solda

Basiç açısı: 45 derece

Toparlak yarıçapı: 10 mm

Hesaplanan değerleri de göster

İşlemi Başlat

Parametreleri Dosyaya Sakla

Parametreleri Dosyadan Yükle

Faz bilgilerini girmek için, ilgili faz satırını tıklayınız:

Faz no	Faz tipi	Yükseklik	Açı	Hareket tipi
1	Çıkış	100	180	Sabit hızlı hareket
2	İniş	0	360	Sabit hızlı hareket

Şekil 4.5. Kam tasarım programının kullanıcı arayüz görüntüsü.

Parametreler kısmındaki veriler ve faz bilgileri girilip işlemi başlat butonuna basıldığında kullanıcı arayüz görüntüsü Şekil 4.6'daki duruma gelir.

Kam Tasarımı

Lütfen parametreleri girip İŞLEMI BAŞLAT butonunu tıklayınız

Parametreler | Yol grafiği | Hız grafiği | İvme grafiği | Kam profili

Proje adı: Örnek kam tasarımı

Faz sayısı: 2

Dönüş hızı: 100 devir/dak

Dönüş yönü
 Saati yönünde
 Saatin tersi yönde

Eksantrisite: 30 mm

Eksantrisite yönü
 Sağda
 Solda

Başıncı açısı: 45 derece

Toparlık yarıçapı: 10 mm

Hesaplanan değerleri de göster

İşlemi Başlat

Parametreleri Dosyaya Sakla

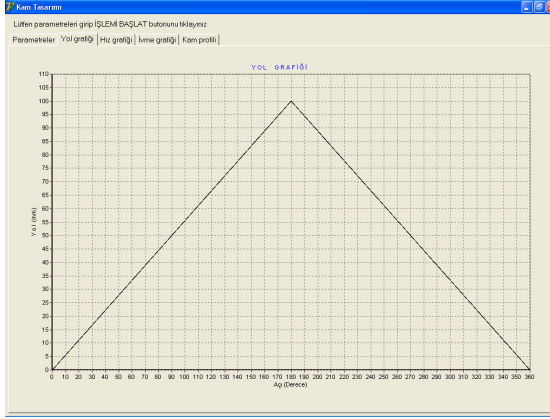
Parametreleri Dosyadan Yükle

Faz bilgilerinizi girmek için ilgili faz satırını tıklayınız:

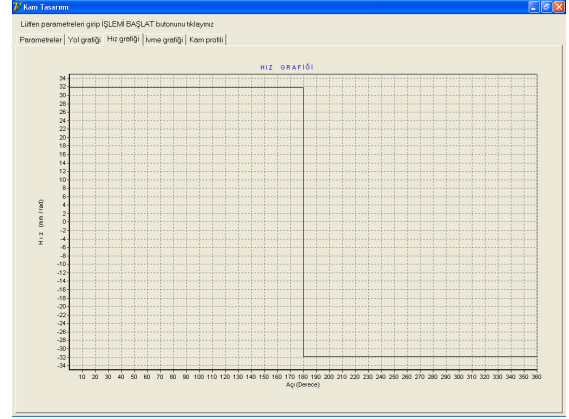
Faz no	Faz tipi	Yükseklik	Açı	Hareket tipi
1	Çıkış	100	180	Sabit hızlı hareket
2	İniş	0	360	Sabit hızlı hareket

Şekil 4.6. Giriş parametrelerinin girilip işlemi başlat'a basıldıktan sonraki arayüz görüntüsü.

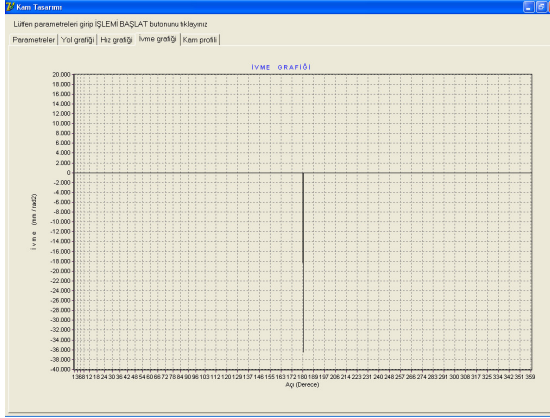
Gerekli olan parametreler programa girilip işlemi başlat düğmesine basılınca program gerekli hesaplamaları yapar ve yol, hız ile ivme grafiklerini çizer ve kam profilini oluşturur. Yol, hız ve ivme grafikleri ile kam profilinin görüntülerine bakmak için program arayüzünde onlar için açılan düğmelere basılması gerekir. Aşağıdaki Şekil 4.7.'de her bir grafiğin ve kam profilinin görüntüleri gösterilmiştir.



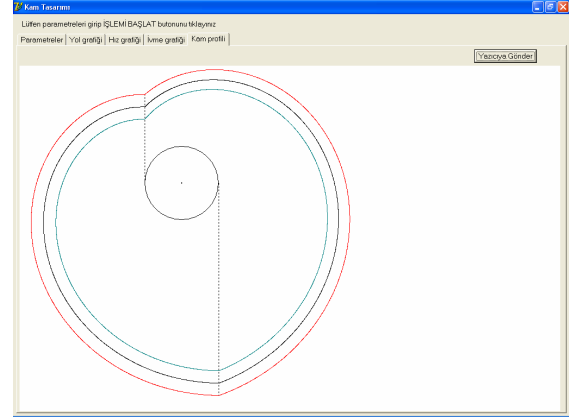
(a)



(b)



(c)



(d)

Şekil 4.7. Yol, hız ve ivme grafikleri ile kam profili görüntüleri.

(a: Yol Grafiği, b: Hız Grafiği, c: İvme Grafiği, d: Kam Profili)

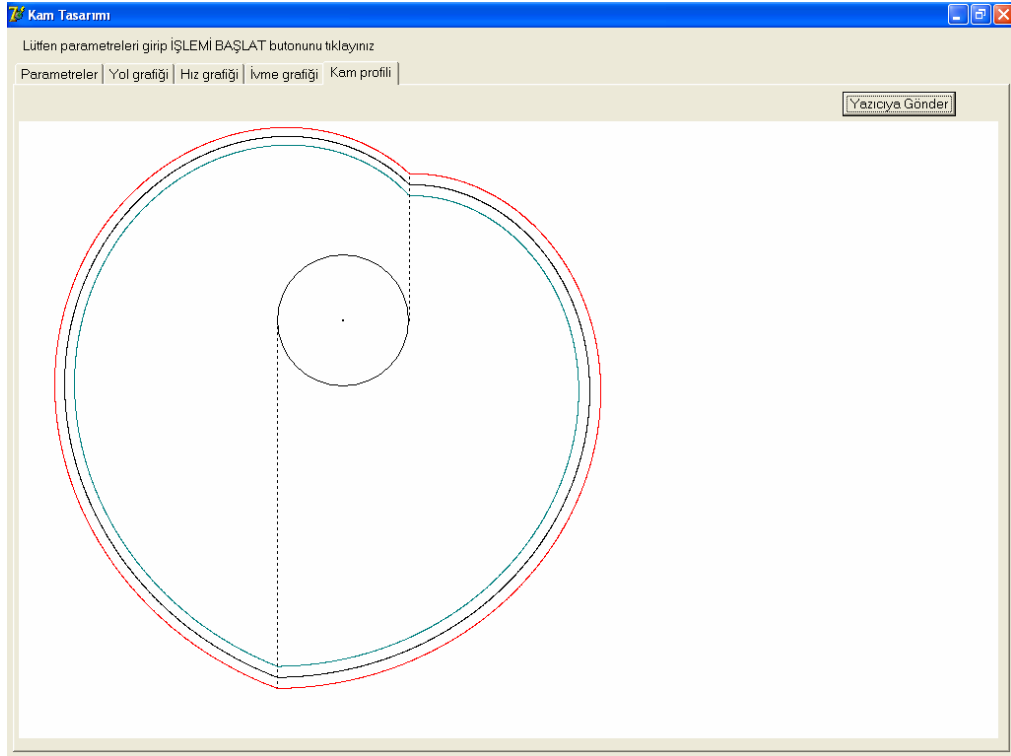
Programın giriş parametreleri değiştirildiğinde kam profilin bundan nasıl etkilendiği direkt olarak görülebilir. Bu parametreler hız değeri, eksantrisite, basınç açısı ve toparlak yarıçapıdır.

Programda ayrıca kullanıcıya tasarımı yapılan kama bir proje ismi vermesi ve bu projeyi diske kayıt etmesi ile tekrar çağırabilmesi olanağı verilmiştir.

Kam tasarım programıyla yapılmış örnek tasarımlar aşağıda verilmiştir.

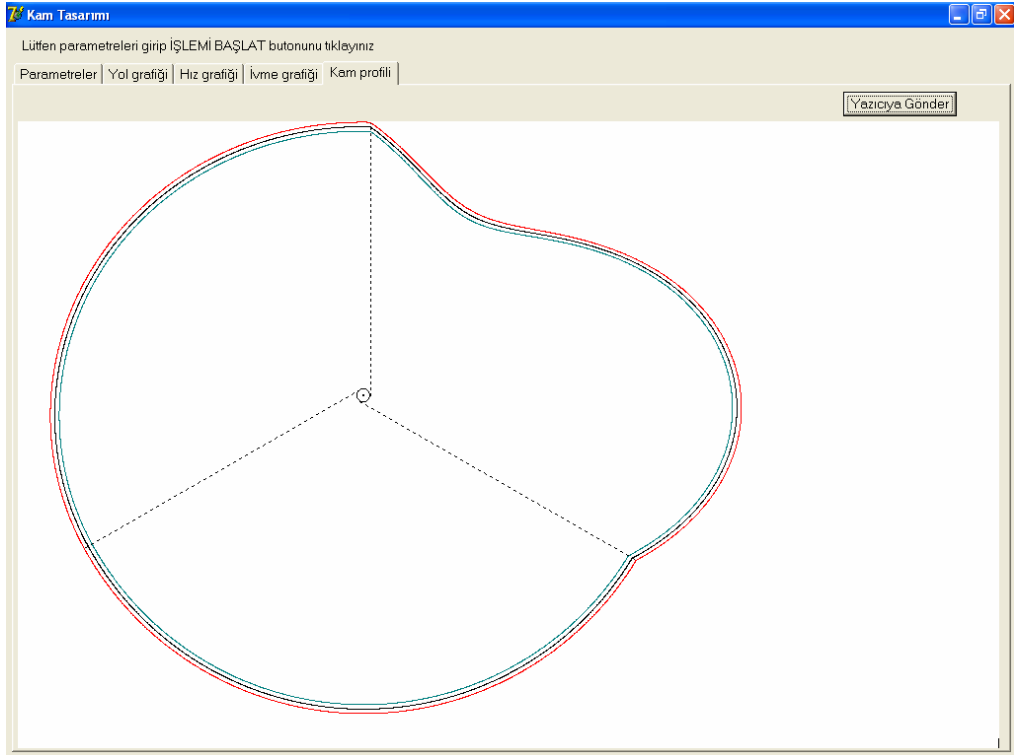
Örnek 1:

Faz Bilgileri				
1. Faz	Hareket Tipi Çıkış	Toplan İlerleme 100 mm	Toplam Dönme 180°	Hareket Denklemi Sabit Hızlı Hareket
2. Faz	Hareket Tipi İniş	Toplan İlerleme 100 mm	Toplam Dönme 180°	Hareket Denklemi Sabit Hızlı Hareket
Giriş Parametreleri				
Devir Sayısı			100 dev/dk	
Basınç Açısı			45°	
Eksantrisite			30 mm	
Toparlık Yarıçapı			5 mm	



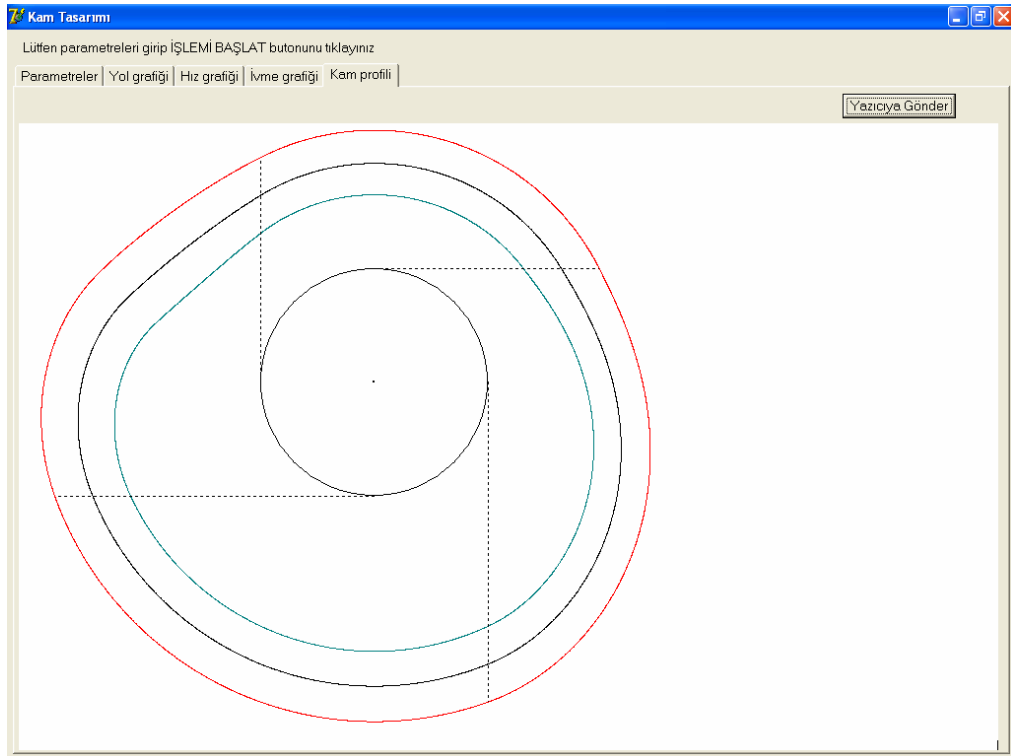
Örnek 2:

Faz Bilgileri				
1. Faz	Hareket Tipi Çıkış	Toplan İlerleme 100 mm	Toplam Dönme 120°	Hareket Denklemi Basit Harmonik Hareket
2. Faz	Hareket Tipi Bekleme	Toplan İlerleme 0 mm	Toplam Dönme 120°	Hareket Denklemi -
3. Faz	Hareket Tipi İniş	Toplan İlerleme 100 mm	Toplam Dönme 120°	Hareket Denklemi Sikloidal Hareket
Giriş Parametreleri				
Devir Sayısı			125 dev/dk	
Basınç Açısı			40°	
Eksantrisite			-15 mm	
Toparlak Yarıçapı			10 mm	



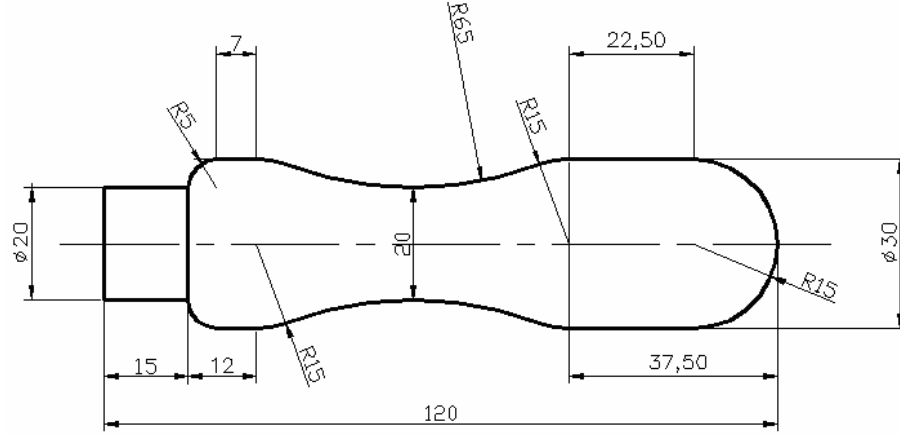
Örnek 3:

Faz Bilgileri				
1. Faz	Hareket Tipi Çıkış	Toplan İlerleme 25 mm	Toplam Dönme 90°	Hareket Denklemi Sabit İvmeli Hareket
2. Faz	Hareket Tipi Bekleme	Toplan İlerleme 0 mm	Toplam Dönme 90°	Hareket Denklemi -
3. Faz	Hareket Tipi İniş	Toplam İlerleme 25 mm	Toplam Dönme 90°	Hareket Denklemi Polinomik Hareket 1
4. Faz	Hareket Tipi Bekleme	Toplan İlerleme 0 mm	Toplam Dönme 90°	Hareket Denklemi -
Giriş Parametreleri				
Devir Sayısı			125 dev/dk	
Basınç Açısı			45°	
Eksantrisite			30 mm	
Toparlık Yarıçapı			10 mm	



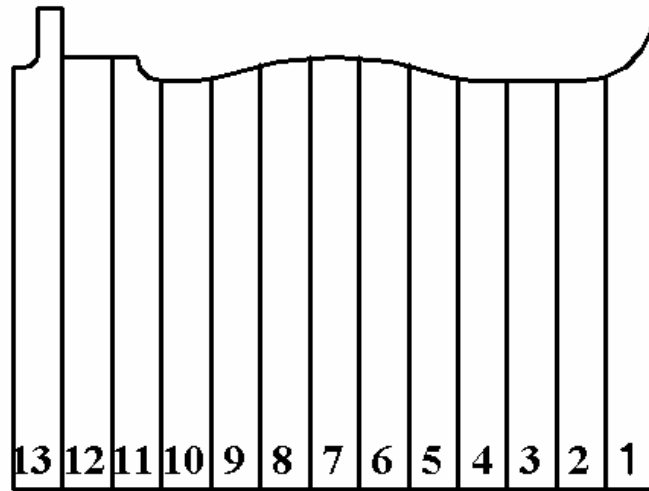
5.ÖRNEK BİR KAM MEKANİZMASININ TASARIMI VE İMALATI

Tasarlanan ve imalatı yapılan kam mekanizması ile eđe sapı üretimi yapılmıştır. Yapılan eđe sapının teknik resmi Şekil 5.1’de verilmiştir.



Şekil 5.1. Eđe sapının teknik resmi.

Mekanizmada eđe sapının profilinin verilmesinde ölçüleri 10x10x125 olan onüç adet torna kalemi kullanılmıştır. Bu kalemlerin kesici yüzeyleri eđe sapının profilini verecek şekilde taşlanmıştır. Kesici yüzeylerinin bilenmesinden sonra kalemlerin eđe sapı profilini nasıl oluşturduđu Şekil 5.2’ gösterilmiştir.



Şekil 5.2. Torna kalemlerine eđe sapı profilinin verilmesi.

