YATAY EKSENLİ RÜZGÂR TÜRBİNİ KANADININ BİLGİSAYAR DESTEKLİ TASARIMI

Murat ÖNDER

YÜKSEK LİSANS TEZİ
ENDÜSTRİYEL TEKNOLOJİ EĞİTİMİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAYIS 2006
ANKARA

Murat ÖNDER tarafından hazırlanan "YATAY EKSENLİ RÜZGÂR TÜRBİNİ KANADININ BİLGİSAYAR DESTEKLİ TASARIMI" adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. H. Güçlü YAVUZCAN

Bu çalışma, jürimiz tarafından Endüstriyel Teknoloji Eğitimi Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Mahmut İZCİLER

Üye : Prof. Dr. H. Güçlü YAVUZCAN (Danışman)

Üye : Doç. Dr. Serdar YÜCESU

Üye :

Üye :

Tarih :12/06/2006

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kuralarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Murat ÖNDER

YATAY EKSENLİ RÜZGÂR TÜRBİNİ KANADININ BİLGİSAYAR DESTEKLİ TASARIMI

(Yüksek Lisans Tezi)

Murat ÖNDER

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Mayıs 2006

ÖZET

Bu çalışmada yatay eksenli rüzgâr türbinleri için istenilen koşullara bağlı olarak kanat tasarımı yapabilecek bir bilgisayar programı geliştirilmesi amaçlanmıştır. Bu amaçla, yatay eksenli rüzgâr türbinlerinin aerodinamiği ile ilgili teorik yaklaşımlar ve kanat elemanı momentum teorisine göre kanat tasarımı sistematik olarak incelenmiştir. Sonuç olarak, yatay eksenli rüzgâr türbin kanat tasarımı için Microsoft Visual C#. NET 2.0 tabanlı olarak KANAT SİMÜLATÖRÜ adlı bir program geliştirilmiştir. Program, kullanıcı bölge ve kanat uzunluğunu girdiği zaman tasarım için gerekli ölçüleri (kanat kiriş ölçüleri, tahmini güç, kanat açısı) verecek şekilde düzenlenmiştir. Program belirli bir bölgedeki rüzgâr verilerine göre yatay eksenli rüzgar türbini kanat tasarımı yapmak isteyenlere fikir verebilecek ve kolaylık sağlayabilecek düzeydedir.

Bilim Kodu : **705.1.071**
Anahtar Kelimeler : **Rüzgar Türbini, Rüzgar Enerjisi, Kanat Tasarımı**
Sayfa Adedi : **98**
Tez Yöneticisi : **Prof. Dr. H. Güçlü YAVUZCAN**

COMPUTER-AIDED DESİGN OF HORIZONTAL AXIS WIND TURBINE BLADE

(M.Sc. Thesis)

Murat ÖNDER

GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

May 2006

ABSTRACT

This study aims to develop software that will be able to perform blade designs depending on the desired conditions for horizontal axis wind turbines. To serve this purpose, theoretical approaches pertaining to aerodynamics of horizontal axis wind turbines and blade design in terms of blade element momentum theory were examined systematically. As a result, software called a wing simulator based Microsoft Visual C#. NET 2.0 was developed for the horizontal axis wind turbines. The software was designed in a manner that will enable to give out required measurements (blade chord measurements, estimated power, blade angle) for the time design when the user inputs local features and blade length. The program is at a level that will give some insight for those willing to perform blade design horizontal axis wind turbines according to the wind data of an area.

Science Code : **705.1.071**
Key Words :**Wind Turbine, Wind Energy, Blade Design**
Page Number : **98**
Adviser : **Prof. Dr. H. Güçlü YAVUZCAN**

TEŞEKKÜR

Bu çalışma sırasında değerli katkılarını gördüğüm Danışmanım Sayın Prof. Dr. H. Güçlü YAVUZCAN'a ve her zaman bana destek olan aileme şükranlarımı sunarım.

İÇİNDEKİLER

Sayfa

ÖZET.....iv

ABSTRACT.....v

TEŞEKKÜR.....vi

İÇİNDEKİLER.....vii

ÇİZELGELERİN LİSTESİ.....x

ŞEKİLLERİN LİSTESİ.....xi

RESİMLERİN LİSTESİ.....xii

HARİTALARIN LİSTESİ.....xiii

SİMGELER VE KISALTMALAR.....xiv

1. GİRİŞ.....1

2. RÜZGAR ENERJİSİ.....2

2.1. Giriş.....2

2.2. Rüzgar Enerjisinin Olumlu Yönleri.....2

2.3. Rüzgar Enerjisinin Maliyeti.....3

2.4. Rüzgar Enerjisinin Maliyetini Etkileyen Faktörler.....4

2.4.1. Rüzgar hızı ve rüzgar tarlasının büyüklüğü.....4

2.4.2. Çevresel maliyetler.....4

2.4.3. Faiz oranı.....5

2.5. Dünya Rüzgar Enerjisi Potansiyeli.....5

2.6. Türkiye'deki Rüzgar Enerji Potansiyeli.....6

3. RÜZGAR TÜRBİNLERİ.....11

3.1. Yatay Eksenli Rüzgar Türbinleri.....11

Sayfa

3.1.1. Önden rüzgar alan türbinler.....12

3.1.2. Arkadan rüzgar alan türbinler.....13

3.2. Düşey Eksenli Rüzgar Türbini.....13

3.3. Rüzgar Türbinlerinin Yapısal Özellikleri.....13

4. KANAT TASARIMINDAKİ AERODİNAMİK ESASLAR.....15

4.1. Sürükleme ve Kaldırma Kuvveti.....15

4.2. Geliştirilmiş olan Modeller.....16

4.3. İdeal Disk Teorisi ve Betz Limiti.....16

4.4. Genel Momentum Teorisi.....21

4.5. Kanat Elemanı Teorisi.....30

4.6. Kanat Elemanı Momentum Teorisi.....34

5. KANAT TASARIMI.....38

5.1. Genel.....38

5.2. Uç Kayıpları.....38

5.3. Kanat Seçimi.....39

5.4. Kanada İlişkin Hesaplamalar.....41

6. KANAT TASARIMI GELİŞTİRİLEN BİLGİSAYAR PROGRAMI.....43

6.1. Genel.....43

6.2. Kanat Simülasyonu.....44

6.3. Kanat Simülatörü Programıyla Kanat Tasarımı.....47

7. SONUÇ VE ÖNERİLER.....52

KAYNAKLAR.....54

EKLER.....56

Sayfa

EK-1 Kanat simülatörü programının kodları.....57

ÖZGEÇMİŞ.....98

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1 Elektrik üretim maliyetleri ……...…3

Çizelge 2.2 Dünyadaki mevcut rüzgar kurulu gücünün ülkelere göre dağılımı….. …5

Çizelge 2.3. Türkiye'deki kurulu gücün yıllar itibari ile gelişimi……6

Çizelge 2.4 Türkiye'nin coğrafi bölgelerindeki ortalama rüzgar hızları .....…6

Çizelge 2.5 Türkiye Rüzgar Atlasında önemli görülen bazı bölgelere ilişkin veriler……8

Çizelge 2.6 Türkiye'deki rüzgar santrali projeleri 9

Çizelge 3.1. Bazı yatay ve düşey eksenli rüzgar türbin tipleri ve kullanım yerleri ve amaçları……..…….……13

Çizelge 6.1. Programın veri tabanında kullanılmak üzere seçilen kanat modelleri ve bu modele ilişkin hesaplanan Reynolds sayısı değerleri.....49

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 4.1. Kanada etki eden kaldırma ve sürükleme kuvveti basit gösterimi…….15

Şekil 4.2. Akım tüpündeki hareketli diskten geçen idealleştirilmiş akış……....17

Şekil 4.3. Akım tüpü boyunca hız ve basınç dağılımı……...……20

Şekil 4.4. Rotordan geçen akışın akım tüpündeki geometrisi…….22

Şekil 4.5. Uç hız oranına göre güç katsayısının gösterimi……..30

Şekil 4.6. Kanat elemanlarının şematik gösterimi……..31

Şekil 4.7. Kanat kirişi…….…….31

Şekil 5.1. Kanat açıları……41

Şekil 6.1 Kanat simülatörü programın akış şeması……...……..45

Şekil 6.2. Programın ara yüzü…….46

Şekil 6.3. Programın girdileri…….….47

Şekil 6.4. Önerilen kanat tipleri…….....48

Şekil 6.5. Seçilen kanat tipleri……..…..48

Şekil 6.6. Programın çıktı görüntüsü……...…...50

Şekil 6.7 Grafik çıktısı……....……51

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 3.1. (a) Önden rüzgar alan yatay eksenli türbin (b) Arkadan rüzgâr alan yatay eksenli türbin……..……12

HARİTALARIN LİSTESİ

Harita **Sayfa**

Harita 2.1. Türkiye rüzgar atlası……..……...…..7

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış olan bazı simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
$\dot{m}$	Rotor düzleminden geçen havanın debisi, m^3/s
ρ	Hava yoğunluğu
Ω	Rüzgâr türbindeki rotorun açısal hızı, m/s
φ	Görünür rüzgâr açısı
α	Hücum açısı
α	Kanadın hücum açısı
θ	Kanat açısı
λ	Uç hız oranı
$\varphi_{opt,i}$	i. kanat elemanı için görünür rüzgar açısı
λ_r	Yerel uç hız oranı
$\lambda_{r,i}$	i. kanat elemanı için uç hız oranı
a	Rotor düzlemindeki eksenel akış faktörü
A	Rüzgâr türbinindeki rotorun alanı, m^2
B	Rotordaki kanat sayısı
c	Kanat kiriş uzunluğu, m
C_d	Kanadın sürükleme katsayısı
c_i	i. kanat elemanın kiriş uzunluğu, m
C_L	Kanadın kaldırma katsayısı
C_p	Rüzgâr türbini rotorunun güç katsayısı
F	Uç kayıp faktörü
F_i	i. kanat elemanı için kayıp faktörü
H_1	Rotor gerisi ile türbin gerisindeki alan arasındaki Bernoulli sabiti

Simgeler	Açıklama
H_o	Serbest rüzgâr ile rotorun önü arasındaki Bernoulli sabiti
N	Jeneratör verimi
N_e	türbinin üreteceği tahmini güç, W
P	rüzgâr türbini rotorundan çıkan güç, W
p'	rotor düzlemindeki basınç düşmesi, Pa
p_d	rotor gerisindeki havanın basıncı, Pa
p_o	türbin önündeki havanın basıncı, Pa
p_u	rotor önündeki havanın basıncı, Pa
Q	rotorun döndürme momenti, Nm
r	rotordaki radyal mesafe, m
R	Rüzgâr türbini rotorunun yarıçapı, m
r_i	kanat yarıçapındaki i. kanat elemanı, m
T	rotorun itme kuvveti, N
u	rotor düzlemindeki eksenel rüzgar hızı, m/s
U_∞	serbest akan rüzgârın hızı, m/s
U_R	rotor düzlemindeki rüzgâr hızı, m/s
u_w	türbin gerisindeki eksenel rüzgâr hızı, m/s
U_w	rotor gerisinde oluşan rüzgâr hızı, m/s
v	rotor düzlemindeki hız bileşeni, m/s
V_a	Görünür rüzgâr hzı, m/s
v_w	türbin gerisindeki radyal rüzgâr hızı, m/s
w	rotor düzlemindeki açısal hız, rad/s
w_w	türbin gerisindeki açısal rüzgâr hızı, rad/s
Re	Reynolds sayısı
δ	Katılık katsayısı
U_{rel}	bağıl rüzgâr hızı, m/s
b	rotor gerisindeki eksenel akış faktörü
γ	Süzülme oranı

1. GİRİŞ

Günümüzde ülkemizin içinde bulunduğu enerji darboğazı giderek artmaktadır. Enerji kaynakları bakımından yeterli çeşitliliğe sahip olan ülkemizde bu kaynakların kullanımı yeterli değildir. Bu nedenle ülkemiz enerji ithalat eden konumundadır. 2000 yılında Türkiye'nin ihtiyaç duyduğu elektrik enerjisi 128 500 GWh iken bu miktarın 3 800 GWh'ni yurtdışından ithal etmiştir. Ülkemizin gelişmekte olan sanayi ye bağlı olarak yıllık enerji talebi % 8 artış göstermektedir [1]. Bundan dolayı dışa bağımlı olmayan enerji kaynaklarına doğru arayışlar başlamıştır. Bunlardan birisi de rüzgâr teknolojisidir. Son dönemlerde rüzgâr enerjisi teknolojisi hızla ilerlemektedir. Rüzgârdan maksimum derecede enerji elde etmek için aerodinamik esaslara uygun tasarım yapılması gereklidir.

Bu çalışmada, aerodinamik esaslar ve kanat elemanı tasarım esasları ışığında istenilen bölge ya da rüzgâr koşullarına bağlı olarak yatay eksenli rüzgâr türbinleri için pratik ve görsellik taşıyan bir kanat tasarımı programı geliştirilmesi amaçlanmıştır. Söz konusu program ile konuya ilgi duyabilecek kullanıcılara yeterli bilgi sunup kolay hesaplama ve karşılaştırma yapabilecekleri bir alt yapı oluşturulması hedeflenmiştir.

2. RÜZGÂR ENERJİSİ

2.1. Giriş

Bilindiği üzere dünyadaki enerji kaynaklarının tamamı güneş tarafından oluşturulmaktadır. Rüzgâr; güneş enerjisinin yeryüzüne ulaşması sonucunda ortaya çıkan bir doğa kuvvetidir. Güneş tarafından dünyaya gönderilen enerjinin yaklaşık %1-2'si rüzgâr enerjisidir [2]. Karaların, denizlerin ve atmosferin kendine özgü özgül ısıları vardır. Bulundukları coğrafik konumlara göre ısıları değişmektedir. Yüzeyde oluşan sıcaklık farkı, basınç farklılıklarına bunun sonucu da rüzgâr oluşumuna neden olur. Yüksek basınçtan, düşük basınca hareket eden havaya rüzgâr denir. Günümüzde rüzgâr enerjisi teknolojisi hızla ilerlemektedir. Rüzgâr enerjisinden elektrik üretiminin maliyeti, geleneksel elektrik enerjisi üretim maliyetleri ile rekabet edebilir duruma gelmiştir.

2.2. Rüzgâr Enerjisinin Olumlu Yönleri

Rüzgâr enerjisi doğal, yenilenebilir, temiz bir enerji kaynağıdır. Rüzgâr enerjisinin bazı üstünlükleri aşağıda sıralanmıştır.

a) Enerji girdisi olarak kullanılan fosil yakıtlardan tasarruf sağlar.

b) Atmosferde bol ve serbest olarak bulunur.

c) Rüzgâr türbinlerinin üzerine kurulduğu alanların hayvancılıkta v.b. gibi kullanımı mümkündür.

d) Hammaddenin zamanla tükenmesi söz konusu değildir.

e) Rüzgâr santralleri için gerekli olan projeler basit ve rüzgâr türbinlerin bakımı kolaydır.

f) Rüzgâr teknolojisi giderek ucuzlamaktadır.

Fakat rüzgârın devamlı olmaması ve rüzgâr santralinin ilk yatırım maliyetinin nispeten yüksek olması rüzgâr enerjisinin olumsuz yönleri olarak göze çarpmaktadır.

2.3. Rüzgâr Enerjisinin Maliyeti

Rüzgâr enerjisinin maliyeti, teknolojideki gelişmelerle beraber azalmaktadır. Günümüzde rüzgâr santralleri kullanımda olan konvansiyonel enerji teknolojileri ile rekabet edebilecek duruma gelmiştir. Rüzgâr enerjisi santrallerinde önemli olan bir nokta ise santral kurulduktan sonra sadece bakım ve onarım masraflarının olmasıdır. Hammadde için herhangi bir ödeme söz konusu değildir. Gelişen teknoloji ile birlikte rüzgârdan elde enerjinin maliyeti düşmektedir [3]. Daha önce rüzgâr türbinlerinin yaklaşık 3000 $/kW olan yatırım maliyeti, günümüzde sistemlerin verimliliğinin artmasından dolayı 980 $/kW civarlarına düşmüştür [4]. Bunun sonucu elektriğin birim üretim maliyeti 20 cent/kWh'den 7 cent/kWh'ya inmiştir. Üretici firmalar bugün düşük hızlarda (<5m/s) ve yüksek hızlarda (>20m/s) bile elektrik üretimi yaparak maliyetleri indirdiklerini ifade etmektedirler [4].

Farklı enerji kaynakları ile kurulan enerji santrallerin birim sistem maliyetleri ve birim enerji maliyetleri karşılaştırmalı olarak Çizelge 2.1'de verilmiştir. Çizelgeden de görüldüğü üzere doğalgaz santrallerinin birim maliyeti en düşüktür. Ancak söz konusu santrallerde hammaddenin dışa bağımlı olduğu unutulmamalıdır.

Çizelge 2.1. Elektrik üretim maliyetleri [4]

Kaynak	Birim Sistem Maliyeti ($/kWh)	Birim Enerji Maliyet (cent/kWh)
Termik Santral (kömür)	1200-1369	4,8-5,5
Doğal Gaz santrali	500-600	3,9-4,4
Rüzgar Santrali	1000-1100	4,0-6,0
Güneş Pili	3000-6000	25,0-100,0
Nükleer Santral	3500-4000	11,1-14,5
Hidroelektrik Santral	800-1000	5,1-11,3

2.4. Rüzgâr Enerjisinin Maliyetini Etkileyen Faktörler

Rüzgârdan elde edilen enerjinin birim maliyeti etkileyen faktörler rüzgâr hızı, rüzgâr tarlasının büyüklüğü, çevresel maliyetler ve faiz oranıdır.

2.4.1 Rüzgâr hızı ve rüzgâr tarlasının büyüklüğü

Rüzgârdan elde edilecek enerjinin miktarı rüzgâr hızının küpü ile orantılı olarak değişir. Bundan dolayı hızdaki değişim enerji üretim miktarı ve birim enerji maliyetinde değişimlere yol açar. Coğrafik olarak rüzgâr hızının yüksek olabileceği yerler;

a) Yağışların sürekli, esen rüzgârlara paralel olduğu vadiler,

b) Yüksek, engebesiz tepe ve platolar,

c) Yüksek basınç gradyanlı geniş düzlükler ve sürekli rüzgâr alan az eğimli vadilerdir [5].

2.4.2. Çevresel maliyetler

Üretilen elektrik enerjisinin gerçek maliyetini belirlemek için üretim esnasında çevreye verilen kirlilik ve sosyal maliyetler hesaba katılmalıdır. İnsan sağlığı ve çevresi üzerinde olumsuz etkileri olan bu maliyete sosyal maliyet adı verilmektedir. Fakat bu maliyetler henüz enerji maliyetlerinde hesaba katılmamaktadır. Örnek olarak petrol çıkarma sırasında yeraltı sularının kirlenmesi, nükleer santrallerde ömrünü tamamlayan atıkların saklanması, kömür çıkarmak için yapılan madenciliğin çevreyi kirletmesi göz ardı edilemez. Bundan dolayı rüzgâr enerjisi çevre dostu bir enerji kaynağı sayılabilmektedir. Enerji üretimi sırasında zararlı emisyon oluşturmaz. Diğer çevresel ve sosyal maliyetler elektrik üretimi maliyetine dâhil edilirse rüzgâr enerjisi teknolojisi diğer enerji teknolojileri ile daha fazla rekabet edebilir [3].

2.4.3. Faiz oranı

Rüzgâr enerjisi ilk yatırım maliyeti yüksek bir teknolojidir. Santralin ilk yatırım maliyeti olarak parçaların üretimi veya satın alınması, santralin kurulması v.b. maliyetleri içerir. Bundan dolayı ilk yatırımda uygulanan faiz oranı geri ödemede önemlidir.

2.5. Dünya Rüzgâr Enerjisi Potansiyeli

Rüzgâr, sahip olduğu kinetik enerji nedeniyle doğal bir enerji potansiyeline sahiptir. Fakat dünyanın sahip olduğu toplam rüzgâr enerjisini tahmin etmek oldukça zordur. Çizelge 2.2'de 2002 itibari ile dünyadaki mevcut kurulu rüzgar gücünün ülkelere göre dağılımı gösterilmiştir. Çizelge 2.2'ye göre türbin teknolojisinde Avrupa ülkeleri ve Amerika Birleşik Devletleri lider konumdadır. Bunun temel nedeni söz konusu ülkelerin rüzgâr teknolojilerini 1960'lı yıllardan bu yana sürekli olarak geliştirmeleridir.

Çizelge 2.2 Dünyadaki mevcut rüzgar kurulu gücünün ülkelere göre dağılımı [6]

No	Ülke	Kurulu Güç (MW)	No	Ülke	Kurulu Güç (MW)
1	Almanya	11968	24	Kosta Rika	71
2	İspanya	5043	25	Belçika	45
3	USA	4674	26	Ukrayna	44
4	Danimarka	2880	27	Finlandiya	39
5	Hindistan	1702	28	Letonya	24
6	İtalya	806	29	Brezilya	19
7	Hollanda	727	30	Türkiye	15
8	İngiltere	570	31	Lüksemburg	14
9	Japonya	486	32	Arjantin	12
10	Çin	473	33	Çek cumhuriyeti	11
11	Yunanistan	462	34	İran	11
12	İsveç	372	35	Tunus	8
13	Kanada	270	36	İsrail	8
14	Portekiz	204	37	Rusya	7
15	Fransa	183	38	Güney Kore	6
16	İrlanda	167	39	İsviçre	5
17	Avustralya	143	40	Meksika	5
18	Avusturya	130	41	İskoçya	7
19	Mısır	125	42	Sri Lanka	2
20	Norveç	97	43	Ürdün	1
21	Yeni Zelanda	82	44	Romanya	1
22	Fas	54	45	Diğer Ülkeler	12
23	Polonya	54	46	Toplam	32039

2.6. Türkiye'deki Rüzgâr Enerjisi Potansiyeli

Türkiye elektrik enerjisi üretimi açısından sorun yaşayan bir ülkedir. Şu an ülkenin içinde bulunduğu darboğazın gelecekte de devam etmesi muhtemel gözükmektedir. Bu nedenle için yeni ve ucuz enerji kaynaklarına ihtiyaç duyulmaktadır. Rüzgâr enerjisi de bunlardan biridir. Türkiye, Avrupa ülkeleri ile eşdeğer düzeyde rüzgâr enerjisi potansiyeline sahiptir. Ülkemizin teorik olarak 83000 MW'lık rüzgâr enerjisi potansiyeline sahip olduğu tahmin edilmektedir. Genel olarak Marmara, Ege, Bozcaada, Gökçeada, Sinop ve İskenderun Türkiye'nin rüzgâr santrali kurmak için potansiyeli en yüksek olan bölgeleridir [7]. Rüzgar enerjisinden elde edilen enerjinin diğer kaynaklardan elde edilen enerji ile yıllara göre karşılaştırılması Çizelge 2.3'de verilmiştir.