5.1. Mekanizmada Kullanılacak Kamın Tasarımı ve İmalatı

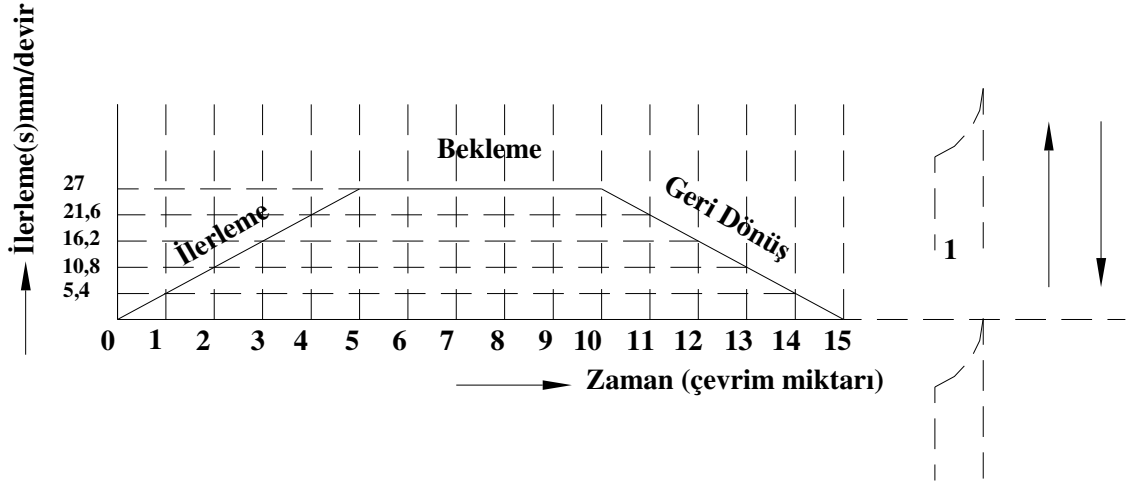
Mekanizmada çıkış uzvu öteleme hareketi yapan kam kullanılmıştır. Kamın sentezi aşağıdaki gibi yapılmıştır.

Kamın devir sayısı (n) = 1 dev/dak

Basınç açısı (β) = 45°

Eksantrisite miktarı (e) = 10 mm dir.

Çıkış uzvuna ait yol-zaman grafiği Şekil 5.3' verilmiştir.



Şekil 5.3. İmalatı yapılacak kama ait yol-zaman grafiği.

Kamın bir devri için geçen süre $T = 60/1 = 60$ sn.

$\Delta t = 60/3 = 20$ sn.

$$\omega = \frac{n_1 \cdot 2\pi}{60} = 0.1047 \text{ rad / sn}$$

$$V_0 = V_5 = \frac{S_5 - S_0}{t_5 - t_0} = \frac{\Delta S}{\Delta t} = \frac{27}{20} = 1.35 \text{ mm / sn}$$

$$V_{15} = V_{10} = \frac{S_{15} - S_{10}}{t_{15} - t_{10}} = \frac{\Delta S}{\Delta t} = \frac{-27}{20} = -1.35 \text{ mm / sn} \quad \text{çıkış ve iniş fazlarındaki sabit hız}$$

değerleri hesaplanır.

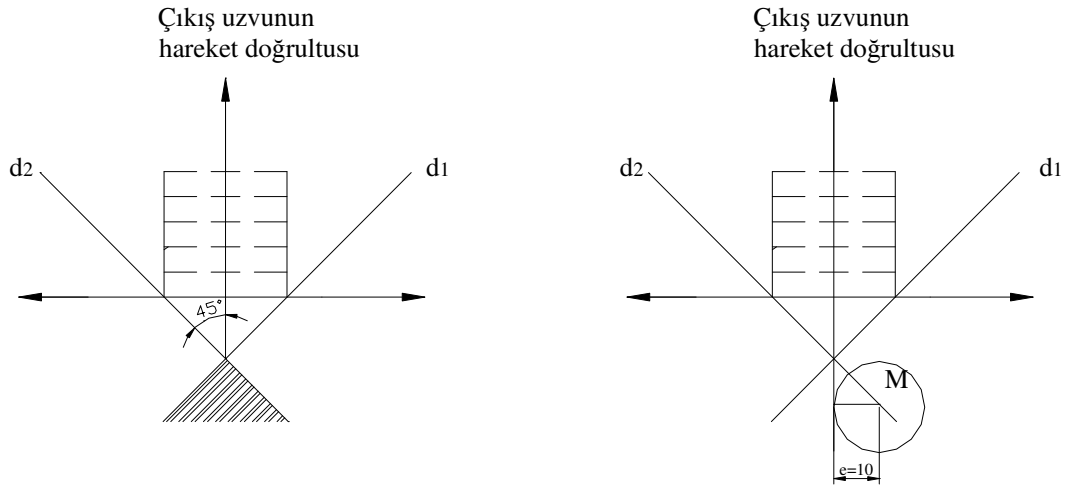
Ani dönme merkezleri yörüngesinin çizilebilmesi için gerekli olan L değerleri hesaplanır.

$$L = \frac{V_i}{\omega}$$

$$L_{0..4} = \frac{1.35}{0.1047} = 12.89mm \quad \text{Çıkış uzvunun sağ tarafında.}$$

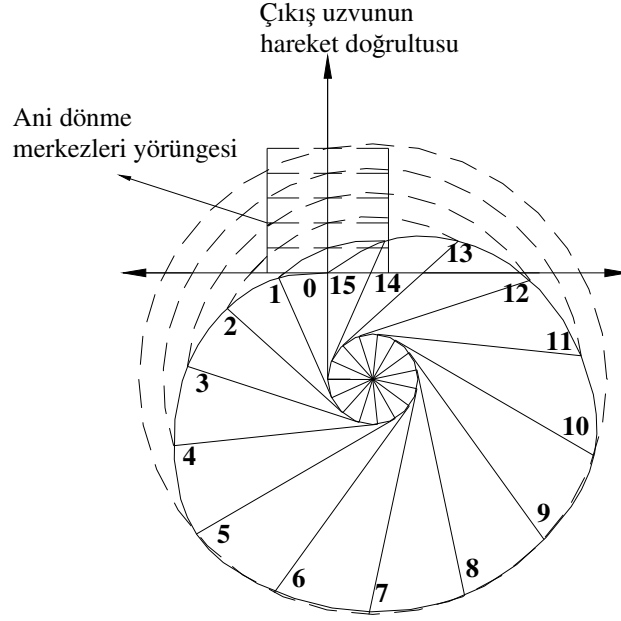
$$L_{0..4} = \frac{-1.35}{0.1047} = -12.89mm \quad \text{Çıkış uzvunun sol tarafında.}$$

Ani dönme merkezleri yörüngesinin çizimi ve kam dönme merkezin bulunması Şekil 5.4'de gösterilmiştir.



Şekil 5.4. Ani dönme merkezlerinin çizimi ve kam dönme merkezinin bulunması

Kam dönme merkezi bulunduktan sonra kam profilinin çizimi Şekil 5.5'de verilmiştir.



Şekil 5.5. İmalatı yapılacak kam profilinin çizimi

Kam profilinin elde edilmesinden sonra bu profil görüntüsü malzeme üzerine markalanarak Simav Teknik Eğitim Fakültesi Atölyesi'nde imal edilmiştir. Ayrıca bu kamın geleneksel yöntemlerle ve tasarım programıyla yapılan tasarımları karşılaştırılmış ve Ek.4'te gösterilmiştir.

5.2. Mekanizmada Kullanılan Diğer Parçalar Ve Mekanizmanın Tanıtımı

Mekanizmada kullanılacak olan kamın tasarımı ve imalatı tamamlandıktan sonra mekanizmada kullanılacak diğer parçaların tasarımları ve imalatı yapılmıştır.

Öncelikle çıkış uzvu olarak kullanılacak olan HSS kalemlerin yan yana dizilerek çalışabileceği mekanizma alt parça tasarlanmıştır. Alt parçanın içi kalemlerin kursları dikkate alınarak boşaltılmıştır. Kalemlerin bu boşaltılan alanın üzerinde çalışabilmeleri için yan destek parçaları ve mekanizma üst parçasının imalatı yapılmıştır.

Kalemlerin kamla temas edecek yüzeyleri aynı hizada olacak şekilde sırandıktan sonra alt taraflarından kanal açılmıştır. Kalemlerin görevlerini yaptıktan sonra geri dönebilmeleri için yaylar kullanılmıştır. Bu yaylar mekanizmanın alt parçası ile kalemlerin altına açılan kanallara yerleştirilen parçalar arasında monte edilmiştir.

Kamın döndürülmesinde kama kanallı mil kullanılmıştır. Ayrıca kamın her bir kalemin arkasına gelebilmesi için mil üzerine adım kadar aralıklarla kanallar açılmıştır. Milin yataklanması için kullanılan parçalar mekanizmanın alt parçasına monte edilmiştir.

Tasarımları yapılan parçaların teknik resim görüntüleri Ek.3 'de verilmiştir ve bu parçaların imalatı Simav Teknik Eğitim Fakültesi Atölyesi'nde gerçekleştirilmiştir.

Parçaların montajı yapıldıktan sonra mekanizma çalıştırılmak üzere torna tezgahının katerliğine bağlanır. Mekanizmada mil adım kadar ilerletilerek kam her bir kalemin arkasına gelecek şekilde konumlandırılır ve kam bir tam tur atacak şekilde saat yönünde döndürülür. Bu sayede her kalem kendi formunu malzeme üzerine işler ve en son kalem malzemeyi keserek işlemi tamamlar. Tasarımı ve imalatı yapılan mekanizmanın montaj resmi Ek 3'de verilmiştir.

6.SONUÇ VE ÖNERİLER

Bu çalışma incelendiğinde kam mekanizmalarının tasarımlarının klasik yöntemlerle yapımının ne kadar güç olduğu anlaşılacaktır. Bunun yanı sıra kam mekanizmalarının çıkış uzvu için seçilen hareket denklemlerinin ve basınç açısı değerlerinin kam profilini nasıl etkilediği görülecektir. Kam mekanizmalarının tasarımlarında çıkış uzvuna ait hareket kanununun ve basınç açısının doğru seçilmesi ile kam mekanizmalarının optimizasyonunda karşılaşılan birçok problemin çözümü gerçekleştirilecektir.

Kam tasarımında yapılan hesaplamaların bilgisayar ortamına aktarılmasıyla, klasik tasarım yöntemlerine göre hızlı ve doğru bilgilerin elde edildiği görülmüştür. Bu nedenle makine elemanlarının tasarımında bilgisayar destekli tasarımın bir ihtiyaç olduğu anlaşılmıştır.

Ayrıca kam mekanizmalarının kullanılmasıyla üretimleri zor olan ve zaman alan parçaların imalatları bilinen yöntemlere göre daha hızlı yapılabildikleri anlaşılmıştır.

Bu çalışmada ayrıca, ege sapı imalatında kullanılacak kamlı mekanizma imal edilerek, hassasiyetini önemli ölçüde kaybetmiş bir torna tezgahına monte edilmiştir. Bu sayede ege sapı üretimi yapılarak hem atölyenin ihtiyaçları karşılanmış hem de bu tezgah faal duruma getirilerek üretime katkısı sağlanmıştır.

Son zamanlarda yapılan çalışmalar kamların boyutlarının minimize edilmesi üzerinedir. Bu nedenle kam mekanizmalarıyla ilgileneceklerin bu konu üzerinde durmaları tavsiye edilmektedir.

KAYNAKLAR DİZİNİ

- [1] Söylemez, E., 2000, Mekanizma tekniği, Prestij Ajans Matbaacılık Basım Yayın Sanayii ve Ticaret Ltd., 434 s.
- [2] www.makineelemanlari.com/makineelemanlari/kam/kamlar-genel_bilgiler-1-htm-28k
- [3] Navora, O., Wu, C.J. and Angeles, J., 2001, The size minimization of planar cam mechanisms, Mechanism and Mechine Theory, 36, 371-386.
- [4] Yao, Y., Zhang, C. And Yan, H.S., 2000, Mation control of cam mechanisms, Mechanism and Mechine Theory, 35, 593-607.
- [5] Kazıhan, K., 1996, Makine ve mekanizmalar teorisi, Beta Yayınevi, 213 s.
- [6] Küçük, T., 1998, Kam Mekanizmalarının Bilgisayar Destekli Tasarımı, Yüksek lisans tezi, OĞÜ Fen Bilimleri Enstitüsü, 91 s.
- [7] Ye, Z. and Smith, M.R., 2002, Synthesis of constant breadth cam mechanisms, Mechanism and Mechine Theory, 37, 941-953.
- [8] <http://www.me.metu.edu.tr/me431/ch8/>
- [9] Küçük, M., 1990, Manine Bilgisi, MEB Basımevi, 815 s.
- [10] Shigley,J.E. and Vicker,J.J., 1980, Theory of machines and mechanism, Mc Graw Hill Book Co., 300 p.
- [11] <http://www.aqurate-group.com/aewimgssscl/-cam>
- [12] <http://floridams.com/Products/Lico10.jpg>
- [13] Nishika, M. And Nishimura, T., 1998, Synthesis of the internal parallel cam mechanism, ProQuest science journals, 212, 7, 577-585.
- [14] Bouzakis, K.D., Mitsi, S. and Tsiafis, J., 1997, Computer aided optimum design and NC milling of planar cam mechanisms, Tools Manufact, 37, 8, 1131-1142.
- [15] Şeker, U., 1987, Kam mekanizmaları ve kamların imalatı ders notu, Gazi Üniversitesi Teknik Eğitim Fakültesi, 51 s.
- [16] Emirdağ, G., 1988, Kam Mekanizmaları ve Bilgisayarla Tasarımı, Yüksek lisans tezi, AÜ Fen Bilimleri Enstitüsü, 61 s.
- [17] Yu, Q. and Lee, H. P., 1998, Size optimization of cam mechanisms with translating roller followers, ProQuest Science Journals, 212, 5, 381-386.s

EKLER

- Ek.1. Kam Mekanizmalarında Kullanılan Hareket Denklemleri
- Ek.2. Kam Tasarım Programının Kodları
- Ek.3. Parçaların Yapım Resimleri Ve Mekanizmanın Montaj Resmi
- Ek.4. Geleneksel Yöntemlerle Ve Tasarım Programıyla Tasarlanan Kamın Karşılaştırılması
- Ek.5. Örnek Kam Mekanizmasına Ait Resimler

EK1- Kam Mekanizmalarında Kullanılan Hareket Denklemleri

Sabit ivmeli hareket

$$0 < \Delta\theta < \beta/2 \Rightarrow S(\theta) = \frac{2 * h * \theta^2}{\beta^2}$$

$$\beta/2 < \Delta\theta < \beta \Rightarrow S(\theta) = h - \frac{2 * h}{\beta^2} * (\beta - \theta)^2$$

Sabit hızlı hareket

$$S = h * \frac{\theta}{\beta}$$

Polinomik hareket denklemleri

$$1-) S = h * \left(-2 * \left(\frac{\theta}{\beta} \right)^3 + 3 * \left(\frac{\theta}{\beta} \right)^2 \right)$$

$$2-) S = h * \left(-\left(\frac{\theta}{\beta} \right)^4 + 2 * \left(\frac{\theta}{\beta} \right)^3 \right)$$

$$3-) S = h * \left(-3 * \left(\frac{\theta}{\beta} \right)^4 + 4 * \left(\frac{\theta}{\beta} \right)^3 \right)$$

$$4-) S = \frac{h}{2} * \left(-3 * \left(\frac{\theta}{\beta} \right)^5 + 5 * \left(\frac{\theta}{\beta} \right)^3 \right)$$

$$5-) S = h * \left(\frac{1}{3} * \left(\frac{\theta}{\beta} \right)^4 - 2 * \left(\frac{\theta}{\beta} \right)^2 + 2.6 * \left(\frac{\theta}{\beta} \right) \right)$$

$$6-) S = h * \left(3 * \left(\frac{\theta}{\beta} \right)^4 - 8 * \left(\frac{\theta}{\beta} \right)^3 + 6 * \left(\frac{\theta}{\beta} \right)^2 \right)$$

$$7-) S = h * \left(6 * \left(\frac{\theta}{\beta} \right)^5 - 15 * \left(\frac{\theta}{\beta} \right)^4 + 10 * \left(\frac{\theta}{\beta} \right)^3 \right)$$

$$8-) S = h * \left(0.375 * \left(\frac{\theta}{\beta} \right)^5 - 1.25 * \left(\frac{\theta}{\beta} \right)^3 + 1.875 * \left(\frac{\theta}{\beta} \right) \right)$$

$$9-) S = h * \left(15 * \left(\frac{\theta}{\beta} \right)^7 - 35 * \left(\frac{\theta}{\beta} \right)^6 + 21 * \left(\frac{\theta}{\beta} \right)^5 \right)$$

EK1- Devami

$$10-) S = h * \left(-20 * \left(\frac{\theta}{\beta} \right)^7 + 70 * \left(\frac{\theta}{\beta} \right)^6 - 84 * \left(\frac{\theta}{\beta} \right)^5 + 35 * \left(\frac{\theta}{\beta} \right)^4 \right)$$

$$11-) S = h * \left(120 * \left(\frac{\theta}{\beta} \right)^7 - 420 * \left(\frac{\theta}{\beta} \right)^6 + 546 * \left(\frac{\theta}{\beta} \right)^5 - 315 * \left(\frac{\theta}{\beta} \right)^4 + 70 * \left(\frac{\theta}{\beta} \right)^3 \right)$$

$$12-) S = h * \left(26.6\bar{6} * \left(\frac{\theta}{\beta} \right)^7 - 93.3\bar{3} * \left(\frac{\theta}{\beta} \right)^6 + 126 * \left(\frac{\theta}{\beta} \right)^5 - 81.6\bar{6} * \left(\frac{\theta}{\beta} \right)^4 + 23.3\bar{3} * \left(\frac{\theta}{\beta} \right)^3 \right)$$

Trigonometrik hareket denklemler

$$1-) S = h * \left(1 - \cos \left(\frac{\pi * \theta}{2} \right) \right)$$

$$2-) S = h * \sin \left(\frac{\pi * \theta}{2} \right)$$

$$3-) S = h * \left(\frac{\theta}{\beta} - \frac{1}{\pi} * \sin \left(\pi * \frac{\theta}{\beta} \right) \right)$$

$$4-) S = h * \left(\frac{\theta}{\beta} + \frac{1}{\pi} * \sin \left(\pi * \frac{\theta}{\beta} \right) \right)$$

$$5-) S = \frac{h}{2} * \left(1 - \cos \left(\pi * \frac{\theta}{\beta} \right) \right)$$

$$6-) S = h * \left(\frac{\theta}{\beta} - \frac{1}{2 * \pi} * \sin \left(2 * \pi * \frac{\theta}{\beta} \right) \right)$$

$$7-) S = h * \left(\frac{\theta}{\beta} - \frac{15}{32 * \pi} * \sin \left(2 * \pi * \frac{\theta}{\beta} \right) - \frac{1}{96 * \pi} * \sin \left(6 * \pi * \frac{\theta}{\beta} \right) \right)$$

$$8-) S = h * \left(\frac{\theta}{\beta} - \frac{1}{2 * \pi} * \left(\frac{27}{28} * \sin \left(2 * \pi * \frac{\theta}{\beta} \right) + \frac{1}{84} * \sin \left(6 * \pi * \frac{\theta}{\beta} \right) \right) \right)$$

9-)

$$S = h * \left(\frac{\theta}{\beta} - \frac{1}{2 * \pi} * \frac{1125}{1192} * \left(\sin \left(2 * \pi * \frac{\theta}{\beta} \right) + \frac{1}{54} * \sin \left(6 * \pi * \frac{\theta}{\beta} \right) + \frac{1}{1250} * \sin \left(10 * \pi * \frac{\theta}{\beta} \right) \right) \right)$$

EK2- Kam Tasarım Programının Kodları

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, StdCtrls, Grids, ExtCtrls, TeEngine, Series, TeeProcs,
  Chart, Printers;
type
  TForm1 = class(TForm)
    PageControl1: TPageControl;
    tabYolGrafigi: TTabSheet;
    tabHizGrafigi: TTabSheet;
    tabIvmeGrafigi: TTabSheet;
    yolChart: TChart;
    Series1: TLineSeries;
    hizChart: TChart;
    Series2: TLineSeries;
    ivmeChart: TChart;
    Series3: TLineSeries;
    tabParametreler: TTabSheet;
    BilgiGrid: TStringGrid;
    Label1: TLabel;
    txtFazSayisi: TEdit;
    updownFazSayisi: TUpDown;
    buttonIslemiBaslat: TButton;
    Label2: TLabel;
    txtDonusHizi: TEdit;
    Label3: TLabel;
    radioDonusYonu: TRadioGroup;
    Label5: TLabel;
    txtEksantirisite: TEdit;
    radioEksantirisiteYonu: TRadioGroup;
    Label6: TLabel;
    txtBasincAcisi: TEdit;
    Label7: TLabel;
    txtToparлакYaricapi: TEdit;
    Label8: TLabel;
    Label9: TLabel;
    Label4: TLabel;
    tabKamProfili: TTabSheet;
    kamProfiliResmi: TImage;
    Label10: TLabel;
    fazGrubu: TGroupBox;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    hareketSecim: TComboBox;
    Label17: TLabel;
    fazSecim: TComboBox;
    txtAci: TEdit;
    txtYukseklık: TEdit;
    bTamam: TButton;
    bVazgeç: TButton;
    updownAci: TUpDown;
    Label18: TLabel;
    txtProjeAdi: TEdit;
    Label19: TLabel;
    buttonYazdir: TButton;
    buttonSakla: TButton;
    buttonYukle: TButton;
    OpenDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    PrintDialog1: TPrintDialog;
    tabLog: TTabSheet;
    memoLog: TMemo;
    checkDebug: TCheckBox;
```

EK-2 Devami

```
procedure FormCreate(Sender: TObject);
  procedure txtFazSayisiChange(Sender: TObject);
  procedure FormActivate(Sender: TObject);
  procedure BilgiGridMouseUp(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure TamsayiTusKontrol(Sender: TObject; var Key: Char);
  procedure KesirliSayiTusKontrol(Sender: TObject; var Key: Char);
  procedure buttonIslemiBaslatClick(Sender: TObject);
  procedure FormResize(Sender: TObject);
  procedure bVazgecClick(Sender: TObject);
  procedure bTamamClick(Sender: TObject);
  procedure buttonSaklaClick(Sender: TObject);
  procedure txtProjeAdiChange(Sender: TObject);
  procedure buttonYukleClick(Sender: TObject);
  procedure buttonYazdirClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
uses Math;
const EnBuyukFazSayisi = 10;
      AraliktakiAdimSayisi = 10;
      ToplamAdimSayisi = 360 * AraliktakiAdimSayisi;
type Nokta = record
  X: extended;
  Y: extended;
end;
type Dogru = record
  Egim: extended;
  Sabit: extended;
end;
type ProfilElemani = record
  X          : extended;
  Y          : extended;
  DonmeMerkezineUzaklik : extended;
  DonmeMerkeziIleYaptigiAci: extended;
  DondurulmusAci      : extended;
end;
type AdimElemani = record
  Yol          : extended;
  Hiz          : extended;
  Ivme         : extended;
  DonusAcisi   : extended;
  AniDonmeMerkeziYorungesi : Nokta;
  TeorikKamProfili : ProfilElemani;
  IcKamProfili    : ProfilElemani;
  DisKamProfili   : ProfilElemani;
end;
type FazElemani = record
  FazTipi: integer;
  Yukseklik: extended;
  BitisAcisi: integer;
  FazFonksiyonu: integer;
  FazFonksiyonuDizini: integer;
end;
{$R *.dfm}
var SecilenFaz: integer;
    Fazlar: array [0..EnBuyukFazSayisi] of FazElemani;
    Adimlar: array [0..ToplamAdimSayisi] of AdimElemani;
    TasarimBasligi: string;
    FazAdet: integer;
    DonusHizi: extended;
    DonusSaatYonunde: boolean;
    DegerleriGoster: boolean;
```


EK-2 Devami

```
Eksantirisite: extended;
EksantirisiteSagda: boolean;
BasincAcisi: extended;
ToparлакYaricapi: extended;
AcisalHiz: extended;
SoldanTegetDogru: Dogru;
SagdanTegetDogru: Dogru;
DonmeMerkezi: Nokta;
FonksiyonAdlari: array [1..20] of string = (
    '10 Sabit hızlı hareket',
    '20 Sabit ivmeli hareket',
    '30 Sikloidal hareket',
    '40 Basit harmonik hareket',
    '51 Polinomik hareket - 1',
    '52 Polinomik hareket - 2',
    '53 Polinomik hareket - 3',
    '54 Polinomik hareket - 4',
    '55 Polinomik hareket - 5',
    '56 Polinomik hareket - 6',
    '57 Polinomik hareket - 7',
    '58 Polinomik hareket - 8',
    '59 Polinomik hareket - 9',
    '61 Trigonometrik hareket - 1',
    '62 Trigonometrik hareket - 2',
    '63 Trigonometrik hareket - 3',
    '64 Trigonometrik hareket - 4',
    '71 Fourier - Gutman I-III Harmonic',
    '72 Fourier - Freudenstein I-III Harmonic',
    '73 Fourier - Freudenstein I-III-IV Harmonic'
);

//-----
function YolHesapla(const FonksiyonNo: integer;
    const h, DeltaTeta, Beta: extended
): extended;
//-----
//
// Hareket fonksiyonu cinsine göre yol hesaplamaları burada yapılır.
// Fonksiyon no değeri, FonksiyonAdlari dizisindeki elemanların ilk 2 karakterinden alınır
//
var X : extended;
begin
    YolHesapla := 0;
    X := DeltaTeta / Beta;
    case FonksiyonNo of
        10: YolHesapla := h * X;
        20: if DeltaTeta < (Beta / 2)
            then YolHesapla := 2 * h * Power(X, 2.0)
            else YolHesapla := h - ( 2 * h * Power(Beta - DeltaTeta, 2.0) / Power(Beta, 2.0));
        30: YolHesapla := h * (X - (0.5 * Pi * Sin(2 * Pi * X)));
        40: YolHesapla := (h / 2) * (1 - Cos(Pi * X));
        51: YolHesapla := h * ((-2 * Power(X, 3.0)) + (3 * Power(X, 2.0)));
        52: YolHesapla := h * ((-1 * Power(X, 4.0)) + (2 * Power(X, 3.0)));
        53: YolHesapla := h * ((-3 * Power(X, 4.0)) + (4 * Power(X, 3.0)));
        54: YolHesapla := (h / 2) * ((-3 * Power(X, 5.0)) + (5 * Power(X, 3.0)));
        55: YolHesapla := h * (((1.0 / 3.0) * Power(X, 4.0)) + (-2 * Power(X, 2.0)) + ((24.0 / 9.0) * X));
        56: YolHesapla := h * ((3 * Power(X, 4.0)) + (-8 * Power(X, 3.0)) + (6 * Power(X, 2.0)));
        57: YolHesapla := h * ((6 * Power(X, 5.0)) + (-15 * Power(X, 4.0)) + (10 * Power(X, 3.0)));
        58: YolHesapla := h * ((0.375 * Power(X, 5.0)) + (-1.25 * Power(X, 3.0)) + (1.875 * X));
        59: YolHesapla := h * ((15 * Power(X, 7.0)) + (-35 * Power(X, 6.0)) + (21 * Power(X, 5.0)));
        61: YolHesapla := h * (1 - Cos(Pi * X * 0.5));
        62: YolHesapla := h * Sin(Pi * X * 0.5);
        63: YolHesapla := h * (X - (Sin(Pi * X) / Pi));
        64: YolHesapla := h * (X + (Sin(Pi * X) / Pi));
        71: YolHesapla := h * (X - ((15.0 * Sin(2 * Pi * X)) / (32 * Pi)) - (Sin(6 * Pi * X) / (96 * Pi)));
        72: YolHesapla := h * (X - ((27.0 * Sin(2 * Pi * X)) / (56 * Pi)) - (Sin(6 * Pi * X) / (168 * Pi)));
        73: YolHesapla := h * (X - ((1125.0 * Sin(2 * Pi * X)) / (2384 * Pi)) - ((1125 * Sin(6 * Pi * X)) / (128736 * Pi)) - ((1125 * Sin(10 * Pi * X)) / (2980000 * Pi)));
    else YolHesapla := 0;
end;
```

EK-2 Devamı

```
end;
end;
//-----
procedure TForm1.FormCreate(Sender: TObject);
//-----
var i: integer;
begin
    DateSeparator := '.';
    DecimalSeparator := ',';
    LongDateFormat := 'dd.mm.yyyy';
    ShortDateFormat := 'dd.mm.yyyy';
    LongTimeFormat := 'hh:nn:ss';
    ShortTimeFormat := 'hh:nn:ss';
    PageControl1.ActivePage := tabParametreler;
    with BilgiGrid do begin
        Cells[0,0]:= 'Faz no ';
        Cells[1,0]:= 'Faz tipi';
        Cells[2,0]:= 'Yükseklik';
        Cells[3,0]:= 'Açı';
        Cells[4,0]:= 'Hareket tipi ';
    end;
    HareketSecim.Clear;
    HareketSecim.Items.Append("");
    for i:=1 to length(FonksiyonAdlari) do
        HareketSecim.Items.Append(copy(FonksiyonAdlari[i],4,100));
    fazGrubu.Visible := false;
    updownFazSayisi.Min := 2;
    updownFazSayisi.Max := EnBuyukFazSayisi;
    updownFazSayisi.Position := 2;
    txtDonusHizi.Text := '0';
    txtEksantirisite.Text := '0';
    txtBasincAcisi.Text := '0';
    txtToparлакYaricapi.Text := '0';
    radioDonusYonu.ItemIndex := 0;
    radioEksantirisiteYonu.ItemIndex := 0;
    with Fazlar[0] do begin
        FazTipi := 1;
        Yukseklik := 0;
        BitisAcisi := 0;
        FazFonksiyonu := 0;
        FazFonksiyonuDizini := 0;
    end;
    tabYolGrafiği.TabVisible := false;
    tabHizGrafiği.TabVisible := false;
    tabIvmeGrafiği.TabVisible := false;
    tabKamProfili.TabVisible := false;
    tabLog.TabVisible := false;
end;
//-----
procedure TForm1.txtFazSayisiChange(Sender: TObject);
//-----
var i: integer;
begin
    try
        i := strtoint(trim(txtFazSayisi.Text));
        if (i > EnBuyukFazSayisi) then raise Exception.Create("");
        BilgiGrid.RowCount := strtoint(trim(txtFazSayisi.Text)) + 1;
        FazAdet := strtoint(trim(txtFazSayisi.Text));
        for i := 1 to BilgiGrid.RowCount do BilgiGrid.Cells[0,i] := inttostr(i);
    except
        MessageDlg('Faz sayısı değeri hatalı. 2 ile '+inttostr(EnBuyukFazSayisi)+' arasında olmalı',mtError,[mbOk],0);
    end;
end;
//-----
procedure TForm1.FormActivate(Sender: TObject);
//-----
begin
    txtFazSayisi.SetFocus;
    txtFazSayisiChange(Sender);
end;
```

EK-2 Devamı

```
txtDonusHizi.Text := '100';
txtEksantirisite.Text := '30';
txtBasincAcisi.Text := '45';
txtToparлакYaricapi.Text := '10';
txtProjeAdi.Text := 'Örnek kam tasarımı';
Fazlar[1].FazTipi := 1;
Fazlar[1].Yukseklık := 100;
Fazlar[1].BitisAcisi := 180;
Fazlar[1].FazFonksiyonu := 10;
Fazlar[1].FazFonksiyonuDizini := 1;
bilgiGrid.Cells[1,1] := fazSecim.Items[1];
bilgiGrid.Cells[2,1] := '100';
bilgiGrid.Cells[3,1] := '180';
bilgiGrid.Cells[4,1] := hareketSecim.Items[1];
Fazlar[2].FazTipi := 3;
Fazlar[2].Yukseklık := 0;
Fazlar[2].BitisAcisi := 360;
Fazlar[2].FazFonksiyonu := 10;
Fazlar[2].FazFonksiyonuDizini := 1;
bilgiGrid.Cells[1,2] := fazSecim.Items[3];
bilgiGrid.Cells[2,2] := '0';
bilgiGrid.Cells[3,2] := '360';
bilgiGrid.Cells[4,2] := hareketSecim.Items[1];
end;
//-----
procedure TForm1.BilgiGridMouseUp(Sender: TObject; Button: TMouseButton;
//-----
Shift: TShiftState; X, Y: Integer);
var ACol,ARow: integer;
begin
BilgiGrid.MouseToCell(X,Y,ACol,ARow);
if ARow >= 0 then begin
SecilenFaz := ARow;
FazSecim.ItemIndex := Fazlar[SecilenFaz].FazTipi;
HareketSecim.ItemIndex := Fazlar[SecilenFaz].FazFonksiyonuDizini;
updownAci.Position := Fazlar[SecilenFaz].BitisAcisi;
txtAci.Text := floattostr(Fazlar[SecilenFaz].BitisAcisi);
txtYukseklık.Text := floattostr(Fazlar[SecilenFaz].Yukseklık);
fazGrubu.Caption := '+' + inttostr(ARow) + '. Faz bilgileri: ';
fazGrubu.Visible := true;
PageControl1.Visible := false;
Label10.Visible := false;
end;
end;
//-----
procedure TForm1.TamsayiTusKontrol(Sender: TObject; var Key: Char);
//-----
begin
case Key of
#8 : Key := Key;
'0'..'9': Key := Key;
else Key := #0;
end;
end;
//-----
procedure TForm1.KesirliSayiTusKontrol(Sender: TObject; var Key: Char);
//-----
begin
case Key of
#8 : Key := Key;
'0'..'9': Key := Key;
'.': Key := Key;
else Key := #0;
end;
end;
//-----
procedure TForm1.buttonIslemiBaslatClick(Sender: TObject);
//-----
var X, Y, Y1, X1 : extended; // Geçici kullanılan değişkenler
```

EK-2 Devamı

Acı, Uzaklık : extended; // Geçici kullanılan değişkenler
AcısalAdım : extended; // Radyan cinsinden her bir adımdaki ilerleme
EnKucukX, EnBuyukX : extended; // Ani dönme merkezleri yönergesinin X sınırları
i,j : integer; // Geçici kullanılan döngü değişkeni
FazNo : integer; // Fazları izleyen döngü değişkeni
AcıNo : integer; // Faz boyunca derece cinsinden açıyı izleyen döngü değişkeni
AdımNo : integer; // Bütün işlem adımlarını izleyen döngü değişkeni
AralıkNo : integer; // Her 1 derecelik adım içindeki aralıkları izleyen döngü değişkeni
Mesafe : extended; // Geçici teğet doğrusu ile ADM noktası arasındaki uzaklık
Beta : extended; // Faz boyunca açı değişimi (radyan)
DeltaTeta : extended; // Faz başlangıcından sonuna kadar açılal ilerleme (radyan)
DeltaH : extended; // Faz boyunca yükseklik değişimi (mm)
EnKisaAralik1 : extended; // Sağ teğet ile ADM noktaları arasındaki en kısa aralık değeri
EnKisaAralik2 : extended; // Sol teğet ile ADM noktaları arasındaki en kısa aralık değeri
TegetlerinKesimNoktasi : Nokta; // Sağ ve Sol teğetlerin kesişim noktası
NoktaAcisi : extended; // ADM noktasını dönme merkezine bağlayan doğrunun açısı
MaxX,MaxY,MinX,MinY : extended; // Kam profilinin sınırları
Katsayi : extended; // Kam profilini çizmek için kullanılacak büyütme/küçültme ölçeği

```
begin
tabYolGrafigi.TabVisible := false;
tabHizGrafigi.TabVisible := false;
tabIvmeGrafigi.TabVisible := false;
tabKamProfili.TabVisible := false;
tabLog.TabVisible := false;
if trim(txtFazSayisi.Text) = '' then begin
    MessageDlg('Faz sayısı girilmemiş',mtError,[mbOk],0);
    exit;
end;
if trim(txtDonusHizi.Text) = '' then begin
    MessageDlg('Dönüş hızı girilmemiş',mtError,[mbOk],0);
    exit;
end;
if trim(txtEksantirisite.Text) = '' then begin
    MessageDlg('Eksantirisite değeri girilmemiş',mtError,[mbOk],0);
    exit;
end;
if trim(txtBasincAcisi.Text) = '' then begin
    MessageDlg('Basınç açısı girilmemiş',mtError,[mbOk],0);
    exit;
end;
if trim(txtToparлакYaricapi.Text) = '' then begin
    MessageDlg('Toparлак yarıçapı girilmemiş',mtError,[mbOk],0);
    exit;
end;
memoLog.Clear;
DegerleriGoster := checkDebug.Checked;
FazAdet := strtoint(trim(txtFazSayisi.Text));
DonusHizi := StrToFloat(trim(txtDonusHizi.Text));
DonusSaatYonunde := radioDonusYonu.ItemIndex = 0;
Eksantirisite := StrToFloat(trim(txtEksantirisite.Text));
EksantirisiteSagda := radioEksantirisiteYonu.ItemIndex = 0;
BasincAcisi := StrToFloat(trim(txtBasincAcisi.Text));
ToparлакYaricapi := StrToFloat(trim(txtToparлакYaricapi.Text));
AcısalHiz := (DonusHizi * 2 * Pi) / 60.0;
if DonusSaatYonunde then AcısalHiz := - AcısalHiz;
if DegerleriGoster then memoLog.Lines.Append('Açılal hız = '+format('10e',[AcısalHiz]));

if (FazAdet < 2) or (FazAdet > 100) then begin
    MessageDlg('Faz sayısı değeri sınırlar dışında; 2 <= Faz sayısı <= 100 olmalı',mtError,[mbOk],0);
    exit;
end;
if (DonusHizi < 1) or (DonusHizi > 10000) then begin
    MessageDlg('Dönüş hızı değeri sınırlar dışında; 1 <= DonusHizi <= 10000 olmalı',mtError,[mbOk],0);
    exit;
end;
if (BasincAcisi < 1) or (BasincAcisi > 63) then begin
    MessageDlg('Basınç açısı değeri sınırlar dışında; 1 <= BasincAcisi <= 63 olmalı',mtError,[mbOk],0);
    exit;
end;
```

EK-2 Devamı