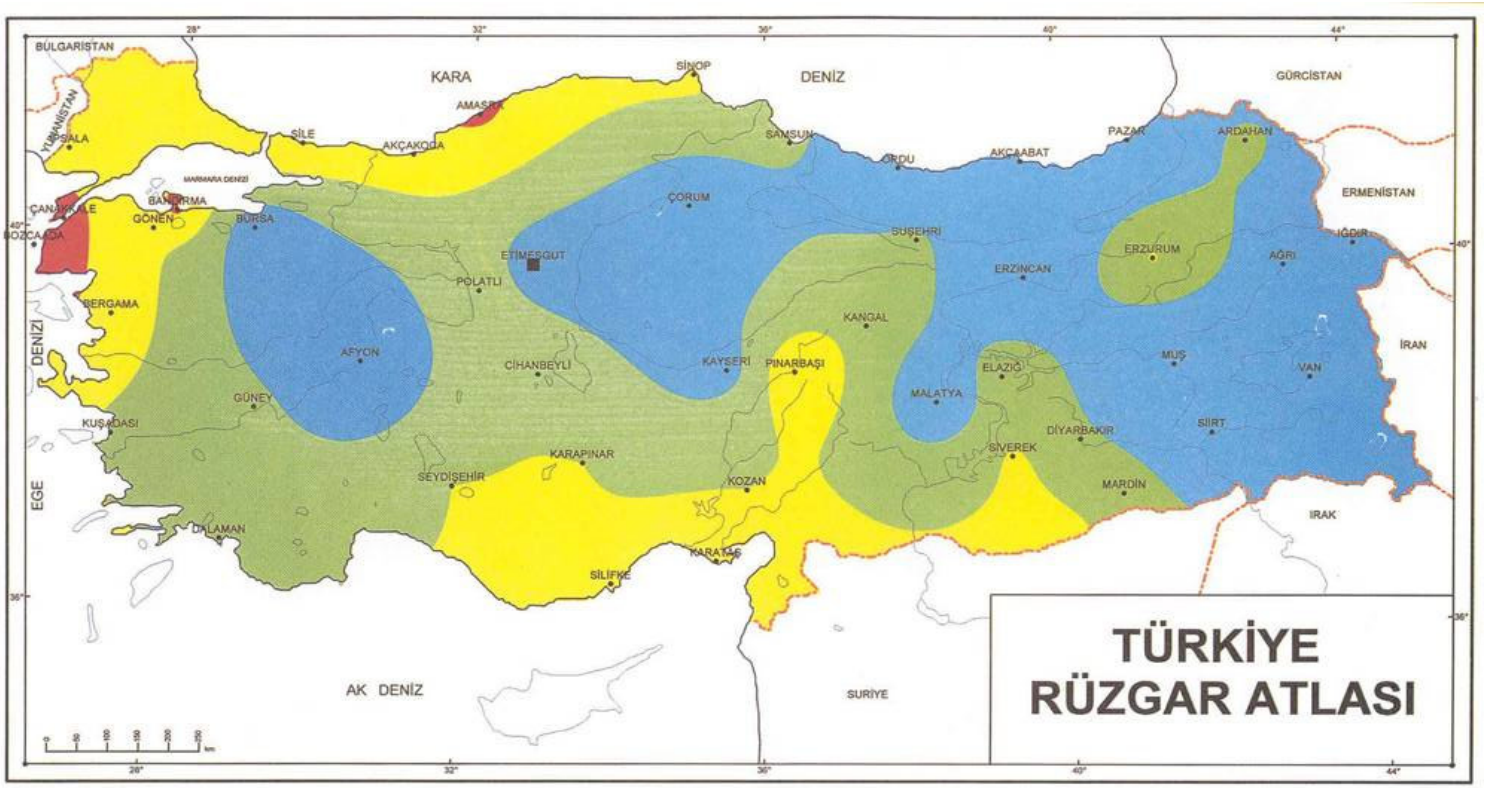
Çizelge 2.3 Türkiye'deki kurulu gücün yıllar itibari ile gelişimi [8]

Yıl	Termik(MW)	Hidrolik(MW)	Rüzgar Enerjisi(MW)	Toplam(MW)
1985	5229,3	3874,8	15	9119,1
1990	9535,8	6764,3	15	16315,1
1995	11074,0	9862,8	15	20951,8
1999	15555,9	10573,2	19	26148,1

Türkiye'nin coğrafi bölgelerindeki ortalama rüzgar hızları Çizelge 2.4'de, tüm ülke için oluşturulan rüzgar atlası ise Şekil 2.1'de gösterilmiştir [9].

Çizelge 2.4. Türkiye'nin coğrafi bölgelerindeki ortalama rüzgar hızları [7]

Bölge Adı	Ort. Rüzgâr Hızı (m/s)
Akdeniz	2,45
İç Anadolu	2,46
Ege	2,65
Karadeniz	2,38
Doğu Anadolu	2,12
Güneydoğu Anadolu	2,69
Marmara	3,29



Beş farklı topoğrafik durum için yer seviyesinden 50 m yükseklikteki rüzgar potansiyelleri[1]

	Kapalı Araziler[2]		Açık Araziler[3]		Kıyılar[4]		Açık Deniz[5]		Tepe ve Bayırlar[6]	
	ms^{-1}	Wm^{-2}	ms^{-1}	Wm^{-2}	ms^{-1}	Wm^{-2}	ms^{-1}	Wm^{-2}	ms^{-1}	Wm^{-2}
	> 6.0	> 250	> 7.5	> 500	> 8.5	> 700	> 9.0	> 800	> 11.5	> 1800
	5.0 – 6.0	150 – 250	6.5 – 7.5	300 – 500	7.0 – 8.5	400 – 700	8.0 – 9.0	600 – 800	10.0-11.5	1200 – 1800
	4.5 – 5.0	100 – 150	5.5 – 6.5	200 – 300	6.0 – 7.0	250 – 400	7.0 – 8.0	400 – 600	8.5-10.0	700 – 1200
	3.5 – 4.5	50 – 100	4.5 – 5.5	100 – 200	5.0 – 6.0	150 – 250	5.5 – 7.0	200 – 400	7.0-8.5	400 – 700
	< 3.5	< 50	< 4.5	< 100	< 5.0	< 150	< 5.5	< 200	< 7.0	< 400

1. Rüzgar potansiyeli, rüzgarın gücünü temsil etmektedir. Rüzgar türbini halihazırdaki potansiyelin % 20 ile % 30 luk bölümünü kullanabilir. Potansiyel hesaplamaları; deniz seviyesinde 1 Atm lik standart basınç ve 15 °C sıcaklığa karşılık gelen 1.23 kg/m³ hava yoğunluğuna göre yapılmıştır.
2. Yerleşim alanları, ormanlar ve rüzgar kırıcıların yoğun olduğu tarım alanları (pürüzlülük sınıfı 3)
3. Az sayıda rüzgar kırıcının olduğu açık araziler (pürüzlülük sınıfı 1). İç bölgelerde en fazla tercih edilen alanlar genellikle bu sınıfta bulunmaktadır.
4. Düzgün kıyı alanları ve çok az sayıda rüzgar kırıcı içeren kara yüzeyleri (pürüzlülük sınıfı 1). Eğer hakim rüzgar yönü deniz tarafından ve sürekli ise, potansiyel daha fazla olabilir. Tam tersi durumda ise potansiyel daha az olabilir.
5. Kıyılardan en az 10 km uzaklıktaki açık denizler (pürüzlülük sınıfı 0).
6. Bütün sınıflarda % 50 ye varan bir hız artışı görülmektedir ve bu sonuç 400 m yüksekliğinde ve 4 km çapındaki simetrik bir tepede yapılan hesaplamalarda elde edilmiştir. Rüzgar hızındaki artış; tepenin yüksekliğine, uzunluğuna ve yapısına bağlıdır.

Harita 2.1. Türkiye rüzgar atlası [9]

Devlet Meteoroloji İşleri Genel Müdürlüğü'nce ölçüm yapılmakta olan istasyonlardan rüzgar şiddeti olarak önemli görülenlere ilişkin bazı veriler Çizelge 2.4'de verilmiştir. Bu çalışma kapsamında gerçekleştirilen bilgisayar programında, Çizelge 2.5'de belirtilen 17 bölge dikkate alınmıştır. Bunun temel nedeni söz konusu 17 bölgede rüzgâr enerjisi santrali kurmanın verimlilik açısından en uygun olacağının düşünülmesidir.

Çizelge 2.5. Türkiye Rüzgar Atlasında önemli görülen bazı bölgelere ilişkin bazı coğrafi veriler [9]

No	İstasyon adı	Enlem	Boylam	Yükseklik (m)	Ölçüm zaman Aralığı	Ortalama Rüzgâr hızı (m/s)
1	Amasra	$41^0$45'02''N	$32^0$23'03''E	73	1989-1998	5,2
2	Bandırma	$40^0$19'54''N	$27^0$59'56''E	58	1989-1998	4,0
3	Bergama	$39^0$07'30''N	$27^0$11'15''E	53	1989-1998	3,0
4	Bozcada	$39^0$50'00''N	$26^0$04'25''E	28	1989-1998	5,8
5	Cihanbeyli	$38^0$39'03''N	$32^0$55'23''E	969	1989-1998	2,9
6	Çanakkale	$40^0$08'33''N	$26^0$24'00''E	6	1989-1998	3,7
7	Diyarbakır	$37^0$54'21''N	$40^0$12'08''E	677	1989-1998	2,8
8	Erzurum	$39^0$57'05''N	$41^0$10'21''E	1758	1989-1998	2,8
9	Güney	$38^0$09'07''N	$29^0$03'34''E	805	1989-1998	4,3
10	İpsala	$40^0$55'06''N	$26^0$22'51''E	10	1989-1998	2,9
11	Karataş	$36^0$34'11'N	$35^0$23'29''E	22	1989-1998	3,1
12	Mardin	$37^0$18'50''N	$40^0$43'37''E	1050	1989-1998	3,9
13	Pınarbaşı	$38^0$43'33''N	$36^0$23'30''E	1500	1989-1998	3,9
14	Sinop	$42^0$01'51'N	$35^0$09'18''E	32	1989-1998	2,9
15	Siverek	$37^0$45'20''N	$39^0$20'00''E	801	1989-1998	2,9
16	Suşehri	$40^0$09'47''N	$38^0$04'26''E	1163	1989-1998	3,2
17	Şile	$41^0$10'13''N	$29^0$36'05''E	31	1989-1998	3,4

Türkiye'de bugüne kadar gerçekleştirilmiş ve gerçekleştirilmesi planlanan rüzgar enerjisi santrallerine ilişkin bilgiler de Çizelge 2.6'de verilmiştir.

Çizelge 2.6. Türkiye'deki rüzgar santrali projeleri [6,9]

No	Projenin Adı	Projeyi geliştiren firma	Yer	Kurulu Güç (MW)
Yapılmış olan projeler				
1	Germiyan R. S.	Delta Plastik	Çeşme Germiyan	1,5
2	Çeşme Alaçatı R.S.	Ares A.Ş.	İzmir-Çeşme-Alaçatı	7,2
3	Bozcaada R.S.	Demirer Holding A.Ş.	Çanakkale	10,2
Devlet Planlama Teşkilatının kararını bekleyen projeler				
4	Kacadağ R.S.	AS Makinsan	Çanakkale	50,4
5	Çanakkale R.S.	AS Makinsan	Çanakkale	30
6	Mazıdağı R.S.	Demirer Holding A.Ş.	İzmir-Çeşme-Alaçatı	39
Fizibilite çalışmaları tamamlanan projeler				
7	Akhisar R.S.	Ak-En (Sasaş İnşaat)	Manisa-Akhisar	12
8	Akhisar R.S.	Demirer Holding A.Ş.	Manisa-Akhisar	30
9	Bandırma R.S.	Atlantis Ticaret	Balıkesir-Bandırma	15
10	Beyoba R.S.	Atlantis Ticaret	Manisa-Akhisar-Beyoba	7,92
11	Çeşme R.S.	Proton	İzmir-Çeşme	12
12	Datça R.S.	Demirer Holding A.Ş.	Muğla-Datça	28,8
13	Datça R.S.	Atlantis Ticaret	Muğla-Datça	12,54
14	İntepe R.S.	Interwind	Çanakkale-İntepe	12
15	İntepe R.S.	Interwind	Çanakkale-İntepe	30
16	Karaburun R.S.	Atlantis Ticaret	İzmir-Karaburun	22,5
17	Yalıkavak R.S.	Atlantis Ticaret	Muğla-Bodrum-Yalıkavak	7,92
Fizibilitesi raporunun yenilenmesi beklenen projeler				
18	Gökçeada R.S.	Simelko	Çanakkale-Gökçeada	5
Fizibilite raporu bekleyen santral planları				
19	Belen R.S.	Teknik Ticaret	Hatay-Belen	20-30
20	Çeres R.S.	Interwind	İzmir-Çeşme	18-25,5
21	Ekinli R.S.	Deryalar Ltd.	Bandırma-Karacabey	39,5
22	Güzeyler R.S.	Enda Enerji Üretim A.Ş.	İzmir-Çeşme	50,4
23	Hacıömerli R.S.	Demirer Holding A.Ş.	İzmir-Hacıömerli	45
24	Kapıdağ R.S.	AS Makinsan	Balıkesir-Erdek	20-35
25	Karabiga R.S.	AS Makinsan	Çanakkale-Karabiga	15-50
26	Kocaali R.S.	Derin Ltd	Tekirdağ-Şarköy	31,2
27	Kocadağ R.S.	Mage A.Ş.	İzmir-Çeşme-Kocadağ	26,25
28	Kumkale R.S.	Demirer Holding A.Ş.	İzmir-Çeşme	12,6
29	Lâpseki R.S.	Atlantis Ticaret	Çanakkale-Lapseki	15
30	Mazıdağı–2 R.S.	Demirer Holding A.Ş.	İzmir-Çeşme	90
31	Mazıdağı–3 R.S.	Yapısan Ltd.	İzmir-Çeşme	39,6
32	Paşalimanı R.S.	AS Makinsan	Marmara-Kapıdağ	9
33	Şenköy R.S.	Akfırat A.Ş.	Hatay-Şenköy	12
34	Seyitali R.S.	Derin Ltd.	İzmir-Aliağa	51
35	Taştepe R.S.	Fora A.Ş.	Bandırma-Taştepe	37,8
36	Topdağı R.S.	Derin Ltd.	Sinop	33
37	Yaylaköy R.S.	Mage A.Ş.	İzmir-Karaburun	15
38	Yellice Belen R.S.	AS Makinsan	Hatay-Belen-Yellice	70-100
39	Yenişarkan R.S.	Yapısan İnşaat Ltd.	İzmir-Aliağa-Bahçedere	54
40	Zeytinbağ R.S.	Deryalar Ltd.	Bursa-Zeytinbağ	30-60

Türkiye'deki özel sektöre ait olan rüzgâr santralinin özellikleri ise şunlardır: Çeşme-Alaçatı rüzgar çiftliğinde; 600 kW'lık, rotor çapı 44 m olan 12 adet Vestas V 44 rüzgar türbini, Çeşme-Germiyan rüzgar çiftliğinde; 500 kW'lık, rotor çapı 40,3 m olan 3 adet Enercon 40 rüzgar türbini, Çanakkale-Bozcaada rüzgar çiftliğinde; 600 kW'lık, rotor çapı 44 m olan 17 adet Enercon 40 rüzgar türbini vardır [10].

3. RÜZGÂR TÜRBİNLERİ

Rüzgâr türbinleri, rüzgârın döndürme kuvvetini kanatları yardımıyla elektrik enerjisine çeviren sistemlerdir. Havanın özgül kütlesi oldukça az olduğundan rüzgârdan elde edilecek enerji esas olarak rüzgâr hızına bağlıdır. Rüzgâr hızı yükseklikle, gücü ise hızının küpü doğru ile orantılı olarak artar. Rüzgâr santrali kurulması planlanan bir bölgenin en az bir yıllık rüzgâr ölçümlerinin yapılması tavsiye edilmektedir. Ortalama rüzgâr hızı topografik özeliklere göre yıldan yıla değişiklik gösterebilir. Rüzgâr hızındaki değişliklerden dolayı, rüzgârdan elde edilecek olan enerji potansiyeli de değişiklik gösterir. Bu yüzden belli bir bölgede, rüzgârdan üretilecek elektrik enerjisinin planlanan üretim miktarı belirlenirken, yıllık ortalama rüzgâr hızı, "Weibull dağılımı" ile hesaplanmış rüzgâr hızı dağılımının kullanılması daha uygundur [2]. Fakat bu çalışmada türbin tasarımı sırasında ortalama rüzgâr hızı kullanılmıştır. Rüzgâr türbinleri ana milin dönüş eksenine göre yatay eksenli veya düşey eksenli olarak imal edilebilir. Bu iki tipten yaygın olarak kullanılanı yatay eksenli türbinlerdir.

3.1.Yatay Eksenli Rüzgâr Türbinleri

Rotorları yüzeye paralel olan türbinlerdir. Geleneksel bir kullanım imkânı sunar. Maksimum güç elde etmek için rotorları rüzgâra dik konumlandırılmalıdır. Yatay eksenli rüzgâr türbinleri önden rüzgâr alan ve arkadan rüzgâr alan olmak üzere iki şekilde imal edilirler.



(a) (b)

Resim 3.1. (a) Önden rüzgar alan yatay eksenli türbin, (b) Arkadan rüzgâr alan yatay eksenli türbin

3.1.1. Önden rüzgâr alan türbinler

Rotorun yüzü rüzgâra doğru konumlandırılmış olan türbinlerdir. Daha yaygın olarak kullanılır. Çünkü bu türbinlerde jeneratörün arkasına yani rotora doğru yapacağı gölgeleme etkisi ortadan kalkmış olur. Fakat kulenin önünde ihmal edilebilecek düzeyde bir rüzgâr gölgelemesi oluşur. Kanatlar kuleye göre eğilimli konumlandırılırlar. Kulenin silindirik yapılması bile kanadın kuleden her geçişinde türbinin ürettiği gücü azaltır. Bundan dolayı kanatların sert malzemeden yapılması ve kuleye göre biraz uzakta konumlandırılması daha uygun olmaktadır.

3.1.2. Arkadan Rüzgâr alan türbinler

Bu türbinlerde ise rotor türbinin arkasında olacak şekilde konumlandırılmıştır. Fakat türbin rotora perdeleme etkisi yaptığı için kullanımı yaygın değildir [5].

3.2. Düşey Eksenli Rüzgâr Türbini

Düşey eksenli türbinlerde, kanatların iç bükey ve dış bükey yüzeyleri arasındaki çekme kuvveti farkı nedeniyle dönme hareketi oluşmaktadır. Bu türbinlerin dönme eksenleri düşeydir. Çalışma basitliğinden dolayı kullanılan en eski rüzgâr türbini çeşididir. Düşey eksenli rüzgâr türbinleri, yatay eksenli rüzgâr türbinlerine göre daha düşük verime sahiptir [5].

Çizelge 3.1'de ise bazı rüzgar türbinlerinin özellikleri verilmiştir.

Çizelge 3.1. Bazı yatay ve düşey eksenli rüzgar türbin tipleri ve kullanım amaçları [11]

Rotor Tipi	λ (kanat uç hız oranı)	C_p (Güç Katsayısı)	Kullanım Amaçları
Pervane Tipi	*3-10*	*0,42-0,57*	*Elektrik Üretimi*
Darrieus Tipi	*5-6*	*0,40*	*Elektrik Üretimi*
Cyclogiro Tipi	*3-4*	*0,45*	*Elektrik Üret. veya su pomp.*
Fan Tipi	*1*	*0,3*	*Su Pompalama*
Çok Kanatlı Tip	*3-4*	*0,35*	*Elektrik Üret. veya su pomp.*
Yelken Kanat Tipi	*4*	*0,35*	*Elektrik Üret. veya su pomp.*
Savonious Tipi	*1*	*0,15*	*Su Pompalama*
Hollanda Tipi	*2-3*	*0,17*	*Su Pomp. veya değirmen*

3.3. Rüzgâr Türbinlerinin Yapısal Özellikleri

Rüzgâr türbinin rotoru, çevredeki engellerin (ev, ağaç v.b.) rüzgâr hızını kesmeyecek yükseklikte konumlandırılmalıdır. Tepelerin üst arka kısımları türbülans nedeniyle konumlandırma için uygun değildir. Malzeme ve işçilik maliyetleri göz önünde bulundurarak türbinlerin enerji nakil hatlarının yakınına yerleştirilmesi gerekir. Rotorun dönüş hızı sistemin yapısına göre sabit veya değişken olabilir. Modern rüzgâr türbinlerinde rotor göbekleri yerden 25–75 m yükseklikte bir kule üzerinde

bulunur. Kuleler konstrüksiyonun sağlamlığı açısından çelik kafes ya da çelik silindirden yapılırlar.

Rüzgâr türbinlerinin kanatları ise polyester ile kuvvetlendirilmiş fiberglas veya epoxy ile güçlendirilmiş fiber karbondan yapılmakta ve çelik omurga ile desteklenmektedir. Kanatlardaki gücün kontrolü için üç yöntem kullanılmaktadır.

a) Kanat açısı kontrolü (pitch kontrol) adı verilen kontrol yönteminde kanat açıları kontrol sistemi tarafından ayarlanmaktadır. Sistem jeneratörden gelen sinyal ile çalışmaktadır. Jeneratör gücü nominal gücün üstüne çıktığı zaman rotor kanatları kendi ekseni etrafında dönerek ve gelen rüzgara göre kanadın hücum açısını değiştirerek aerodinamik kuvvetleri azaltır. Böylece frenleme veya durdurma gerçekleştirilir, türbinin güç çıkışı azaltılmış olur.

b) "Stall kontrol" yönteminde ise ayarlanacak hareketli parça yoktur. Hava akımının az veya hücum açısının fazla olması dolayısıyla pervane kanadının kaldırma kuvvetinin azalıp havada asılı kalmayacak hale gelmesine "stall kontrol" denir. Rotor kanadın doğal aerodinamik özelliği ile durdurulur.

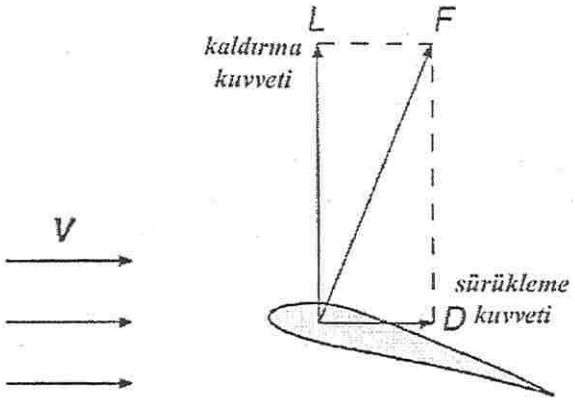
c) "Pitch kontrol" ve "stall kontrolün" birleşimi olarak gerçekleştirilen aktif stall kontrol sistemidir.

Rüzgâr türbinlerinin ömrü, türbin kalitesine ve yerel iklim özelliklerine göre 20 ile 25 yıl arasında değişmektedir. Rotor yüksekliğindeki türbülans yoğunluğu ve bundan dolayı oluşacak mekanik yorulma, türbin ömrünü etkileyen önemli faktörlerdir.

4. KANAT TASARIMININ AERODİNAMİK ESASLARI

4.1. Sürükleme ve Kaldırma Kuvveti

Günümüzde rüzgâr türbinlerinin kanat tasarımı yapılırken, etkili olan iki önemli aerodinamik kuvvet bulunmaktadır. Bu kuvvetler kanadın sürükleme ve kaldırma kuvvetleridir [12]. Kanadın sürükleme kuvveti, kanada akış yönünde etkiyen kuvvettir. Örnek olarak, kanat yapısını düz bir plaka olarak düşünürsek plaka üstünde oluşacak maksimum sürükleme kuvveti plakanın akış yönüne 90^0 derece dik konumlandırıldığı zaman oluşmaktadır. Minimum sürükleme kuvveti ise, kanat akış yönüne paralel olarak konumlandırıldığı meydana gelir. İdeal bir kanatta sürükleme kuvvetinin az olması tercih edilir. Şekil 4.1'de kanat üstündeki kaldırma sürükleme kuvvetinin etkisi gösterilmiştir.