```
end;
for i := 1 to FazAdet do begin
  if Fazlar[i].FazTipi = 0 then begin
    MessageDlg(inttostr(i)+' fazın tipi belirlenmemiş',mtError,[mbOk],0);
    exit;
  end;
  if (Fazlar[i].FazTipi = 1) and (Fazlar[i - 1].Yukseklk >= Fazlar[i].Yukseklk) then begin
    MessageDlg(inttostr(i)+' fazın tipi çıkış olduğu halde, yükseklik değeri önceki fazdan düşük',mtError,[mbOk],0);
    exit;
  end;
  if (Fazlar[i].FazTipi = 2) and (Fazlar[i - 1].Yukseklk <> Fazlar[i].Yukseklk) then begin
    MessageDlg(inttostr(i)+' fazın tipi bekleme olduğu halde, yükseklik değeri önceki fazdan farklı',mtError,[mbOk],0);
    exit;
  end;
  if (Fazlar[i].FazTipi = 3) and (Fazlar[i - 1].Yukseklk <= Fazlar[i].Yukseklk) then begin
    MessageDlg(inttostr(i)+' fazın tipi iniş olduğu halde, yükseklik değeri önceki fazdan büyük',mtError,[mbOk],0);
    exit;
  end;
  if (Fazlar[i].BitisAcisi < 0) or (Fazlar[i].BitisAcisi > 360) then begin
    MessageDlg(inttostr(i)+' faz bitiş açısı değeri sınırlar dışında; 0 <= Açık <= 360 olmalı',mtError,[mbOk],0);
    exit;
  end;
  if Fazlar[i].BitisAcisi <= Fazlar[i - 1].BitisAcisi then begin
    MessageDlg(inttostr(i)+' faz bitiş açısı, önceki fazdan büyük değil',mtError,[mbOk],0);
    exit;
  end;
end;
if Fazlar[FazAdet].Yukseklk <> 0 then begin
  MessageDlg(inttostr(FazAdet)+' faz son faz olduğu halde, yükseklik değeri 0 değil',mtError,[mbOk],0);
  exit;
end;
Series1.Clear;
Series2.Clear;
Series3.Clear;
AcisalAdim := (2 * Pi) / ToplamAdimSayisi;
AdimNo := 0;
Series1.AddXY(0,0,'cBlack');
Adimlar[0].Yol := 0;
Adimlar[0].Hiz := 0;
Adimlar[0].Ivme:= 0;
Adimlar[0].DonusAcisi:= 0;
for FazNo := 1 to FazAdet do begin
  Beta := DegToRad(Fazlar[FazNo].BitisAcisi - Fazlar[FazNo - 1].BitisAcisi);
  DeltaTeta := 0;
  DeltaH := Abs(Fazlar[FazNo].Yukseklk - Fazlar[FazNo - 1].Yukseklk);
  for Acino := Fazlar[FazNo - 1].BitisAcisi + 1 to Fazlar[FazNo].BitisAcisi do begin
    for AralikNo := 1 to AraliktakiAdimSayisi do begin
      DeltaTeta := DeltaTeta + AcisalAdim;
      AdimNo := AdimNo + 1;
      //
      // Yol değeri hareketin tipine göre hesaplanır
      //
      if Fazlar[FazNo].FazTipi = 1
      then Adimlar[AdimNo].Yol := Fazlar[FazNo - 1].Yukseklk + YolHesapla(Fazlar[FazNo].FazFonksiyonu, DeltaH,
DeltaTeta, Beta);
      if Fazlar[FazNo].FazTipi = 2 then Adimlar[AdimNo].Yol := Fazlar[FazNo - 1].Yukseklk;
      if Fazlar[FazNo].FazTipi = 3
      then Adimlar[AdimNo].Yol := Fazlar[FazNo - 1].Yukseklk - YolHesapla(Fazlar[FazNo].FazFonksiyonu, DeltaH,
DeltaTeta, Beta);
      //
      // Hız ve ivme değerlerini önceki adım ile aradaki yol farkından hesaplayabiliriz
      // Türevlere bakmak çok gerekli değil
      //
      Adimlar[AdimNo].Hiz := (Adimlar[AdimNo].Yol - Adimlar[AdimNo - 1].Yol) / AcisalAdim;
      Adimlar[AdimNo].Ivme:= (Adimlar[AdimNo].Hiz - Adimlar[AdimNo - 1].Hiz)
/ AcisalAdim;
      //
      // Bu adımdaki dönüş açısını saklayalım ileride gerekli olacak
      //
    end;
  end;
end;
```

EK-2 Devamı

```
Adimlar[AdimNo].DonusAcisi := Adimlar[AdimNo - 1].DonusAcisi + AcisalAdim;
//
// Hesaplanan deęerleri ilgili grafiklere koyalım
//
Series1.AddXY(AdimNo / AraliktakiAdimSayisi,Adimlar[AdimNo].Yol,inttostr(AciNo),clBlack);
Series2.AddXY(AdimNo / AraliktakiAdimSayisi,Adimlar[AdimNo].Hiz,inttostr(AciNo),clBlack);
Series3.AddXY(AdimNo / AraliktakiAdimSayisi,Adimlar[AdimNo].Ivme,inttostr(AciNo),clBlack);
if DegerleriGoster then memoLog.Lines.Append(format('%5.5d . adım: Yol = %15.10e Hız = %15.10e Aç1 =
%15.10e',[AdimNo,Adimlar[AdimNo].Yol,Adimlar[AdimNo].Hiz,Adimlar[AdimNo].DonusAcisi]));
end;
end;
end;
//
// Grafiklerin alt ve üst tarafında biraz boşluk kalması için ayarlama yap
//
Series1.GetVertAxis.Maximum:=false;
Series1.GetVertAxis.Minimum := Series1.MaxYValue * 1.1;
Series2.GetVertAxis.Maximum:=false;
Series2.GetVertAxis.Minimum:=false;
Series2.GetVertAxis.Maximum := Series2.MaxYValue * 1.1;
Series2.GetVertAxis.Minimum := Series2.MinYValue * 1.1;
Series3.GetVertAxis.Maximum:=false;
Series3.GetVertAxis.Minimum:=false;
Series3.GetVertAxis.Maximum := Series3.MaxYValue * 1.1;
Series3.GetVertAxis.Minimum := Series3.MinYValue * 1.1;
//
// Ani dönme merkezi yörüngelerini hesapla
//
EnKucukX := 999999.0;
EnBuyukX := -999999.0;
for i := 1 to ToplamAdimSayisi do begin
  Adimlar[i].AniDonmeMerkeziYorungesi.X := Adimlar[i].Hiz / (AcisalHiz / Abs(AcisalHiz));
  Adimlar[i].AniDonmeMerkeziYorungesi.Y := Adimlar[i].Yol;
  if (Adimlar[i].AniDonmeMerkeziYorungesi.X < 0) and (Adimlar[i].AniDonmeMerkeziYorungesi.X < EnKucukX)
  then EnKucukX := Adimlar[i].AniDonmeMerkeziYorungesi.X;
  if (Adimlar[i].AniDonmeMerkeziYorungesi.X >= 0) and (Adimlar[i].AniDonmeMerkeziYorungesi.X > EnBuyukX)
  then EnBuyukX := Adimlar[i].AniDonmeMerkeziYorungesi.X;
  if DegerleriGoster then memoLog.Lines.Append(format('%5.5d . adım ADM yörüngesi: X = %15.10e Y =
%15.10e',[i,Adimlar[i].AniDonmeMerkeziYorungesi.X,Adimlar[i].AniDonmeMerkeziYorungesi.Y]));
end;
//
// Sağdan ve soldan teęet doęrularının formüllerini bul
// Geçici olarak sabit deęerleri yörüngenin dış tarafında al
//
SoldanTegetDogru.Egim := Tan(DegToRad(90.0 + BasincAcisi));
SoldanTegetDogru.Sabit := EnKucukX;
SagdanTegetDogru.Egim := Tan(DegToRad(90.0 - BasincAcisi));
SagdanTegetDogru.Sabit := - EnBuyukX;
if DegerleriGoster then memoLog.Lines.Append('Soldan teęet (başlangıç): Eęim = '+floattostr(SoldanTegetDogru.Egim)+' Sabit
='+floattostr(SoldanTegetDogru.Sabit));
if DegerleriGoster then memoLog.Lines.Append('Saędan teęet (başlangıç): Eęim = '+floattostr(SagdanTegetDogru.Egim)+' Sabit
='+floattostr(SagdanTegetDogru.Sabit));
//
// Teęet doęrular üzerindeki noktaların, yörünge üzerindeki noktalara uzaklıklarını hesapla
// Soldan ve saędan en küçük aralık farklarını bul
//
EnKisaAralik1 := 99999999;
EnKisaAralik2 := 99999999;
for i := 1 to ToplamAdimSayisi do begin
  X := Adimlar[i].AniDonmeMerkeziYorungesi.X;
  Y := Adimlar[i].AniDonmeMerkeziYorungesi.Y;
  if X >= 0 then begin
    X1 := (Y - SagdanTegetDogru.Sabit) / SagdanTegetDogru.Egim;
    if (X1 - X) < EnKisaAralik1 then EnKisaAralik1 := X1 - X;
  end
  else begin
    X1 := (Y - SoldanTegetDogru.Sabit) / SoldanTegetDogru.Egim;
    if X1 < 0 then Mesafe := Abs(X1 - X);
    if X1 >= 0 then Mesafe := Abs(X) + X1;
  end
end
```

EK-2 Devamı

```
if X - X1 < EnKisaAralik2 then EnKisaAralik2 := X - X1;
end;
end;
//
// Teğet doğruların sabit değerlerini, bulunan farkları kullanarak düzelt
//
SagdanTegetDogru.Sabit := SagdanTegetDogru.Sabit + EnKisaAralik1;
SoldanTegetDogru.Sabit := SoldanTegetDogru.Sabit + EnKisaAralik2;
if DegerleriGoster then memoLog.Lines.Append('Soldan teğet (sonuç): Eğim = '+floattostr(SoldanTegetDogru.Egim)+' Sabit = '+floattostr(SoldanTegetDogru.Sabit));
if DegerleriGoster then memoLog.Lines.Append('Sağdan teğet (sonuç): Eğim = '+floattostr(SagdanTegetDogru.Egim)+' Sabit = '+floattostr(SagdanTegetDogru.Sabit));
//
// Teğet doğruların kesişim noktasının koordinatlarını bul
//
TegetlerinKesimNoktasi.X := (SoldanTegetDogru.Sabit - SagdanTegetDogru.Sabit) / (SagdanTegetDogru.Egim - SoldanTegetDogru.Egim);
TegetlerinKesimNoktasi.Y := (SagdanTegetDogru.Egim * TegetlerinKesimNoktasi.X) + SagdanTegetDogru.Sabit;
if DegerleriGoster then memoLog.Lines.Append('Kesişim noktası: X = '+floattostr(TegetlerinKesimNoktasi.X)+' Y = '+floattostr(TegetlerinKesimNoktasi.Y));
//
// Dönme merkezinin koordinatlarını bul
// Eksantirisitenin ne tarafta olduğunu
// Teğet doğruların kesişim noktasının mecburi eksantirisite yaratıp yaratmadığımı
// dikkate al
//
if TegetlerinKesimNoktasi.X = 0 then begin
//
// Kesişim noktası çıkış uzvu doğrultusunda; verilen eksantirisite değerleri kullanılabilir
//
if EksantirisiteSagda then begin
DonmeMerkezi.X := Eksantirisite;
DonmeMerkezi.Y := (SoldanTegetDogru.Egim * DonmeMerkezi.X) + SoldanTegetDogru.Sabit;
end
else begin
DonmeMerkezi.X := -1 * Eksantirisite;
DonmeMerkezi.Y := (SagdanTegetDogru.Egim * DonmeMerkezi.X) + SagdanTegetDogru.Sabit;
end;
end;
if TegetlerinKesimNoktasi.X > 0 then begin
//
// Kesişim noktası çıkış uzvu doğrultusunun sağında; mecburi eksantirisite var
//
if EksantirisiteSagda then begin
if Eksantirisite > TegetlerinKesimNoktasi.X then begin
//
// Eksantirisite değeri mecburi eksantirisiteden büyük, bu durumda bizi etkilemez
//
DonmeMerkezi.X := Eksantirisite;
DonmeMerkezi.Y := (SoldanTegetDogru.Egim * DonmeMerkezi.X) + SoldanTegetDogru.Sabit;
end
else begin
//
// Eksantirisite değeri mecburi eksantirisiteden küçük, eksantirisite noktası değişmeli
//
MessageDlg('Verilen eksantirisite değerine göre dönme merkezi teğetlerin kesim noktasının üstünde kalıyor. Eksantirisite > '+floattostr(TegetlerinKesimNoktasi.X) + ' olmalı',mtError,[mbOk],0);
exit;
end;
end
else begin
//
// Eksantirisitenin solda olması isteniyordu, bu durumda bizi etkilemez
//
DonmeMerkezi.X := -1 * Eksantirisite;
DonmeMerkezi.Y := (SagdanTegetDogru.Egim * DonmeMerkezi.X) + SagdanTegetDogru.Sabit;
end;
end;
if TegetlerinKesimNoktasi.X < 0 then begin
```

EK-2 Devamı

```
//
// Kesişim noktası çıkış uzvu doğrultusunun solunda; mecburi eksantirisite var
//
if EksantirisiteSagda then begin
//
// Eksantirisitenin sağda olması isteniyordu, bu durumda bizi etkilemez
//
DonmeMerkezi.X := Eksantirisite;
DonmeMerkezi.Y := (SoldanTegetDogru.Egim * DonmeMerkezi.X) + SoldanTegetDogru.Sabit;
end
else begin
if Eksantirisite > Abs(TegetlerinKesimNoktasi.X) then begin
//
// Eksantirisite değeri mecburi eksantirisiteden büyük, bu durumda bizi etkilemez
//
DonmeMerkezi.X := -1 * Eksantirisite;
DonmeMerkezi.Y := (SagdanTegetDogru.Egim * DonmeMerkezi.X) + SagdanTegetDogru.Sabit;
end
else begin
//
// Eksantirisite değeri mecburi eksantirisiteden küçük, eksantirisite noktası değişmeli
//
MessageDlg('Verilen eksantirisite değerine göre dönme merkezi tegetlerin kesim noktasının üstünde kalıyor. Eksantirisite < '
+ floatostr(TegetlerinKesimNoktasi.X) + ' olmalı',mtError,[mbOk],0);
exit;
end;
end;
end;
if DegerleriGoster then memoLog.Lines.Append('Dönme merkezi: X = '+floatostr(DonmeMerkezi.X)+' Y =
'+floatostr(DonmeMerkezi.Y));
//
// Kam profilini çıkart
//
for i := 1 to ToplamAdimSayisi do begin
//
// Teorik kam profili koordinatlarını hesapla
//
Adimlar[i].TeorikKamProfili.DonmeMerkezineUzaklik := Power(Power(DonmeMerkezi.Y - Adimlar[i].Yol, 2.0) +
Power(DonmeMerkezi.X, 2.0), 0.5);
if EksantirisiteSagda
then Adimlar[i].TeorikKamProfili.DonmeMerkeziIleYaptigiAci := Pi - ArcCos(Abs(DonmeMerkezi.X) /
Adimlar[i].TeorikKamProfili.DonmeMerkezineUzaklik)
else Adimlar[i].TeorikKamProfili.DonmeMerkeziIleYaptigiAci := ArcCos(Abs(DonmeMerkezi.X) /
Adimlar[i].TeorikKamProfili.DonmeMerkezineUzaklik);
if DonusSaatYonunde
then Adimlar[i].TeorikKamProfili.DondurulmusAci := Adimlar[i].TeorikKamProfili.DonmeMerkeziIleYaptigiAci +
Adimlar[i].DonusAcisi
else Adimlar[i].TeorikKamProfili.DondurulmusAci := Adimlar[i].TeorikKamProfili.DonmeMerkeziIleYaptigiAci -
Adimlar[i].DonusAcisi;

Adimlar[i].TeorikKamProfili.X := DonmeMerkezi.X + (Adimlar[i].TeorikKamProfili.DonmeMerkezineUzaklik *
Cos(Adimlar[i].TeorikKamProfili.DondurulmusAci));
Adimlar[i].TeorikKamProfili.Y := DonmeMerkezi.Y + (Adimlar[i].TeorikKamProfili.DonmeMerkezineUzaklik *
Sin(Adimlar[i].TeorikKamProfili.DondurulmusAci));
if DegerleriGoster then memoLog.Lines.Append(format('%5.5d . adım Kam Profili: X = %15.10e Y = %15.10e Açısı =
%15.10e Uzaklık =
%15.10e',i,Adimlar[i].TeorikKamProfili.X,Adimlar[i].TeorikKamProfili.Y,Adimlar[i].TeorikKamProfili.DondurulmusAci,Adimlar
[i].TeorikKamProfili.DonmeMerkezineUzaklik));
//
// Toparlık var ise gerçek kam profili koordinatlarını hesapla
//
if ToparlıkYaricapi > 0 then begin
Adimlar[i].IcKamProfili.DonmeMerkezineUzaklik := Power(Power(DonmeMerkezi.Y - (Adimlar[i].Yol - ToparlıkYaricapi),
2.0) + Power(DonmeMerkezi.X, 2.0), 0.5);
if EksantirisiteSagda
then Adimlar[i].IcKamProfili.DonmeMerkeziIleYaptigiAci := Pi - ArcCos(Abs(DonmeMerkezi.X) /
Adimlar[i].IcKamProfili.DonmeMerkezineUzaklik)
else Adimlar[i].IcKamProfili.DonmeMerkeziIleYaptigiAci := ArcCos(Abs(DonmeMerkezi.X) /
Adimlar[i].IcKamProfili.DonmeMerkezineUzaklik);
```


EK-2 Devamı

```
    if DonusSaatYonunde
        then Adimlar[i].IcKamProfili.DondurulmusAci := Adimlar[i].IcKamProfili.DonmeMerkeziIleYaptigiAci +
        Adimlar[i].DonusAcisi
        else Adimlar[i].IcKamProfili.DondurulmusAci := Adimlar[i].IcKamProfili.DonmeMerkeziIleYaptigiAci -
        Adimlar[i].DonusAcisi;
        Adimlar[i].IcKamProfili.X := DonmeMerkezi.X + (Adimlar[i].IcKamProfili.DonmeMerkezineUzaklik *
        Cos(Adimlar[i].IcKamProfili.DondurulmusAci));
        Adimlar[i].IcKamProfili.Y := DonmeMerkezi.Y + (Adimlar[i].IcKamProfili.DonmeMerkezineUzaklik *
        Sin(Adimlar[i].IcKamProfili.DondurulmusAci));

        Adimlar[i].DisKamProfili.DonmeMerkezineUzaklik := Power(Power(DonmeMerkezi.Y - (Adimlar[i].Yol +
        ToparlakYaricapi), 2.0) + Power(DonmeMerkezi.X, 2.0), 0.5);
        if EksantirisiteSagda
            then Adimlar[i].DisKamProfili.DonmeMerkeziIleYaptigiAci := Pi - ArcCos(Abs(DonmeMerkezi.X) /
            Adimlar[i].DisKamProfili.DonmeMerkezineUzaklik)
            else Adimlar[i].DisKamProfili.DonmeMerkeziIleYaptigiAci := ArcCos(Abs(DonmeMerkezi.X) /
            Adimlar[i].DisKamProfili.DonmeMerkezineUzaklik);
            if DonusSaatYonunde
                then Adimlar[i].DisKamProfili.DondurulmusAci := Adimlar[i].DisKamProfili.DonmeMerkeziIleYaptigiAci +
                Adimlar[i].DonusAcisi
                else Adimlar[i].DisKamProfili.DondurulmusAci := Adimlar[i].DisKamProfili.DonmeMerkeziIleYaptigiAci -
                Adimlar[i].DonusAcisi;
                Adimlar[i].DisKamProfili.X := DonmeMerkezi.X + (Adimlar[i].DisKamProfili.DonmeMerkezineUzaklik *
                Cos(Adimlar[i].DisKamProfili.DondurulmusAci));
                Adimlar[i].DisKamProfili.Y := DonmeMerkezi.Y + (Adimlar[i].DisKamProfili.DonmeMerkezineUzaklik *
                Sin(Adimlar[i].DisKamProfili.DondurulmusAci));
            end
            else begin
                Adimlar[i].IcKamProfili := Adimlar[i].TeorikKamProfili;
                Adimlar[i].DisKamProfili := Adimlar[i].TeorikKamProfili;
            end;
        end;
    //
    // Kam profilini çizimi
    //
    with kamProfiliResmi.Canvas do begin
        //
        // Resim alanını temizle
        //
        kamProfiliResmi.Picture.Graphic.Width := kamProfiliResmi.Width;
        kamProfiliResmi.Picture.Graphic.Height := kamProfiliResmi.Height;
        Brush.Color := clWhite;
        Brush.Style := bssolid;
        FillRect(rect(0,0,kamProfiliResmi.Width - 1,kamProfiliResmi.Height - 1));
        //
        // Teorik profili düzgün çizebilmemiz için min ve max değerleri bilmemiz gerek
        //
        MinX:= 999999;
        MinY:= 999999;
        MaxX:=-999999;
        MaxY:=-999999;
        for i := 1 to ToplamAdimSayisi do begin
            if MaxX < Adimlar[i].DisKamProfili.X then MaxX := Adimlar[i].DisKamProfili.X;
            if MinX > Adimlar[i].DisKamProfili.X then MinX := Adimlar[i].DisKamProfili.X;
            if MaxY < Adimlar[i].DisKamProfili.Y then MaxY := Adimlar[i].DisKamProfili.Y;
            if MinY > Adimlar[i].DisKamProfili.Y then MinY := Adimlar[i].DisKamProfili.Y;
        end;
        MinX := MinX * 1.1;
        MaxX := MaxX * 1.1;
        MinY := MinY * 1.1;
        MaxY := MaxY * 1.1;
        //
        // Kam profilinin resim alanına tam uyması için ne kadar
        // büyütme ya da küçültme yapmamız gerektiğini hesaplayalım
        //
        if (kamProfiliResmi.Width / (MaxX - MinX)) < (kamProfiliResmi.Height / (MaxY - MinY))
            then Katsayi := kamProfiliResmi.Width / (MaxX - MinX)
            else Katsayi := kamProfiliResmi.Height / (MaxY - MinY);
```

EK-2 Devamı

```
//
//
// Teorik profili çiz
//
Pen.Color := clBlack;
Pen.Style := psSolid;
X1 := Katsayi * (Adimlar[1].TeorikKamProfili.X - MinX);
Y1 := Katsayi * (MaxY - Adimlar[1].TeorikKamProfili.Y);
MoveTo(round(X1),round(Y1));
for i := 2 to ToplamAdimSayisi do begin
  X := Katsayi * (Adimlar[i].TeorikKamProfili.X - MinX);
  Y := Katsayi * (MaxY - Adimlar[i].TeorikKamProfili.Y);
  LineTo(round(X), round(Y));
end;
LineTo(round(X1), round(Y1));
if ToparlakYaricapi > 0 then begin
  //
  // İç profili çiz
  //
  Pen.Color := clTeal;
  Pen.Style := psSolid;
  X1 := Katsayi * (Adimlar[1].IcKamProfili.X - MinX);
  Y1 := Katsayi * (MaxY - Adimlar[1].IcKamProfili.Y);
  MoveTo(round(X1),round(Y1));
  for i := 2 to ToplamAdimSayisi do begin
    X := Katsayi * (Adimlar[i].IcKamProfili.X - MinX);
    Y := Katsayi * (MaxY - Adimlar[i].IcKamProfili.Y);
    LineTo(round(X), round(Y));
  end;
  LineTo(round(X1), round(Y1));
  //
  // Dış profili çiz
  //
  Pen.Color := clRed;
  Pen.Style := psSolid;
  X1 := Katsayi * (Adimlar[1].DisKamProfili.X - MinX);
  Y1 := Katsayi * (MaxY - Adimlar[1].DisKamProfili.Y);
  MoveTo(round(X1),round(Y1));
  for i := 2 to ToplamAdimSayisi do begin
    X := Katsayi * (Adimlar[i].DisKamProfili.X - MinX);
    Y := Katsayi * (MaxY - Adimlar[i].DisKamProfili.Y);
    LineTo(round(X), round(Y));
  end;
  LineTo(round(X1), round(Y1));
end;
//
// Dönme merkezini işaretle
//
Brush.Color := clBlack;
Brush.Style := bsSolid;
i := round(Katsayi * (DonmeMerkezi.X - MinX));
j := round(Katsayi * (MaxY - DonmeMerkezi.Y));
FillRect(rect(i-1,j-1,i+1,j+1));
Brush.Color := clWhite;
Brush.Style := bsClear;
//
// Varsa eksantirisite çemberini çiz
//
if Eksantirisite > 0 then begin
  Pen.Color := clBlack;
  Pen.Style := psSolid;
  Ellipse( round(Katsayi * (DonmeMerkezi.X - MinX - Eksantirisite)),
            round(Katsayi * (MaxY - DonmeMerkezi.Y - Eksantirisite)),
            round(Katsayi * (DonmeMerkezi.X - MinX + Eksantirisite)),
            round(Katsayi * (MaxY - DonmeMerkezi.Y + Eksantirisite))
          );
end;
//
// Fazların bitişlerini işaretle
```

EK-2 Devamı

```
//
Pen.Color := clBlack;
Pen.Style := psDot;
if Eksantirisite = 0 then begin
//
// Eksantirisite çemberi ve teğet yok, sadece dönme merkezi ile birleştir
//
X1 := Katsayi * (DonmeMerkezi.X - MinX);
Y1 := Katsayi * (MaxY - DonmeMerkezi.Y);
for i := 1 to FazAdet do begin
//
// Faz bitişi noktasının hangi adıma isabet ettiğini bul
//
j := Fazlar[i].BitisAcisi * AraliktakiAdimSayisi;
//
// Merkez ile teorik/dış kam profili noktasını birleştir
//
X := Katsayi * (Adimlar[j].DisKamProfili.X - MinX);
Y := Katsayi * (MaxY - Adimlar[j].DisKamProfili.Y);
MoveTo(round(X),round(Y));
LineTo(round(X1), round(Y1));
end;
end
else begin
for i := 1 to FazAdet do begin
//
// Faz bitişi noktasının hangi adıma isabet ettiğini bul
//
j := Fazlar[i].BitisAcisi * AraliktakiAdimSayisi;
//
// Teğet noktasına ulaşmak için ne kadarlık bir açı ile ilerleyeceğiz?
//
Aci := ArcCos(Eksantirisite / Adimlar[j].DisKamProfili.DonmeMerkezineUzaklik);
//
// Teğet noktası ile dış kam profil noktasını birleştir
//
if EksantirisiteSagda then begin
X := DonmeMerkezi.X + (Eksantirisite * Cos(Adimlar[j].DisKamProfili.DondurulmusAci + Aci));
Y := DonmeMerkezi.Y + (Eksantirisite * Sin(Adimlar[j].DisKamProfili.DondurulmusAci + Aci));
end
else begin
X := DonmeMerkezi.X + (Eksantirisite * Cos(Adimlar[j].DisKamProfili.DondurulmusAci - Aci));
Y := DonmeMerkezi.Y + (Eksantirisite * Sin(Adimlar[j].DisKamProfili.DondurulmusAci - Aci));
end;
end;
X := Katsayi * (X - MinX);
Y := Katsayi * (MaxY - Y);
X1 := Katsayi * (Adimlar[j].DisKamProfili.X - MinX);
Y1 := Katsayi * (MaxY - Adimlar[j].DisKamProfili.Y);
MoveTo(round(X),round(Y));
LineTo(round(X1), round(Y1));
end;
end;
end;
tabYolGrafigi.TabVisible := true;
tabHizGrafigi.TabVisible := true;
tabIvmeGrafigi.TabVisible := true;
tabKamProfili.TabVisible := true;
if DegerleriGoster then tabLog.TabVisible := true;
end;
//-----
procedure TForm1.FormResize(Sender: TObject);
//-----
begin
PageControl1.Width := Form1.Width - 20;
PageControl1.Height := Form1.Height - 75;
PageControl1.Invalidate;
yolChart.Width := tabKamProfili.Width - 20;
yolChart.Height := tabKamProfili.Height - 20;
hizChart.Width := tabKamProfili.Width - 20;
```

EK-2 Devami

```
hizChart.Height := tabKamProfili.Height - 20;
ivmeChart.Width := tabKamProfili.Width - 20;
ivmeChart.Height := tabKamProfili.Height - 20;
kamProfiliResmi.Width := tabKamProfili.Width - 10;
kamProfiliResmi.Height := tabKamProfili.Height - 10;
bilgiGrid.Width := tabKamProfili.Width - 320;
bilgiGrid.Height := tabKamProfili.Height - 100;
bilgiGrid.ColWidths[4] := bilgiGrid.Width - 350;
end;
//-----
procedure TForm1.bVazgecClick(Sender: TObject);
//-----
begin
  PageControl1.Visible := true;
  Label10.Visible := true;
  fazGrubu.Visible := false;
end;
//-----
procedure TForm1.bTamamClick(Sender: TObject);
//-----
begin
  if trim(txtYukseklk.Text) = '' then begin
    MessageDlg('Ulaşılacak yükseklik değeri girilmemiş',mtError,[mbOk],0);
    exit;
  end;
  if fazSecim.ItemIndex = 0 then begin
    MessageDlg('Faz tipi seçilmemiş',mtError,[mbOk],0);
    exit;
  end;
  if (hareketSecim.ItemIndex = 0) and (fazSecim.ItemIndex <> 2) then begin
    MessageDlg('Hareket fonksiyonu seçilmemiş',mtError,[mbOk],0);
    exit;
  end;
  try
    Fazlar[SecilenFaz].FazTipi := fazSecim.ItemIndex;
    if Fazlar[SecilenFaz].FazTipi = 2 then begin
      Fazlar[SecilenFaz].FazFonksiyonuDizini := 0;
      Fazlar[SecilenFaz].FazFonksiyonu := 0;
    end
  else begin
    Fazlar[SecilenFaz].FazFonksiyonuDizini := hareketSecim.ItemIndex;
    Fazlar[SecilenFaz].FazFonksiyonu := strtoint(copy(FonksiyonAdlari[hareketSecim.ItemIndex],1,2));
  end;
  Fazlar[SecilenFaz].Yukseklk := strtfloat(trim(txtYukseklk.Text));
  Fazlar[SecilenFaz].BitisAcisi := strtoint(trim(txtAci.Text));
  with bilgiGrid do begin
    Cells[1,SecilenFaz] := fazSecim.Text;
    Cells[2,SecilenFaz] := trim(txtYukseklk.Text);
    Cells[3,SecilenFaz] := trim(txtAci.Text);
    Cells[4,SecilenFaz] := hareketSecim.Text;
  end;
  PageControl1.Visible := true;
  Label10.Visible := true;
  fazGrubu.Visible := false;
except
  MessageDlg('Ulaşılacak yükseklik değeri hatalı',mtError,[mbOk],0);
end;
end;
//-----
procedure TForm1.buttonSaklaClick(Sender: TObject);
//-----
var Cikis: TextFile;
i: integer;
begin
  if SaveDialog1.Execute() then begin
    AssignFile(Cikis,SaveDialog1.FileName);
    Rewrite(Cikis);
    WriteLn(Cikis,'KamTasarımıParametreleri');
    WriteLn(Cikis,trim(TasarimBasligi));
```

EK-2 Devamı

```
WriteLn(Cikis,trim(txtFazSayisi.Text));
WriteLn(Cikis,trim(txtDonusHizi.Text));
WriteLn(Cikis,trim(txtEksantirisite.Text));
WriteLn(Cikis,trim(txtBasincAcisi.Text));
WriteLn(Cikis,trim(txtToparлакYaricapi.Text));
WriteLn(Cikis,inttostr(radioEksantirisiteYonu.ItemIndex));
WriteLn(Cikis,inttostr(radioDonusYonu.ItemIndex));
for i := 1 to FazAdet do begin
  WriteLn(Cikis,inttostr(Fazlar[i].FazTipi));
  WriteLn(Cikis,inttostr(Fazlar[i].FazFonksiyonuDizini));
  WriteLn(Cikis,inttostr(Fazlar[i].BitisAcisi));
  WriteLn(Cikis,floattostr(Fazlar[i].Yukseklik));
end;
CloseFile(Cikis);
MessageDlg('Bilgiler kaydedildi',mtInformation,[mbOk],0);
end;
end;
//-----
procedure TForm1.txtProjeAdiChange(Sender: TObject);
//-----
begin
  TasarimBasligi := trim(txtProjeAdi.Text);
end;
//-----
procedure TForm1.buttonYukleClick(Sender: TObject);
//-----
var Giris: TextFile;
    i: integer;
    s: string;
begin
  if OpenFileDialog1.Execute() then begin
    AssignFile(Giris,OpenDialog1.FileName);
    Reset(Giris);
    ReadLn(Giris,s);
    if trim(s) <> 'KamTasarimParametreleri' then begin
      MessageDlg('Dosya önceden oluşturulmuş bir saklama dosyası değil',mtInformation,[mbOk],0);
      CloseFile(Giris);
      exit;
    end;
    ReadLn(Giris,s); txtProjeAdi.Text := s;
    ReadLn(Giris,s); txtFazSayisi.Text := s;
    ReadLn(Giris,s); txtDonusHizi.Text := s;
    ReadLn(Giris,s); txtEksantirisite.Text := s;
    ReadLn(Giris,s); txtBasincAcisi.Text := s;
    ReadLn(Giris,s); txtToparлакYaricapi.Text := s;
    ReadLn(Giris,s); radioEksantirisiteYonu.ItemIndex := strtoint(trim(s));
    ReadLn(Giris,s); radioDonusYonu.ItemIndex := strtoint(trim(s));
    FazAdet := strtoint(trim(txtFazSayisi.Text));
    BilgiGrid.RowCount := FazAdet + 1;
    for i := 1 to FazAdet do begin
      ReadLn(Giris,s); Fazlar[i].FazTipi := strtoint(trim(s));
      BilgiGrid.Cells[1,i] := fazSecim.Items[Fazlar[i].FazTipi];
      ReadLn(Giris,s); Fazlar[i].FazFonksiyonuDizini := strtoint(trim(s));
      BilgiGrid.Cells[4,i] := copy(FonksiyonAdlari[Fazlar[i].FazFonksiyonuDizini],4,100);
      Fazlar[i].FazFonksiyonu := strtoint(copy(FonksiyonAdlari[Fazlar[i].FazFonksiyonuDizini],1,2));
      ReadLn(Giris,s); Fazlar[i].BitisAcisi := strtoint(trim(s));
      BilgiGrid.Cells[3,i] := s;
      ReadLn(Giris,s); Fazlar[i].Yukseklik := strtfloat(trim(s));
      BilgiGrid.Cells[2,i] := s;
    end;
    CloseFile(Giris);
  end;
end;
end;
procedure TForm1.buttonYazdirClick(Sender: TObject);
var X, Y, Y1, X1, Acı : extended; // Geçici kullanılan değişkenler
    i,j : integer; // Geçici kullanılan döngü değişkeni
    MaxX,MaxY,MinX,MinY : extended; // Kam profilinin sınırları
    Katsayi : extended; // Kam profilini çizmek için kullanılacak büyütmek/küçültme ölçüğü
```

EK-2 Devamı