Şekil 4.1. Kanada etki eden kaldırma ve sürükleme kuvveti basit gösterimi

Sürükleme kuvvetini minimuma indirmek için kanat tasarımında yağmur damlasını andıran tasarımlar yapılmaktadır. Bunlara "akış hatlı" cisimler adı verilir. Kanadın kaldırma kuvveti ise, kanada akış yönüne dik olarak etkiyen kuvvettir. Örnek olarak uçakların havalanması kanadın kaldırma kuvveti ile sağlanmaktadır. Maksimum kaldırma kuvveti kanat akış yönüne paralel konumlandırıldığında meydana gelir. İdeal bir kanat tasarımında kaldırma kuvvetinin yüksek olması tercih edilir. Havanın akış yönüne göre, kanat ile yapacağı açılardan dolayı akış şiddeti artar. Artan akış şiddetinden dolayı düşük basınçlı bölgeler oluşur. Bu bölgelere akışaltı (downstream)

denir. Dolayısıyla, hava akış hızı ile basınç arasında bir ilişki vardır. Yani hava akış hızı hızlandıkça basınç düşer, hava akışı yavaşladıkça basınç artar. Bu ilişkiye "Bernoulli etkisi" denir. Bu ilişki sonucunda, kanat üzerinde kaldırma kuvveti oluşmuş olur [12].

4.2. Geliştirilmiş Olan Modeller

Rüzgâr türbini üretimi için geliştirilen modeller güç tahmini için önem taşımaktadır. İdeal disk teorisi geliştirilmiş olan ilk teorilerdendir. Bu teoride koşullar ideal olarak düşünülmüştür. Bu teorinin temelinde sonsuz sayıda kanada sahip ve sıfır kalınlıkta bir rotor bulunmaktadır [7, 13, 14].

Kanat Elemanı Momentum Teorisi ise İdeal Disk Teorisi ve Kanat Elemanı Teorisinin birleşimidir [13]. Şu anda geçerliliği olan bir teoridir. Kanat Elemanı Teorisi ile kanatların şekli tanımlanabilir ve kanattaki diferansiyel (eğrisel) kısmın analizi yapılabilir. Glauert ise akım tüpü analizini kullanarak tek boyutlu momentum teorisi ve Kanat Elemanı Teorisini kullanarak kendi modelini geliştirmiştir [15]. Yukarıda bahsedilen teorilerde 2 boyutlu akış olduğu kabul edilerek analizler yapılmıştır.

Prandtl ise kanatlar için "Kaldırma Çizgisi Teorisini" geliştirmiştir. Bu model ile yatay eksenli rüzgâr türbinlerinin performans tahmininde yeni bir boyut ortaya çıkmıştır. Kaldırma Çizgisi teorisi, kanat üzerindeki kaldırmanın dağılımının hesaplanması için geliştirilmiş yoldur. "Kaldırma yüzeyleri metodu" ise karmaşık kanat şekilleri için yararlı bir hesaplamadır. Değişik serbest rüzgâr hızları için yapılan diğer bir çalışma ise "Girdap Teorisi" dir [13].

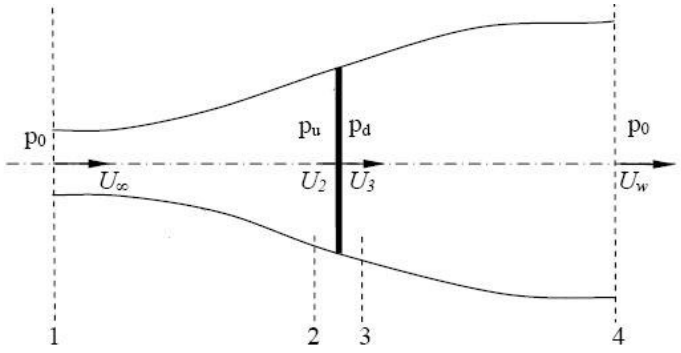
4.3. İdeal Disk Teorisi ve Betz Limiti

Rüzgârdan enerji elde etmek için kurulacak türbinin boyutları aerodinamik yapı ile doğrudan ilişkilidir. Sistem ne kadar ideal olursa olsun rüzgârdan elde edilecek enerjinin bir üst limiti vardır. Betz tarafından 1919–1926 yılında belirlenen bu limite

Betz limiti denir [11, 16, 17, 14]. Betz söz konusu teoriyi hesaplarken hareketli diskin önünde, üzerinde ve gerisindeki hava akımları için enerjinin korunumu kanunlarını kullanmıştır. Betz'in ideal disk teorisine göre;

Diskten geçen havanın akış hızı, disk alanının her noktasında eşit olmasına rağmen basınç ani olarak düşer. Bu basınç farkından dolayı diskin hareket enerjisi artmış olur. Betz teoremine göre hesaplama yapılırken, akışın sürekli, homojen ve sıkıştırılamaz olduğu, disk üzerindeki basınç değişiminin disk üzerindeki her noktada aynı olduğu, sonsuz sayıda kanat olduğu ve diskin önünde ve arkasında türbülans olmadığı kabullenilmektedir.

İdeal disk teorisinin analizi, kontrol hacmi alınarak yapılır. Burada kontrol hacminin sınırları şekilde gösterildiği gibi akım tüpünün sınırları ve akım tüpünün iki ucudur (Şekil 4.2).



Şekil 4.2. Akım tüpündeki hareketli diskten geçen idealleştirilmiş akış [7,16]

Burada diskin içinden geçen hava akış hızının sürekli olduğu varsayılırsa;

$$U_2 = U_3 = U_R \tag{4.1}$$

yazılır. Sabit debili bir akışta diskten geçen havanın debisi ise;

$$\dot{m} = \rho A U_R \tag{4.2}$$

olur. Tüm sistemi çevreleyen kontrol hacminde lineer momentumun korunumu ilkesi uygulanır, böylece kontrol hacminin içerisindeki net kuvvetler bulunur. Bernoulli fonksiyonu, hareketli diskin her iki tarafındaki kontrol hacminde de kullanılabilir. Serbest akış ve rotorun ön tarafı arasında ve rotorun arka tarafı ile oluşan dalga arasında sırasıyla Bernoulli fonksiyonu uygulanırsa;

$$p_o + \frac{1}{2}\rho U_\infty^2 = p_u + \frac{1}{2}\rho U_R^2 \tag{4.3}$$

$$p_d + \frac{1}{2}\rho U_R^2 = p_O + \frac{1}{2}\rho U_w^2 \tag{4.4}$$

Burada T rotordaki rüzgâr kuvvetidir ve itme kuvveti olarak adlandırılmaktadır. İtme kuvveti, hareketli diskin her iki tarafındaki kuvvetlerin toplamı olarak da aşağıdaki şekilde ifade edilebilir:

$$T = Ap' \tag{4.5}$$

Burada *p'* hareketli disk yüzeyindeki basınç azalmasını ifade etmektedir.

$$p' = (p_u - p_d) \tag{4.6}$$

Eş. 4.3 ve Eş. 4.4 nolu Bernoulli denklemini kullanılarak basınç azalması;

$$p' = \frac{1}{2}\rho(U_\infty^2 - U_w^2) \tag{4.7}$$

olarak yazılabilir. Eş. 4.7, Eş. 4.5 nolu eşitlikte kullanırsa;

$$T = \frac{1}{2}\rho A(U_\infty^2 - U_w^2) \tag{4.8}$$

olur. Rotor düzlemindeki hız ise;

$$U_R = \frac{U_\infty + U_w}{2} \tag{4.9}$$

olarak ifade edilebilir. Böylece, rotor düzlemindeki rüzgâr hızı basit bir model kullanılarak ortalama olarak bulunur. Eş. 4.9 nolu eşitlikten görüldüğü gibi hareketli disk üzerindeki akış hızı, diskin önündeki ve arkasındaki hızların aritmetik ortalamasıdır. Eş. 4.9 nolu eşitliğe göre $U_w<U_R$ ve $U_R<U_\infty$ olur. Eksenel akış faktörü *a* ise aşağıdaki şekilde ifade edilebilir.

$$a = \frac{U_\infty - U_R}{U_\infty} \tag{4.10}$$

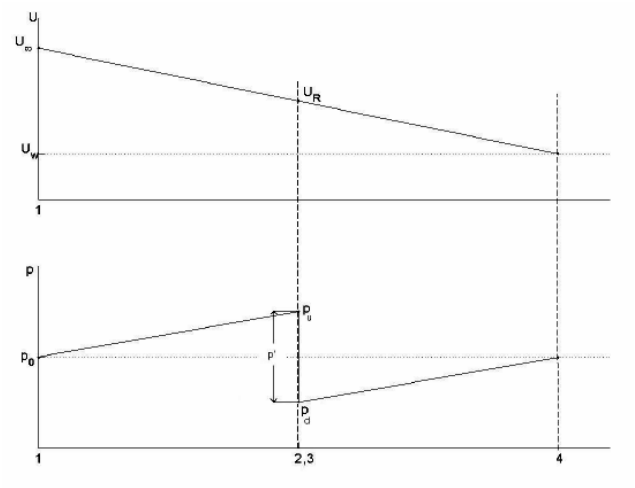
Eksenel akış faktörü 0-1 arasında değişen boyutsuz bir sayıdır. Buradan rotor düzlemindeki hız ile eksenel akış faktörü arasındaki ilişki;

$$U_R = U_\infty(1-a) \tag{4.11}$$

ve diskten sonraki akış hızı ile eksenel akış faktörü arasındaki ilişki ise;

$$U_w = U_\infty(1-2a) \tag{4.12}$$

olarak belirlenir. Akım tüpü boyunca hız ve basınç dağılımına ilişkin örnek Şekil 4.3'de gösterilmiştir.



Şekil 4.3. Akım tüpü boyunca hız ve basınç dağılımı [7, 17]

Rotordaki güç çıkışı (P) ise, itme kuvvetinin ve rotor düzlemindeki rüzgâr hızının çarpımına eşit olarak şu şekilde belirlenir;

$$P = TU_R \tag{4.13}$$

Yazılır. Eş. 4.8 nolu eşitlik Eş. 4.13 nolu eşitlikte kullanılırsa;

$$P = \frac{1}{2}\rho A(U_\infty^2 - U_w^2)U_R \tag{4.14}$$

eşitliği elde edilir. Daha önce yazılan Eş. 4.11 ve Eş. 4.12 nolu eşitliklerden U_R ve U_w değerleri Eş. 4.14 nolu eşitlikte yerine yazılırsa;

$$P = 2\rho Aa(1-a)^2 U_\infty^3 \tag{4.15}$$

eşitliği elde edilir. Türbin güç katsayısı ise;

$$c_p = \frac{P}{\frac{1}{2}\rho U_\infty^3 \pi R^2} \tag{4.16}$$

eşitliği ile ifade edilebilir. Güç katsayısı c_p ise, Eş. 4.15 nolu eşitlik Eş. 4.16 nolu eşitlikte kullanılırsa,

$$c_p = 4a(1-a)^2 \tag{4.17}$$

haline gelir. Maksimum güç katsayısını c_p bulmak için Eş. 4.17 nolu eşitliğin türevi alınıp, sıfıra eşitlenirse (a=1/3 olarak alındığında) ;

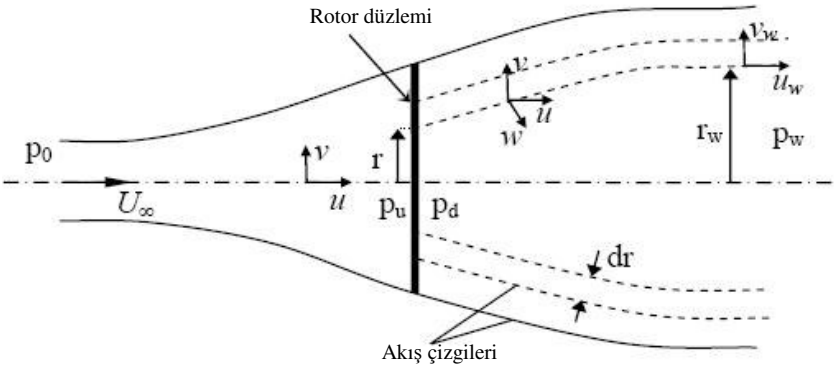
$$(c_p)_{\max} = \frac{16}{27} = 0.5926$$

elde edilir. Bulunan bu değer, ideal ortam şartlarında, rüzgârdan çekilebilecek enerjinin teorik olarak en üst limitidir [16]. İdeal çalışma şartlarında rotor serbest şekilde akan rüzgârın ancak 2/3 ününden faydalanabilir. Fakat pratik kullanımda istenilen ideal çalışma şartları sağlanamayacağından bu limite ulaşılması zordur. Bunun temel nedenleri; rotor gerisindeki dairesel dalgalar, sınırlı sayıdaki kanat sayısı ve kanat uç kayıpları ile aerodinamik sürtünmedir.

4.4. Genel Momentum Teorisi

İdeal disk teorisi üzerine geliştirilen bu teoride kanat arkasındaki akış çizgilerinde akışın döner hareket etmediği kabul edilir. Ayrıca hareketli diskin yerine türbin kanatları bulunmaktadır. Bu durum hızda herhangi bir değişme olmadan akışta ani basınç düşmesine neden olmaktadır. Genellikle, akış, kanat maksimum döndürme momentine ulaşınca döngüsel hareket alır. Söz konusu döngüsel hareket enerji kaybını ifade etmektedir. Döngüsel hareketin etkilerini içeren teoriyi genişletilerek, eksenel ve radyal bileşenlerin değişmeden kaldığı zamanda dönel bileşenin bulunması gerekmektedir.

Akım tüpü analizi kullanarak, dalga hızı ve rotordaki rüzgâr hızı arasındaki ilişki eşitliklerle ifade edebilir. Türbin rotorundan geçen akışın akım tüpündeki geometrisi Şekil 4.4'de verilmiştir.



Şekil 4.4. Rotordan geçen akışın akım tüpündeki geometrisi [7]

Şekil 4.4'de şekle göre, r rotor düzlemindeki herhangi bir yarıçaptaki kanat elemanının mesafesini, *u* ve *v* ise akıştaki yatay ve düşey hızları göstermektedir. p_u ise iç akıştaki basıncı, p′ ise rotor düzleminden çıkan akışın açısal hızla bağlı olarak azalan basıncı göstermektedir. Dalgada ise p_w basıncı, u_w eksenel hızı ve w_w akış tüpü ekseninden r_w yarıçap mesafesindeki açısal hızı göstermektedir.

Rotordaki dairesel eleman için akışın sürekliliği uygulanırsa;

$$u_w r_w dr_w = urdr \tag{4.18}$$

Pervaneden geçen rüzgâr akışın açısal momentumunun sabit olduğu kabul edilirse;

$$w_w r_w^2 = wr^2 \tag{4.19}$$

Dolayısıyla kanat elemanın döndürme momenti,

$$dQ = \rho u w r^2 dA \tag{4.20}$$

Olur. Burada $dA = 2\pi r dr$

Bernoulli denklemi, akış tüpünde diskten önce ve sonraki enerjiler düşünülerek yazıldığında;

$$H_o = p_o + \frac{1}{2}\rho U_\infty^2$$

$$= p_u + \frac{1}{2}\rho(u^2 + v^2)$$

$$H_1 = p_d + \frac{1}{2}\rho(u^2 + v^2 + w^2 r^2)$$

$$= p_w + \frac{1}{2}\rho(u_w^2 + w_w^2 r_w^2)$$

Böylece

$$H_o - H_1 = p' - \frac{1}{2}\rho(w^2 r^2) \tag{4.21}$$

olur. Eş. 4.21 nolu eşitlik, kanat elemanı üzerinden geçen toplam basınçtaki azalmayı göstermektedir. Toplam basınç ise;

$$p_o - p_w = \frac{1}{2}\rho(u_w^2 - U_\infty^2) + \frac{1}{2}\rho w_w^2 r_w^2 + (H_o - H_1)$$

$$= \frac{1}{2}\rho(u_w^2 - U_\infty^2) + \frac{1}{2}\rho(w_w^2 r_w^2 - w^2 r^2) + p' \tag{4.22}$$

eşitliği ile bulunabilir. Ani basınç düşüşünü (p') bulmak için, Bernoulli eşitliği kanadın giriş ve çıkış akışına uygulanabilir. Kanatlar Ω açısal hızı ile dönmektedir. Rotor arkasındaki akış, rotordaki akışın döndürme momentinden dolayı rotora ters yönde döner. Bundan dolayı kanattaki havanın açısal hızı Ω' dan (Ω+w)'ya artar ancak hızın eksenel bileşenleri sabit kalır. Bu sonuç;

$$p' = \frac{1}{2}\rho[(\Omega + w)^2 - \Omega^2]r^2$$

$$= \rho(\Omega + \frac{w}{2})wr^2 \tag{4.23}$$

şeklinde ifade edilebilir. Sonuçta, Eş. 4.23 nolu eşitlik ile Eş. 4.19 ve Eş. 4.22 nolu eşitlikler birleştirilirse, kanat gerisindeki basınç düşüşü;

$$p_o - p_w = \frac{1}{2}\rho(u_w^2 - U_\infty^2) + \rho(\Omega + \frac{w}{2})w_w r_w^2 \tag{4.24}$$

olur. Kanat gerisindeki basınç grandyantı, akıştaki merkezkaç kuvvetini dengeler ve aşağıdaki eşitliğe dönüştürülür:

$$\frac{dp_w}{dr_w} = \rho w_w^2 r_w \tag{4.25}$$

Eş. 4.24 nolu eşitliğin r_w ye göre diferansiyeli alınır ve Eş. 4.26 nolu eşitliğe eşitlenir. Diferansiyel eşitlik düzlem sonrasındaki eksenel ve dönel hızlara bağlı olarak aşağıdaki gibi bulunur;

$$\frac{1}{2}\frac{d}{dr_w}\left[U_\infty^2 - u_w^2\right] = \left(w_w + \Omega\right)\frac{d}{dr_w}\left(w_w r_w^2\right) \tag{4.26}$$

Daha sonra kanat elemanı için eksenel momentum eşitliği aşağıdaki gibi olur.

$$T = \int \rho u_w \left(U_\infty - u_w\right) dA_w + \int \left(p_o - p_w\right) dA_w \tag{4.27}$$

Eş. 4.27 nolu eşitliğin diferansiyeli alınmış hali ise aşağıdaki gibidir:

$$dT = \rho u_w (U_\infty - u_w) dA_w + (p_o - p_w) dA_w \tag{4.28}$$

Rotor düzlemindeki basınç düşüşü, dT aşağıdaki gibi yazılabilir,

$$dT = p'dA \tag{4.29}$$

Eş. 4.23 nolu eşitlikteki değer Eş. 4.29 nolu eşitlikte yerine konulursa,

$$dT = \rho\left(\Omega + \frac{w}{2}\right) w r^2 dA \tag{4.30}$$

ifadesi elde edilir. Sonuç olarak Eş. 4.18, Eş. 4.22, Eş. 4.28 ve Eş. 4.30 nolu eşitliklerin birleştirilmesiyle;

$$\frac{1}{2}\left[U_\infty - u_w\right]^2 = \left[\frac{\Omega + \frac{w_w}{2}}{u_w} - \frac{\Omega + \frac{w}{2}}{U_\infty}\right] u_w w_w r_w^2 \tag{4.31}$$

Elde edilir. Eş. 4.18, Eş. 4.19, Eş. 4.26 ve Eş. 4.31 nolu eşitliklerin, kanadın itme kuvveti ve döndürme momenti hakkında karar vermeye yeterli olmaktadır. Örnek olarak, eğer açısal hız w_w yarıçapın (r) fonksiyonu olarak istenirse, Eş. 4.31 nolu eşitlikten eksenel hız u_w bulunur. Sonra Eş. 4.31 nolu eşitlik, Eş. 4.18 ve Eş. 4.26 nolu eşitliklerle birlikte kullanılarak rotor düzlemindeki eksenel ve dairesel hızlar bulunabilir. Rotorun itme kuvveti ve döndürme momenti Eş. 4.30 nolu eşitlik ve Eş. 4.20 nolu eşitlikten bulunur. Bu eşitliklerin karmaşıklığından dolayı, akış çizgisindeki döngüsel hızlar kesin sonuçlar bulmak için çok küçüktür bu yüzden ihmal edilebilir.