```
begin
  if not PrintDialog1.Execute() then exit;
  Printer.BeginDoc;
  //
  // Kam profilini çizimi
  //
  with Printer.Canvas do begin
    Brush.Color := clWhite;
    Brush.Style := bssolid;
    //
    // Teorik profili düzgün çizebilmemiz için min ve max değerleri bilmemiz gerek
    //
    MinX:= 999999;
    MinY:= 999999;
    MaxX:=-999999;
    MaxY:=-999999;
    for i := 1 to ToplamAdimSayisi do begin
      if MaxX < Adimlar[i].DisKamProfili.X then MaxX := Adimlar[i].DisKamProfili.X;
      if MinX > Adimlar[i].DisKamProfili.X then MinX := Adimlar[i].DisKamProfili.X;
      if MaxY < Adimlar[i].DisKamProfili.Y then MaxY := Adimlar[i].DisKamProfili.Y;
      if MinY > Adimlar[i].DisKamProfili.Y then MinY := Adimlar[i].DisKamProfili.Y;
    end;
    MinX := MinX * 1.1;
    MaxX := MaxX * 1.1;
    MinY := MinY * 1.1;
    MaxY := MaxY * 1.1;
    //
    // Kam profilinin resim alanına tam uyması için ne kadar
    // büyütme ya da küçültme yapmamız gerektiğini hesaplayalım
    //
    if (Printer.PageWidth / (MaxX - MinX)) < (Printer.PageHeight / (MaxY - MinY))
      then Katsayi := Printer.PageWidth / (MaxX - MinX)
      else Katsayi := Printer.PageHeight / (MaxY - MinY);
    //
    //
    // Teorik profili çiz
    //
    Pen.Color := clBlack;
    Pen.Style := pssolid;
    X1 := Katsayi * (Adimlar[1].TeorikKamProfili.X - MinX);
    Y1 := Katsayi * (MaxY - Adimlar[1].TeorikKamProfili.Y);
    MoveTo(round(X1),round(Y1));
    for i := 2 to ToplamAdimSayisi do begin
      X := Katsayi * (Adimlar[i].TeorikKamProfili.X - MinX);
      Y := Katsayi * (MaxY - Adimlar[i].TeorikKamProfili.Y);
      LineTo(round(X), round(Y));
    end;
    LineTo(round(X1), round(Y1));
    //
    // İç profili çiz
    //
    Pen.Color := clTeal;
    Pen.Style := pssolid;
    X1 := Katsayi * (Adimlar[1].IcKamProfili.X - MinX);
    Y1 := Katsayi * (MaxY - Adimlar[1].IcKamProfili.Y);
    MoveTo(round(X1),round(Y1));
    for i := 2 to ToplamAdimSayisi do begin
      X := Katsayi * (Adimlar[i].IcKamProfili.X - MinX);
      Y := Katsayi * (MaxY - Adimlar[i].IcKamProfili.Y);
      LineTo(round(X), round(Y));
    end;
    LineTo(round(X1), round(Y1));
    //
    // Dış profili çiz
    //
    Pen.Color := clRed;
    Pen.Style := pssolid;
    X1 := Katsayi * (Adimlar[1].DisKamProfili.X - MinX);
    Y1 := Katsayi * (MaxY - Adimlar[1].DisKamProfili.Y);
```

EK-2 Devamı

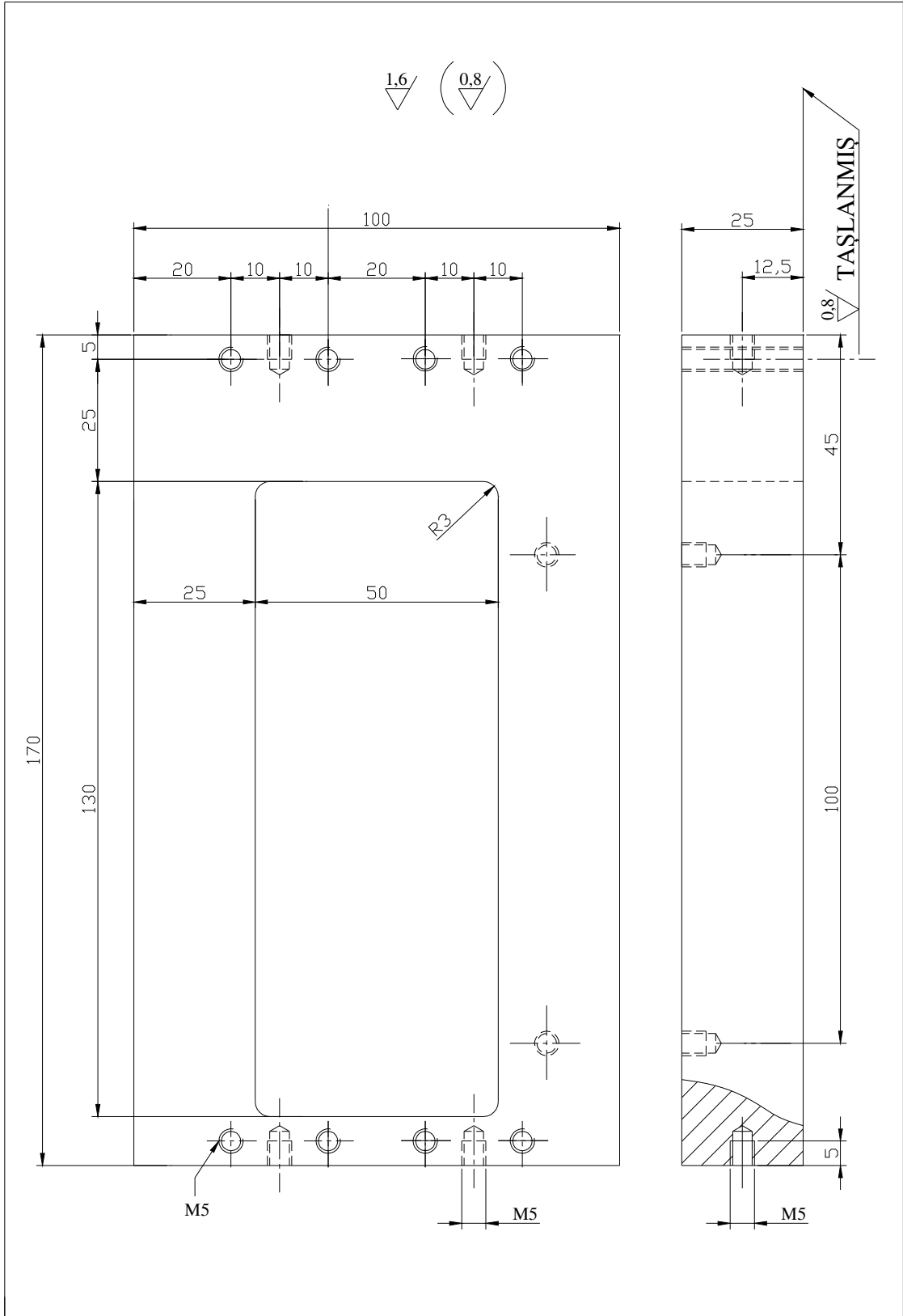
```
MoveTo(round(X1),round(Y1));
for i := 2 to ToplamAdimSayisi do begin
  X := Katsayi * (Adimlar[i].DisKamProfili.X - MinX);
  Y := Katsayi * (MaxY - Adimlar[i].DisKamProfili.Y);
  LineTo(round(X), round(Y));
end;
LineTo(round(X1), round(Y1));
//
// Dönme merkezini işaretle
//
Brush.Color := clBlack;
Brush.Style := bsSolid;
i := round(Katsayi * (DonmeMerkezi.X - MinX));
j := round(Katsayi * (MaxY - DonmeMerkezi.Y));
FillRect(rect(i-1,j-1,i+1,j+1));
Brush.Color := clWhite;
Brush.Style := bsClear;
//
// Varsa eksantirisite çemberini çiz
//
if Eksantirisite > 0 then begin
  Pen.Color := clBlack;
  Pen.Style := psSolid;
  Ellipse( round(Katsayi * (DonmeMerkezi.X - MinX - Eksantirisite)),
           round(Katsayi * (MaxY - DonmeMerkezi.Y - Eksantirisite)),
           round(Katsayi * (DonmeMerkezi.X - MinX + Eksantirisite)),
           round(Katsayi * (MaxY - DonmeMerkezi.Y + Eksantirisite))
         );
end;
//
// Fazların bitişlerini işaretle
//
Pen.Color := clBlack;
Pen.Style := psDot;
if Eksantirisite = 0 then begin
  X1 := Katsayi * (DonmeMerkezi.X - MinX);
  Y1 := Katsayi * (MaxY - DonmeMerkezi.Y);
  for i := 1 to FazAdet do begin
    j := Fazlar[i].BitisAcisi * AraliktakiAdimSayisi;
    X := Katsayi * (Adimlar[j].DisKamProfili.X - MinX);
    Y := Katsayi * (MaxY - Adimlar[j].DisKamProfili.Y);
    MoveTo(round(X),round(Y));
    LineTo(round(X1), round(Y1));
  end;
end
else begin
  for i := 1 to FazAdet do begin
    j := Fazlar[i].BitisAcisi * AraliktakiAdimSayisi;
    Acı := ArcCos(Eksantirisite / Adimlar[j].DisKamProfili.DonmeMerkezineUzaklik);
    if EksantirisiteSagda then begin
      X := DonmeMerkezi.X + (Eksantirisite * Cos(Adimlar[j].DisKamProfili.DondurulmusAcı + Acı));
      Y := DonmeMerkezi.Y + (Eksantirisite * Sin(Adimlar[j].DisKamProfili.DondurulmusAcı + Acı));
    end
    else begin
      X := DonmeMerkezi.X + (Eksantirisite * Cos(Adimlar[j].DisKamProfili.DondurulmusAcı - Acı));
      Y := DonmeMerkezi.Y + (Eksantirisite * Sin(Adimlar[j].DisKamProfili.DondurulmusAcı - Acı));
    end;
    end;
    X := Katsayi * (X - MinX);
    Y := Katsayi * (MaxY - Y);
    X1 := Katsayi * (Adimlar[j].DisKamProfili.X - MinX);
    Y1 := Katsayi * (MaxY - Adimlar[j].DisKamProfili.Y);
    MoveTo(round(X),round(Y));
    LineTo(round(X1), round(Y1));
  end;
end;
//
// Proje adını ve tarih-saati yazdır
//
Font.Assign(Form1.Font);
```

EK-2 Devamı

```
TextOut(100,Printer.PageHeight - (TextHeight('x') * 5),TasarimBasligi);
TextOut(100,Printer.PageHeight - (TextHeight('x') * 4),'Tarih: '+DateToStr(Date));
TextOut(trunc(Printer.PageWidth * 0.8),Printer.PageHeight - (TextHeight('x') * 4),'Saat: '+TimeToStr(Time));
end;
Printer.EndDoc;
MessageDlg('Bilgiler yazdırıldı',mtInformation,[mbOk],0);
end;

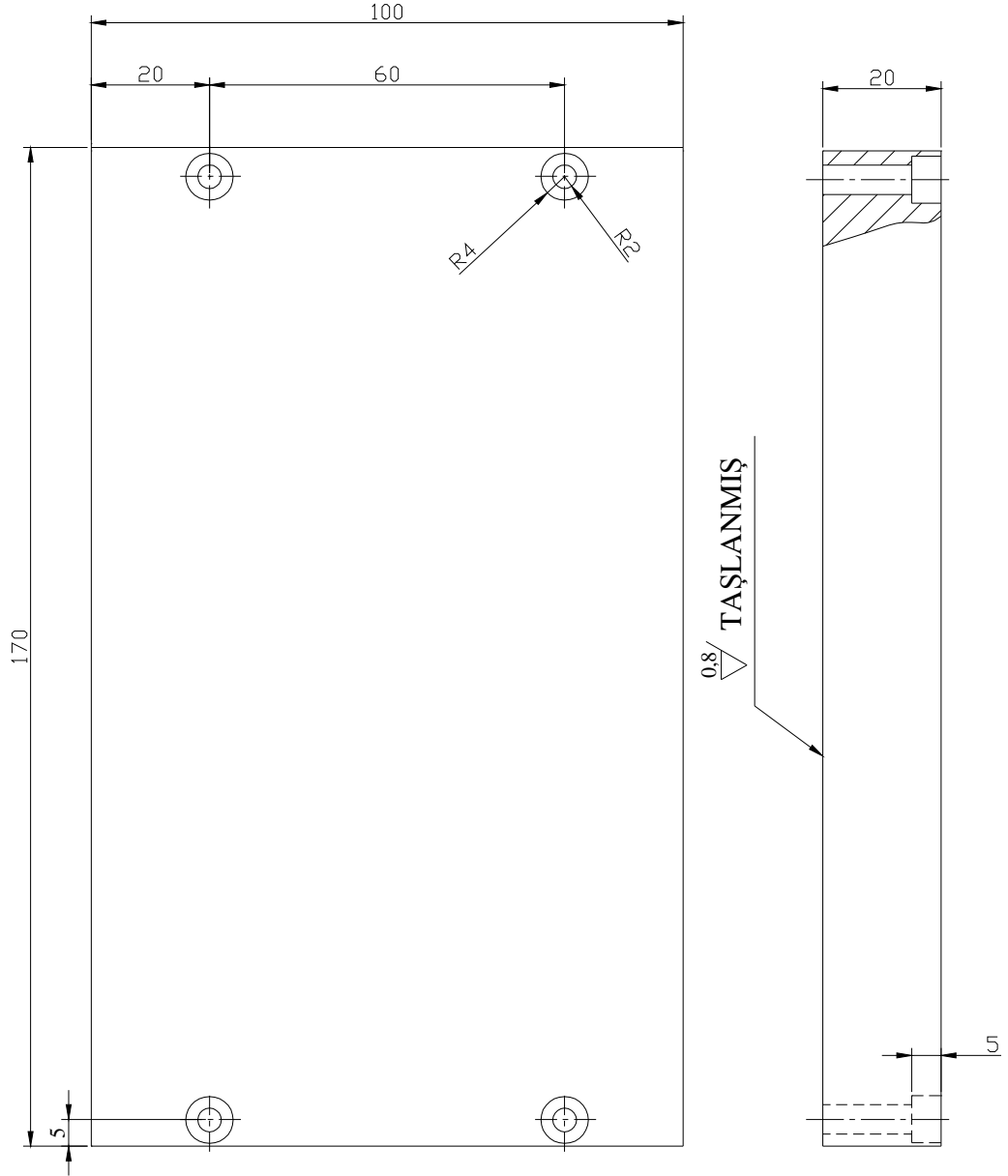
end.
```


EK3- Parçaların Yapım Resimleri Ve Mekanizmanın Montaj Resmi



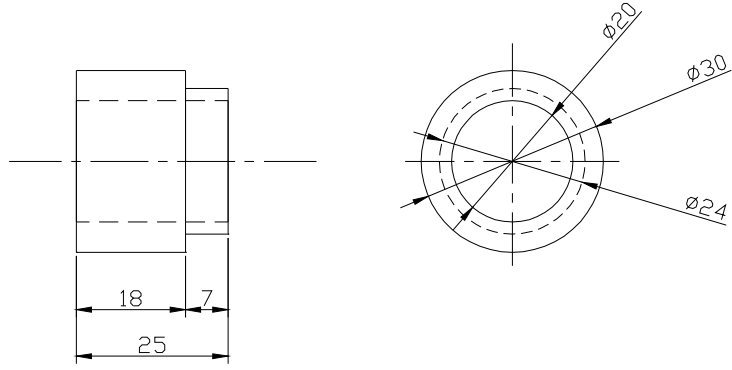
Ç 1040	1:1	1	MEKANİZMA ALT PARÇA	Murat KOYUNBAKAN	KM_1
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

1.6/ (0.8/)



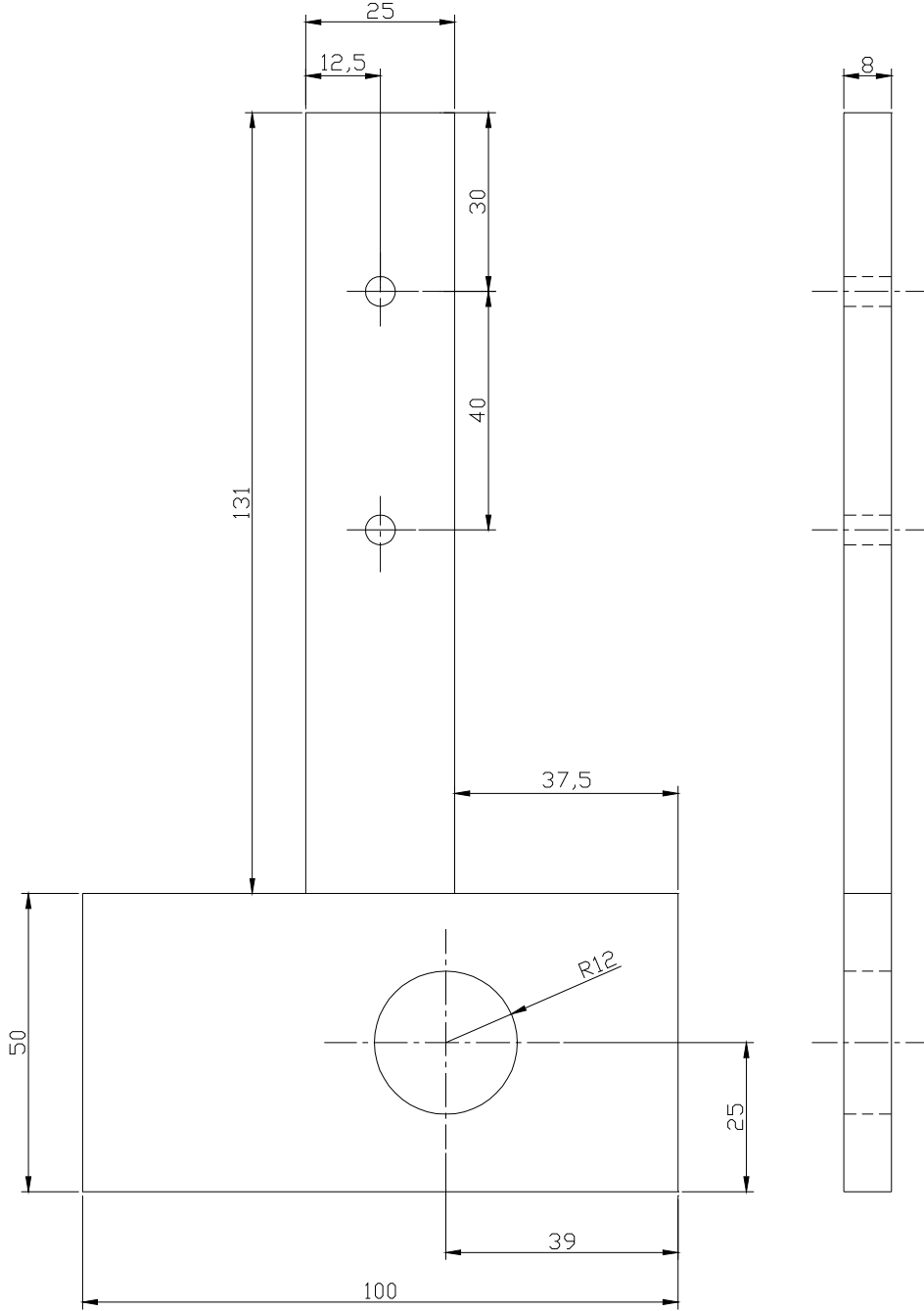
Ç 1040	1:1	1	MEKANİZMA ÜST PARÇA	Murat KOYUNBAKAN	KM_2
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

1,6/



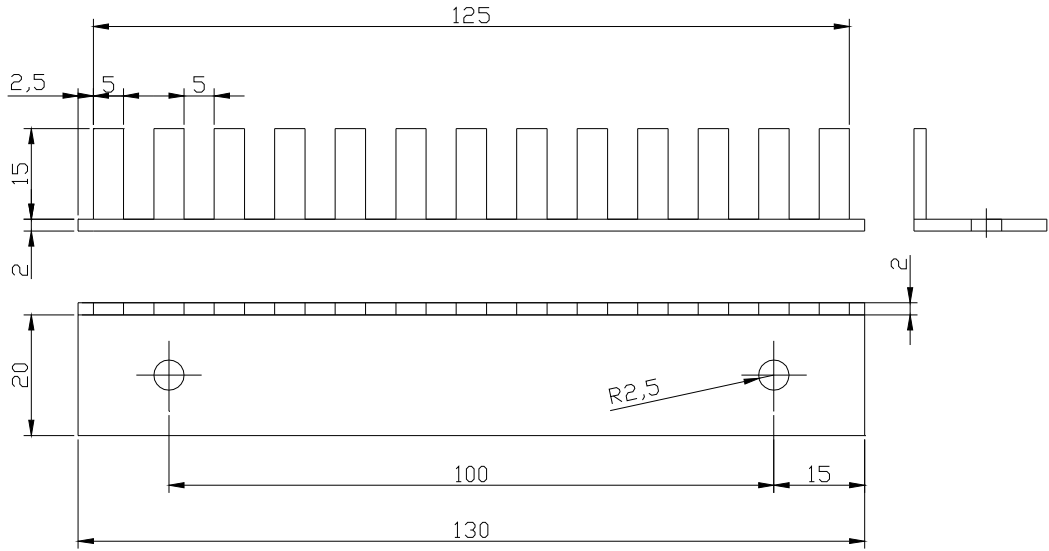
Pirinç	1:1	2	MİL YATAĞI	Murat KOYUNBAKAN	KM_3
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

1,6/
▽

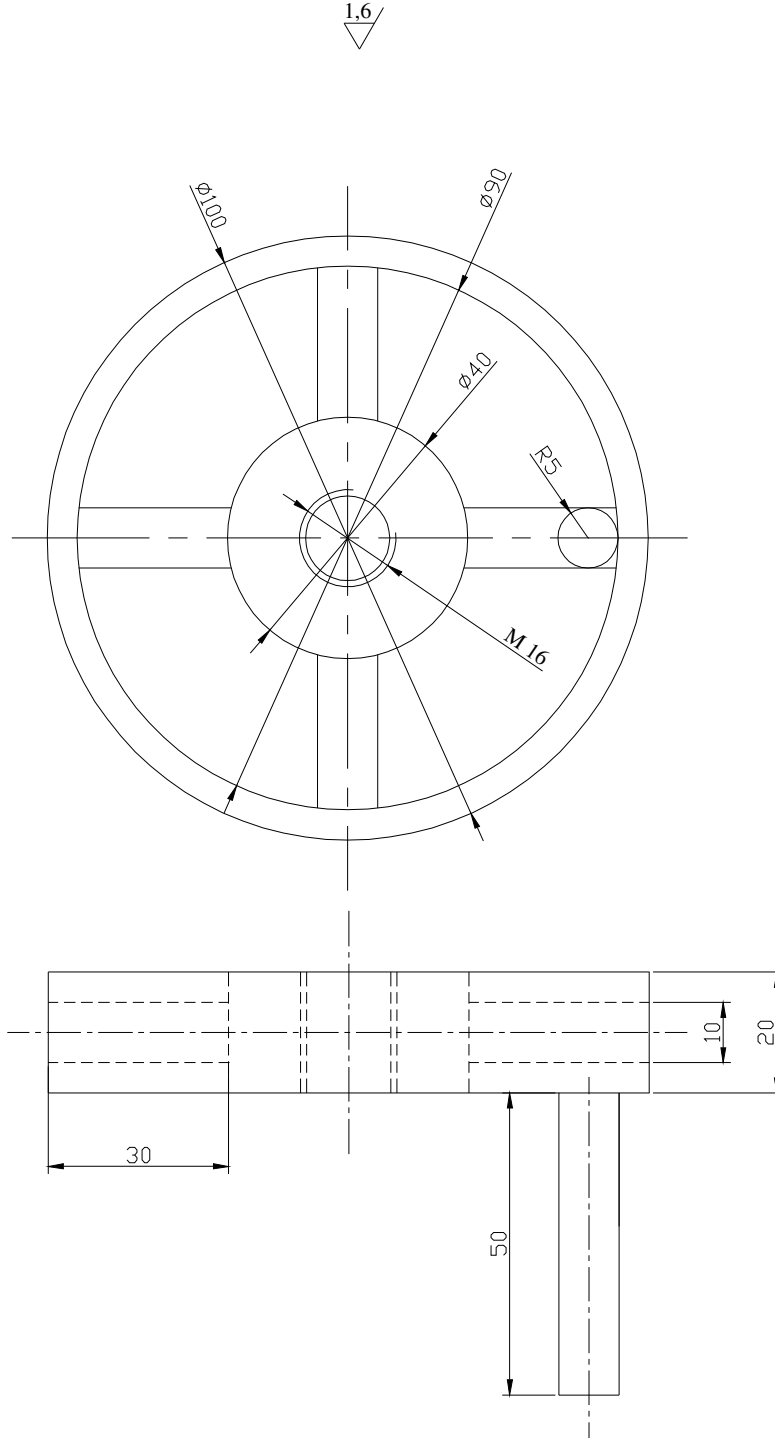


Ç 1040	1:1	2	MİL YAN YATAKLAMA PARÇASI	Murat KOYUNBAKAN	KM_4
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

1.6/



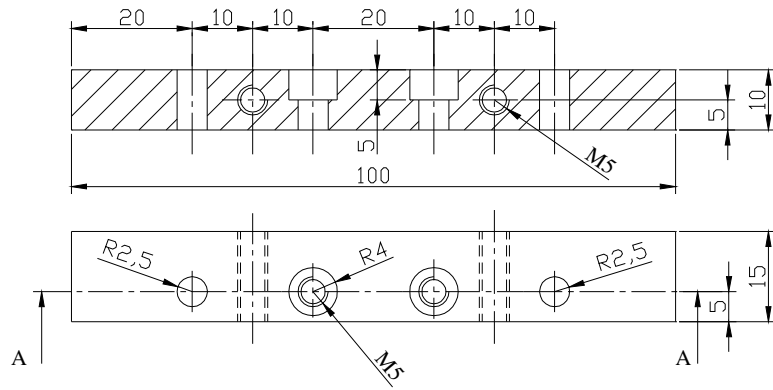
Ç 1040	1:1	1	YAY TUTUCU	Murat KOYUNBAKAN	KM_5
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr



Ç 1040	1:1	1	ÇEVİRME KOLU	Murat KOYUNBAKAN	KM_6
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

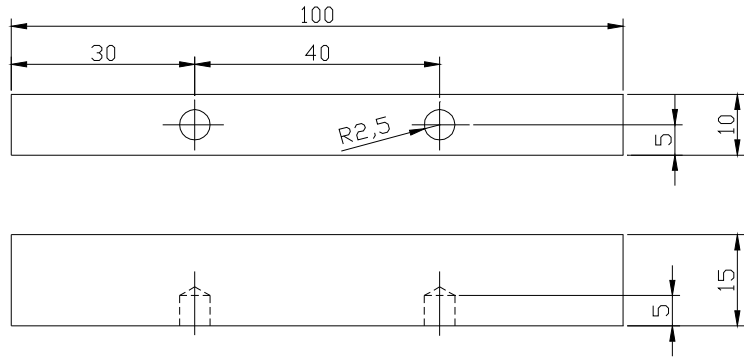
1,6/

A-A KESİTİ



Ç 1040	1:1	1	SOL YAN DESTEK PARÇA	Murat KOYUNBAKAN	KM_7
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

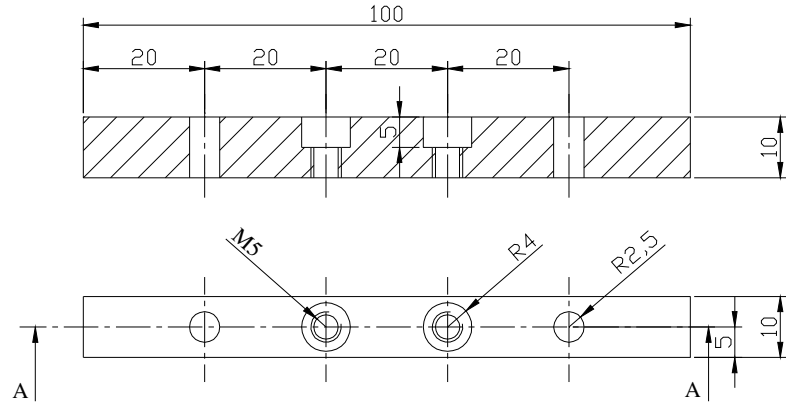
1.6/



Ç 1040	1:1	1	KALEM HİZALAMA PARÇASI	Murat KOYUNBAKAN	KM_8
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

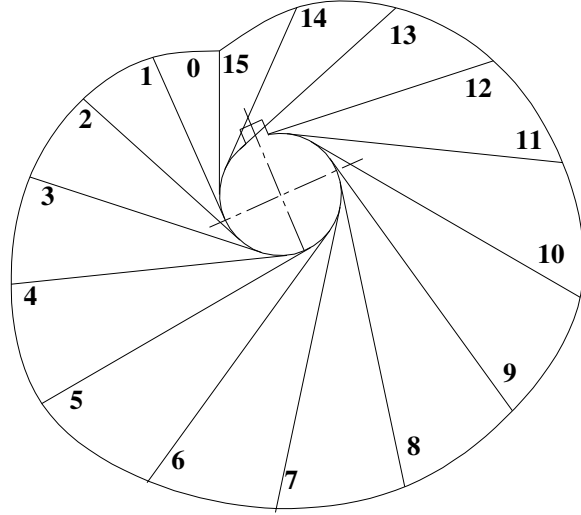
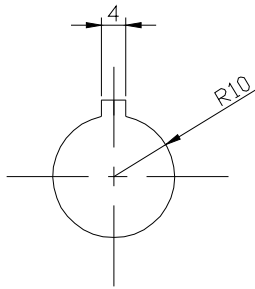
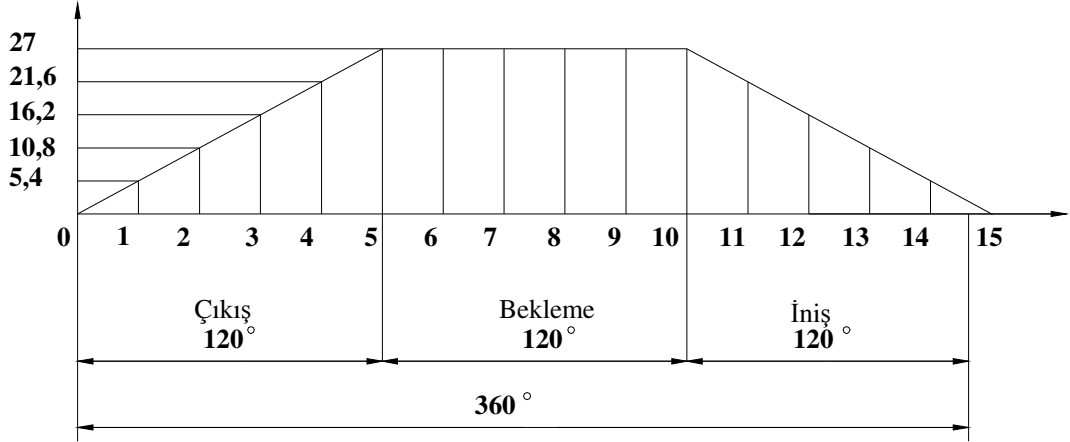
1,6

A-A KESİTİ

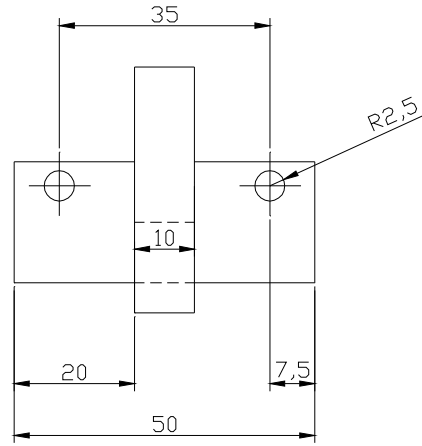
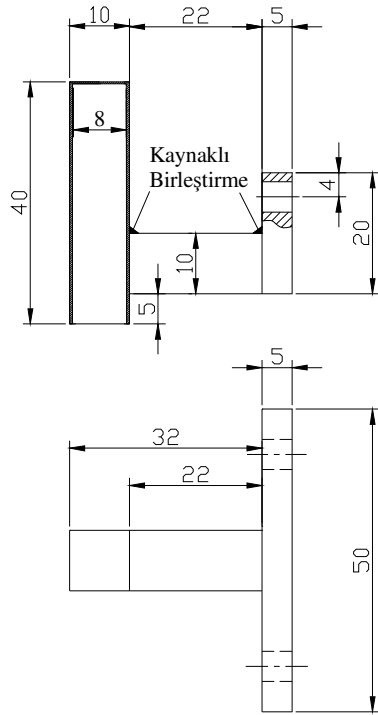


Ç 1040	1:1	1	SAĞ YAN DESTEK PARÇA	Murat KOYUNBAKAN	KM_9
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

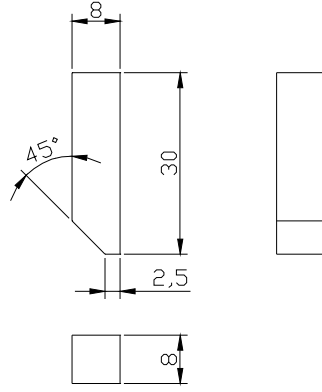
1,6/



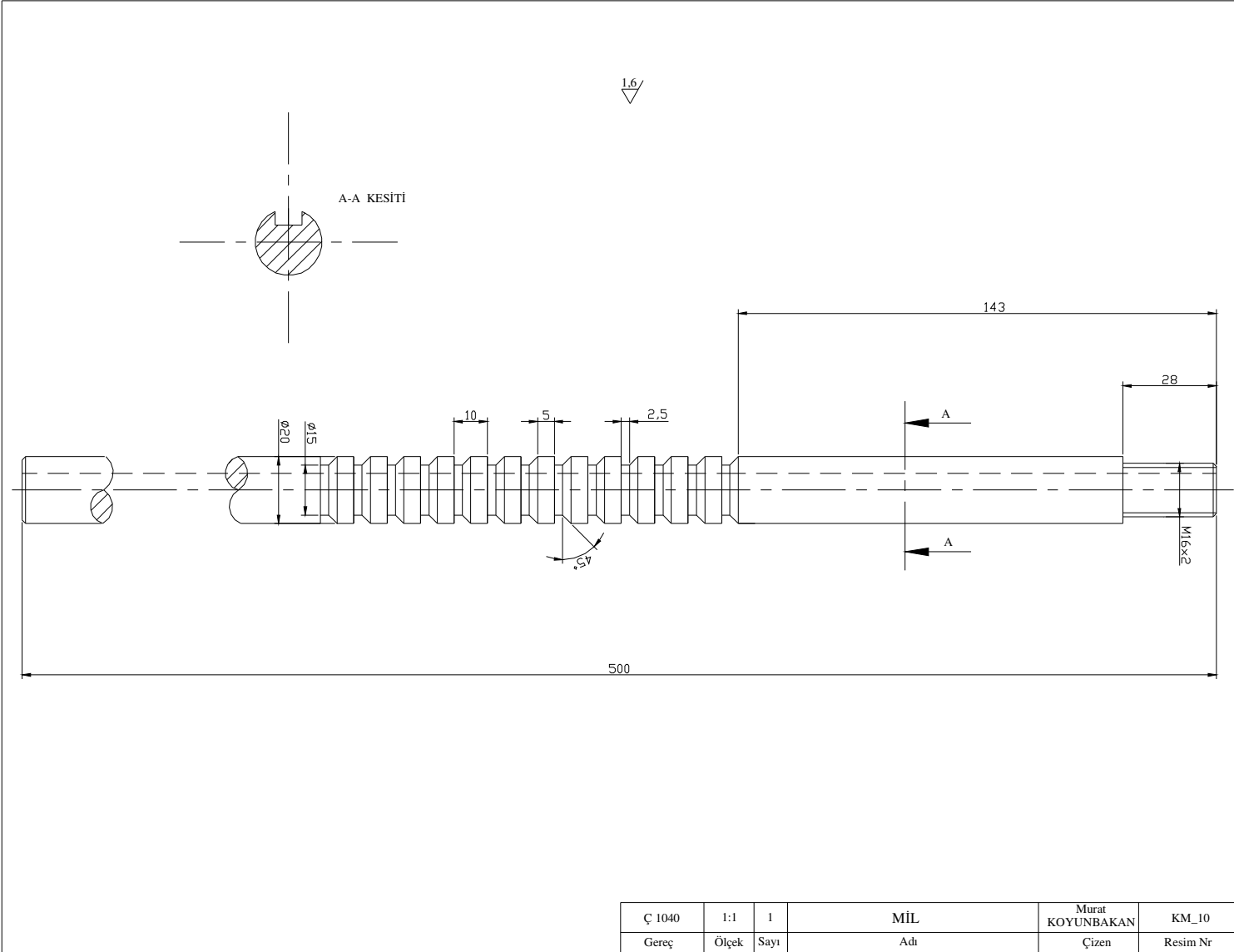
Ç 1040	1:1	1	Sabit Hızlı Kam	Murat KOYUNBAKAN	KM_11
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr



Ç 1040	1:1	1	Adım Ayar Parçası	Murat KOYUNBAKAN	KM_16
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr



Ç 1040	1:1	1	Adım İlerletme Parçası	Murat KOYUNBAKAN	KM_17
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr



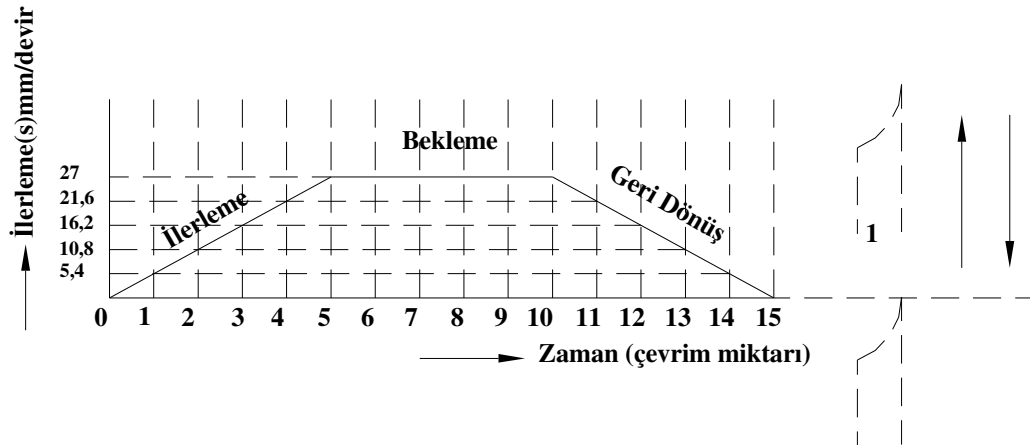