Genel momentum eşitliklerinin kesin çözümü, kanat arkasındaki akışın eksen boyunca döngüsel olmadığı zaman bulunabilir. Bu koşullar, döngüsel momentumun wr^2 tüm radyal elemanlar için aynı değere sahip olduğunda kabul edilir. Eş. 4.29 nolu eşitliğin etkisiyle, eksenel hız, u_w eşitliğin sağ tarafı sıfır olduğundan dolayı yarıçap boyunca sabittir.

u ve u_w eksenel hızları aşağıdaki gibi yazıldığında,

$$u = U_\infty(1-a)$$

$$u_w = U_\infty(1-b)$$

u ve u_w değerlerini veren eşitlikler, Eş. 4.31 nolu eşitlikte yerine konulduğunda;

$$a = \frac{b}{2}\left[1 - \frac{(1-a)b^2}{4\lambda^2(b-a)}\right] \tag{4.32}$$

elde edilir. Yaklaşık çözüm için, varsayımlar yapılır. Açısal hız w kanatın açısal hızı Ω ile karşılaştırıldığında çok küçüktür ve bu yüzden genel eşitlikte w^2 ihmal edilerek basitleştirmek mümkündür. Rotor gerisindeki havanın basıncı p_w rotor öncesindeki ilk basınca p_o a eşittir ve rotor diskten geçen basıncın azalması p' toplam basınç azalmasına $(H_o\text{-}H_1)$eşittir. İtme kuvveti ve eksenel hız arasındaki ilişki basit momentum teorisi ile aynıdır, rotor düzlemindeki eksenel hız u $[u = U_\infty(1-a)]$, serbest akan rüzgârın hızına U_∞ cinsinden yazılırsa kanat elemanın itme kuvveti, Eş. 4.33 nolu eşitlikteki gibi elde edilir,

$$dT = 2\rho u(u - U_\infty)dA$$

$$= 4\pi\rho U_\infty^2 a(1-a)rdr \tag{4.33}$$

Eş. 4.23 nolu eşitlik Eş. 4.33 nolu eşitliğin içinde yazılırsa;

$$dT = p'dA$$

$$2\pi\rho(\Omega + \frac{w}{2})wr^3dr$$

Açısal akım faktörü $a' = w/2\Omega$

$$dT = 4\pi\rho\Omega^2 a'(1+a')r^3 dr \tag{4.34}$$

Eş. 4.33 ve Eş. 4.34 nolu eşitliklerde verilen itme kuvvetleri için iki ifade eşitlendiğinde, eksenel akış faktörü *a* ve açısal akış faktörü *a'* arasındaki ilişki bulunur:

$$\frac{a(1-a)}{a'(1-a')} = \frac{\Omega^2 r^2}{U_\infty^2} = \lambda_r^2 \tag{4.35}$$

Rotordaki her elemanın döndürme momenti Eş. 4.20 nolu eşitlikten bulunur;

$$dQ = \rho u w r^2 dA$$

$$= 4\pi\rho U_\infty \Omega a'(1-a) r^3 dr \tag{4.36}$$

Her radyal eleman için oluşturulan güç, *dP* aşağıdaki eşitlikle ifade edilir;

$$dP = \Omega dQ \tag{4.37}$$

dQ Eş. 4.36 nolu eşitlikten, Eş. 4.37 nolu eşitliğin içine yerleştirildiğinde, lokal uç hız oranı λ_r tanımlanırsa, kanattaki her bir radyal eleman için güç,

$$dP = \frac{1}{2}\rho A U_\infty^3 \left[\frac{8}{\lambda^2} a'(1-a)\lambda_r^3 d\lambda_r \right] \tag{4.38}$$

Şeklinde ifade edilir. Her bir halkadaki artan güç katsayısı;

$$dC_p = \frac{dP}{\frac{1}{2}\rho U_\infty^3 A} \tag{4.39}$$

eşitliği ile ifade edilir. Eş. 4.38 nolu eşitlik Eş. 4.39 nolu eşitliğin içinde konulduğunda ve elemantal güç katsayısının göbekteki lokal uç hız oranından λ_h, uç hız oranına kadar integrali alınırsa C_p,

$$C_p = \frac{8}{\lambda^2}\int_{\lambda_h}^{\lambda} a'(1-a)\lambda_r^3 d\lambda_r \tag{4.40}$$

olur. Eş. 4.40 nolu eşitliğin integralini almak için a, a', ve λ_r değişkenleri arasındaki ilişkinin bilinmesine ihtiyaç duyulmaktadır. Eş. 4.35 nolu eşitliğin çözülerek a', a cinsinden aşağıdaki gibi yazılır:

$$a' = -\frac{1}{2} + \frac{1}{2}\sqrt{1 + \frac{4}{\lambda_r^2}a(1-a)} \tag{4.41}$$

Eş. 4.40 nolu eşitlikteki $a'(1-a)$ değeri maksimum olduğu zaman mümkün olan maksimum güç üretimi için aerodinamik koşullar meydan gelmektedir. $a'(1-a)$ terimi Eş. 4.41 nolu eşitliğin içine a' değerini bulmak için kullanılırsa ve a ya göre türevi alınıp sıfıra eşitlendiğinde:

$$\lambda_r^2 = \frac{(1-a)(4a-1)^2}{(1-3a)} \tag{4.42}$$

olur. Eş. 4.24 nolu eşitlik Eş. 4.17 nolu eşitliğin içine konulursa, her bir halka için maksimum güç aşağıdaki gibi bulunur;

$$a' = \frac{(1-3a)}{(4a-1)} \tag{4.43}$$

Eğer Eş. 4.42 nolu eşitlik a ya göre çarpanlarına ayrılırsa, $d\lambda_r$ ve da arasındaki matematiksel eşitlik aşağıdaki gibi olur,

$$2\lambda_r d\lambda_r = \left[\frac{6(4a-1)(1-2a)^2}{(1-3a)^2}\right] da \tag{4.44}$$

Daha sonra Eş. 4.42, Eş. 4.43, Eş. 4.44 nolu eşitlikler Eş. 4.40 nolu eşitliğin içine yerleştirilirse,

$$C_{p,\max} = \frac{24}{\lambda^2}\int_{a_1}^{a_2}\left[\frac{(1-a)(1-2a)(1-4a)}{(1-3a)}\right]^2 da \tag{4.45}$$

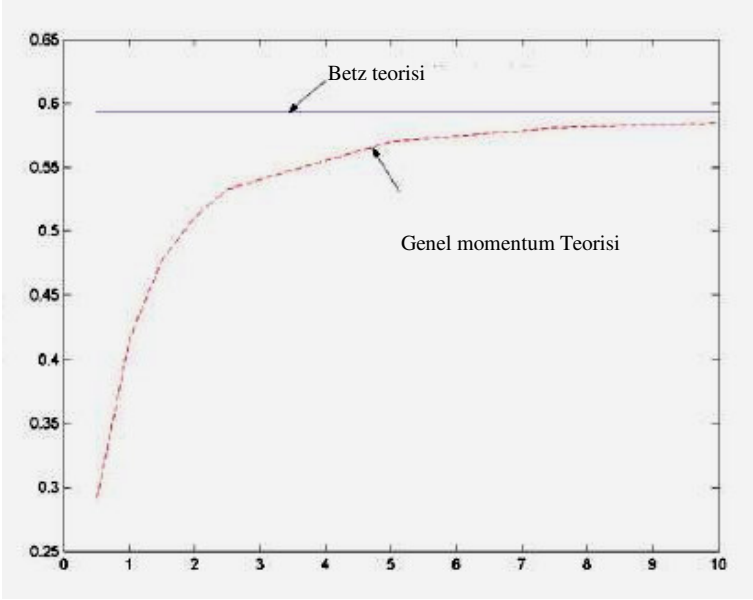
elde edilir. Eş. 4.42 nolu eşitlik kullanılarak;

$$\lambda^2 = \frac{(1-a_2)(1-4a_2)}{(1-3a_2)} \tag{4.46}$$

Eş. 4.46 nolu eşitlikte a_2=1/3 olduğunda, maksimum güç katsayısına ulaşılmış olur.

Eğer λ=5 olarak kabul edilip çözüm yapılırsa a_2=0,3324 ve c_{pmax} ise 0,57 olur.

Genel momentum teorisinin sonuçları Şekil 4.5'da verilmiştir. Grafikte, önceki bölümde analiz edilen lineer momentuma göre ideal türbin için Betz limitleri de gösterilmiştir. Sonuç olarak uç hız oranı artıkça, maksimum güç katsayısının da artığı görülmektedir [16].



Şekil 4.5. Uç hız oranına göre güç katsayısının gösterimi [7]

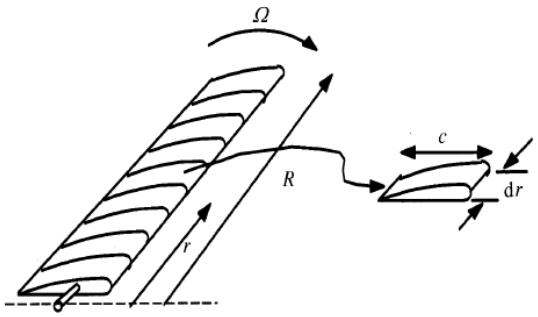
4.5. Kanat Elemanı Teorisi

Rüzgardan maksimum derecede güç çekilebilmesi için rüzgar türbinin kanat yapısının optimum düzeyde tasarlanması gereklidir. Glauert çalışmalarında Betz'den farklı olarak, rotor sonrasında oluşan girdaplı akımı da hesaba katmıştır. Böylece teori optimum kanat tasarımı da önem kazanmıştır [17]. Glauert'in teorisine göre;

- Hava akımı homojen, türbülans sız ve süreklidir,
- Rotorum gerisinde silindirik bir iz meydana gelmektedir.
- Kanatın her bir elemanı bir sonraki kanat elemanından bağımsız hareket etmektedir.
- Uygulamada kullanılacak olan profilin karakteristikleri her bir elemana adapte edilmiştir.

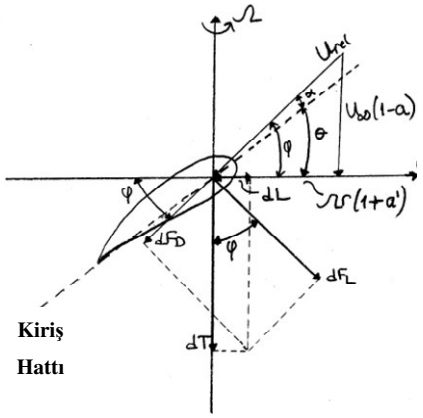
Kanat elemanı teorisi iki ana esasa dayanır. Kanat elemanına etki eden kuvvet ve momentler; kanat profilindeki kaldırma ve sürükleme kuvvetlerine bağlıdır. Her bir kanat elemanının bağımsız hareket etmesi kabulünden dolayı her bir kanat elemanı

üzerine gelen kuvvetler, bölgesel rüzgâr hızları ile hesaplanır [14]. Kanat elemanı teorisi, hava içindeki hareketinden dolayı kanatların davranışını analiz eden alternatif bir yöntemdir. Bu analizde Şekil 4.6'de gösterildiği gibi, kanadın N tane parçaya veya elemana bölündüğü kabul edilmektedir ve her bir kanat elemanı için kabul edilen aerodinamik kuvvet, kanat elemanları için adapte edilen aynı kesitin uygun kanat profilindeki kuvvetler olarak düşünülmektedir. Her bir elemanın davranışı, aynı kanadın bitişik elemanları tarafından etkilenmez, kanat ucundaki kuvvet kanat boyunca tüm elemanlarındaki kuvvetlerin birbirine eklenmesiyle bulunabilir [7]. Kanat elemanı şematik görünüşü Şekil 4.6'da verilmiştir.



Şekil 4.6. Kanat elemanlarının şematik görünüşü

Kanat kirişine etkiyen kuvvetlerin analizi Şekil 4.7'de gösterilmiştir.



Şekil 4.7. Kanat kirişi

Şekil 4.7'e dayanılarak aşağıdaki eşitlikleri elde etmek mümkündür.

$$U_{rel} = \frac{U_{\infty}(1-a)}{\sin(\varphi)} \tag{4.47}$$

$$\tan(\varphi) = \frac{U_{\infty}(1-a)}{\Omega r(1+a')} = \frac{(1-a)}{(1+a')\lambda_r} \tag{4.48}$$

$$dF_D = C_D \frac{1}{2}\rho U_{rel}^2 c dr \tag{4.49}$$

$$dF_L = C_L \frac{1}{2}\rho U_{rel}^2 c dr \tag{4.50}$$

$$dT = dF_L \cos\varphi + dF_D \sin\varphi \tag{4.51}$$

$$dL = dF_L \sin\varphi - dF_D \cos\varphi \tag{4.52}$$

Eğer rotorda B adet kanat varsa, elemanın r mesafesinde etkiyen tüm normal ve teğetsel kuvvetler Eş. 4.51, Eş. 4.52 nolu eşitlikler ile Eş. 4.47, Eş. 4.49, ve Eş. 4.50 nolu eşitlikler kullanılarak aşağıdaki şekilde yeniden düzenlenebilir:

$$dT = B\frac{1}{2}\rho U_{rel}^2 (C_L \cos\varphi + C_D \sin\varphi) c dr \tag{4.53}$$

$$dL = B\frac{1}{2}\rho U_{rel}^2 (C_L \sin\varphi - C_D \cos\varphi) c dr \tag{4.54}$$

Teğetsel kuvvetler nedeniyle oluşan moment (dL) merkezden r mesafesindeki çalışmalarda aşağıdaki eşitlikle bulunabilir:

$$dQ = r dL \tag{4.55}$$

Böylece Eş. 4.54 nolu eşitlik Eş. 4.55 nolu eşitliğe eklenerek dönme momenti aşağıdaki şekilde bulunur:

$$dQ = B\frac{1}{2}\rho U_{rel}^2 (C_L \sin\varphi - C_D \cos\varphi) c r dr \tag{4.56}$$

Şekil 4.7 deki kanada ait hız çizelgesinden, bağıl rüzgar hızı aşağıdaki gibi yazılabilir,

$$U_{rel} = \frac{U_\infty (1-a)}{\sin\varphi} \tag{4.57}$$

δ katılık katsayısı ise;

$$\delta = \frac{Bc}{2\pi r} \tag{4.58}$$

ifadesi ile tanımlanabilir ve Eş. 4.57 ve Eş. 4.58 nolu eşitlikler Eş. 4.53 ve Eş. 4.56 nolu eşitliğin içerisine eklenerek, itme kuvveti ve dönme momentinin eşitlikleri aşağıdaki genel yapıya dönüştürülebilir:

$$dT = \sigma\pi\rho \frac{U_\infty^2 (1-a)^2}{\sin^2\varphi} (C_L \cos\varphi - C_D \sin\varphi) r dr \tag{4.59}$$

$$dQ = \sigma\pi\rho \frac{U_\infty^2 (1-a)^2}{\sin^2\varphi} (C_L \sin\varphi - C_D \cos\varphi) r^2 dr \tag{4.60}$$

Eş. 4.59 ve Eş. 4.60 nolu eşitlikler; rotor bölümündeki normal itme kuvvetini ve teğetsel kuvvetleri (dönme momenti); kanatların akış açılarının ve kanat profili karakteristikleri fonksiyonu olarak tanımlanmaktadır.

Kanat elemanı teorisinin geliştirilmesi için aşağıdaki kabullenmelerin yapılması gereklidir:

a) Elemanın davranışı aynı kanattaki bitişik elemandan etkilenmez.

b) Kanat profillerinin özellikleri kanat elemanları için geçerlidir.

Kanat elemanı teorisi ve sonra geliştirilen teorilerde kabul edilen bağımsız kanat elemanları ile genel momentum teorisindeki kabullenmeler birbirine benzerdir. Bu teoride rotordaki itme kuvveti

$$dT = 2\rho u(u - U_\infty)da \tag{4.61}$$

olarak ifade edilir. Bu yaklaşımla, rotor merkezine r mesafesindeki kanat elemanındaki itme kuvveti, kendisinden sonra gelen kanat elemanın itme kuvvetinden bağımsızdır.

4.6. Kanat Elemanı Momentum Teorisi

Kanat Elemanı Momentum Teorisi oldukça yaygın geçerliliği olan teorilerden biridir. Genel momentum teorisi ve kanat elemanı teorisinin birleşiminden oluşmaktadır.

$$\frac{a}{(1-a)} = (\sigma C_L)\frac{\cos\theta}{4\sin^2\theta}\left[1 + \left(\frac{C_D}{C_L}\right)\tan\theta\right] \tag{4.62}$$

Genel momentum teorisi ve kanat elemanı teorisindeki eşitliklerin türevi alınırsa;

$$\frac{a'}{(1-a)} = \frac{(\sigma C_L)}{4\lambda_r \sin\theta}\left[1 - \left(\frac{C_D}{C_L}\right)\cot\theta\right] \tag{4.63}$$

bulunur. Hızların oluşturduğu geometriden;

$$\tan\theta = \frac{(1-a)U_\infty}{(1+a')r\Omega}$$

elde edilir ve Eş. 4.48 nolu eşitlik kullanılarak Eş. 4.63 nolu eşitliğin içinde kullanılırsa Eş. 4.64 nolu eşitlik elde edilir;

$$\frac{a'}{(1+a')}=\frac{(\sigma C_L)}{4\cos\theta}\left[1-\left(\frac{C_D}{C_L}\right)\cot\theta\right] \tag{4.64}$$

eşitliği bulunur. Daha önceki bölümde de ifade edildiği gibi kanat profili performansının iyi olabilmesi için kaldırma katsayısının yüksek, sürükleme katsayısının düşük olması gerekmektedir. Dolayısıyla düşük C_D/C_L oranı ihmal edilirse;

$$\frac{a}{1-a}=(\sigma C_L)\frac{\cos\theta}{4\sin^2\theta} \tag{4.65}$$

eşitliği elde edilir. Bu eşitlik aşağıdaki gibi yeniden düzenlenebilir.

$$\frac{a'}{1+a'}=\frac{(\sigma C_L)}{4\lambda_r\sin\theta}$$

$$\frac{a'}{(1-a)}=\frac{(\sigma C_L)}{4\lambda_r\sin\varphi} \tag{4.66}$$

Türbin kanat tasarımı için kullanılan diğer bir ifade ise uç hız oranıdır. Rotor yarıçapı R ise, uç hız oranı;

$$\lambda=\frac{\Omega R}{U_\infty} \tag{4.67}$$

yazılabilir. Kanat tasarımındaki her bir parça birbirinden bağımsız hareket ettiği için yerel uç hız oranı ise;

$$\lambda_r=\lambda\frac{r}{R}=\frac{\Omega r}{U_\infty} \tag{4.68}$$

olur. Eş. 4.65 ve Eş. 4.66 nolu eşitlikler kullanılarak aşağıdaki eşitliklerin elde edilmesi mümkündür.

$$c_L = \frac{4\sin\theta}{\sigma}\frac{\left(\cos\theta - \lambda_r \sin\theta\right)}{\left(\sin\theta + \lambda_r \cos\theta\right)} \tag{4.69}$$

$$a = \frac{1}{\left[1+\left[\frac{4\sin^2\theta}{\left(\sigma C_L\right)\cos\theta}\right]\right]} \tag{4.70}$$

$$a' = \frac{1}{\left[\left[\frac{4\cos\theta}{\left(\sigma C_L\right)}\right]-1\right]} \tag{4.71}$$

$$\frac{a}{a'} = \frac{\lambda_r}{\tan\theta} \tag{4.72}$$

Rüzgâr türbininin güç katsayısını belirlemek için, her yarıçaptaki güç katkısının bulunması ve kanat boyuna göre integralinin alınması gerekir Aşağıdaki eşitlik kullanılarak güç katsayısının c_p bulunması mümkündür:

$$c_p = \frac{P}{\frac{1}{2}\rho U_\infty^3 \pi R^2} \tag{4.73}$$

Her bir kanat elemanının gücü, aşağıdaki eşitlikle tanımlanır;

$$dP = \Omega dQ \tag{4.74}$$

ve rotordaki toplam güç de aşağıdaki eşitlikle ifade edilirse:

$$P = \int_{r_h}^{R} dP = \int_{r_h}^{R} \Omega dQ \tag{4.75}$$

Eş. 4.73 nolu eşitlik ve Eş. 4.75 nolu eşitlik aşağıdaki gibi tekrar yazılabilir:

$$c_p = \frac{P}{\frac{1}{2}\rho U_\infty^3 A} = \frac{\int_{r_h}^{R} \Omega dQ}{\frac{1}{2}\rho U_\infty^3 \pi R^2} \tag{4.76}$$

Elemental tork ifadesi kullanılarak güç katsayısı;

$$c_p = \frac{2}{\lambda^2}\int \sigma C_L \frac{(1-a)^2}{\sin\theta}\left[1-\left(\frac{C_D}{C_L}\right)\cot\theta\right]\lambda_r^2 d\lambda_r \tag{4.77}$$

eşitliği ile belirlenir. Eş. 4.65 ve Eş. 4.72 nolu eşitlikler Eş. 4.77 nolu eşitlikte kullanıldığında, güç katsayısının genel formu aşağıdaki gibi ifade edilebilir:

$$c_p = \frac{8}{\lambda^2}\int_{\lambda_h}^{\lambda} \lambda_r^3 a'(1-a)\left[1-\left(\frac{C_D}{C_L}\right)\cot\theta\right] d\lambda_r \tag{4.78}$$

$C_D \approx 0$ olduğu zaman, c_p genel momentum eşitliğindeki halini alır. Daha değişik bir ifade ile Eş. 4.69 ve Eş. 4.70 nolu eşitlikler Eş. 4.77 nolu eşitliğin içinde kullanıldığında;

$$c_p = \frac{8}{\lambda^2}\int_{\lambda_h}^{\lambda} \sin^2\theta(\cos\theta - \lambda_r \sin\theta)(\sin\theta + \lambda_r \cos\theta)\left[1-\left(\frac{C_D}{C_L}\right)\cot\theta\right]\lambda_r^2 d\lambda_r \tag{4.79}$$

elde edilir.

5. KANAT TASARIMI

5.1. Genel

Kanat Elemanı Momentum Teorisini kullanarak kanat kirişi hesaplaması günümüzde en yaygın olarak kullanılan yöntemdir. Söz konusu yöntemde tahmini bir kanat kiriş değeri belirlenerek Reynolds sayısı hesaplanmakta ve Reynolds sayısına göre kanadın kaldırma ve sürükleme katsayısı belirlenerek maksimum güç bulunmaktadır [18].

Jung et al. [19] 30 kW'lık iki rotora sahip olan bir türbinin boyutlandırmasında Glauert momentum teorisinden faydalanmışlardır. Badreddinne [15] ise kaldırma çizgisi teorisini esas alarak girdap teorisi ile boyutlandırma yapmış, bulduğu sonuçları kanat elemanı momentum teorisi ile karşılaştırmıştır. Daha önceki bölümde bahsedilen kanat elemanı momentum teorisi, rotordaki kanat sayısının performansa olan etkisini tam olarak belirleyememektedir. Bu nedenle performans analizinde kanatlardaki uç kayıplarının da dikkate alınması daha uygun olmaktadır.

5.2. Uç Kayıpları

Daha önce bahsedilen varsayımlarda rotor kanadının sonsuz sayıda olduğundan hareket edilmiştir. Bu varsayım yardımıyla rotor düzleminden ve iz bölgesinden geçen akışın radyal hızı ihmal edilerek eşitlikler türetildi. Ancak, akış hattının sınırında, kanat uçlarından dolaşan havanın hızı da önemlidir. Çünkü kanadın emme yüzeyindeki basıncı, basınç yüzeyindekinden düşüktür. Dolayısıyla kanat ucuna yaklaştıkça kaldırma kuvveti ve güç rüzgârın uç kısımlardan kaybolmasından dolayı azalmaktadır. Söz konusu radyal akışın ve uç kayıplarının etkilerini tahmin etmek için Prandtl tarafından bir model önerilmiştir [7, 14, 16].

Prandtl, bu yöntemde türbin arkasında oluşan helisel girdap tabakalarının, dış akıma göre eksenel hıza eşit bir hızda, katı düzlemler şeklinde hareket ettiğini belirtmiştir.

Hesap kolaylığı açısından bu düzlemleri, aralarında *d'* kadar mesafe bulunan paralel ince levhalar olarak incelemiştir [17].