Ç 1040	1:1	1	MİL	Murat KOYUNBAKAN	KM_10
Gereç	Ölçek	Sayı	Adı	Çizen	Resim Nr

EK4. Geleneksel Yöntemlerle Ve Tasarım Programıyla Tasarlanan Kamın Karşılaştırılması

Aşağıdaki çizelgede tasarımı yapılan kamın giriş parametreleri verilmiştir.

Faz Bilgileri				
1. Faz	Hareket Tipi Çıkış	Toplan İlerleme 27 mm	Toplam Dönme 120°	Hareket (Tipi) Denklemi Sabit Hızlı Hareket
2. Faz	Hareket Tipi Bekleme	Toplan İlerleme 0 mm	Toplam Dönme 120°	Hareket (Tipi) Denklemi Sabit Hızlı Hareket
3. Faz	Hareket Tipi İniş	Toplan İlerleme 27 mm	Toplam Dönme 120°	Hareket (Tipi) Denklemi Sabit Hızlı Hareket
Giriş Parametreleri				
Devir Sayısı		1 dev/dk		
Basınç Açısı		45°		
Eksantrisite		10 mm		
Toparlık Yarıçapı		0		

Yukarıdaki parametrelere göre geleneksel yöntemlere göre çizilmiş kama ait yol grafiği aşağıdaki gibidir.



EK4-Devamı

Giriş parametreleri tasarım parogramına girildiğinde ve program çalıştığındaki ekran görüntüsü ve yol grafiği ise aşağıdaki gibidir.

Kam Tasarımı

Lütfen parametreleri girip İŞLEMİ BAŞLAT butonunu tıklayınız

Parametreler | Yol grafiği | Hız grafiği | İvme grafiği | Kam profili

Proje adı: Örnek kam tasarımı

Faz sayısı: 3

Dönüş hızı: 1 devir/dak

Dönüş yönü
 Saat yönünde
 Saatin tersi yönde

Eksantrisite: 10 mm

Eksantrisite yönü
 Sağda
 Solda

Başıncı açısı: 45 derece

Toprak yarıçapı: 0 mm

Hesaplanan değerleri de göster

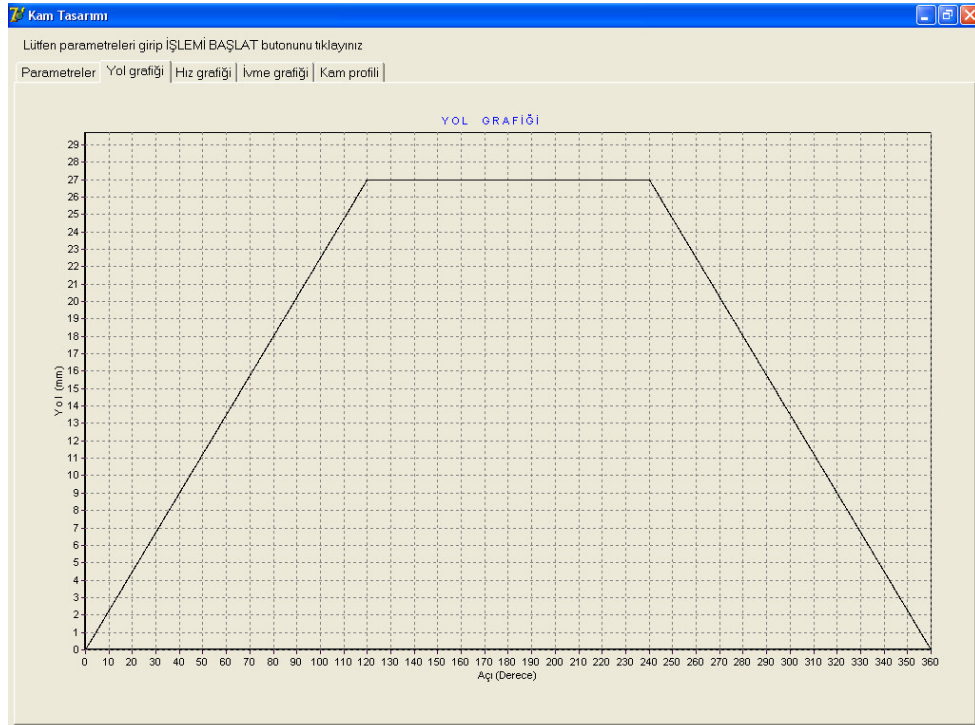
İşlemi Başlat

Parametreleri Dosyaya Sakla

Parametreleri Dosyadan Yükle

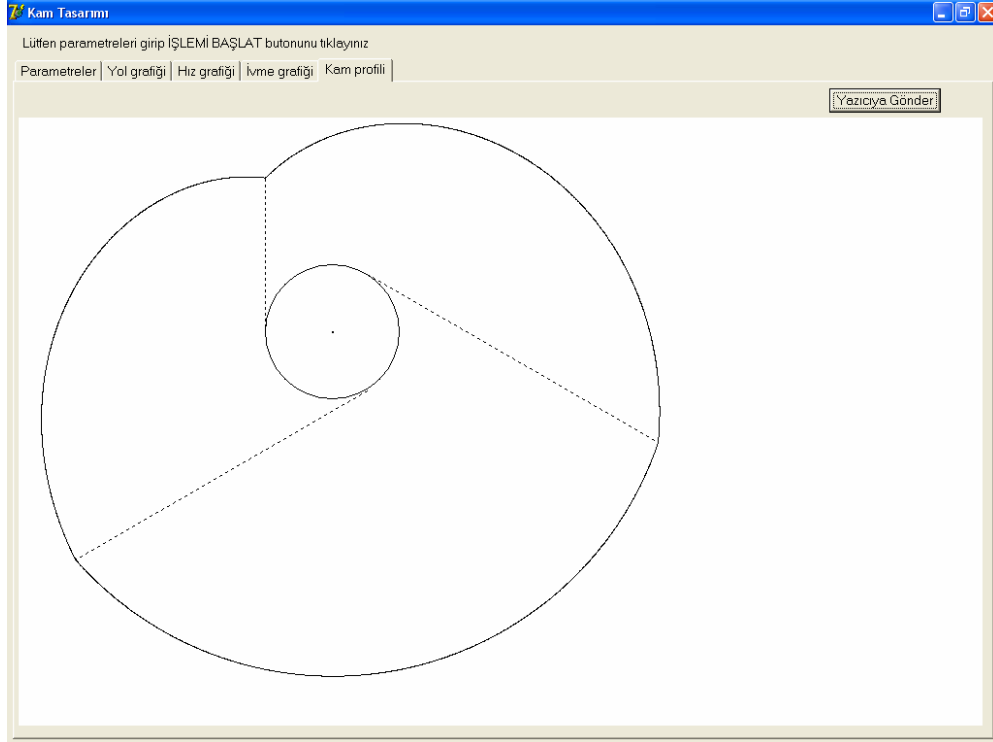
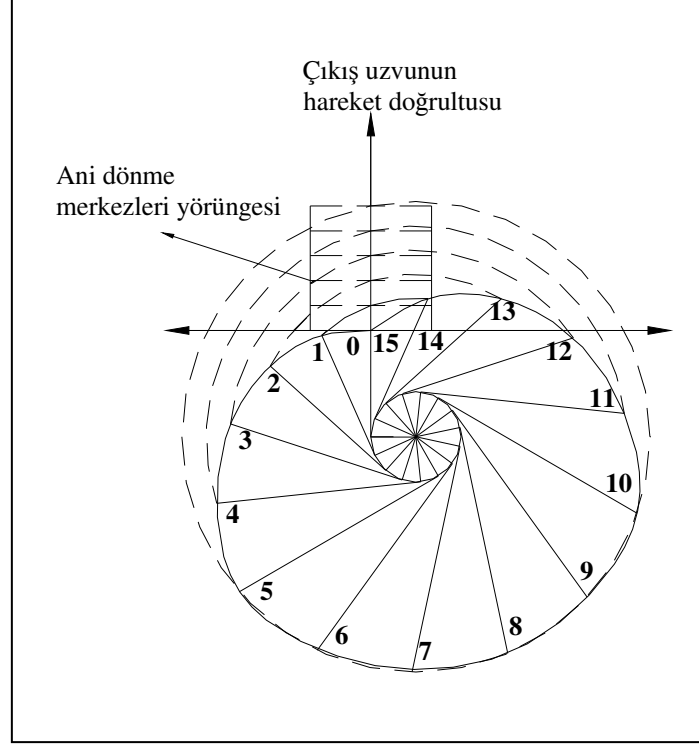
Faz bilgilerinin girilmesi için ilgili faz satırını tıklayınız:

Faz no	Faz tipi	Yükseklik	Açı	Hareket tipi
1	Çıkış	27	120	Sabit hızlı hareket
2	Bekleme	27	240	Sabit hızlı hareket
3	İniş	0	360	Sabit hızlı hareket

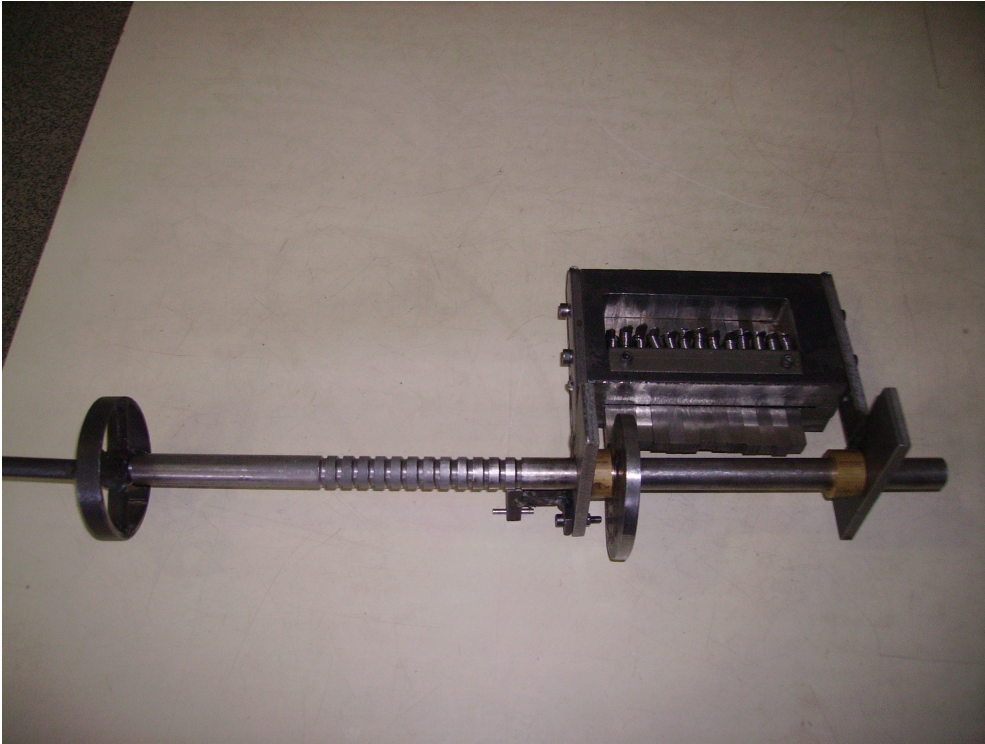
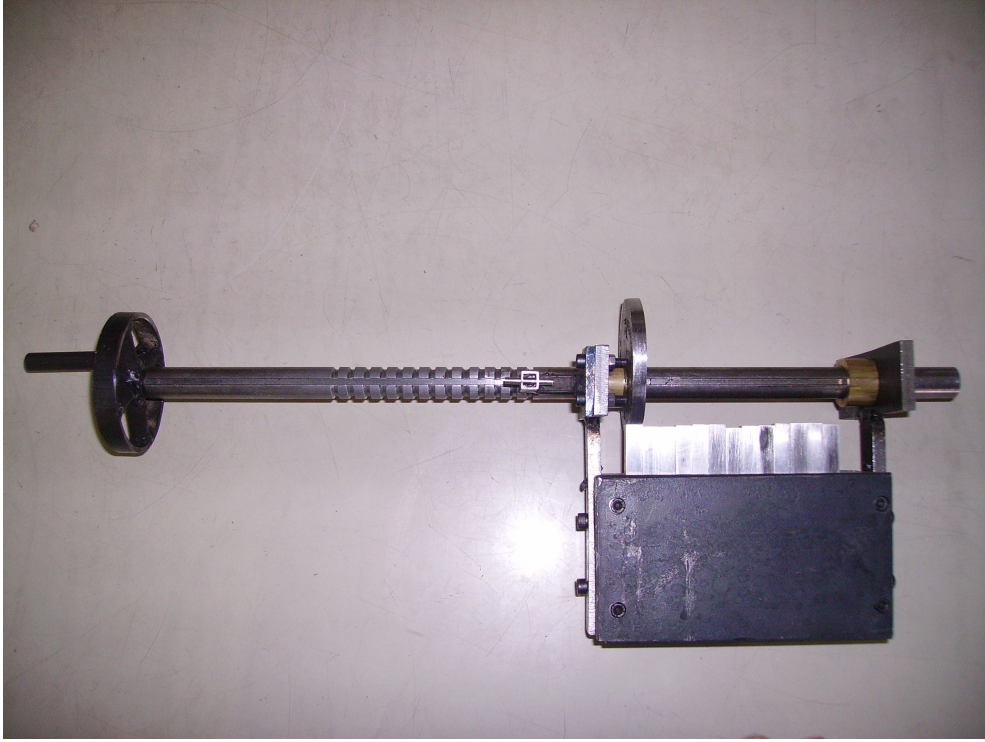


EK4-Devamı

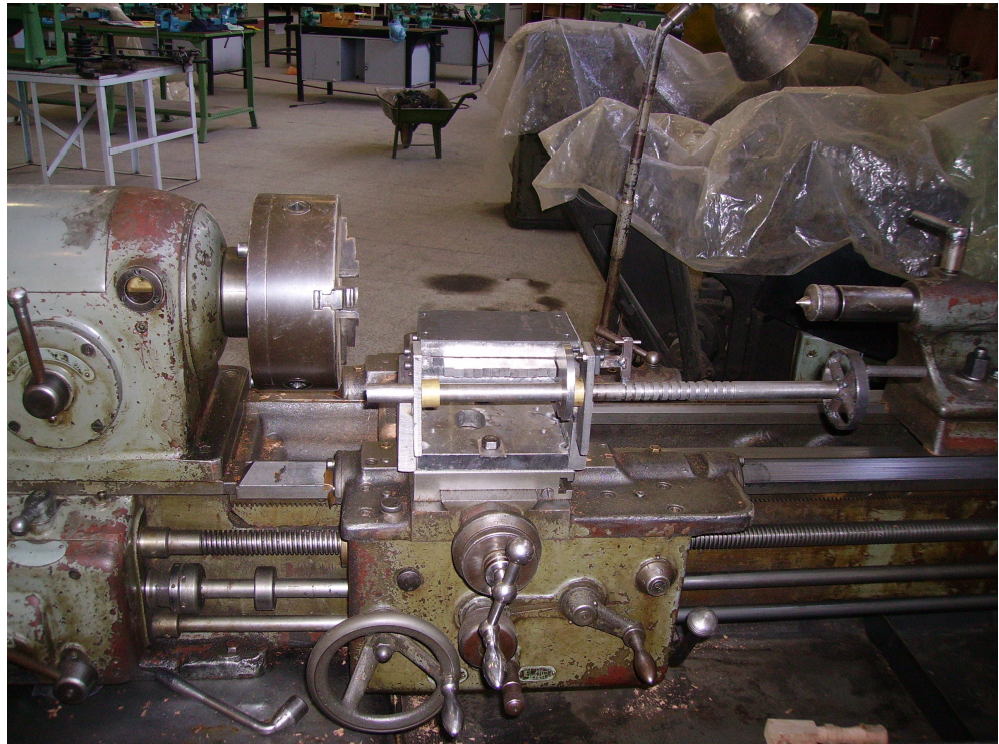
Geleneksel yöntemlerle ve tasarım programıyla tasarlanan kama ait profil görüntüleri aşağıda verilmiştir.



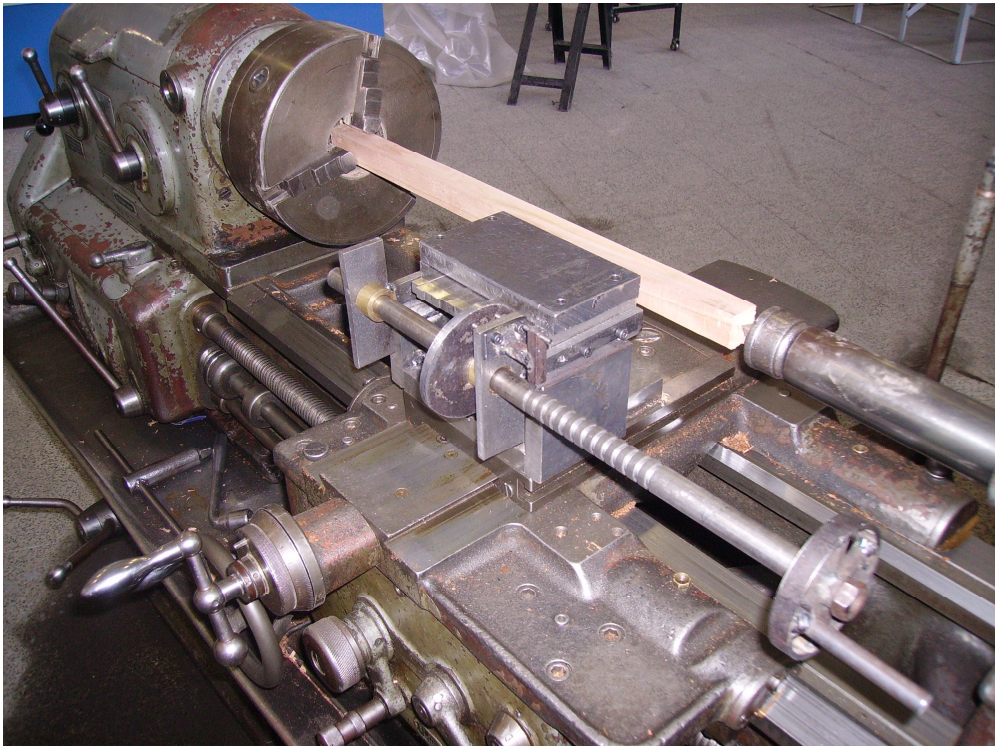
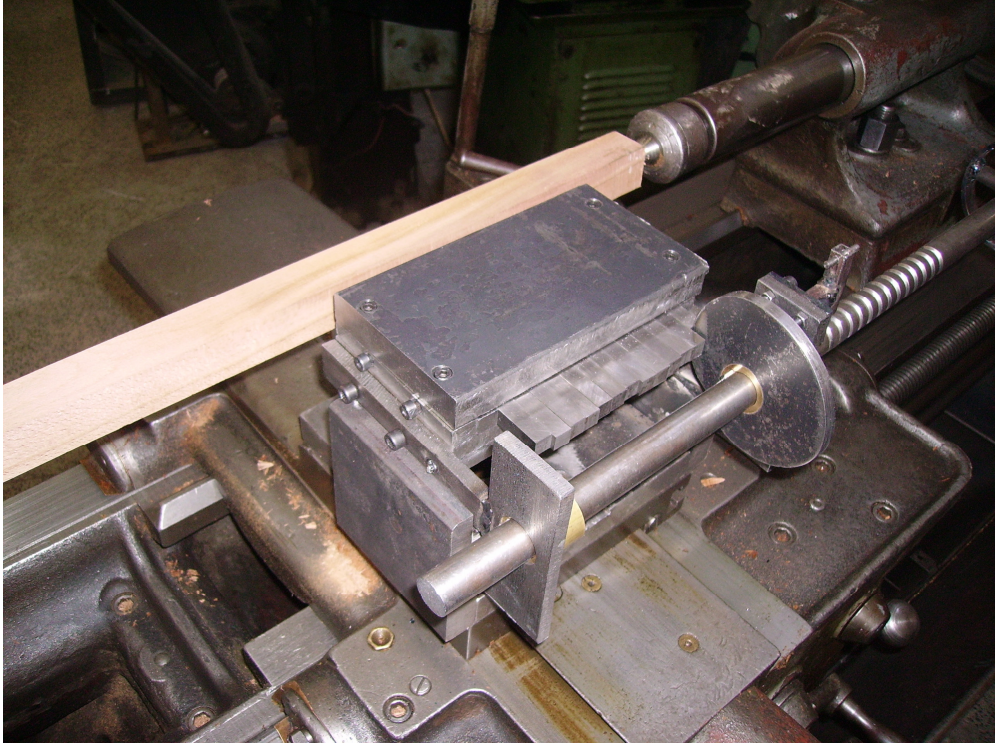
EK5- Örnek Kam Mekanizmasına Ait Resimler



EK-5 Devami



EK-5 Devamı



EK5- Devamı