Levhaların ucuna doğru iki levha arasındaki potansiyel farkın azalma miktarının, rotor palasının çevresindeki sirkülâsyonunun azalma miktarına eşit olduğu varsayılarak *F* uç düzeltme katsayısı, aşağıdaki gibi belirlenmiştir [20]:

$$F = \left(\frac{2}{\pi}\right)\cos^{-1}\left[\exp\left[\frac{-\left(\frac{B}{2}\right)\left[1-\left(\frac{r}{R}\right)\right]}{\left(\frac{r}{R}\right)\sin\theta}\right]\right] \tag{5.1}$$

İşlem kolaylığı açısından F aşağıda belirtilen koşullarda 0 veya 1 olarak alınır[7, 11],

$0<r<r_e$ ise F=1,

$r_e \leq r \leq R$ ise F=0,

$r_e \approx 0{,}97$ R

5.3. Kanat Seçimi

Kanatlar, türbinlerde mekanik gücün geliştirilmesi amacıyla kullanılan elemanlardır. Kanadın kiriş ve uzunluğu aerodinamik performansını etkiler. Dolayısıyla kanat profillerinin özellikleri yapılacak olan dizaynı etkiler. Kanat seçimindeki en önemli etken Reynolds sayısıdır. Çünkü en uygun hücum açısını (α) bulmak performans belirlemesi için gerekli bir husustur. Kaldırma ve sürükleme katsayısı hem kanat kesitine hem de Reynolds sayısına bağlıdır. Modern türbinlerde kullanılan kanatların kiriş ölçüleri MegaWatt büyüklüğündeki türbinlerde 2 m'ye kadar ulaşmaktadır. Kanatların uç hız oranları (λ) ise 0,5 den 10'a kadar değişmektedir. Dolayısıyla Reynolds sayısı da 0.3 milyondan ile 10 milyon arasında değişmektedir. Reynolds sayısını bulmak için öncellikle görünür rüzgâr hızının bulunması gerekir.

$$V_a = \frac{\left(\frac{r}{R}\right)\lambda U_\infty}{\cos\varphi} \tag{5.2}$$

Uç hız oranı (λ) üç kanatlı rüzgâr türbinleri için yaklaşık 5 olarak alınabilir. Rüzgârın rotor düzlemine çarptığı açı φ ise, aşağıdaki gibi hesaplanır:

$$\varphi = \left(\frac{2}{3}\right)\tan^{-1}\left(\frac{R}{r\lambda}\right) \tag{5.3}$$

olur. Reynolds sayısının hesaplanabilmesi için tahmini bir kiriş uzunluğunun bilinmesi gereklidir. Kiriş mesafesi, *c*;

$$c = \frac{16\pi R\left(\frac{R}{r}\right)}{9\lambda^2 B} \tag{5.4}$$

eşitliği ile elde edilir. Bulunan bu değerler Eş. 5.5 nolu eşitlikte yerine konulduğunda Reynolds sayısı elde edilmiş olur,

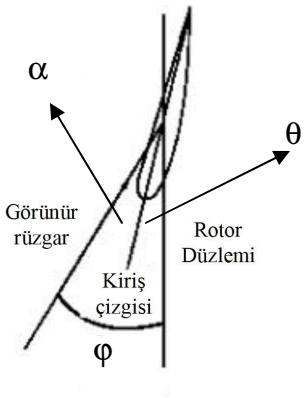
$$\mathrm{Re} = 68500 \times c \times V_a \tag{5.5}$$

Yazılan programda ise Reynolds sayısının hesabında kanadın orta noktası esas alınmıştır. Hesaplanan Reynolds sayısına bağlı olarak belirlenen kanat tiplerinin sürükleme-kaldırma katsayılarına bakılarak en uygun kanat seçimi yapılır;

$$\gamma = \frac{C_D}{C_L} \tag{5.6}$$

5.4. Kanada İlişkin Hesaplamalar

Kanat şeklinin tasarımı, rotordan maksimum güç elde edilecek şekilde bilinen parametrelerle yapılır. Kanat uzunluğu boyunca kiriş-uzunluk (c-R) ve kanat açısı (θ) hesaplanır. Bu nedenle hesaplamalara başlamadan önce uç hız oranına karar verilmelidir. Uç hız oranına bağlı olarak, maksimum güç katsayısı (c_p) ise genel momentum teorisinde uç hız oranı λ=5 kabul edilerek 0,57 olarak belirlenir. Kanat profili seçilmeden önce kanadın verimine sürükleme/kaldırma oranına bağlı olarak kanat verimi hakkında yaklaşımda bulunulması gerekir. Şekil 5.1 de ise kanat düzleminde oluşan açılar gösterilmiştir.



Şekil 5.1. Kanat açıları

Rüzgâr açısı ve bölgesel uç hız oranı arasındaki ilişkiden dolayı,

$$\varphi_{opt} = \left(\frac{2}{3}\right)\tan^{-1}\left(\frac{1}{\lambda_r}\right) \tag{5.7}$$

olur. Optimum rüzgâr açısı için bölgesel uç-hız oranının bulunması gerekir. Kanat elemanı momentum teorisi ile birlikte uç-kayıp katsayısı hesaba katılırsa seçilen kanat için rotordan maksimum güç elde edecek şekilde boyutlandırma yapılır. Boyutlandırma işlemi verilen kanat boyunun N parçaya bölünerek her bir istasyon için ayrı ayrı hesaplama yapılarak gerçekleştirilir. Kanat elemanı için bölgesel uç kayıp katsayısı,

$$\lambda_{r,i} = \lambda\left(\frac{r_i}{R}\right) \tag{5.8}$$

olur. Eş. 5.7 nolu eşitliğe göre her bir kanat elemanı için optimum rüzgar açısı,

$$\varphi_{opt,i} = \left(\frac{2}{3}\right)\tan^{-1}\left(\frac{1}{\lambda_{r,i}}\right) \tag{5.9}$$

olarak hesaplanır. Eş. 5.10 nolu eşitlikten her bir kanat elemanı için uç kayıp faktörü,

$$F_i = \left(\frac{2}{\pi}\right)\cos^{-1}\left[\exp\left[\frac{-\left(\frac{B}{2}\right)\left[1-\left(\frac{r_i}{R}\right)\right]}{\left(\frac{r_i}{R}\right)\sin\varphi_{opt,i}}\right]\right] \tag{5.10}$$

bulunur. Kanat boyunca kiriş uzunluğunun dağılımı, her bir eleman için,

$$c_i = \frac{8\pi r_i F_i \sin\varphi_{opt,i}}{BC_L}\frac{\left(\cos\varphi_{opt,i} - \lambda_{r,i}\sin\varphi_{opt,i}\right)}{\left(\sin\varphi_{opt,i} + \lambda_{r,i}\cos\varphi_{opt,i}\right)} \tag{5.11}$$

olur. Kanat açısı ise,

$$\theta_i = \varphi_{opt,i} - \alpha \tag{5.12}$$

eşitliği ile belirlenir. Eş. 5.13 nolu eşitlik kullanılarak türbinin üreteceği tahmini güç bulunabilir:

$$Ne = 0,6 \times C_p \times N \times U_\infty^3 \tag{5.13}$$

eşitlikte jeneratör verimi (N) 0,8 alınmıştır.

6. KANAT TASARIMI GELİŞTİRİLEN BİLGİSAYAR PROGRAMI

6.1. Genel

Bu çalışma kapsamında; önceki bölümlerde belirtilen kanat tasarım parametreleri ve kanat elemanı momentum teorisi çerçevesinde yatay eksenli rüzgâr türbini kanat tasarımını bilgisayar destekli olarak gerçekleştirecek bir bilgisayar programı gerçekleştirilmiştir. Geliştirilen program sayesinde kanadın kiriş uzunluğu, kanat açısı ve elde edilecek tahmini güç hesaplanmış olacaktır. Kanat kiriş mesafesi ve kanat açısı ne kadar küçük aralıklarla hesaplanırsa elde edilen sonuç o kadar gerçekçi olacaktır. Microsoft Visual C#.NET 2.0 tabanlı olarak Kanat Simülasyonu programı kodlanmıştır. Programın geliştirilmesinde söz konusu yazılımın kullanılmasının nedenleri şunlardır;

a) XML tabanlı bir yapıya sahiptir (metin tabanlı olduğu için kullanımı kolaydır).

b) Oluşturulan veritabanıyla kolay iletişim sağlamaktadır.

c) Program geliştirme sırasında kolaylık sağlamakta ve içerdiği nesneler yardımıyla fazla kod yazmaya gerek duyulmamaktadır.

d) Programın derlenmesi (çalıştırılması) oldukça hızlıdır.

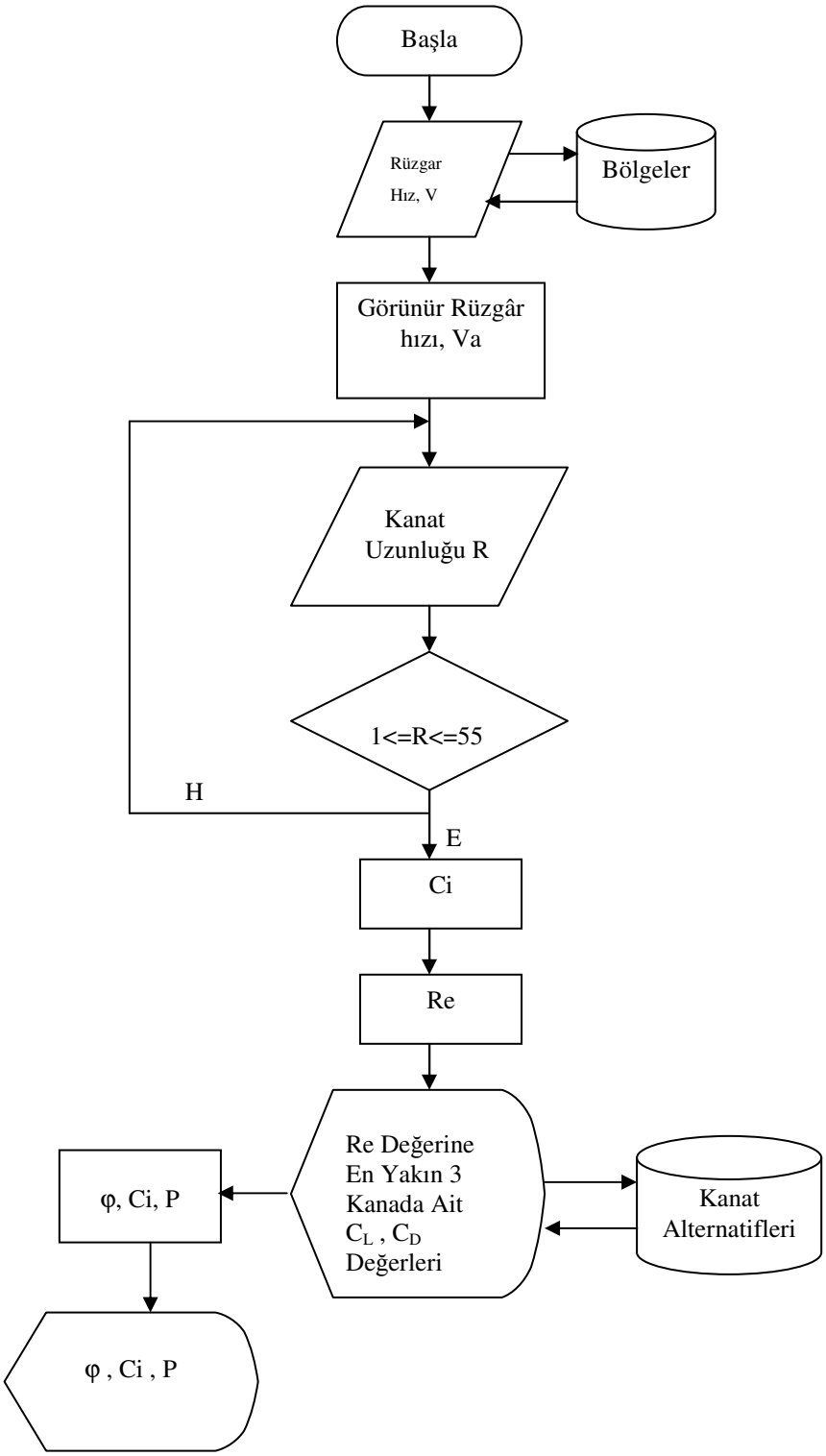
e) Veritabanı ile bağlantı sırasında gerekli olan kütüphane dosyaları programın içerisinde yer aldığından dolayı kullanım sırasında farklı bir veritabanı bağlantı yazılımına gerek duyulmamaktadır.

f) Grafik yazımı için ek nesnelere gereksinim duyulmamaktadır. Yazılımın kendi içindeki nesneler bu husus için yeterlidir.

Geliştirilen program ile değişik kanat tiplerinin tasarım şekillerinin karşılaştırılabilmesi mümkün olacaktır. Öncellikle program için geliştirilen algoritmadan bahsedilecek ve daha sonra programa ilişkin yapılan örnek bir hesaplama açıklanacaktır.

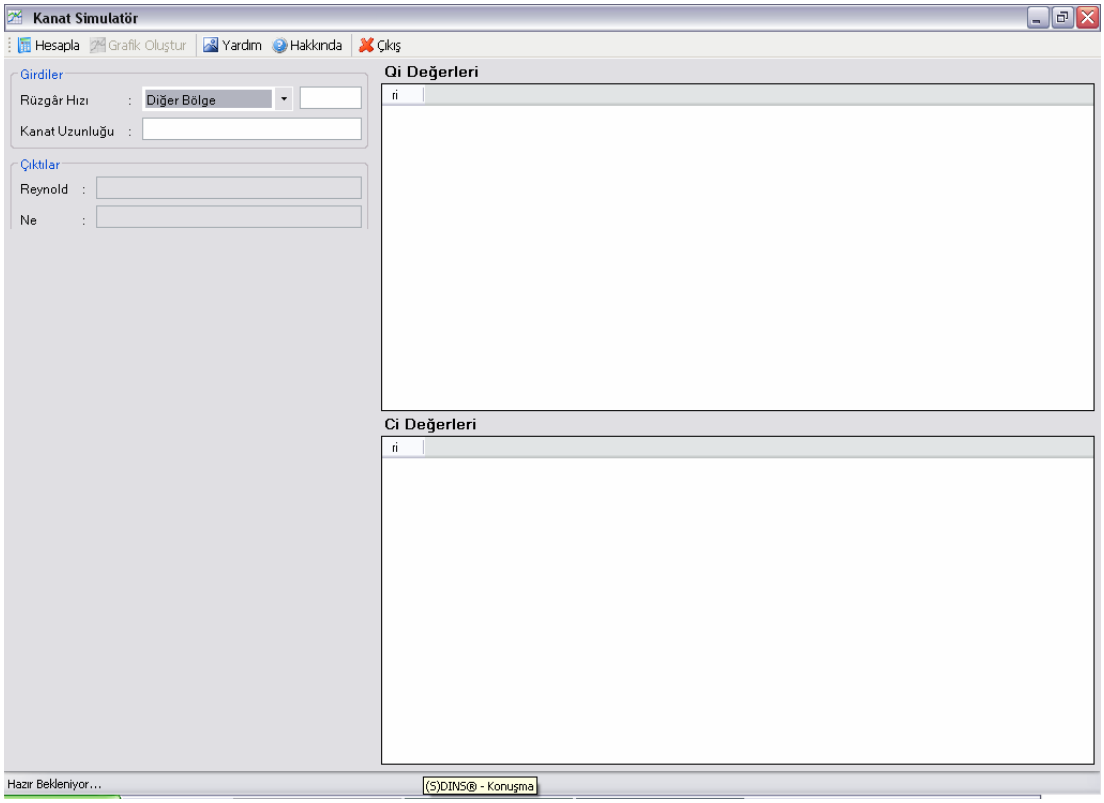
6.2. Kanat Simülasyonu

Gerçekleştirilen bilgisayar programının temel amacı Türkiye'nin herhangi bir bölgesinde veya rüzgâr hızı bilinen bir yerde kurulması istenilen rüzgâr türbinin kanat uzunluğuna bağlı olarak gerekli boyutlandırmasının yapılmasıdır. Yapılan boyutlandırmanın programın önereceği kanat tipleriyle karşılaştırılması da mümkün olabilecektir. Programın akış diyagramı Şekil 6.1'de verilmiştir.



Şekil 6.1 Kanat simülatörü programın akış şeması

Programın ara yüzü Şekil 6.2'de gösterilmiştir.



Şekil 6.2. Programın ara yüzü

Program hakkındaki genel bilgiler aşağıda özetlenmiştir:

a) Program Microsoft Visual C#.NET 2.0 da kodlanmıştır.

b) Programın iki girdisi, dört çıktısı bulunmaktadır.

c) Programın girdisi olarak bölge veya rüzgâr hızı ve istenen kanat boyu, çıktı olarak kanat kiriş uzunluğu, kanat açısı, üretilmesi planlanan güç ile Reynolds sayısıdır.

d) Ayrıca programda hesaplatılan kanat kiriş değerlerini grafik olarak elde etmek de mümkün olacaktır.

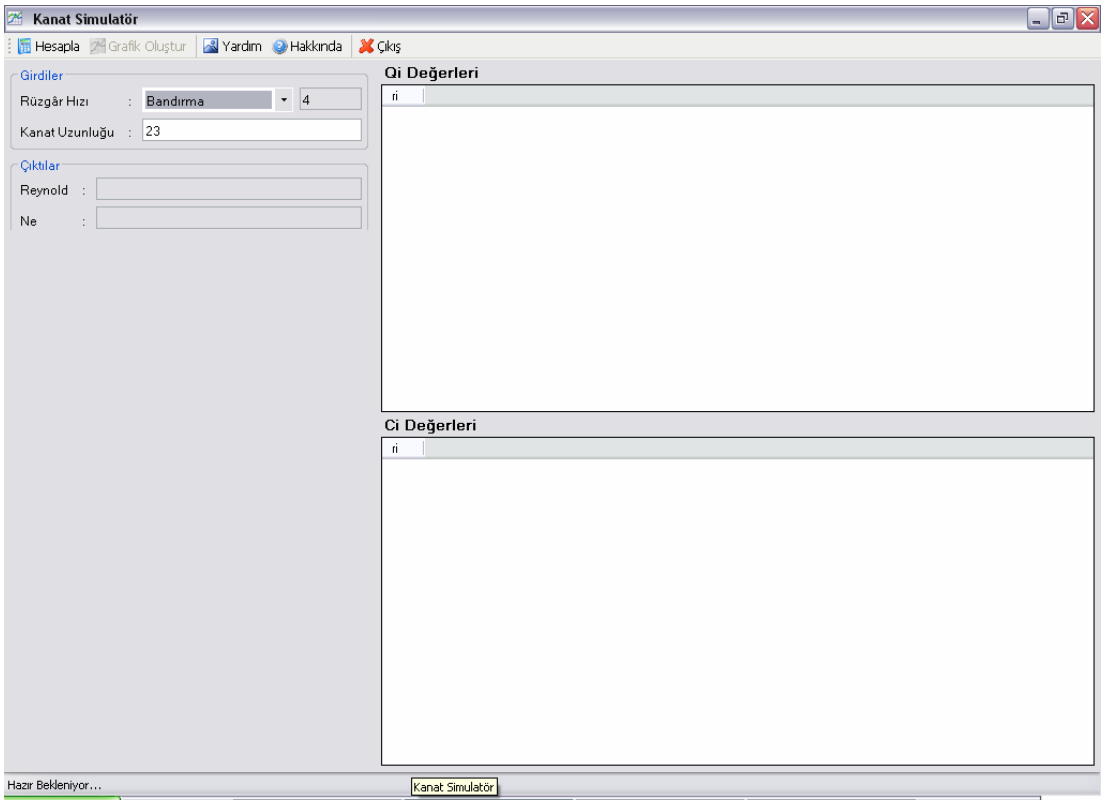
e) Kanadın kiriş mesafesi ve oturma açısı 0,01 m aralıklarla hesaplanacaktır.

f) Program istediğimiz giriş değerlerine göre hesaplama yaparak bulduğu Reynolds sayısına yakın olan üç kanat tipinin üçünü, ikisini veya birini seçme imkânı vermektedir.

g) Eğer istenirse seçilen kanat tiplerinin veya tipinin grafiğini verebilir.

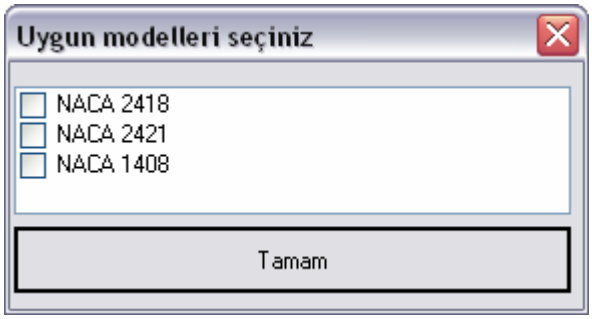
6.3. Kanat Simülatörü Programıyla Kanat Tasarımı

Şekil 6.3'de görüldüğü üzere; programa giriş verileri olarak rüzgâr hızı ve istenen kanat uzunluğu değerleri girilir. Rüzgâr hızı için istenirse programın bilgi bankasında kayıtlı olan bölgelerden biri seçilebilir veya doğrudan sayısal değer yazılabilir. Programın bilgi bankasında bulunan bölgeler Çizelge 1.4 de verilmiştir. Programda girilebilecek kanat uzunluğu ise 1 ile 55 metre arasında sınırlandırılmıştır. Bunun nedeni ticari olarak yapılan rotorların çaplarının yaklaşık 40 ila 65 metre olmasıdır. Ufak çapta yapılanlar ise 2 ila 9 metre arasında değişmektedir.



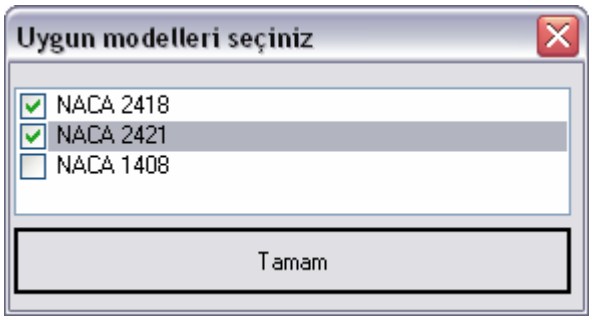
Şekil 6.3. Programın girdileri

Şekil 6.3'de örnek giriş verileri yazılmıştır. Şekilde rüzgar türbini planlaması yapılan yer olarak Bandırma seçilmiştir. Kanat boyunun ise 23 metre olması istenmiştir. Program yaygın olarak kullanımından dolayı verilen girdi koşullarında 3 kanatlı türbin tasarımı için simülasyon yapılacaktır. Üç kanatlı türbinler için ideal olan uç hız oranında kanadın boyutlandırılması yapılır. Sonuç olarak rotorun çapı hesaba katılarak güç hesabı yapılır. Programdaki hesapla butonuna basıldığında program akış diyagramında gösterildiği gibi hesaplanan Reynolds sayısına göre 3 adet kanat tipi önermektedir (Şekil 6.4). Programın veri tabanına girilen ve piyasada yaygın kullanım alanı bulan kanat modelleri ile söz konusu modellere ilişkin hesaplanan Reynolds sayısı değerleri Çizelge 6.1'de verilmiştir.



Şekil 6.4. Önerilen kanat tipleri

Programın önerdiği 3 kanat tipinden istenilenler seçilerek (Şekil 6.5) ve "Tamam" butonuna basılarak kanat boyunca kirişi 0,01m aralıklarla kiriş genişliği ve kanat açısı değerleri hesaplanmaktadır (Şekil 6.6).

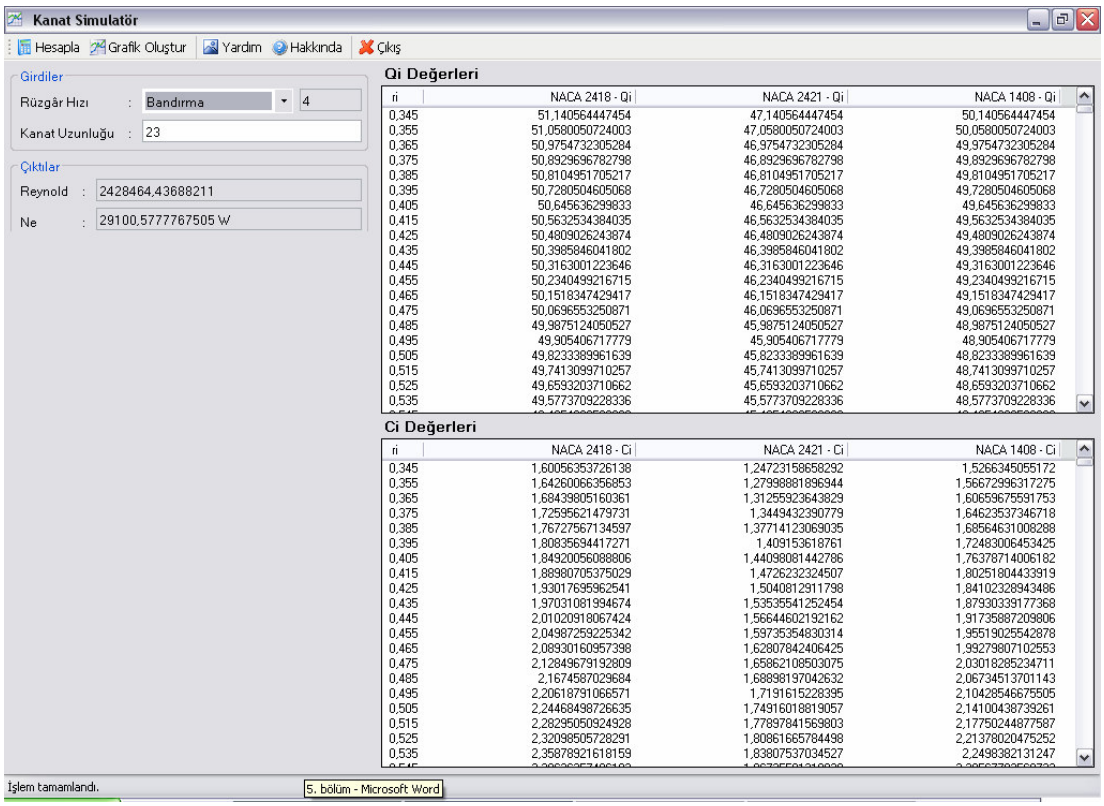


Şekil 6.5. Seçilen kanat tipleri

Çizelge 6.1. Programın veri tabanında kullanılmak üzere seçilen kanat modelleri ve bu modele ilişkin hesaplanan Reynolds sayısı değerleri

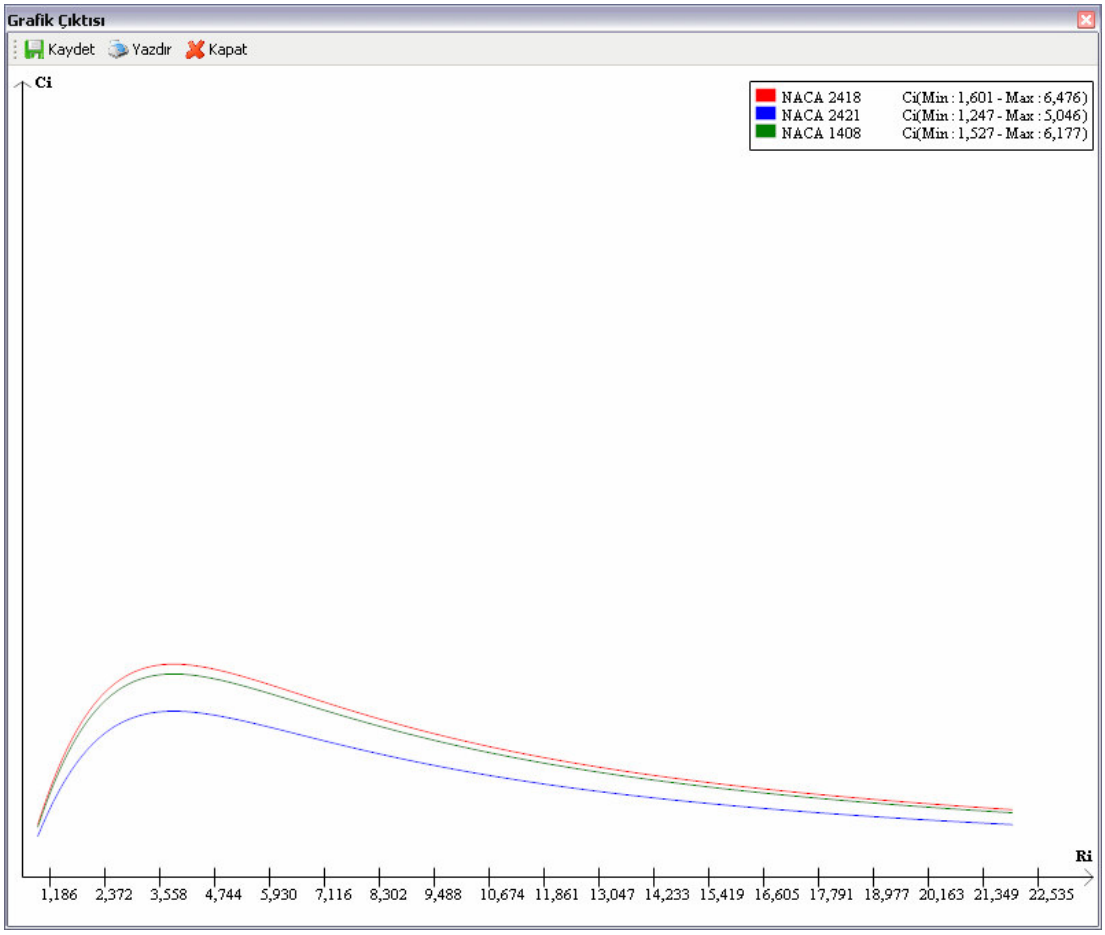
Kanatlar	Reynolds sayıları
A18 SM	203000, 300000, 302600
AQUILA SM	203900, 301100
GOE 417 A	297500, 398800
ESA 40	200600, 301600, 398100, 403800
DAVIS SM	198600, 297900
J 5012	201900, 303100
CLARK Y	203800, 301200
FALCON 56 MK 2	200000, 301300, 401700
K 3311 SM	201800, 302900
R 140 SM	203000, 199900, 300300
PT 40	200100, 299800, 399100
SPICA SM	202300, 301500
USNPS 4	199300, 299000, 398300
TRANIER 60	200000, 301800, 400000
WASP SM	201600, 302600
WB 140	200500, 307600
WB 13535 SM	204000, 302700
CLARK YSM	200000, 299800, 399900
NACA 2414	200500
NACA 2414	301000
NACA 2415	200600
NACA 2415	301100
NACA 25411	200600
NACA 25411	303300
NACA 6409	200100
NACA 2418	2900000
NACA 2412	3100000
NACA 2421	2900000
NACA 1408	3000000
NACA 4412	3000000

Programda kayıtlı olan bölge ve kanat tipleri dışında veri eklenmesi istenirse programın yardım menüsünde söz konusu verilerin nasıl ekleneceğine ilişkin açıklamalar yapılmıştır.



Şekil 6.6. Programın çıktı görüntüsü

Program, kanat kiriş mesafesini ve kanat açısını kanat elemanının başlangıcındaki %2'lik kısmında ve kanat sonundaki %2'lik kısımda hesaplamamaktadır. Bunun nedeni, kanadın rotora oturması için gerekli işlem kolaylığının sağlanması ve kanat uç kısmının yuvarlatılmasıdır. Üç ya da iki kanat tipi seçilerek aralarındaki karşılaştırma yapılması mümkündür. Ayrıca görüntü olarak karşılaştırma yapılmak istenirse "grafik oluştur" butonuna basılarak kanat grafiklerinin elde edilmesi de mümkündür (Şekil 6.7).



Şekil 6.7 Kanat uzunluğu boyunca kiriş genişliğinin değişimini gösteren grafik çıktısı

Ekran çıktısında seçilen her bir kanadın kendine ait ayrı bir rengi bulunmakta ve renklerin hangi modellere ait olduğu ekranın sağ üst köşesinde gösterilmektedir. Böylece kanatları birbirinden ayırt etmek mümkün olmaktadır. Ekran çıktısı sayesinde seçilen kanadın nasıl bir şekle sahip olduğu kolaylıkla görülebilmektedir.

7. SONUÇ VE ÖNERİLER

Bu çalışmada, yatay eksenli rüzgâr türbinlerinin kanat tasarımındaki aerodinamik esaslar tanıtılmış ve *kanat simülatörü* isimli paket bir program hazırlanarak 3 kanatlı yatay eksenli rüzgâr türbinlerinin tasarım parametrelerinin belirlenmesi amaçlanmıştır.

Rüzgâr türbinlerindeki kanatlar elektrik üretimi için önemlidir. Kanatlar rüzgârdaki enerjiyi alan parçalardır. Dolayısıyla rüzgâr ila kanat arasındaki aerodinamik kuvvetler tasarım için oldukça önemlidir. Rüzgâr türbininden üretilecek enerji aerodinamik koşullar ile sınırlıdır. Kanat tasarımı ile ilgili bugüne kadar pek çok teori geliştirilmiştir. Öncelikle ideal disk teorisinden bahsedilmiştir. Burada türbin veriminin ideal koşullarda ancak % 59 olabileceği dolayısıyla da ideal disk teorisinin temel bir model olmasına rağmen rüzgâr türbinlerinin ideal tasarımında gerçekçi sonuçlar vermediği belirtilmiştir.

Daha sonra, genel momentum teorisinde rotordaki ve rotor gerisindeki hızların kanat tasarımına olan etkileri analiz edilmiştir. Ancak, genel momentum teorisinin kanat geometrisi açısından uygun sonuçlar vermediği belirtilmiştir.

Kanat elemanı teorisinde ise kanat geometrisindeki katsayılar kullanılarak rotorun dönme momenti ve itme kuvveti analiz edilmiştir. Söz konusu teori ile rotorun performansı ve rotor geometrisi arasındaki ilişkinin desteklendiği analiz edilmiştir.

Sonuç olarak yatay eksenli rüzgâr türbininin kanat tasarımı için genel momentum teorisinin ve kanat elemanı teorisinin birleştirilmesi gerektiği; bu teoriye ise kanat elemanı momentum teorisi adı verildiği belirtilmiştir. Sonuçları doğrusal hale dönüştürmek için uç düzeltme metodu (Prandtl'ın uç kayıp faktörü) da hesaba katılmıştır. Böylece rotordaki kanat sayısının etkisi hesaba dâhil edilmiştir.

Ayrıca değişik rüzgâr hızları için çeşitli kanat tipleri verilerek bölgeye uygun kanat tipinin belirlenmesi sağlanmıştır. Kanat tipinin kolay seçilebilmesi için kanat simülatörü programı kodlanmıştır. Programda rüzgâr hızı ve kanat uzunluğunun girilmesi yeterlidir. Daha sonra program, en uygun 3 kanat tipinden birisini kullanıcıya seçtirme imkânı sağlamakta ve seçilen kanat tipine göre gerekli boyutlandırmayı yapmaktadır. İstenildiğinde sonuçları da grafiksel olarak da elde edebilmektedir.

Eğer kullanıcılar bu programdan daha fazla oranda yararlanmak isterlerse veri tabanındaki kanat profili tiplerini artırabilmektedirler, Bunun için gerekli açıklamalar yardım bölümü altında belirtilmiştir. Böylelikle istenilen kanat tiplerinin karşılaştırmasını yapabilmek mümkün olmaktadır.

Bilindiği üzere, kanat profillerinin kaldırma ve sürükleme davranışları rüzgâr tünellerinde kanat hareketsiz haldeyken ölçülmektedir. Ancak gerçek koşullarda, rüzgâr türbinlerindeki kanatlar hareket halindeyken eğilme meydana gelmektedir. İleride yapılacak çalışmalarda söz konusu eğilmeler de dikkate alınarak programın geliştirilmesi ve uygulamaya daha yakın sonuçlar vermesi sağlanabilir.

Kanat elemanı momentum teorisi bir yaklaşımı yansıtmaktadır. Değişik şartlar altında kanat tasarımı yapmak için yeni yaklaşımlar geliştirilebilir ve söz konusu yeni yaklaşımlar ışığında benzer paket programlar hazırlanarak karşılaştırma yapılabilir.

Bu çalışmada, rüzgâr hızının değişmediği kabul edilmiştir. Ancak normal şartlarda rüzgâr hızı değişkendir. Ani rüzgâr hızı değişimleri kanatlarda zorlamalara dolayısıyla hesaplanan kanat boyutlarının ilgili bölge için geçerli olabilecek maksimum rüzgâr hızları da kontrol edilmesi faydalı olacaktır.

KAYNAKLAR

1. Özerdem, B., "Türkiye'de Rüzgar Enerjisi Uygulamalarının Gelişimi ve Geleceği", ***Mühendis ve Makine***, 526: 10-25 (2003).

2. Daradeli, S., "Rüzgar Enerjisi", ***Temiz Enerji Vakıfı,*** Ankara, 5-22 (2001).

3. Demirtola , M., "Yenilenebilir Enerji Kaynakları", ***Enerji Dünyası***, 42-43: 7-22 (2002).

4. Köse, F., Özgören, M., "Rüzgar Enerjisi Potansiyeli Ölçümü ve Rüzgar Türbini Seçimi", ***Mühendis ve Makine***, 551:20-30(2005).

5. Çelik, B., "Yatay Eksenli Rüzgar Türbini Tasarım ve Prototip İmalatı", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü,*** Ankara, 9-47(2001).

6. Kenisarin, M., Karslı, V. M., Çağlar, M., "Wind power engineering in the world and perspectives of its development in Turkey", ***Renewable&Sustainable Energy Reviews***, 10: 341-369 (2006) .

7. Duran, S., "Computer-Aided Design of Horizontal-Axis Wind Turbine Blades", Yüksek Lisans Tezi, ***Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 26-66 (2005).

8. Atılgan, İ., "Türkiye'nin Enerji Potansiyeline Bakış", ***Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi***, 15(1):31-47 (2000).

9. Dündar, C., Canbaz, M., Akgün, N., Ural, G., "Türkiye Rüzgar Atlası", ***EİE&DMİ***, 27-205 (2002).

10. Özgener, Ö., "Rüzgar Gülü Kanadı ve Güç Eldesi Üzerine Bir Araştırma", Yüksek Lisans Tezi, ***Ege Üniversitesi Fen Bilimleri Enstitüsü***, İzmir, 12-36 (2002).

11. Walker, J., F., Jenkins, N., "Wind Energy Technology", ***John Wiley & Sons***, New York, 17-74 (1997).

12. Durak, M., "Rüzgar Enerjisi Teknolojisi ve Türkiye Uygulaması: Akhisar Rüzgar Elektrik Santrali", ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 26-31 (2000).

13. Yücel, B., "Performance Prediction of Horizontal Axis Wind Turbines Using Vortex Teory", Yüksek Lisans Tezi***, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 5-14 (2004).

14. Onat, C., "Rüzgar Türbini Pervanesi Dizaynı", Yüksek Lisans Tezi, ***İnönü Üniversitesi Fen Bilimleri Enstitüsü***, Malatya, 61-98 (2001).

15. Badredinne, K., Ali, H., David, A., "Optimum project for horizontal axis wind turbines 'OPHWT'", ***Renewable Energy***, 30, 2019-2043 (2005).

16. Çiçek, A., "Yatay Eksenli Rüzgar Türbini Dizayn ve İmalatı", Yüksek Lisans Tezi, ***Niğde Üniversitesi Fen Bilimleri Enstitüsü***, Niğde, 22-35 (2002).

17. Wilson, R., E., "Aerodynamic Behavior of Wind Turbines", Wind turbine Technology Fundamental Concepts of Wind Turbine Engineering, Spera, D., A., ***Asme Pres***, New York, 215-282 (1998).

18. Mejia, J., M., Chejne, F., Smith, R., Rodriguez, L., F., Fernandez, O., Dyner, I., "Simulation of wind energy output at Guajira, Colombia", ***Renewable Energy***, 31, 383-399 (2006).

19. Jung, S., N., No, T., Ryu, K., "Aerodynamic performance prediction of a 30 kW counter-rotating win turbine system", ***Renewable Energy***, 30: 631-644 (2005).

20. Manwell, J., F., McGowan, J., G., Rogers., A., L., "Wind Energy Explained: Theory, Design and Application", ***John Wiley&Sons Ltd***., London, 83-137 (2002).

EKLER

EK-1 Kanat simülatörü programının kodları

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities;
namespace SBB.External.WingSimulator.Windows.Business.XmlHelper
{    public sealed class StationsXmlHelper
   {        private XmlDocument document;
      public StationsXmlHelper(string documentName)
      {            document = new XmlDocument();
         document.Load(documentName);
      }
      public StationEntityList GetAllStations()
      {            StationEntityList retList = new StationEntityList();
         XmlNodeList stationList =
this.document.SelectNodes("/StationsData/Stations");
         foreach (XmlNode node in stationList)
         {               retList.Add(new StationEntity(node));
         }
         return retList;
      }
   }
}
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities;
namespace SBB.External.WingSimulator.Windows.Business.XmlHelper
{    public sealed class WingsXmlHelper
   {        private XmlDocument document;
      private WingEntityList wingList;
      public WingsXmlHelper(string documentName)
      {            document = null;
         wingList = null;
         document = new XmlDocument();
         document.Load(documentName);
      }
      public WingEntityList GetWings()
      {            WingEntityList wingEntityList = new WingEntityList();
         XmlNodeList wingDataList = document.SelectNodes("/WingDatas/Wings");
foreach (XmlNode node in wingDataList)
            wingEntityList.Add(new WingEntity(node));
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
 return wingEntityList;
        }
        public WingEntityList SearchNearestWings(double reynold)
        {           if (wingList == null)
              wingList = GetWings();
           WingEntityList returnList = new WingEntityList();
           long num1 = 10;
           while (returnList.Count < 3)
           {               foreach (WingEntity entity in wingList)
              {                  if (returnList.Count == 3)
                    break;
                 if (!((entity.RValue <= (reynold - num1) || entity.RValue >= (reynold +
num1)) || isIn(returnList, entity)))
                    returnList.Add(entity);               }
              num1 += 10;           }
           return returnList;       }
        private bool isIn(WingEntityList list, WingEntity entity)
        {           foreach (WingEntity currentEntity in list)
           {               if (currentEntity.RValue == entity.RValue && currentEntity.Rate
== entity.Rate && currentEntity.Rid == entity.Rid && currentEntity.WingName ==
entity.WingName)
                 return true;           }
           return false;       }    }}
using System;
using System.Collections.Generic;
using System.Text;
namespace SBB.External.WingSimulator.Windows.Business.Interfaces
{    public interface IValuedList
    {
    }
}
using System;
using System.Collections.Generic;
using System.Text;
namespace SBB.External.WingSimulator.Windows.Business.Interfaces
{    public interface IValuedListItem
    {       double Index
        {           get;
           set; }
        double Value
        {           get;
           set;
        }
    }
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
}
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
namespace SBB.External.WingSimulator.Windows.Business.Exceptions
{    public class ExceptionManager
   {        public static void ManageException(Exception exc)
      {            MessageBox.Show(exc.Message, "Wing Simulator",
MessageBoxButtons.OK, MessageBoxIcon.Error);
      }
   }
}
using System;
using System.Collections.Generic;
using System.Text;
using SBB.Mathematics;
namespace SBB.External.WingSimulator.Windows.Business.Calculator
{    public class SimulationCalculator
   {        #region Private - Public Declerations
      private double b;
      private double c;
      private double io;
      private double lambda;
      private long prm_ReynoldFactor;
      private double r;
      private double reynold;
      private double v;
      private double va;
      public double B
      {            get
         {               return b;
         }
         set
         {               b = value;
}
      }
      public double C
      {
         get
         {               return c;
         }

  }
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        public double IO
        {           get
           {            return io;
           }
        }
        public double Lambda
        {           get
           {            return lambda;
           }          set
           {            lambda = value;
           }
        }
        public double R
        {           get
           {            return r;
           }          set
           {            r = value;
           }
        }
        public double Reynold
        {           get
           {            return reynold;
           }
        }
        public double V
        {           get
           {            return v;
           }          set
           {            v = value;
           }
        }

        public double Va
{           get
        {            return va;
           }
        }
        #endregion

        public SimulationCalculator()
        {           v = 5.2;
           r = 5;
           prm_ReynoldFactor = 68500;
           lambda = 5;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            b = 3;
        }
        public void CalculateReynold()
        {           c = (16 * Math.PI * R * (R / (R / 2))) / ((9 * Math.Pow(Lambda, 2)) *
B);
            io = MathematicConversion.ToDegree(Math.Atan((2 * R) / (3 * (R / 2) *
lambda)));
            va = (((R / 2) / R) * Lambda * v) /
Math.Cos(MathematicConversion.ToRadian(IO));
            reynold = prm_ReynoldFactor * c * va;
        }
    }
}
using System;
using System.Collections.Generic;
using System.Text;
namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
{    public sealed class BEntity
    {       #region Private - Public Declerations
        private double index;
        private double value;
        public double Value
        {           get
            {               return value;
            }
            set
            {               this.value = value;
            }
        }
        public double Index
        {           get
         {              return index;
  }
            set
            {               index = value;
            }
        }
        #endregion
        #region Constructors
        public BEntity(double ix, double val)
        {           this.index = ix;
            this.value = val;
        }
        #endregion
    }
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
}
using System;
using System.Collections.Generic;
using System.Text;
using SBB.External.WingSimulator.Windows.Business.Interfaces;
namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
{    public sealed class CEntity : IValuedListItem
   {       #region Private - Public Declerations
      private double index;
      private double value;
      public double Value
      {           get
         {              return value;
         }
         set
         {              this.value = value;
         }
      }
      public double Index
      {           get
         {              return index;
         }
         set
         {              index = value;
         }
  }
      #endregion
      #region Constructors
      public CEntity(double ix, double val)
{           this.index = ix;
         this.value = val;
      }
      #endregion
   }
}
using System;
using System.Collections.Generic;
using System.Text;

namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
{    public sealed class CHelperEntity
   {       #region Private - Public Declerations
      private double firstPartUp;
      private double secondPart;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        private double b;
        private double index;
        public double B
        {           get
           {              return b;
           }
        }
        public double Index
        {           get
           {              return index;
           }
        }
        public double FirstPartUp
        {           get
           {              return firstPartUp;
           }
        }
        public double SecondPart
        {           get
           {
  return secondPart;
           }
        }
        #endregion
        #region Constructors
        public CHelperEntity(double index, double firstPartUp, double secondPart,
double b)
{           this.index = index;
           this.firstPartUp = firstPartUp;
           this.secondPart = secondPart;
           this.b = b;
        }
        #endregion
    }
}
using System;
using System.Collections.Generic;
using System.Text;

namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
{    public sealed class LambdaREntity
    {       #region Private - Public Declerations
        private double index;
        private double value;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        public double Value
        {           get
           {              return value;
           }
           set
           {              this.value = value;
           }
        }
        public double Index
        {           get
           {              return index;
           }
           set
           {              index = value;
           }
        }
  #endregion
        #region Constructors
        public LambdaREntity(double index, double value)
        {           this.index = index;
           this.value = value;
        }       #endregion
    }
}
using System;
using System.Collections.Generic;
using System.Text;
using SBB.External.WingSimulator.Windows.Business.Interfaces;
namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
{    public sealed class QEntity : IValuedListItem
    {       #region Private - Public Declerations
        private double index;
        private double value;
        public double Value
        {           get
           {              return value;
           }
           set
           {              this.value = value;
           }
        }
        public double Index
        {           get
           {              return index;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            }
            set
            {           index = value;
            }
        }
        #endregion
        #region Constructors
        public QEntity(double ix, double val)
        {           this.index = ix;
      this.value = val;
        }
        #endregion
    }
}
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
{    public sealed class StationEntity
    {       #region Private - Public Declarations
        private int rid;
        private string stationName;
private double yearlyAverageSpeed;
        public int Rid
        {           get
            {           return rid;
            }
        }
        public string StationName
        {           get
            {           return stationName;
            }
        }
        public double YearlyAverageSpeed
        {           get
            {           return yearlyAverageSpeed;
            }
        }

        #endregion
        #region Constructors
        public StationEntity(XmlNode nodeX)
        {           XmlNodeList list = nodeX.ChildNodes;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            foreach (XmlNode node in list)
            {               switch (node.Name)
               {                    case "Rid":
                        {
            rid = ParseValue.ToInt(node.InnerText);
                           break;
                        }
                     case "StationName":
                        {                          stationName = node.InnerText;
                           break;
                        }
                     case "YearlyAverageSpeed":
                        {    yearlyAverageSpeed = ParseValue.ToDouble(node.InnerText);
                           break;
                        }
               }
            }
         }
         #endregion
      }
   }
   using System;
   using System.Collections.Generic;
   using System.Text;
   using System.Xml;
   using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
   namespace SBB.External.WingSimulator.Windows.Business.BusinessEntities
   {    public sealed class WingEntity
      {       #region Private - Public Declerations
         private int attackAngle;
         private double liftingFactor;
         private double rate;
         private int rid;
         private long rValue;
         private double sweepingFactor;
         private string wingName;
         private QEntityList qValues;
         private CEntityList cValues;

         public CEntityList CValues
         {           get
            {
               return cValues;
            }
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        }
        public QEntityList QValues
        {           get
           {            return qValues;
           }
    }
        public int AttackAngle
        {           get
           {            return attackAngle;
           }
        }
        public double LiftingFactor
        {           get
           {            return liftingFactor;
           }
        }
        public double Rate
        {           get
           {            return rate;
     }
        }
        public int Rid
        {           get
           {            return rid;
           }
        }
        public long RValue
        {           get
           {            return rValue;
           }
        }
        public double SweepingFactor
        {           get
           {            return sweepingFactor;
           }
        }
        public string WingName
        {           get
           {            return wingName;
           }
    }
        #endregion
        #region Constructors
        public WingEntity(XmlNode nodeX)
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
{           XmlNodeList list = nodeX.ChildNodes;
   foreach (XmlNode node in list)
   {              switch (node.Name)
      {                 case "Rid":
            {                       rid = ParseValue.ToInt(node.InnerText);
               break;
            }
         case "WingName":
            {                       wingName = node.InnerText;
               break;
            }
         case "RValue":
            {                       rValue = ParseValue.ToLong(node.InnerText);
               break;
            }
         case "AttackAngel":
            {                       attackAngle = ParseValue.ToInt(node.InnerText);
   break;                  }
         case "LiftingFactor":
            {                       liftingFactor =
ParseValue.ToDouble(node.InnerText);
               break;
            }
         case "SweepingFactor":
            {                       sweepingFactor =
ParseValue.ToDouble(node.InnerText);
               break;
            }
         case "Rate":
            {                       rate = ParseValue.ToDouble(node.InnerText);
               break;
            }
      }
   }
   }
   #endregion
   #region Public Methods
   public void CalculateQValues(BEntityList bValues)
   {           qValues = new QEntityList();
      foreach (BEntity bEntity in bValues)
         qValues.Add(new QEntity(bEntity.Index, bEntity.Value -
this.attackAngle));       }
   public void CalculateCValues(CHelperEntityList cHelperValues)
   {           cValues = new CEntityList();
      foreach (CHelperEntity cHelperEntity in cHelperValues)
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            cValues.Add(new CEntity(cHelperEntity.Index,
((cHelperEntity.FirstPartUp / (cHelperEntity.B * this.liftingFactor) )  *
cHelperEntity.SecondPart) ));
        }
        #endregion
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities;
namespace SBB.External.WingSimulator.Windows.UI
{    public partial class WingSelector : Form
    {        WingEntityList wingList;
        bool close = false;
        public WingSelector()
        {            InitializeComponent();
        }
        public void SetWings(WingEntityList list)
        {            wingList = list;
            foreach (WingEntity entity in list)
                mlbxChecked.Items.Add(entity.WingName, false);
        }
        public WingEntityList GetSelectedWings()
        {            WingEntityList retList = new WingEntityList();
            for (int i = 0; i < mlbxChecked.CheckedItems.Count; i++)
            {                foreach(WingEntity entit in wingList)
                    if (entit.WingName == mlbxChecked.CheckedItems[i].ToString())
                        retList.Add(entit);
            }
            return retList;
        }
        private void mbtnOk_Click(object sender, EventArgs e)
        {
            if (mlbxChecked.CheckedItems.Count < 1)
            {                MessageBox.Show("En az 1 model seçmelisiniz", "Wing
Simulator", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                close = false;
                return;
            }
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            close = true;        }
        private void WingSelector_FormClosing(object sender, FormClosingEventArgs
e)
        {            if (!close)
                e.Cancel = true;
            else
            {                this.DialogResult = DialogResult.OK;            }
        }
    }
}
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
namespace SBB.External.WingSimulator.Windows.UI
{    internal class UIHelper
    {        public static void FillListView(ListView lstview, CEntityList entityList,
bool first)
        {            for (int i = 0; i < entityList.Count; i++)
            {                if (first)
                    lstview.Items.Add(String.Format("{0:#,###0.000}",
entityList[i].Index));
                lstview.Items[i].SubItems.Add(entityList[i].Value.ToString());            }
        }
        public static void FillListView(ListView lstview, QEntityList entityList, bool
first)
        {            for (int i = 0; i < entityList.Count; i++)
            {                if (first)
                    lstview.Items.Add(String.Format("{0:#,###0.000}",
entityList[i].Index));
                lstview.Items[i].SubItems.Add(entityList[i].Value.ToString());
            }
        }
        public static void CleanColumnsAndItems(ListView lstview)
        {            for (int i = lstview.Columns.Count - 1; i > 0; i--)
            lstview.Columns.Remove(lstview.Columns[i]);
            lstview.Items.Clear();
        }

        public static ColumnHeader BuildColumnHeader(ListView lw, string
headerText, int columnCount)
        {            ColumnHeader header = new ColumnHeader();
            header.TextAlign = HorizontalAlignment.Right;
            header.Text = headerText;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            header.Width = (lw.Width - lw.Columns[0].Width) / (columnCount) - 10;
            return header;
        }
    }
}
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using SBB.Windows.UI.Threading.UI;
using System.Globalization;
namespace SBB.External.WingSimulator.Windows.UI
{    static class Program
    {        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {            Application.EnableVisualStyles();
            FreezerManager.FreezerFormType = typeof(DrawingForm);
            ParseValue.FormatProvider = new CultureInfo("en-US");
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new MainForm());
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SBB.Windows.UI.Threading;
using System.Threading;
using SBB.External.WingSimulator.Windows.Business.XmlHelper;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
using SBB.External.WingSimulator.Windows.Business.Calculator;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities;
using SBB.External.WingSimulator.Windows.Business.Exceptions;
using SBB.Mathematics;
using System.Configuration;
using SBB.External.WingSimulator.Windows.Business.Interfaces;
using System.Globalization;
using System.IO;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
namespace SBB.External.WingSimulator.Windows.UI
{    public partial class MainForm : Form
   {       private WingsXmlHelper wingXmlHelper;
      private StationsXmlHelper stationXmlHelper;
      private StationEntityList stations;
      private WingEntityList wings;
      private DrawingForm drawingForm = new DrawingForm();
      public MainForm()
 {           InitializeComponent();
      }
      private void calculate()
      {            try
         {              if (String.IsNullOrEmpty(mnbxKanatUzunlugu.Text))
               throw new Exception("Kanat uzunluğu belirtilmelidir.");
            if (ParseValue.ToDouble(mnbxKanatUzunlugu.Text) < 1 ||
ParseValue.ToDouble(mnbxKanatUzunlugu.Text) > 55)
               throw new Exception("Kanat uzunluğu 1m ile 55m arasında olmalıdır.");
            if (mcbxWingSpeed.SelectedIndex == 0 &&
String.IsNullOrEmpty(mnbxRuzgarHizi.Text) ||
(ParseValue.ToDouble(mnbxRuzgarHizi.Text) < 0 ||
ParseValue.ToDouble(mnbxRuzgarHizi.Text) > 20))
               throw new Exception("Geçerli bir rüzgâr hızı belirtmelisiniz!");
            mtbsProgressBar.Maximum = 10;
            mtbsProgressBar.Value = 0;
            mtbsProgressBar.Visible = true;
            setStatusText("Hesaplama İşlemi başladı.");
            increaseProgressBar();
            SimulationCalculator calculator = new SimulationCalculator();
            double fi =
ParseValue.ToDouble(ConfigurationManager.AppSettings["Fi"]);
            calculator.R = ParseValue.ToDouble(mnbxKanatUzunlugu.Text);
            foreach (StationEntity entity in stations)
            {                  if (entity.StationName ==
mcbxWingSpeed.Items[mcbxWingSpeed.SelectedIndex].ToString())
               {                     calculator.V = entity.YearlyAverageSpeed;
                  break;
               }
            }
            increaseProgressBar();
            calculator.CalculateReynold();
            setStatusText("Uygun kanat modelleri belirleniyor.");
            increaseProgressBar();
            WingEntityList wingList =
wingXmlHelper.SearchNearestWings(calculator.Reynold);
            setStatusText("Uygun kanat modelleri belirlendi.");
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            increaseProgressBar();
            WingSelector selector = new WingSelector();
            selector.SetWings(wingList);
            if(selector.ShowDialog() == DialogResult.OK)
            {                   increaseProgressBar();
               WingEntityList selectedWings = selector.GetSelectedWings();
  double bVal = ParseValue.ToDouble(ConfigurationManager.AppSettings["B"]);
               LambdaREntityList lambdaRList = new LambdaREntityList();
               BEntityList bEntityList = new BEntityList();
               CHelperEntityList cHelperEntityList = new CHelperEntityList();
               for (double i = (1.5 * calculator.R) / 100; i < (98 * calculator.R / 100);
i+=0.01)
               {                   LambdaREntity lambdaREntity = new
LambdaREntity(i, calculator.Lambda * (i / calculator.R));
                  BEntity bEntity = new BEntity(i,
MathematicConversion.ToDegree(Math.Atan(1 / lambdaREntity.Value)) * 2 / 3);
                  double radianB = MathematicConversion.ToRadian(bEntity.Value);
                  double firsPartUp = 8 * Math.PI * i * fi * Math.Sin(radianB);
                  double secondPart = (Math.Cos(radianB) - lambdaREntity.Value *
Math.Sin(radianB)) / (Math.Sin(radianB) + lambdaREntity.Value *
Math.Cos(radianB));
                  cHelperEntityList.Add(new CHelperEntity(i, firsPartUp, secondPart,
bVal));
                  lambdaRList.Add(lambdaREntity);
                  bEntityList.Add(bEntity);
               }
               setStatusText("Bilgiler tabloya dolduruluyor.");
               increaseProgressBar();
               UIHelper.CleanColumnsAndItems(mlwWingCiDatas);
               increaseProgressBar();
               UIHelper.CleanColumnsAndItems(mlwWingQiDatas);
               increaseProgressBar();
               CEntityList[] cPointList = new CEntityList[selectedWings.Count];
               for (int i = 0; i < selectedWings.Count; i++ )
               {                   selectedWings[i].CalculateQValues(bEntityList);
                  selectedWings[i].CalculateCValues(cHelperEntityList);


mlwWingQiDatas.Columns.Add(UIHelper.BuildColumnHeader(mlwWingQiDatas,
String.Format("{0} - Qi", selectedWings[i].WingName), selectedWings.Count));

mlwWingCiDatas.Columns.Add(UIHelper.BuildColumnHeader(mlwWingCiDatas,
String.Format("{0} - Ci", selectedWings[i].WingName), selectedWings.Count));
                  UIHelper.FillListView(mlwWingQiDatas, selectedWings[i].QValues,
i == 0);
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
                UIHelper.FillListView(mlwWingCiDatas, selectedWings[i].CValues,
i == 0);
                cPointList[i] = selectedWings[i].CValues;
            }
            increaseProgressBar();
            drawingForm.CEntities = cPointList;
            drawingForm.SelectedWings = selectedWings;
        }
        increaseProgressBar();
        mnbxReynold.Text = calculator.Reynold.ToString();
        double a = Math.PI * Math.Pow(calculator.R, 2);
        double v = Math.Pow(calculator.V, 3);
        mnbxNe.Text = String.Format("{0} W", a * v *
ParseValue.ToDouble(ConfigurationManager.AppSettings["N"]) *
ParseValue.ToDouble(ConfigurationManager.AppSettings["Cp"]) * 0.6);
        mtbsProgressBar.Visible = false;
        mtsbtnGraphic.Enabled = true;
        setStatusText("İşlem tamamlandı.");
      }
      catch (Exception exc)
      {           ExceptionManager.ManageException(exc);
      }
    }
    private void setStatusText(string text)
    {       mtbsStatus.Text = text;
      mtbsProgressBar.Size = new Size(mssMain.Width - mtbsStatus.Width - 20,
mtbsProgressBar.Size.Height);
      Application.DoEvents();
    }
    private void increaseProgressBar()
    {       mtbsProgressBar.Value++;
      Application.DoEvents();
    }
    private void MainForm_Load(object sender, EventArgs e)
    {       try
      {           if (!File.Exists("Stations.xml") || !File.Exists("Wings.xml"))
        {               MessageBox.Show("Programın çalışabilmesi için gerekli
bileşen dosyaları bulunamadı.\nProgramı tekrar yükleyiniz.", this.Text,
MessageBoxButtons.OK, MessageBoxIcon.Warning);
          Application.Exit();
        }
        new AboutBackGroundForm(this).Execute(true);
        this.Icon = WingSimulatorResources.LineChart;
        mtsbtnCalculate.Image =
WingSimulatorResources.Calculator.ToBitmap();
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            mtsbtnExit.Image = WingSimulatorResources.Delete.ToBitmap();
            mtsbtnGraphic.Image = WingSimulatorResources.LineChart.ToBitmap();
            mtsbtnAbout.Image = WingSimulatorResources.Information.ToBitmap();
            mtsbHelp.Image = WingSimulatorResources.Help.ToBitmap();
            wingXmlHelper = new WingsXmlHelper("Wings.xml");
            stationXmlHelper = new StationsXmlHelper("Stations.xml");
            wings = wingXmlHelper.GetWings();
            stations = stationXmlHelper.GetAllStations();
            mcbxWingSpeed.Items.Add("Diğer Bölge");
            foreach (StationEntity entity in stations)
               mcbxWingSpeed.Items.Add(entity.StationName);
            mcbxWingSpeed.SelectedIndex = 0;
         }
         catch (Exception exc)
         {              ExceptionManager.ManageException(exc);
         }
      }
      private void mcbxWingSpeed_SelectedIndexChanged(object sender, EventArgs
e)
      {           if (mcbxWingSpeed.SelectedIndex == 0)
         {              mnbxRuzgarHizi.ReadOnly = false;
            mnbxRuzgarHizi.Text = String.Empty;
         }
         else
         {              mnbxRuzgarHizi.ReadOnly = true;
            foreach (StationEntity entity in stations)
            {                 if (entity.StationName ==
mcbxWingSpeed.Items[mcbxWingSpeed.SelectedIndex].ToString())
               {                    mnbxRuzgarHizi.Text =
entity.YearlyAverageSpeed.ToString(new CultureInfo("en-US"));
                  break;
               }
            }
         }
      }
      private void MainForm_Paint(object sender, PaintEventArgs e)
      {           for(int i = 1; i<mlwWingQiDatas.Columns.Count; i++)
            mlwWingQiDatas.Columns[i].Width = (mlwWingQiDatas.Width -
mlwWingQiDatas.Columns[0].Width) / mlwWingQiDatas.Columns.Count - 10;
         for (int i = 1; i < mlwWingCiDatas.Columns.Count; i++)
            mlwWingCiDatas.Columns[i].Width = (mlwWingCiDatas.Width -
mlwWingCiDatas.Columns[0].Width) / mlwWingCiDatas.Columns.Count - 10;
      }
      private void mtsbtnCalculate_Click(object sender, EventArgs e)
      {
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            calculate();
        }
        private void mtsbtnExit_Click(object sender, EventArgs e)
        {
            this.Close();
        }
        private void mtsbtnAbout_Click(object sender, EventArgs e)
        {
            new AboutBackGroundForm(this).Execute();
        }
        private void mtsbtnGraphic_Click(object sender, EventArgs e)
        {
            drawingForm.ShowDialog();
        }
        private void mtsbHelp_Click(object sender, EventArgs e)
        {
            new HelpForm().ShowDialog();
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Configuration;
namespace SBB.External.WingSimulator.Windows.UI
{    public partial class HelpForm : Form
    {        public HelpForm()
        {            InitializeComponent();
        }
        private void mbtnClose_Click(object sender, EventArgs e)
        {            Close();
        }

        private void HelpForm_Load(object sender, EventArgs e)
        {            mtbxContent.Text =
String.Format(WingSimulatorResources.HelpText, Application.StartupPath);
        }
    }
}
using System;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities.List;
using System.Collections;
using SBB.External.WingSimulator.Windows.Business.BusinessEntities;
using System.Drawing.Imaging;
using SBB.External.WingSimulator.Windows.Business.Exceptions;
namespace SBB.External.WingSimulator.Windows.UI
{    public partial class DrawingForm : Form
   {        private CEntityList[] cEntities = null;
      private WingEntityList selectedWings = null;
      private Bitmap bmp;
      public WingEntityList SelectedWings
      {           get
         {              return selectedWings;
         }
         set
         {              selectedWings = value;
         }
      }
      public CEntityList[] CEntities
      {           get
         {              return cEntities;
         }
         set
         {              cEntities = value;
         }
      }
      public DrawingForm()
      {           InitializeComponent();
         mtsbKaydet.Image = WingSimulatorResources.SaveDisk.ToBitmap();
         mtsbKapat.Image = WingSimulatorResources.Delete.ToBitmap();
         mtsbtnPrint.Image = WingSimulatorResources.Printer.ToBitmap();
      }
      private void DrawingForm_Paint(object sender, PaintEventArgs e)
      {           this.Height = 675;
         this.Width = 800;
         Color[] lineColors = { Color.Red, Color.Blue, Color.Green, Color.Bisque };
         Font writingFont = new Font("Times New Roman", 8, FontStyle.Regular);
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        Font writingFontBold = new Font("Times New Roman", 8,
FontStyle.Regular | FontStyle.Bold);
        SolidBrush blackSolidBrush = new SolidBrush(Color.Black);
        int width = 800;
        int height = 650;
        Graphics gx = this.CreateGraphics();
        bmp = new Bitmap(width, height);
        Pen blackPen = new Pen(Color.Black);
        Pen bluePen = new Pen(Color.Blue);
        Graphics drawingGx = Graphics.FromImage(bmp);
        drawingGx.FillRectangle(new SolidBrush(Color.White), new Rectangle(0, 0,
width, height));
        drawingGx.CompositingQuality =
System.Drawing.Drawing2D.CompositingQuality.HighQuality;
        drawingGx.InterpolationMode =
System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;
        drawingGx.SmoothingMode =
System.Drawing.Drawing2D.SmoothingMode.AntiAlias;
        //dikey ok
        drawingGx.DrawLine(blackPen, new Point(10, 10), new Point(10, height -
60));
        drawingGx.DrawLine(blackPen, new Point(16, 14), new Point(10, 10));
        drawingGx.DrawLine(blackPen, new Point(4, 14), new Point(10, 10));
        drawingGx.DrawString("Ci", writingFontBold, blackSolidBrush, 17.0F,
4.0F);
        //yatay ok
        drawingGx.DrawLine(blackPen, new Point(10, height - 60), new Point(width
- 10, height - 60));
        drawingGx.DrawLine(blackPen, new Point(width - 16, height - 66), new
Point(width - 10, height - 60));
        drawingGx.DrawLine(blackPen, new Point(width - 16, height - 54), new
Point(width - 10, height - 60));
        drawingGx.DrawString("Ri", writingFontBold, blackSolidBrush, width - 24,
height - 82);
        //sağ üst kutu
        drawingGx.DrawRectangle(blackPen, new Rectangle(width - 260, height -
640, 250, 50));


        //yatay noktalar ve yazıları
        int pointCounter = 0;
        for (int i = 30; i < 790; i += 40)
        {               pointCounter++;
           drawingGx.DrawLine(blackPen, new Point(i, height - 67), new Point(i,
height - 53));
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
            drawingGx.DrawString(String.Format("{0:#,###0.000}",
CEntities[0][CEntities[0].Count - 1].Index / 19 * pointCounter), writingFont,
blackSolidBrush, new PointF(i - 8, height - 54));
         }
         for (int i = 0; i < CEntities.Length; i++)
         {              PointF[] points = new PointF[CEntities[i].Count];
            Double minValue = CEntities[i][0].Value, maxValue =
CEntities[i][0].Value;
            Pen tmpPen = new Pen(lineColors[i]);
            double xFactor = (73000 / points.Length);
            double yFactor = (54283 / points.Length);
            for (int j = 0; j < CEntities[i].Count; j++)
            {                  if (CEntities[i][j].Value > maxValue)
               maxValue = CEntities[i][j].Value;
              if (CEntities[i][j].Value < minValue)
               minValue = CEntities[i][j].Value;
              points[j] = new PointF(Convert.ToSingle(CEntities[i][j].Index * xFactor
+ 10), Convert.ToSingle((height - 60) - CEntities[i][j].Value * yFactor));
            }
            drawingGx.FillRectangle(new SolidBrush(lineColors[i]), new
Rectangle(width - 256, height - (635 - i * 13), 15, 10));
            drawingGx.DrawString(selectedWings[i].WingName.Substring(0,
selectedWings[i].WingName.Length > 16 ? 16 :
selectedWings[i].WingName.Length), writingFont, blackSolidBrush, width - 238,
height - (635 - i * 13));
            drawingGx.DrawString(String.Format("Ci(Min : {0:#,###0.000} - Max :
{1:#,###0.000})", minValue, maxValue), writingFont, blackSolidBrush, width - 150,
height - (635 - i * 13));
                        drawingGx.DrawLines(tmpPen, points);
         }
         drawingGx.Dispose();
         gx.DrawImageUnscaled(bmp, 0, 25);
      }
      private void DrawingForm_KeyPress(object sender, KeyPressEventArgs e)
      {           this.Close();
      }
      private void DrawingForm_MouseDown(object sender, MouseEventArgs e)
      {           this.Close();
      }
      private void mtsbKaydet_Click(object sender, EventArgs e)
      {           try
         {              mdlgSaveFile.Filter = "Bitmap Dosyası| *.bmp";
            if (mdlgSaveFile.ShowDialog() == DialogResult.OK)
            {                  bmp.Save(mdlgSaveFile.FileName, ImageFormat.Bmp);
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
                MessageBox.Show("Dosya başarıyla kaydedilmiştir.", this.Text,
MessageBoxButtons.OK, MessageBoxIcon.Information,
MessageBoxDefaultButton.Button1);
            }
        }
        catch(Exception exc)
        {           ExceptionManager.ManageException(exc);
        }
    }
    private void mtsbKapat_Click(object sender, EventArgs e)
    {       Close();
    }
    private void mtsbtnPrint_Click(object sender, EventArgs e)
    {
    }
  }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Threading;
namespace SBB.External.WingSimulator.Windows.UI
{   public partial class AboutForm : Form
  {     internal EventHandler<AboutEventArgs> eventWriter;
    private bool startingShow = false;
    internal bool StartingShow
    {       get
      {           return startingShow;
      }
      set
      {           startingShow = value;
      }
    }
    public AboutForm()
    {       InitializeComponent();
    }
    private void mpbxBackGround_MouseUp(object sender, MouseEventArgs e)
    {       if(!startingShow)
          this.Close();
    }
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        private void AboutForm_MouseUp(object sender, MouseEventArgs e)
        {           if (!startingShow)
                this.Close();
        }
        private void mtbxMessage_MouseUp(object sender, MouseEventArgs e)
        {           if (!startingShow)
                this.Close();
        }
        protected override void OnClosing(CancelEventArgs e)
        {           eventWriter("MainForm", new AboutEventArgs());
           base.OnClosing(e);
        }
        protected override void OnShown(EventArgs e)
        {           mlblLoading.Visible = startingShow;
           mpbxLoading.Visible = startingShow;
           mtbxMessage.Visible = !startingShow;
           if (startingShow)
           {               mpbxLoading.Value = 0;
              mpbxLoading.Maximum = 10;
              for (int i = 0; i < 10; i++)
              {                   Thread.Sleep(200);
                 mpbxLoading.Value++;
                 Application.DoEvents();
              }
              this.Close();
           }
           else
           {               mtbxMessage.Text = "Bu program Arş. Gör. Murat ÖNDER'in
master araştırmasının gerçekleştirilmesi  amacıyla Mehmet Akif ÇAKAR tarafından
kodlanmıştır.\n Teknik Destek için Web Adresi http://www.sbbsoftware.com";
           }
        }
        #region Internal Class
        internal class AboutEventArgs : EventArgs
        {           public AboutEventArgs()
              : base()
           {
           }
        }

        #endregion
     }
}
using System;
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using SBB.Windows.UI.Constants;
namespace SBB.External.WingSimulator.Windows.UI
{    internal partial class AboutBackGroundForm : Form
   {       #region Private - Public Declerations
      private AboutForm aboutForm;
      private Form parent;
      private bool startingShow;
      #endregion
      #region Constructors
      public AboutBackGroundForm()
      {           InitializeComponent();
      }
      public AboutBackGroundForm(Form parentForm)
      {           parent = parentForm;
         InitializeComponent();
      }
      #endregion
      #region Events
      protected override void OnLoad(EventArgs e)
      {           base.OnLoad(e);
         aboutForm = new AboutForm();
         aboutForm.eventWriter += result;
      }
      protected override void OnShown(EventArgs e)
      {           aboutForm.StartingShow = startingShow;
         aboutForm.ShowDialog();
      }
      private void result(object sender,
SBB.External.WingSimulator.Windows.UI.AboutForm.AboutEventArgs e)
      {           DialogResult = DialogResult.OK;
      }
      #endregion

      #region Calling Types
      public bool Execute()
      {return Execute(false);
      }
      public bool Execute(bool start)
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
        {           startingShow = start;
            if(start)
                this.Size = new Size(Screen.PrimaryScreen.Bounds.Width,
Screen.PrimaryScreen.Bounds.Height);
            else
                this.Size = parent.Size;
            return ShowDialog(parent) == DialogResult.OK;
        }
        #endregion
    }
}
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="Fi" value="1" />
    <add key="Cp" value="0.57" />
    <add key="N" value="0.8" />
    <add key="B" value="3" />
  </appSettings>
</configuration>
<?xml version="1.0" encoding="UTF-8"?>
<StationsData>
  <Stations>
    <Rid>1</Rid>
    <StationName>Amasra</StationName>
    <YearlyAverageSpeed>5.2</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>2</Rid>
    <StationName>Bandırma</StationName>
    <YearlyAverageSpeed>4</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>3</Rid>
    <StationName>Bergama</StationName>
    <YearlyAverageSpeed>3</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>4</Rid>
    <StationName>Bozcada</StationName>
    <YearlyAverageSpeed>5.8</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>5</Rid>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <StationName>Cihanbeyli</StationName>
  <YearlyAverageSpeed>2.9</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>6</Rid>
  <StationName>Çanakkale</StationName>
  <YearlyAverageSpeed>3.7</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>7</Rid>
  <StationName>Diyarbakır</StationName>
  <YearlyAverageSpeed>2.8</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>8</Rid>
  <StationName>Erzurum</StationName>
  <YearlyAverageSpeed>2.8</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>9</Rid>
  <StationName>Güney</StationName>
  <YearlyAverageSpeed>4.3</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>10</Rid>
  <StationName>İpsala</StationName>
  <YearlyAverageSpeed>2.9</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>11</Rid>
  <StationName>Karataş</StationName>
  <YearlyAverageSpeed>3.1</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>12</Rid>
  <StationName>Mardin</StationName>
  <YearlyAverageSpeed>3.9</YearlyAverageSpeed>
 </Stations>
 <Stations>
  <Rid>13</Rid>
  <StationName>Pınarbaşı</StationName>
  <YearlyAverageSpeed>3.9</YearlyAverageSpeed>
 </Stations>
 <Stations>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
    <Rid>14</Rid>
    <StationName>Sinop</StationName>
    <YearlyAverageSpeed>2.9</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>15</Rid>
    <StationName>Siverek</StationName>
    <YearlyAverageSpeed>2.9</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>16</Rid>
    <StationName>Suşehri</StationName>
    <YearlyAverageSpeed>3.2</YearlyAverageSpeed>
  </Stations>
  <Stations>
    <Rid>17</Rid>
    <StationName>Şile</StationName>
    <YearlyAverageSpeed>3.4</YearlyAverageSpeed>
  </Stations>
</StationsData>
<?xml version="1.0" encoding="UTF-8"?>
<WingDatas>
  <Wings>
    <Rid>1</Rid>
    <WingName>A18 SM</WingName>
    <RValue>203000</RValue>
    <AttackAngel>4</AttackAngel>
    <LiftingFactor>0.8</LiftingFactor>
    <SweepingFactor>0.0111</SweepingFactor>
    <Rate>72.07</Rate>
  </Wings>
  <Wings>
    <Rid>2</Rid>
    <WingName>A18 SM</WingName>
    <RValue>300000</RValue>
    <AttackAngel>7</AttackAngel>
    <LiftingFactor>0.98</LiftingFactor>
    <SweepingFactor>0.024</SweepingFactor>
    <Rate>40.83</Rate>
  </Wings>
  <Wings>
    <Rid>3</Rid>
    <WingName>A18 SM</WingName>
    <RValue>302600</RValue>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <AttackAngel>4</AttackAngel>
  <LiftingFactor>0.76</LiftingFactor>
  <SweepingFactor>0.0094</SweepingFactor>
  <Rate>80.85</Rate>
 </Wings>
 <Wings>
  <Rid>4</Rid>
  <WingName>AQUILA SM</WingName>
  <RValue>203900</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>0.92</LiftingFactor>
  <SweepingFactor>0.015</SweepingFactor>
  <Rate>61.33</Rate>
 </Wings>
 <Wings>
  <Rid>5</Rid>
  <WingName>AQUILA SM</WingName>
  <RValue>301100</RValue>
  <AttackAngel>7</AttackAngel>
  <LiftingFactor>1.1</LiftingFactor>
  <SweepingFactor>0.0152</SweepingFactor>
  <Rate>72.36</Rate>
 </Wings>
 <Wings>
  <Rid>6</Rid>
  <WingName>GOE 417 A</WingName>
  <RValue>297500</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>1.04</LiftingFactor>
  <SweepingFactor>0.0126</SweepingFactor>
  <Rate>82.5</Rate>
 </Wings>
 <Wings>
  <Rid>7</Rid>
  <WingName>GOE 417 A</WingName>
  <RValue>398800</RValue>
  <AttackAngel>2</AttackAngel>
  <LiftingFactor>0.8</LiftingFactor>
  <SweepingFactor>0.014</SweepingFactor>
  <Rate>57.14</Rate>
 </Wings>
 <Wings>
  <Rid>8</Rid>
  <WingName>ESA 40</WingName>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
    <RValue>200600</RValue>
    <AttackAngel>10</AttackAngel>
    <LiftingFactor>0.98</LiftingFactor>
    <SweepingFactor>0.024</SweepingFactor>
    <Rate>40.8</Rate>
  </Wings>
  <Wings>
    <Rid>9</Rid>
    <WingName>ESA 40</WingName>
    <RValue>301600</RValue>
    <AttackAngel>10</AttackAngel>
    <LiftingFactor>1.02</LiftingFactor>
    <SweepingFactor>0.02</SweepingFactor>
    <Rate>51</Rate>
  </Wings>
  <Wings>
    <Rid>10</Rid>
    <WingName>ESA 40</WingName>
    <RValue>398100</RValue>
    <AttackAngel>10</AttackAngel>
    <LiftingFactor>1.05</LiftingFactor>
    <SweepingFactor>0.0177</SweepingFactor>
    <Rate>59.32</Rate>
  </Wings>
  <Wings>
    <Rid>11</Rid>
    <WingName>ESA 40</WingName>
    <RValue>403800</RValue>
    <AttackAngel>5</AttackAngel>
    <LiftingFactor>0.5</LiftingFactor>
    <SweepingFactor>0.0118</SweepingFactor>
    <Rate>42.37</Rate>
  </Wings>
  <Wings>
    <Rid>12</Rid>
    <WingName>DAVIS SM</WingName>
    <RValue>198600</RValue>
    <AttackAngel>3</AttackAngel>
    <LiftingFactor>0.95</LiftingFactor>
    <SweepingFactor>0.0155</SweepingFactor>
    <Rate>61.29</Rate>
  </Wings>
  <Wings>
    <Rid>13</Rid>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <WingName>DAVIS SM</WingName>
  <RValue>297900</RValue>
  <AttackAngel>3</AttackAngel>
  <LiftingFactor>0.98</LiftingFactor>
  <SweepingFactor>0.016</SweepingFactor>
  <Rate>61.25</Rate>
 </Wings>
 <Wings>
  <Rid>14</Rid>
  <WingName>J 5012</WingName>
  <RValue>201900</RValue>
  <AttackAngel>4</AttackAngel>
  <LiftingFactor>0.47</LiftingFactor>
  <SweepingFactor>0.12</SweepingFactor>
  <Rate>39.16</Rate>
 </Wings>
 <Wings>
  <Rid>15</Rid>
  <WingName>J 5012</WingName>
  <RValue>303100</RValue>
  <AttackAngel>4</AttackAngel>
  <LiftingFactor>0.47</LiftingFactor>
  <SweepingFactor>0.0106</SweepingFactor>
  <Rate>44.33</Rate>
 </Wings>
 <Wings>
  <Rid>16</Rid>
  <WingName>CLARK Y</WingName>
  <RValue>203800</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.95</LiftingFactor>
  <SweepingFactor>0.0144</SweepingFactor>
  <Rate>65.97</Rate>
 </Wings>
 <Wings>
  <Rid>17</Rid>
  <WingName>CLARK Y</WingName>
  <RValue>301200</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>0.87</LiftingFactor>
  <SweepingFactor>0.012</SweepingFactor>
  <Rate>72.5</Rate>
 </Wings>
 <Wings>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <Rid>18</Rid>
  <WingName>FALCON 56 MK 2</WingName>
  <RValue>200000</RValue>
  <AttackAngel>8</AttackAngel>
  <LiftingFactor>0.9</LiftingFactor>
  <SweepingFactor>0.018</SweepingFactor>
  <Rate>50</Rate>
 </Wings>
 <Wings>
  <Rid>19</Rid>
  <WingName>FALCON 56 MK 2</WingName>
  <RValue>301300</RValue>
  <AttackAngel>8</AttackAngel>
  <LiftingFactor>0.98</LiftingFactor>
  <SweepingFactor>0.0155</SweepingFactor>
  <Rate>63.2</Rate>
 </Wings>
 <Wings>
  <Rid>20</Rid>
  <WingName>FALCON 56 MK 2</WingName>
  <RValue>401700</RValue>
  <AttackAngel>9</AttackAngel>
  <LiftingFactor>1.08</LiftingFactor>
  <SweepingFactor>0.0154</SweepingFactor>
  <Rate>70.1</Rate>
 </Wings>
 <Wings>
  <Rid>21</Rid>
  <WingName>K 3311 SM</WingName>
  <RValue>201800</RValue>
  <AttackAngel>8</AttackAngel>
  <LiftingFactor>1.14</LiftingFactor>
  <SweepingFactor>0.0157</SweepingFactor>
  <Rate>72.61</Rate>
 </Wings>
 <Wings>
  <Rid>22</Rid>
  <WingName>K 3311 SM</WingName>
  <RValue>302900</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.93</LiftingFactor>
  <SweepingFactor>0.0114</SweepingFactor>
  <Rate>81.5</Rate>
 </Wings>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
<Wings>
  <Rid>23</Rid>
  <WingName>R 140 SM</WingName>
  <RValue>203000</RValue>
  <AttackAngel>4</AttackAngel>
  <LiftingFactor>0.48</LiftingFactor>
  <SweepingFactor>0.0123</SweepingFactor>
  <Rate>39.02</Rate>
</Wings>
<Wings>
  <Rid>24</Rid>
  <WingName>RG 14</WingName>
  <RValue>199900</RValue>
  <AttackAngel>4</AttackAngel>
  <LiftingFactor>0.6</LiftingFactor>
  <SweepingFactor>0.0106</SweepingFactor>
  <Rate>56.6</Rate>
</Wings>
<Wings>
  <Rid>25</Rid>
  <WingName>RG 14</WingName>
  <RValue>300300</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>0.72</LiftingFactor>
  <SweepingFactor>0.0109</SweepingFactor>
  <Rate>66.05</Rate>
</Wings>
<Wings>
  <Rid>26</Rid>
  <WingName>PT 40</WingName>
  <RValue>200100</RValue>
  <AttackAngel>7</AttackAngel>
  <LiftingFactor>0.945</LiftingFactor>
  <SweepingFactor>0.0184</SweepingFactor>
  <Rate>51.3</Rate>
</Wings>
<Wings>
  <Rid>27</Rid>
  <WingName>PT 40</WingName>
  <RValue>299800</RValue>
  <AttackAngel>7</AttackAngel>
  <LiftingFactor>0.97</LiftingFactor>
  <SweepingFactor>0.0173</SweepingFactor>
  <Rate>56.06</Rate>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
</Wings>
<Wings>
  <Rid>28</Rid>
  <WingName>PT 40</WingName>
  <RValue>399100</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.9</LiftingFactor>
  <SweepingFactor>0.0146</SweepingFactor>
  <Rate>61.6</Rate>
</Wings>
<Wings>
  <Rid>29</Rid>
  <WingName>SPICA SM</WingName>
  <RValue>202300</RValue>
  <AttackAngel>10</AttackAngel>
  <LiftingFactor>1.17</LiftingFactor>
  <SweepingFactor>0.018</SweepingFactor>
  <Rate>65</Rate>
</Wings>
<Wings>
  <Rid>30</Rid>
  <WingName>SPICA SM</WingName>
  <RValue>301500</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>0.75</LiftingFactor>
  <SweepingFactor>0.0112</SweepingFactor>
  <Rate>66.96</Rate>
</Wings>
<Wings>
  <Rid>31</Rid>
  <WingName>USNPS 4</WingName>
  <RValue>199300</RValue>
  <AttackAngel>4</AttackAngel>
  <LiftingFactor>0.93</LiftingFactor>
  <SweepingFactor>0.0135</SweepingFactor>
  <Rate>68.8</Rate>
</Wings>
<Wings>
  <Rid>32</Rid>
  <WingName>USNPS 4</WingName>
  <RValue>299000</RValue>
  <AttackAngel>3</AttackAngel>
  <LiftingFactor>0.87</LiftingFactor>
  <SweepingFactor>0.0104</SweepingFactor>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <Rate>83.6</Rate>
 </Wings>
 <Wings>
  <Rid>33</Rid>
  <WingName>USNPS 4</WingName>
  <RValue>398300</RValue>
  <AttackAngel>3</AttackAngel>
  <LiftingFactor>0.85</LiftingFactor>
  <SweepingFactor>0.0098</SweepingFactor>
  <Rate>86.7</Rate>
 </Wings>
 <Wings>
  <Rid>34</Rid>
  <WingName>TRANIER 60</WingName>
  <RValue>200000</RValue>
  <AttackAngel>8</AttackAngel>
  <LiftingFactor>0.8</LiftingFactor>
  <SweepingFactor>0.0189</SweepingFactor>
  <Rate>42.32</Rate>
 </Wings>
 <Wings>
  <Rid>35</Rid>
  <WingName>TRANIER 60</WingName>
  <RValue>301800</RValue>
  <AttackAngel>10</AttackAngel>
  <LiftingFactor>1.07</LiftingFactor>
  <SweepingFactor>0.02</SweepingFactor>
  <Rate>53.5</Rate>
 </Wings>
 <Wings>
  <Rid>36</Rid>
  <WingName>TRANIER 60</WingName>
  <RValue>400000</RValue>
  <AttackAngel>10</AttackAngel>
  <LiftingFactor>1.067</LiftingFactor>
  <SweepingFactor>0.018</SweepingFactor>
  <Rate>59.27</Rate>
 </Wings>
 <Wings>
  <Rid>37</Rid>
  <WingName>WASP SM</WingName>
  <RValue>201600</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.93</LiftingFactor>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <SweepingFactor>0.148</SweepingFactor>
  <Rate>62.8</Rate>
 </Wings>
 <Wings>
  <Rid>38</Rid>
  <WingName>WASP SM</WingName>
  <RValue>302600</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.93</LiftingFactor>
  <SweepingFactor>0.0124</SweepingFactor>
  <Rate>75</Rate>
 </Wings>
 <Wings>
  <Rid>39</Rid>
  <WingName>WB 140</WingName>
  <RValue>200500</RValue>
  <AttackAngel>7</AttackAngel>
  <LiftingFactor>1.06</LiftingFactor>
  <SweepingFactor>0.0152</SweepingFactor>
  <Rate>69.7</Rate>
 </Wings>
 <Wings>
  <Rid>40</Rid>
  <WingName>WB 140</WingName>
  <RValue>307600</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.96</LiftingFactor>
  <SweepingFactor>0.0121</SweepingFactor>
  <Rate>79.33</Rate>
 </Wings>
 <Wings>
  <Rid>41</Rid>
  <WingName>WB 13535 SM</WingName>
  <RValue>204000</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.982</LiftingFactor>
  <SweepingFactor>0.0157</SweepingFactor>
  <Rate>62.54</Rate>
 </Wings>
 <Wings>
  <Rid>42</Rid>
  <WingName>WB 13535 SM</WingName>
  <RValue>302700</RValue>
  <AttackAngel>3</AttackAngel>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
  <LiftingFactor>0.73</LiftingFactor>
  <SweepingFactor>0.0115</SweepingFactor>
  <Rate>63.47</Rate>
 </Wings>
 <Wings>
  <Rid>43</Rid>
  <WingName>CLARK YSM</WingName>
  <RValue>200000</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>0.85</LiftingFactor>
  <SweepingFactor>0.0129</SweepingFactor>
  <Rate>65.89</Rate>
 </Wings>
 <Wings>
  <Rid>44</Rid>
  <WingName>CLARK YSM</WingName>
  <RValue>299800</RValue>
  <AttackAngel>6</AttackAngel>
  <LiftingFactor>0.945</LiftingFactor>
  <SweepingFactor>0.012</SweepingFactor>
  <Rate>78.75</Rate>
 </Wings>
 <Wings>
  <Rid>45</Rid>
  <WingName>CLARK YSM</WingName>
  <RValue>399900</RValue>
  <AttackAngel>5</AttackAngel>
  <LiftingFactor>8.882</LiftingFactor>
  <SweepingFactor>0.01</SweepingFactor>
  <Rate>88.2</Rate>
 </Wings>
 <Wings>
  <Rid>46</Rid>
  <WingName>NACA 2414</WingName>
  <RValue>200500</RValue>
  <AttackAngel>7</AttackAngel>
  <LiftingFactor>0.88</LiftingFactor>
  <SweepingFactor>0.0146</SweepingFactor>
  <Rate>60.27</Rate>
 </Wings>
 <Wings>
  <Rid>47</Rid>
  <WingName>NACA 2414</WingName>
  <RValue>301000</RValue>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
    <AttackAngel>7</AttackAngel>
    <LiftingFactor>0.867</LiftingFactor>
    <SweepingFactor>0.0126</SweepingFactor>
    <Rate>68.8</Rate>
  </Wings>
  <Wings>
    <Rid>48</Rid>
    <WingName>NACA 2415</WingName>
    <RValue>200600</RValue>
    <AttackAngel>7</AttackAngel>
    <LiftingFactor>0.95</LiftingFactor>
    <SweepingFactor>0.0162</SweepingFactor>
    <Rate>58.64</Rate>
  </Wings>
  <Wings>
    <Rid>49</Rid>
    <WingName>NACA 2415</WingName>
    <RValue>301100</RValue>
    <AttackAngel>6</AttackAngel>
    <LiftingFactor>0.867</LiftingFactor>
    <SweepingFactor>0.0126</SweepingFactor>
    <Rate>68.8</Rate>
  </Wings>
  <Wings>
    <Rid>50</Rid>
    <WingName>NACA 25411</WingName>
    <RValue>200600</RValue>
    <AttackAngel>5</AttackAngel>
    <LiftingFactor>0.715</LiftingFactor>
    <SweepingFactor>0.0137</SweepingFactor>
    <Rate>52.18</Rate>
  </Wings>
  <Wings>
    <Rid>51</Rid>
    <WingName>NACA 25411</WingName>
    <RValue>303300</RValue>
    <AttackAngel>4</AttackAngel>
    <LiftingFactor>0.615</LiftingFactor>
    <SweepingFactor>0.01045</SweepingFactor>
    <Rate>58.85</Rate>
  </Wings>
  <Wings>
    <Rid>52</Rid>
    <WingName>NACA 6409</WingName>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
    <RValue>200100</RValue>
    <AttackAngel>7</AttackAngel>
    <LiftingFactor>1.26</LiftingFactor>
    <SweepingFactor>0.0161</SweepingFactor>
    <Rate>78.26</Rate>
  </Wings>
  <Wings>
    <Rid>53</Rid>
    <WingName>NACA 2418</WingName>
    <RValue>2900000</RValue>
    <AttackAngel>6</AttackAngel>
    <LiftingFactor>0.826</LiftingFactor>
    <SweepingFactor>0.00876</SweepingFactor>
    <Rate>95.3</Rate>
  </Wings>
  <Wings>
    <Rid>54</Rid>
    <WingName>NACA 2412</WingName>
    <RValue>3100000</RValue>
    <AttackAngel>5</AttackAngel>
    <LiftingFactor>0.756</LiftingFactor>
    <SweepingFactor>0.00743</SweepingFactor>
    <Rate>101.74</Rate>
  </Wings>
  <Wings>
    <Rid>55</Rid>
    <WingName>NACA 2421</WingName>
    <RValue>2900000</RValue>
    <AttackAngel>10</AttackAngel>
    <LiftingFactor>1.06</LiftingFactor>
    <SweepingFactor>0.0114</SweepingFactor>
    <Rate>92.3</Rate>
  </Wings>
  <Wings>
    <Rid>56</Rid>
    <WingName>NACA 1408</WingName>
    <RValue>3000000</RValue>
    <AttackAngel>7</AttackAngel>
    <LiftingFactor>0.866</LiftingFactor>
    <SweepingFactor>0.0094</SweepingFactor>
    <Rate>92.78</Rate>
  </Wings>
  <Wings>
    <Rid>57</Rid>
```

EK-1 (Devam) Kanat simülatörü programının kodları

```
    <WingName>NACA 4412</WingName>
    <RValue>3000000</RValue>
    <AttackAngel>5</AttackAngel>
    <LiftingFactor>0.95</LiftingFactor>
    <SweepingFactor>0.0076</SweepingFactor>
    <Rate>125</Rate>
  </Wings>
</WingDatas>
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: ÖNDER, Murat
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 03.02.1980
Medeni hali	: Bekar
Telefon	: 0 312 212 67 67 / 401
e-mail	: muratonder@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	Niğde Üniversitesi /Makine Müh.	2002
Lise	Özel Alp Lisesi	1997

İş Deneyimi

Yıl	Yer	Görev
2003-.....	Gazi Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce