

İZMİR KÂTİP CELEBİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**RÜZGAR TÜRBİNLERİNDE DENEYSEL TORK ÖLÇÜMÜ İÇİN FARKLI
KONTROL ALGORİTMALARININ KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

Haydar ÜLKER

Sistem Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Ziya Haktan KARADENİZ

HAZİRAN 2017

İZMİR KÂTİP CELEBİ ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**RÜZGAR TÜRBİNLERİNDE DENEYSEL TORK ÖLÇÜMÜ İÇİN FARKLI
KONTROL ALGORİTMALARININ KARŞILAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

**Haydar ÜLKER
(600115023)**

Sistem Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Ziya Haktan KARADENİZ

HAZİRAN 2017

İKÇÜ, Fen Bilimleri Enstitüsü'nün **600115023** numaralı Yüksek Lisans Öğrencisi **Haydar ÜLKER**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**RÜZGAR TÜRBİNLERİNDE DENEYSEL TORK ÖLÇÜMÜ İÇİN FARKLI KONTROL ALGORİTMALARININ KARŞILAŞTIRILMASI**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Yrd. Doç. Dr. Ziya Haktan KARADENİZ**
İzmir Katip Çelebi Üniversitesi

Jüri Üyeleri : **Yrd. Doç. Dr. Ziya Haktan KARADENİZ**
İzmir Katip Çelebi Üniversitesi

Doç. Dr. Alpaslan TURGUT
Dokuz Eylül Üniversitesi

Yrd. Doç. Dr. Sercan ACARER
İzmir Katip Çelebi Üniversitesi

Teslim Tarihi : **15 Haziran 2017**
Savunma Tarihi : **30 Haziran 2017**

ÖNSÖZ

Günümüz dünyasında enerjiye olan ihtiyacın gün geçtikçe artması, aynı zamanda enerji üretiminde kullanılan katı ve fosil yakıtların çevreye verdiği zararları da artırmaktadır. Sürdürülebilir ve yenilenebilir enerji kaynaklarına geçiş artık zorunlu hale gelmiştir. Tarih boyunca çeşitli amaçlar için kullanılan rüzgar, sürdürülebilirlik ve yenilenebilirlik enerji kavramlarında temiz enerji olarak 1900'lü yılların sonunda 2000'li yılların başında ön plana çıkmıştır.

Rüzgar enerjisinden faydalanma verimliliğini artırmak için çeşitli tipte rüzgar türbinleri tasarlanmıştır. Bunlar genel olarak yatay ve dikey rüzgar türbinleri olarak sınıflandırılabilirler.

Bu çalışmada dikey rüzgar türbinleri tasarlama aşamasında tork ölçüm metodunun geliştirilmesi amaçlanmıştır.

Tez aşamasında yardım ve desteklerini esirgemeyen Yrd. Doç. Dr. Ziya Haktan KARADENİZ hocama, lisans bitirme projeleri kapsamında bu tezin alt yapısına yaptıkları katkıları için Fatih Tolga Özdemir, İshak Alakuş'a, İzmir Katip Çelebi Üniversitesi çalışanlarına ve Makina Mühendisleri Odası İzmir Şubesi Kalibrasyon Laboratuvarı ve Metroloji Eğitim Merkezine (MMO KALMEM) teşekkür ederim.

Özellikle çalışmalarım boyunca desteklerini her zaman hissettiğim eşim Ceyda'ya çocuklarım Berat Ege ve Alperen'e teşekkür ederim.

Haziran 2017

Haydar ÜLKER

İÇİNDEKİLER

Sayfa

ÖNSÖZ	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
SEMBOLLER	xi
ÇİZELGE LİSTESİ	xiii
ŞEKİL LİSTESİ	xv
ÖZET	xix
SUMMARY	xxi
1. TANIMLAR	1
1.1 Rüzgarın Oluşumu ve Önemi.....	1
1.2 Dikey Eksenli Darrieus Rüzgar Turbinleri	1
1.3 Literatür Araştırması.....	9
1.3.1 Rüzgar türbinlerinin laboratuvarında tork ölçümü	10
1.3.2 Rüzgar türbinlerinde kullanılan kontrol yöntemleri	21
2. METERYAL VE METODLAR	25
2.1 Kanat Yapısı.....	25
2.2 Rüzgar Test Tüneli	26
2.3 FN38 DC Motor ve Encoder.....	28
2.4 Güç Kaynağı ve Motor Sürücü Kartı.....	29
2.5 Sinyal Genişlik Modülasyonu (Pulse Width Modulation (PWM))	32
2.6 Arduino Mega 2560 ve PWM uygulaması	35
2.7 Denetim Yöntemi	37
2.7.1 Oransal- İntegral-Türevsel (PID) Denetim yöntemi	37
2.7.2 Oransal-İntegral (PI) Denetim yöntemi	38
2.8 Arduinio PID ve Visual Studio Arayüz Programı.....	40
3. Bulgular ve Değerlendirme	49
3.1 Deney Yöntemi	49
3.2 Veri İşleme	54
3.3 Farklı Denetim Yöntemlerinden Alınan Sonuçların Karşılaştırılması	60
3.4 Türbin Karakterizasyonu	68
3.4.1 Küçük kanat türbin karakterizasyonu	68
3.4.2 Büyük kanat türbin karakterizasyonu	75
4. SONUÇ OLARAK	81
4.1 Sonuç ve Değerlendirme.....	81
KAYNAKLAR	83
EKLER	85
ÖZGEÇMİŞ	101

KISALTMALAR

CFD	: Computational fluid dynamics
HAWT	: Horizontal axis wind turbine
LEV	: Leading edge vortex
TEV	: Trailing edge vortex
VAWT	: Vertical axis wind turbine
DETR	: Düşey eksenli rüzgar türbini
TSR	: Tip speed ratio
PGHÖ	: Parçacık görüntülemeli hız ölçümü

SEMBOLLER

A	: Kanat süpürme alanı
c	: Kanat veter uzunluğu
C_{power}	: güç katsayısı
N	: Kanat sayısı
P	: Güç
Q	: Rotor torku
R	: Rotor çapı
Re	: Reynold sayısı
I_{rig}	: Rotor dönüşü atalet momenti
L	: Kanat uzunluğu
P_B	: Kanat gücü
P_w	: Rüzgar gücü
T_{app}	: Uygulanan fren torku
T_B	: Kanat torku
T_{res}	: Direnç torku
r_f	: Akışkan yoğunluğu
V_w	: Serbes akış rüzgar hızı
v_s	: Ayarlanan hız
μ	: Akışkan viscositesi
x	: Rotor açısal hızlanması
ρ	: Hava yoğunluğu
s	: Rotor katılık oranı, Nc/r
λ	: Uç hız oranı
U_{local}	: Serbest rüzgar hızı
U_{eff}	: Etkin rüzgar hızı

ÇİZELGE LİSTESİ

Sayfa

Çizelge 1.1	: Deneysel ve sayısal olarak motor sürücülü deneyler	10
Çizelge 2.1	: Motor sürücü kartının giriş ve çıkış pinleri	31
Çizelge 3.1	: Kanat uç hız oranı Çalışma Sahası	52
Çizelge 3.2	: 100-1200 RPM Aralığında çizgisel Hız Re Sayısı	53
Çizelge 3.3	: 1300-2100 RPM Aralığında çizgisel Hız Re Sayısı	53
Çizelge 3.4	: Deney uygulama planı	61
Çizelge 3.5	: Büyük kanat PID ve PI değerlendirmesi	62
Çizelge 3.6	: Küçük kanat PID ve PI değerlendirmesi	63

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1	: Yatay eksenli rüzgar türbinleri.....	2
Şekil 1.2	: Küçük ölçekli yatay eksenli rüzgar türbini.....	3
Şekil 1.3	: Darrieus rüzgar türbini.....	3
Şekil 1.4	: Darrieus H-rotor rüzgar türbini.....	4
Şekil 1.5	: Kanat profili üzerindeki oluşan hız ve kuvvet vektörleri.....	5
Şekil 1.6	: Dikey eksenli rüzgar türbininde kanada etkiyen hız ve kuvvet bileşenleri.....	6
Şekil 1.7	: Farklı rüzgar türbinleri için kanat uç hız oranına göre (λ) güç Katsayısı (C_p) grafikleri.....	9
Şekil 1.8	: Su Kanalı Tesisi.....	10
Şekil 1.9	: (a) Türbin Doğal Eğrisi, sabir bir Re sayısı için türbin doğal eğrisinin (b) altında kalan ve (c) üstünde kalan ölçüm noktaları.....	11
Şekil 1.10	: Akış sürücülü ve motor sürücülü durumlar için ölçülen tork değerleri.....	12
Şekil 1.11	: NACA0021 kanat profili ile yapılan iki kanatlı dikey rüzgar türbin testi.....	13
Şekil 1.12:	(a) NACA0021 kanat profili C_p /kanat uç hız oranı (Qing'an Li vd. 2016), (b) Re sayısı C_p /kanat uç hız oranı(Qing'an Li vd. 2016).....	14
Şekil 1.13	: Birleştirilmiş C_p /kanat uç hız oranı grafiği.....	14
Şekil 1.14	: Farklı rüzgar hızlarında pürüzlü ve pürüzsüz kanatlara ait C_p /kanat uç hız oranı.....	15
Şekil 1.15	: Sheffield üniversitesi rüzgar tüneli tesisi.....	16
Şekil 1.16	: (a) Kanat 1 modeli için üç rüzgar hızındaki test sonucu güç katsayısı – kanat uç hız oranı grafiği. (b) Kanat 2 modeli için üç rüzgar hızındaki test sonucu güç katsayısı –kanat uç hız oranı grafiği. (c) Kanat 1 ve 2 modeli için C_p /kanat uç hız oranı grafiği.....	18
Şekil 1.17	: Düz üç kanatlı DERT ve spiral üç kanatlı DERT testleri.....	19
Şekil 1.18	: CFD simülasyonu Tork/zaman ve C_p /kanat uç hız oranı grafiği.....	20
Şekil 1.19	: CFD simülasyonu Tork/RPM grafiği.....	20
Şekil 1.20	: Tekli DERT.....	21
Şekil 1.21	: Çoklu DERT.....	21
Şekil 2.1	: NACA 0018 profilli türbin modeli ve sehba kaidesi.....	25
Şekil 2.2	: Rüzgar Test Tüneli.....	26
Şekil 2.3	: Darrieus tipi türbinin ölçüleri ve test oryantasyonu.....	26
Şekil 2.4	: Rüzgar Test Tüneli boyutları.....	27
Şekil 2.5	: Canon FN-38 DC Motor+Encoder Özellikleri.....	28
Şekil 2.6	:(a) TT-Technic marka RXN-3020D modeli güç kaynağı. (b) HP-250 20A H-Köprü DC Motor Sürücü devresi.....	29
Şekil 2.7	: Motor sürücü devresi açık şeması.....	30

Şekil 2.8	: Kare dalga.	33
Şekil 2.9	:Sinyal Oranı (Duty Cycle) Arduino PWM çıkışları % 25, % 50, % 75,% 100 olduğunda kara dalga genişlikleri	34
Şekil 2.10	: Arduino Mega 2560	36
Şekil 2.11	: PID prensip şeması	37
Şekil 2.12	: PI prensip şeması	38
Şekil 2.13	: Program Akış Şeması.....	39
Şekil 2.14	: Motor sürücü devresi Motor+Encoder ve arduino bağlantı şeması....	40
Şekil 2.15	: Arayüz programı	42
Şekil 2.16	: Bağlantı ayarları bölümü	43
Şekil 2.17	: PID parametreleri ayarlama bölümü	44
Şekil 2.18	: Motor ayarları bölümü	44
Şekil 2.19	: Anlık veri göstergeleri Kısmı	45
Şekil 2.20	:Veri görselleştirme bölümü	45
Şekil 2.21	: Listeler sekmesi	46
Şekil 3.1	: Makine Mühendisleri Odası KALMEM Rüzgar Tüneli.....	49
Şekil 3.2	: Deney Setinin Prensip Şeması.....	50
Şekil 3.3	: Çalışma Ortamındaki karışıklığı önlemek için elektronik donanımlar kutu içerisine yerleştirilmiştir.....	51
Şekil 3.4	: Örnek Okuma Hatası İçeren RPM Grafiği.....	54
Şekil 3.5	: Yüksüz durum için PID kontrolcü ile 100 dev/d'ya sabitlenen dönme hızının zamanla değişimi.....	55
Şekil 3.6	: Yüksüz durum için PID kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için gerilimin zamanla değişimi.....	55
Şekil 3.7	: Yüksüz durum için PID kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için akımın zamanla değişimi	55
Şekil 3.8	: Yüksüz durum için PI kontrolcü ile 100 dev/d'ya sabitlenen dönme hızının zamanla değişimi.....	56
Şekil 3.9	: Yüksüz durum için PI kontrolcü ile 100 dev/d'ya sabitlenen dönmehızı için gerilimin zamanla değişimi.....	56
Şekil 3.10	: Yüksüz durum için PI kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için akımın zamanla değişimi.	56
Şekil 3.11	: (a) PID ve (b) PI kontrolcülerinde 100 dev/d için yazılım tarafından hesaplanan güç değerleri.	57
Şekil 3.12	: (a) PID ve (b) PI kontrolcülerinde 100 dev/d için yazılım tarafından hesaplanan Tork değerleri.	58
Şekil 3.13	: Te: Elektriksel Tork, Tm: Mekaniki Tork, Ta: Aerodinamik Tork oluşum bölgeleri.	59
Şekil 3.14	: Boşta Büyük Kanat PID ve PI dev/d verileri.	62
Şekil 3.15	: Boşta Küçük Kanat PID ve dev/d verileri.....	62
Şekil 3.16	: 3 m/s rüzgar hızında PID ve PI kontrollü Tork- dev/d verileri.	63
Şekil 3.17	: 5 m/s rüzgar hızında PID ve PI kontrollü Tork- dev/d verileri.	64
Şekil 3.18	: (a) 7 m/s PI, (b) 8 m/s PID, (c) 10 m/s PI, (d) 11 m/s PID, (e) 15 m/s PID, (f) 14 m/s PID Kontrolcülü Tork- dev/d verileri.	65
Şekil 3.19	: DC motor güç verileri.	66
Şekil 3.20	: (a) 1000 dev/d , (b) 1500 dev/d , (c) 2000 dev/d artan rüzgar hızında tork verileri.....	.67
Şekil 3.21	: 300 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.....	69

Şekil 3.22	: 600 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.....	71
Şekil 3.23	: 1200 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.....	72
Şekil 3.24	: 1500 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.....	74
Şekil 3.25	: Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.	75
Şekil 3.26	: Farklı RPM değerlerinde Darrieus Boşta Çalışma Güç Tüketimi.	76
Şekil 3.27	: 3 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.	76
Şekil 3.28	: 5 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği	76
Şekil 3.29	: 8 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.	77
Şekil 3.30	: 11 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.	77
Şekil 3.31	: 14 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği	77
Şekil 3.32	: Büyük Kanat PID kontrolcü Darrieus Re sayısının Cp/kanat uç hız oranı Grafiği.....	79

RÜZGAR TÜRBİNLERİNDE DENEYSEL TORK ÖLÇÜMÜ İÇİN FARKLI KONTROL ALGORİTMALARININ KARŞILAŞTIRILMASI

ÖZET

Enerjinin korunum yasasına göre rüzgardan kaynaklanan hareket enerjisini rüzgar türbinleri vasıtasıyla jeneratöre aktarıp, jeneratörün tasarımı sayesinde bu hareket enerjisini, elektrik enerjisine dönüştürme işlemine rüzgar türbinlerinden elektrik üretme olarak tanımlıyoruz.

1980'den günümüze kadar özellikle son yıllarda rüzgar türbinlerinin verimliliğini artırmak için yapılan bilimsel çalışmalarda elektrik üretimi için kullanılan rüzgar türbinlerini yatay ve dikey olarak iki gruba ayırmak mümkündür.

Yatay olarak tasarlanan rüzgar türbinleri daha büyük güç üretmeleri için tasarlanmış ticari işletmeler tarafından yönetilen tesislerde kullanılmaktadırlar.

Dikey rüzgar türbinleri ise boyutları genelde küçük, düşük güç üretmeleri için tasarlanmış kurulum ve kullanım kolaylıkları bakımından bireysel ihtiyaçlara cevap verecek niteliktedirler. Bu özelliklerinden dolayı bu tip rüzgar türbinlerinin gelecek yıllarda kullanımının artacağı değerlendirilmektedir.

Rüzgar türbinlerinin önemli bir karakteristiği kanat uç hız oranı ile güç ve verim ilişkisini veren grafikleridir. Bu grafikler, kanat uç hızı ve rüzgar hızı değerlerinin oranı sabit tutularak rüzgar yükünün doğurduğu torkun ölçülmesi veya bu şartlar altında yapılan matematiksel analizler ile torkun hesaplanması ile elde edilmektedir. Bu eğriler deneysel olarak ve gerçek çalışma şartlarında ölçülmeye çalışıldığında, ölçülecek tork değerlerinin ve mekanik yapıların büyüklüğü direk tork ölçümünü zorlaştırmaktadır. Direk tork ölçümünün gerçek ölçekteki türbinlerde uygulanmasının zor olması sebebi ile araştırmacılar, mevcut sistemlerin farklı alt sistemlerindeki ölçülebilir büyüklükleri kullanarak rüzgar torkunun gerçek zamanlı izlenmesi için gözlemleyeciler önermişlerdir. Bu yöntemlerin en yaygını elektrik makinalarının kontrolü için rüzgar torkunun bulunması için tasarlanan

gözlemleyecilerdir. Hassas Tork kontrolünün amacı elektrik makinasının veriminin yükseltilmesidir.

Bu tezde deneysel türbin karakterizasyonunda tork metre kullanmadan kontrol algoritması metodu ile tork ölçme ve sistemin torkunu hesapladıktan sonra türbin karakterizasyonunu hesaplama yöntemi denenmiştir.

Bu kontrol algoritması hassas hız kontrollü motor ile sürülen PID ve PI sistemi ile oluşturulmuştur. Bu sayede istenilen kanat uç hız oranının sağlanması ile ilgili var olan sistemlerden daha yüksek bir başarımla sağlanmıştır. Hassas hız kontrolü için kullanılacak bu sistemler, anlık olarak rüzgar yükünü belirleyeceğinden sistemde ayrıca bir tork sensörü bağlanması gerekmemektedir. Her iki sistemin sonuçları karşılaştırılıp türbin deneylerindeki en uygun yöntem belirlenmiştir.

Sistem için en iyi kontrol algoritması belirlendikten sonra deneyin ikinci aşaması olan türbin karakterizasyon işlemi iki kanat yapısı ile farklı rüzgar hızlarında sabit RPM'de ve Sabit Reynold sayısı için sabit rüzgar hızlarında olacak şekilde deneyler yapılmış bu deneylerin sonuçları C_p /kanat uç hız oranı grafikleri ile yorumlanmıştır. Sonuç olarak tezin kurgusu olan rüzgar türbinlerinde deneysel tork ölçümü için farklı kontrol algoritmalarının karşılaştırılmış, sistemin torku tork metre olmadan ölçülmüş ve ikinci aşamada farklı türbin yapılarının karakterizasyon ölçümleri yapılmıştır.

Bu çalışmada dikkat edilmesi gereken konu ölçeklendirilen kanat modeline uygun tepkimeler verecek güçte motor ve seçilen bu motora uygun elektronik ekipmanlar seçilmelidir.

COMPARISON OF DIFFERENT CONTROL ALGORITHMS FOR EXPERIMENTAL TORQUE MEASUREMENT IN WIND TURBINES

SUMMARY

According to the conservation law of energy, we define the movement energy originating from the wind as a generator through wind turbines and design this generator as the generation of electricity from the wind turbines to convert this movement energy into electric energy.

It is possible to separate two wind turbines horizontally and vertically from the wind turbines used for electricity generation in the scientific studies to increase the efficiency of the wind turbines, especially from 1980 to the present day.

Horizontally designed wind turbines are used in installations managed by commercial enterprises designed to generate greater power.

Vertical wind turbines will respond to individual needs in terms of installation and ease of use, which are usually designed for small, low power generation. Due to these features, it is estimated that the use of such wind turbines will increase in the coming years.

An important characteristic of wind turbines is the graph that shows the relationship between blade tip speed ratio and power and efficiency. These graphs are obtained by measuring the torque generated by the wind load by keeping the ratio of the blade tip speed and the wind speed constant, or by calculating the torque by the mathematical analysis made under these conditions. When trying to measure these curves experimentally and under actual operating conditions, the torque values to be measured and the size of the mechanical structures make direct torque measurement difficult. Because direct torque measurement is difficult to implement in real-world turbines, researchers have suggested that observers use real-time tracking of wind torque using measurable quantities in different subsystems of existing systems. The most common of these methods are observers designed to find wind turbines for

control of electric machines. Precision Torque control is the purpose of increasing the efficiency of the electric machine.

The experimental part of the comparison of control algorithms for experimental torque measurement in wind turbines consists of two parts, mechanically and electronically. The mechanical part is the interface program written in Darrieus 3-wing wind turbine and shovel, and the electronic part is motor, encoder, motor drive, arduino mega 2560, arduino software and visual studio. In order for the thesis to be able to draw conclusions from the experimental section, it is necessary that these two parts work in harmony with one another. Before the test phase, 2 different power engines and 2 different types of wings were tested and the engine set which responded to the wind speeds according to the turbine scale was selected.

In the experiments, firstly different control algorithms were concluded to be more successful than PID algorithm and PID algorithm compared to PID algorithm and PI algorithm. This result was then passed to the second stage and the PID controller for the turbine characterization, the fixed RPM variable wind speed for the Darrieus turbine with a veterinary length of 2 cm (small wing) and a veterinary length of 4 cm (large wing) These data of DC motor RPM, current, voltage, torque, PWM information are recorded in txt format and converted to graphics with excel program. During this process, the obvious reading errors caused by the synchronization error between the interface and the arduino pragma have been corrected manually.

Power efficiency (C_p) is obtained by dividing the turbine power into wind power. The Reynolds number and the number of Reynolds numbers that determine the characterization of the flow that causes the fluid flow through the fluid are the basic parameters used in the wind turbine characterization. The power is calculated by multiplying the current and the voltage by the data taken in the experiment and the power and rpm values are calculated in the field. In this experiment, torque measurement without torque meter was performed in experimental turbine tests, and C_p / tip speed ratio graphs and turbine characteristics were investigated in the second stage.

The C_p / tip speed ratio graphs for the turbine characterization were generated at pre-planned RPM and wind speeds. Negative and positive points were obtained according to the tip speed ratio of C_p , but the transition points from the positive to the negative region were not precisely determined. In future studies, the graphics resolution can

be increased by taking more data between transition zones. The experiments in this thesis for the next study are an important data quality.

The interface program written in visual studio in the controller section of the system can be improved and the reading error can be reduced to minimum due to the processor speed of the arduino mega card and the requirements of the motor + encoder system in the current system, with the belief that the C_p / TSR graphs obtained for future studies can be used as important data. A 96-pulse encoder was used. 400 pulse encoder for more precise operation of the installed system and operation with this data processing speed

1. TANIMLAR

1.1 Rüzgarın Oluşumu ve Önemi

Hava hareketlerinin temel prensibi, güneşten alınan ısı nedeniyle mevcut atmosfer basıncının bölgeler arasında değişmesidir. Rüzgar, alçak basınçla yüksek basınç bölgeleri arasında yer değiştiren hava akımıdır. Rüzgar daima yüksek basınç alanından alçak basınç alanına doğru gider. İki bölge arasındaki basınç farkı ne kadar büyük olursa, hava akımı hızı o kadar fazla olur. Rüzgar enerjisinden günümüzde modern rüzgar türbinleri ile elektrik enerjisi üretimi de gerçekleştirilmektedir. Enerjinin korunum yasasına göre rüzgardan kaynaklanan hareket enerjisini rüzgar türbinleri vasıtasıyla jeneratöre aktarır, jeneratörün tasarımı sayesinde bu hareket enerjisini, elektrik enerjisine dönüştürme işlemine rüzgar türbinlerinden elektrik üretme olarak tanımlıyoruz. Üretilen elektrik enerjisi bölgesel kullanılabildiği gibi aynı zamanda elektrik şebekesine bağlantısı gerçekleştirerek birlikte kullanılabilmektedir.

1.2 Rüzgar Türbinleri

Rüzgar türbinleri, rüzgarın kinetik gücünü kanatlar ve rotor ile kullanılabilir güce dönüştüren makinelerdir.

Rüzgar gücüyle hareket eden gemiler, değirmenler, su pompaları ve harman makineleri de rüzgardan elde edilen gücün uygulamalarına diğer örneklerdir.

Teknoloji ve sanayi devrimiyle, 20. yüzyılın sonlarında rüzgar gücünü verimli şekilde kullanan makinelerin ortaya çıktığı görülmektedir.

a. Rüzgar Türbinleri Çeşitleri

Rüzgar türbinleri genellikle rotor dönme ekseninin rüzgara ile yaptığı açı ile, yatay ve düşey eksenli rüzgar türbinleri olarak sınıflandırılır.

a1. Yatay Eksenli Rüzgar Türbinleri

Bu tip türbinlerde, dönme eksenini rüzgar yönüne paralel, kanatlar rüzgar yönüne diktir. Yatay eksenli rüzgar türbinleri Şekil 1.1'de görüldüğü gibi günümüzde kullanımı en yaygın olan rüzgar türbinleri çeşididir.



Şekil 1.1: Yatay eksenli rüzgar türbinleri, Sarpıncık, İzmir (Fotoğraf, Z. Haktan Karadeniz).

Yatay eksenli rüzgar türbinlerinde kanatların rüzgarı dik alabilmesi için yönlendirme mekanizması vardır. Bu yönlendirme mekanizması mega-watt ölçeğindeki büyük türbinlerde naselin üzerinde rüzgar yönünü ölçen anemometre ile ölçülüp, rüzgarın yönüne göre kanatlar dişli mekanizmaları ile rüzgara dik açığa getirilmektedir. Küçük ölçekli yatay eksenli rüzgar türbinlerinde ise bu işlem kanatların arkasındaki kuyruk ile sağlanmaktadır (Şekil 1.2).



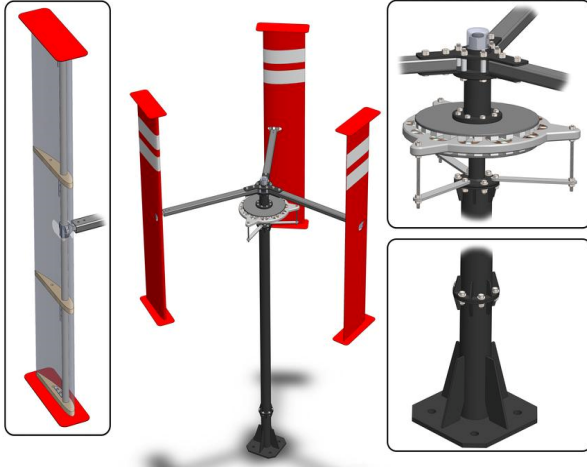
Şekil 1.2: Küçük ölçekli yatay eksenli rüzgar türbini.

a2. Düşey Eksenli Rüzgar Türbinleri

Bu tip rüzgar türbinlerinde rotor eksenli rüzgar yönüne diktir. Bu sayede rüzgara karşı yön alma gibi bir düzeneğe ihtiyaç yoktur rüzgar kanatlara hangi yönden gelirse gelsin önemli değildir. Şekil 1.3'te Darrieus tipi Şekil 1.4'te Darrieus H-rotor tipi rüzgar türbini gösterilmiştir.



Şekil 1.3: Darrieus rüzgar türbini.



(a)



(b)



(c)



(d)

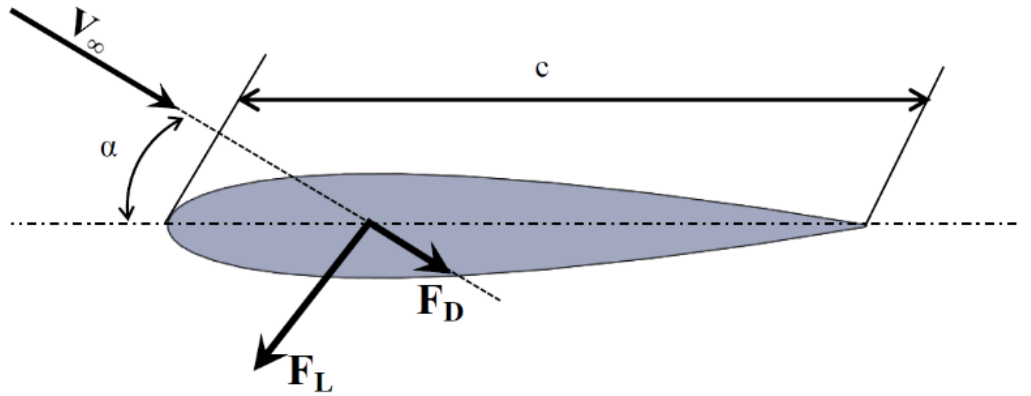
Şekil 1.4: Darrieus H-rotor rüzgar türbinleri (Url-1).

Düşey eksenli rüzgar türbinleri kentsel kullanım imkanı ve fazla alana kaplamaması sayesinde rüzgardan elektrik üretiminde gittikçe önem kazanan türbinlerdir. Ancak düşey eksenli rüzgar türbinlerinin yatay eksenli rüzgar türbinlerine kıyasla daha az verimli olduğu bilinmektedir.

Bir çeşit dikey eksenli rüzgar türbini olan Darrieus rüzgar türbini geniş hız aralıklarında işletmeye elverişlidir.

b. Darrieus Rüzgar Türbin Aerodinamiği

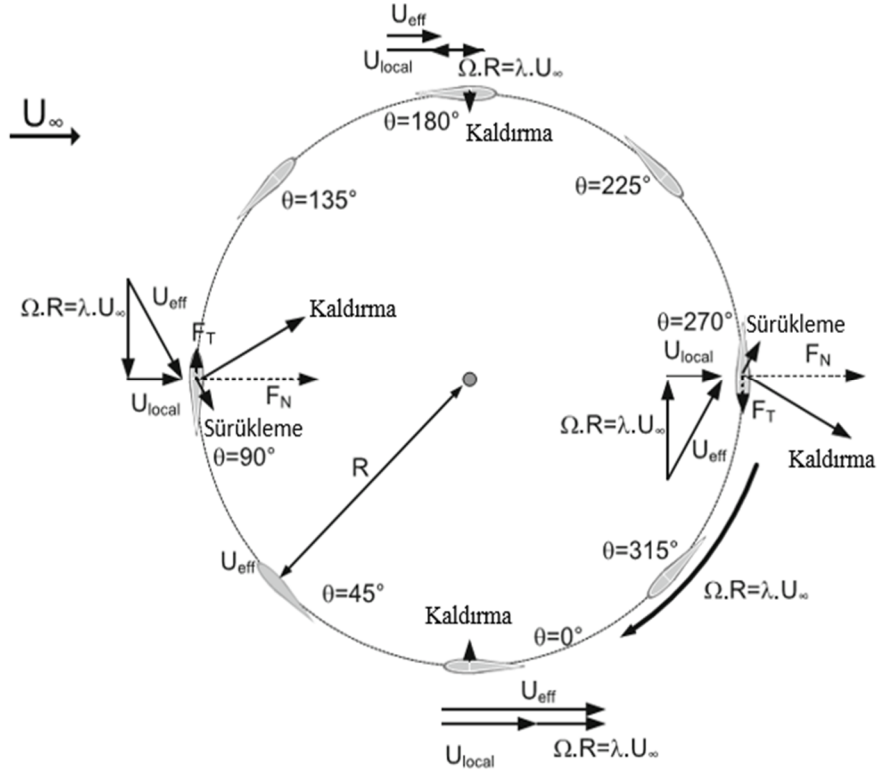
Modern rüzgar türbinlerinin kanat profilleri, kanat üzerinde meydana gelen yükleri optimum güç elde edilmesi amacıyla geliştirilmiş olan özel profillerden (airfoil) seçilmektedir. Bu profillerin geliştirilmesi ve seçilmesinde amaç kanadın kaldırma kuvvetinin iyileştirilmesidir. Profiller, bir giriş hattı üzerinde ve altında tanımlanmış iki özel eğriden oluşmaktadır. Hücum açısına bağlı olarak, rüzgar bu eğriler üzerinde yüksek ve düşük basınç alanları oluşturmakta ve bu basınç alanları arasındaki fark kanada kuvvet olarak yansımaktadır. Bir serbest akım içerisinde ve belirli bir hücum açısında (α) örnek bir profilin maruz kaldığı kuvvetler Şekil 1.5'te görülmektedir. Şekil 1.5'te görüldüğü gibi α hücum açısında gelen bir rüzgar hızı, kendisine paralel yönde bir sürükleme kuvvetine ve kendisine dik açı yapacak şekilde bir kaldırma kuvvetine sebep olmaktadır.



Şekil 1.5: Kanat profili üzerindeki oluşan hız ve kuvvet vektörleri (Dağdevir 2015).

Darrieus rüzgar türbininde hız ve kuvvet vektörleri detaylı olarak Şekil 1.6'da gösterilmiştir. Burada iki adet önemli hız bileşeni vardır; birincisi, "c" veter uzunluğuna her zaman paralel olan ve rotorun açısal hızı ile rotor yarıçapı çarpımına eşit olan kanadın çizgisel hızı ($\omega \cdot R$), ikincisi kanatlara tesir eden ve yönü ve büyüklüğü sabit olarak kabul edilen serbest rüzgar hızı (U_{local}). Bu iki hızın sonucu olarak kanada etki eden bağıl hız (U_{eff}) ortaya çıkmaktadır (Denklem 1.1.)

$$\vec{U}_{eff} = \vec{U}_{local} + \vec{\omega} * \vec{R} \quad (1.1)$$



Şekil 1.6: Dikey eksenli rüzgar türbininde kanada etkiyen hız ve kuvvet bileşenleri (Ferreira 2008).

Tek bir kanat üzerinde gerçekleşen kaldırma ve sürüklenme kuvvetlerini etkileyen diğer parametre ise hücum açısıdır. Simetrik olan kanat profillerinde hücum açısı 0° olduğunda kaldırma kuvveti yoktur ve sürüklenme kuvveti en azdır. Hücum açısının değişimi ile kanada etki eden kaldırma ve sürüklenme kuvvetleri değişir. Darrieus tipi rüzgar türbinlerinde kanat sabit bir nokta etrafında döndüğünden, kanada etki eden bağıl hız vektörü kanadın konumuna bağlı olarak değişir. Değişim periyodiktir bu nedenle bir tur boyunca hücum açısının değişimi belirlendiğinde, kanada etkiyen kaldırma ve sürüklenme kuvvetleri de kanadın merkez etrafındaki konumuna göre belirlenebilir.

Şekil 1.6’da görüldüğü gibi kanatlara etkiyen sürüklenme kuvveti, kaldırma kuvvetine dik olacak şekilde oluşmaktadır. Ancak türbin mili etrafındaki momentin hesaplanabilmesi için, kanadın her açısal konumunda kanat yörüngesine teğet kuvvetin belirlenmesi gerekir. Bu nedenle, her bir açısal konum için kanat üzerindeki kaldırma ve sürüklenme kuvvetleri, teğetsel ve normal bileşenlerine ayrılarak kanadın üzerindeki toplam teğetsel ve toplam normal kuvvetler belirlenir. Normal kuvvetin, rotor mili üzerinde döndürme etkisi yoktur. Konuma bağlı teğetsel kuvvet değeri belirlendikten sonra, bir tam tur için teğetsel kuvvetin rotor mili üzerinde oluşturduğu ortalama moment (tork) hesaplanarak türbinin üreteceği güç belirlenebilir.

c. Performans Parametreleri

Rüzgar türbinleri tasarlanırken göz önüne alınması gereken birçok parametre söz konusudur. Bunları türbinin kurulacağı düşünüldüğü yerin rüzgar potansiyeli, bu rüzgar potansiyeline uygun kanat profili, hücum açısı, iklim şartları, türbinden elde edilmek istenen güç miktarı ve kullanım amacı olarak sıralayabiliriz. Rüzgar türbinleri genellikle birimsiz performans parametreleri ile karakterize edilirler.

Kanat uç hız oranı, kanadın çizgisel hızının ($\omega \cdot R$) serbest rüzgar hızına oranı olarak tanımlanır ve λ ile sembolize edilir (Denklem 1.2). Akış karakteristiğini belirlemede temel gösterge olan Reynolds sayısı denklem 1.3’ te verilmiştir.

$$\lambda = \frac{\omega R}{V_{\infty}} \quad (1.2)$$

$$Re = \frac{\rho c U}{\mu} \quad (1.3)$$

burada ω türbinin açısal dönme hızı, R rotor yarıçapını, V_{∞} serbest rüzgar hızını (Denklem 1.2), ρ akışkan yoğunluğu, U_{eff} kanada etkiyen bağlı hızı (Denklem 1.1), c karakteristik uzunluk olarak kanadın veter uzunluğunu, μ ise akışkanın dinamik viskozite değerini (Denklem 1.3) ifade etmektedir.

c1. Rüzgardan Alınan Güç

Rüzgar sahip olduğu hız nedeniyle kinetik enerjiye sahiptir. Rüzgarın bu gücü, rüzgarın hızı, yoğunluğu, havanın sıcaklığı ve geçtiği kesit alanına bağlıdır. Kinetik enerji ve debi denklemlerinden rüzgarın gücü denklemi kolaylıkla elde edilebilir. Kinetik enerji:

$$E = \frac{1}{2} m V_{\infty}^2 \quad (1.4)$$

$$\dot{E} = P_{rüzgar} = \frac{1}{2} \dot{m} V_{\infty}^2 \quad (1.5)$$

Kütleli debi:

$$\dot{m} = \rho A \frac{dx}{dt} = \rho A V_{\infty} \quad (1.6)$$

Değerler yerine yazıldığında kuramsal olarak V rüzgar hızındaki A alanından geçen rüzgarın sahip olduğu güç:

$$P_{rüzgar} = \frac{1}{2} \rho A V_{\infty}^3 \quad (1.7)$$

Bu denklemden gücün yoğunluk ve süpürülen alanın birinci, hızın üçüncü kuvvetiyle orantılı olduğu görülebilmektedir. Denklem C_p güç katsayısı ile çarpılarak rüzgardan türbine aktarılan güç oranını ifade etmektedir.

$$P_{türbin} = \frac{1}{2} \rho A V^3 C_p \quad (1.8)$$

Türbinin verimini ifade eden güç katsayısı (C_p) denklem 1.9'da tanımlanmıştır.

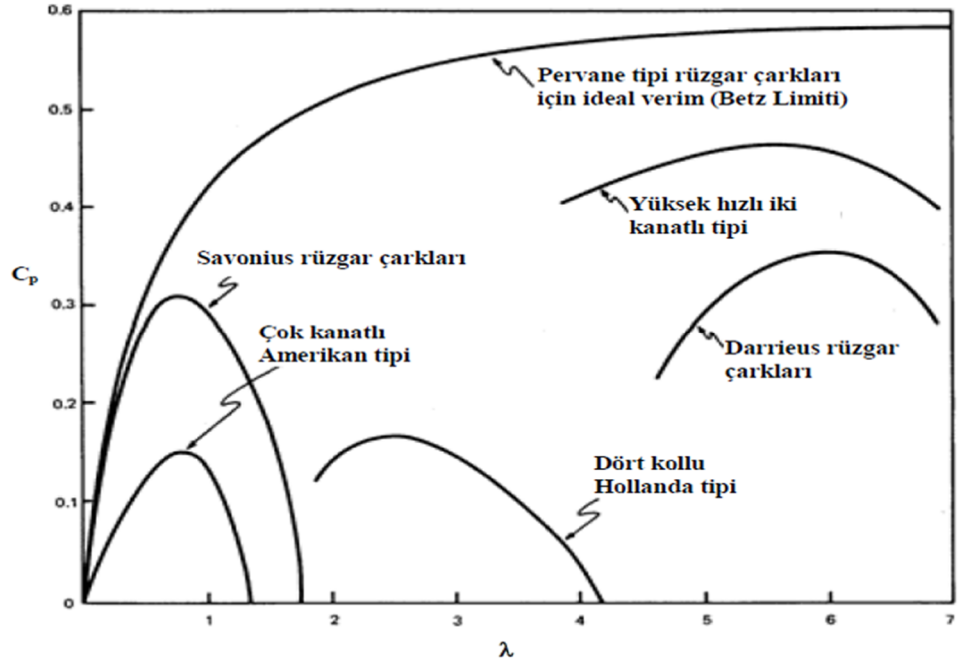
$$C_p = \frac{P_{türbin}}{P_{rüzgar}} = \frac{T\omega}{\frac{1}{2} \rho A V_{\infty}^3} \quad (1.9)$$

Burada T ; türbinden elde edilen tork, ω türbinin açısal dönme hızı, A türbinin süpürdüğü rüzgar alanıdır. Günümüzde kullanımı olan rüzgar türbinlerini güç katsayılarının, kanat uç hızı oranı ile değişimi Şekil 1.7’de verilmiştir. T torkuyla dönen bir türbinin tork katsayısı Denklem 1.10’da ifade edilmiştir. Burada da R türbinin rotor yarıçapını ifade etmektedir.

$$C_T = \frac{T}{\frac{1}{2}\rho R A V_\infty^2} \quad (1.10)$$

Türbinin geometrik parametrelerinden bir diğeri de katılık oranıdır. Katılık oranı σ ile sembol edilir ve kanatların alanın kanatların süpürdüğü toplam alana oranı olarak tanımlanır:

$$\sigma = \frac{N \cdot c}{R} \quad (1.11)$$



Şekil 1.7: Farklı rüzgar türbinleri için kanat uç hız oranına göre (λ) güç katsayısı (C_p) grafikleri.(Ur1-2)

1.3 Literatür Araştırması

Darrieus tipi dikey eksenli rüzgar türbininin deneysel tork ölçümünde farklı kontrol algoritmalarının karşılaştırmasını yapmak için temel bilgiler ve türbinin çalışma

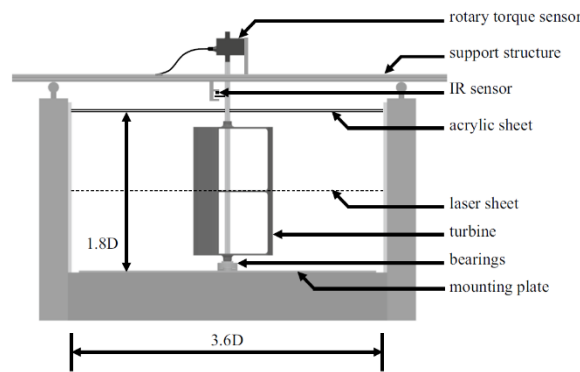
prensibi bir önceki kısımda anlatılmıştır. Literatürdeki deneysel ve sayısal olarak yapılan çalışmalardan, tork belirleyerek türbinin performansını ölçmek için güç katsayısı belirleyen bazı çalışmalar Çizelge 1.1’de verilmiştir. Bu çalışmalar ile ilgili ayrıntılar aşağıda sunulmuştur.

Çizelge 1.1: Deneysel ve sayısal olarak motor sürücülü deneyler.

	YERT	DERT	Sayısal	Deneysel	Akış Sürücülü	Motor sürücülü	Tork Gücü Raporu
Duque et al. (2003)	X		X			X	X
Bahaj et al. (2007)	X		X	X	X	X	X
Chao and Dam	X		X			X	X
Lida et al. (2007)		X	X			X	X
Howell et al. (2010)		X	X	X	X	X	X
Martinelli and Smits		X	X			X	X
Bachant and Wosnik		X	X	X	X	X	X
Korobenko et al.		X	X			X	X
Le et al. (2014)		X	X		X	X	X
Araya et al. (2015)		X	X	X	X	X	X
Qing'an Li vd.(2016)		X		X	X		X
Howell vd. (2016)		X	X	X	X		X

1.3.1 Rüzgar türbinlerinde laboratuvarında tork ölçümü

Araya vd. (2015) akış ve motor sürücülü türbini, belirlenen hızlarda su kanalı tesisinde döndürülmüş ve türbin türbülansının deneysel sonuçları karşılaştırılmıştır.

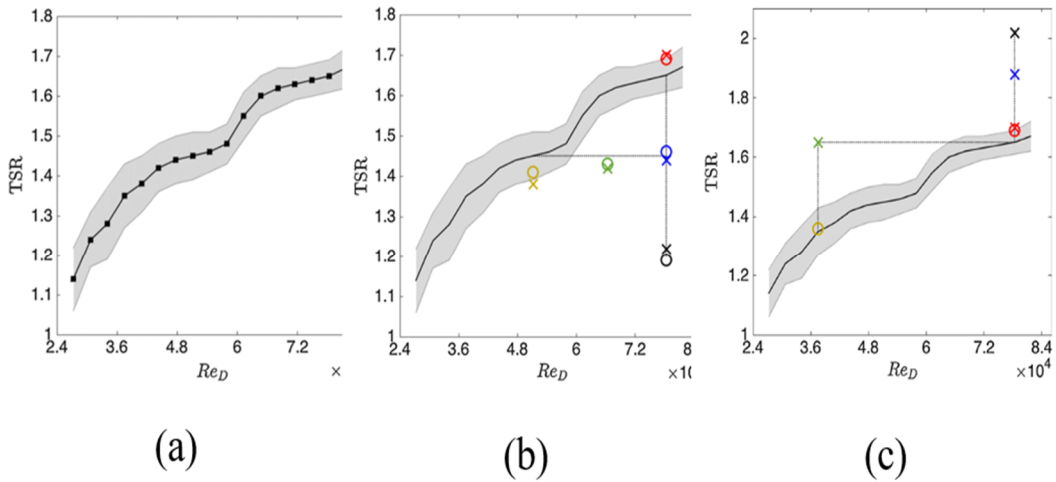


Şekil 1.8: Su Kanalı Tesisi (Araya 2015).

Çalışmada, Şekil 1.8’de gösterilen su kanalı tesisi kurulmuş bu kanala 3 kanatlı türbin şekildeki gibi yerleştirilmiş, akış ve motor sürücülü farklı hız gruplarında üç kanatlı düşey eksenli rüzgar türbini (DERT) çevresindeki akış alanını incelemek için 2D

parçacık görüntülemeli hız ölçümü (PGHÖ) kullanılmış ve akış ile sürülen ve motor ile sürülen rotor durumları için ortalama hız, hız-güç spektrumları, türbülans sirkülasyonu ve rotor torku ölçümleri karşılaştırılmıştır. Bu çalışmada rotor torkunu ölçmek için türbin milinin üstüne dijital enkoderli (Futek TRS705) döner bir tork sensörü bağlanmıştır.

Gerçek saha koşullarında ve türbin devrinin kontrol edilmediği laboratuvar çalışmalarında (akışla sürülen rotor durumu) kanat uç hız oranı, her bir farklı serbest akış hızı için rotorun aerodinamik yapısı, mil ve bağlantı elemanlarının mekanik ataleti ve türbinin bağlı olduğu jeneratörün elektriksel ataleti ile doğal olarak oluşan bir değerde sabitlenir buna doğal kanat uç hız oranı denir. Şekil 1.9 (a)'da Araya vd. (2015) türbin çapına göre reynolds sayısı ve her akış hızında kanat uç hız oranı ortalama hızı ile karşılaştırılması ve kanat uç hız oranı'nın standart sapma ölçüm değerlerinin sınırları taralı bölge ile gösterilmiştir. Bu türbin doğal eğrisi, motorla sürülen çalışmalarda sabit bir Re sayısı için türbin doğal eğrisinin altındaki ve üstündeki kanat uç hız oranı çalışmalarını ayırt etmek için kullanılmıştır.

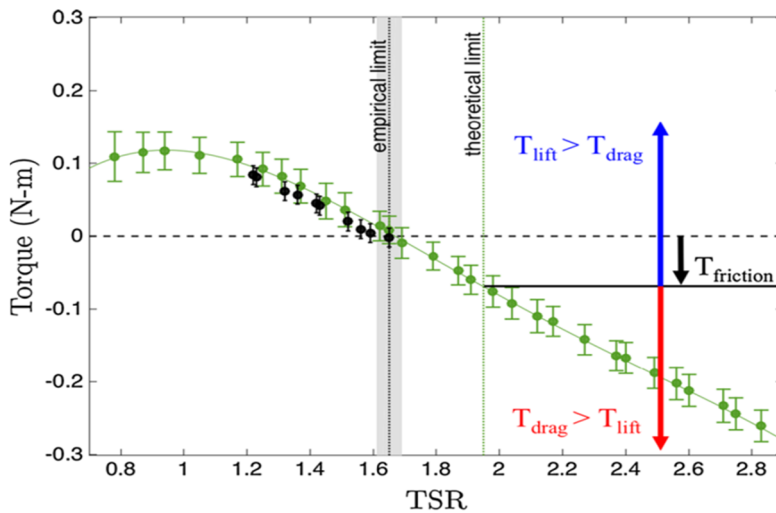


Şekil 1.9: (a) Türbin Doğal Eğrisi, sabit bir Re sayısı için türbin doğal eğrisinin (b) altında kalan ve (c) üstünde kalan ölçüm noktaları (Araya 2015).

Şekil 1.9 (b)'de motorla veya akışla sürülen durumlar için kanat uç hız oranı doğal eğrisi altında beş tane motor sürücülü ve beş tane akış sürücülü ölçüm noktası belirlenmiş bu ölçüm noktaları arasında akış ('O' ile gösterilen) ile motor sürücülü ('X' ile gösterilen) ölçüm sonucu arasında yakın bir uyumluluk görülmüştür. İki ayrı çalışma durumu için türbin başarımını karşılaştırmak için önce doğal kanat uç hız

oranı eğrisi çıkarılmış, bu bölgenin altında kanat uç hız oranı'na sabit tutup değişken Re ile üç nokta ve Re sayısını sabit tutup değişken kanat uç hız oranı ile üç nokta belirlenmiştir. Bu noktalar için yapılan motor ile ve akış ile sürülen test sonuçları benzerdir. Şekil 1.9 (c)'de kanat uç hız oranı doğal eğrisi üzerinde noktalar belirlenmiş ve sonuçlar Şekil 1.9 (b)'den farklı çıkmıştır.

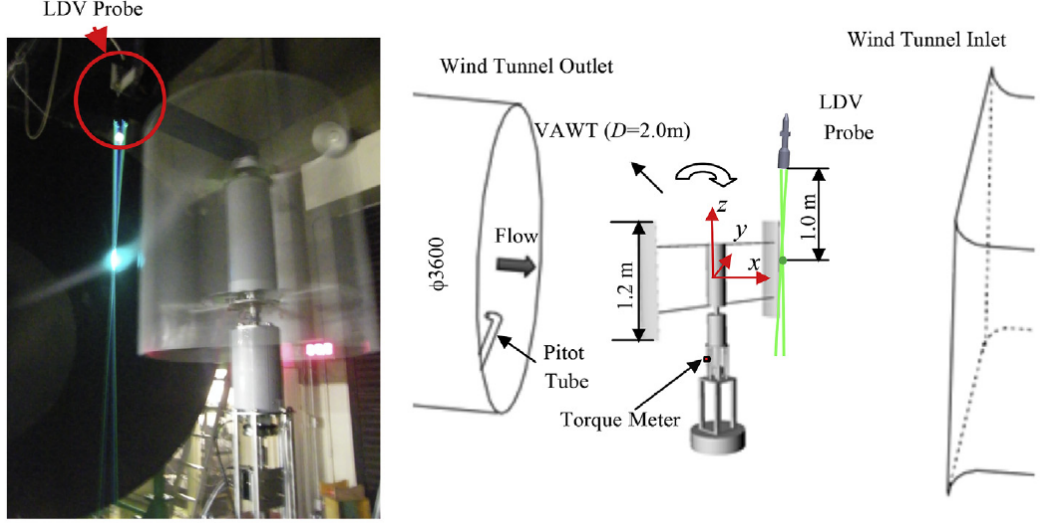
Kanat uç hız oranı doğal eğrisi üzerinde sabit Re sayısına göre değişken kanat uç hız oranı ve sabit kanat uç hız oranı'na göre değişken Re sayılarına karşılık gelen noktalar belirlenmiştir. Akış ile sürülen durumda, kanat uç hız oranı türbin doğal eğrisinin üzerine çıkamadığı için sadece TSR standart sapması içerisinde kalan Re $3,7 \cdot 10^4$ ve $7,9 \cdot 10^4$ değerlerindeki iki noktada yapılabilmektedir. Buna karşılık motor sürücülü deneyde motorun istenen kanat uç hız oranına ayarlanabilmesi nedeniyle, kanat uç hız oranı türbin doğal eğrisinin dışındaki noktalarda da türbin testler yapılabilmektedir.



Şekil 1.10: Akış ile sürülen ve motor ile sürülen durumlar için ölçülen tork değerleri (Araya 2015).

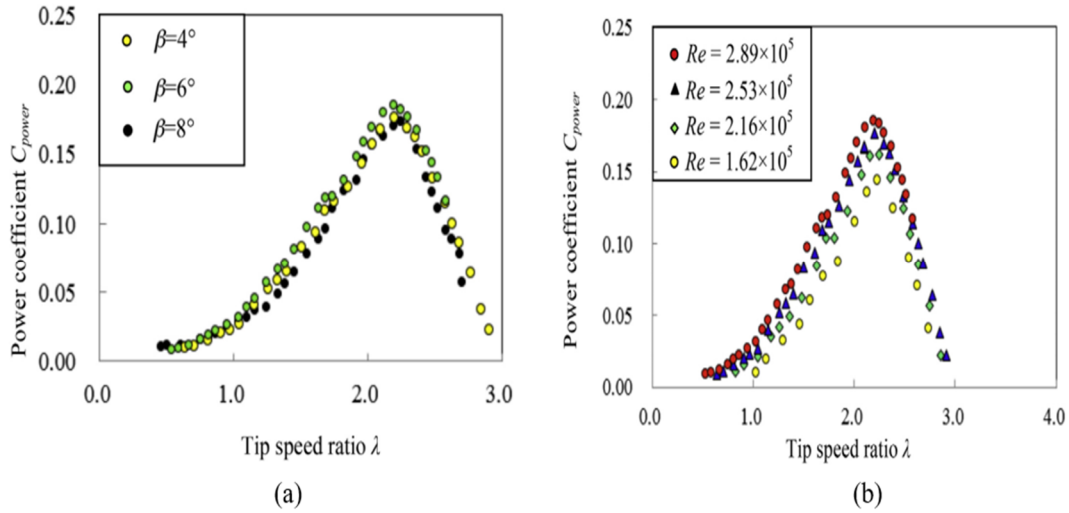
Şekil 1.10 akış ile sürülen (siyah) ve motor ile sürülen (yeşil) durumlar için ölçülen tork değerlerini göstermektedir. Rotor tork ölçümleri akış tarafından ayarlanan kanat uç hız oranı eşik değeri üzerine karşılık gelen sinyal değişikliklerini göstermiştir. Sonuçlar türbin kanatlarını bağımsız hareket ettirmek suretiyle, türbin kinematik ve aerodinamik özelliklerini yöneten yegane faktörün türbülans dinamikleri olduğu iddiasını desteklemiştir. Bu sonuçlar ışığında türbin kinematiklerini analiz etmek için kanat uç hız oranı önemli bir parametre olduğu ve kanat uç hız oranını kontrol etmek için motor sürücülü deney yöntemlerinin ön plana çıktığı gözlemlenmiştir. Burada rüzgar türbin testlerinde önemli parametre olan tork ölçümü tork metre ile yapılmıştır.

Qing'an Li vd. (2016) NACA0021 kanat profili ile yapılan iki kanatlı dikey rüzgar türbin testi güç performansı rüzgar tüneli ve arazi testleri ile yapılmıştır (Şekil 1.11). Testlerde pitch açısı, Reynolds sayısı ve rüzgar hızının DERT güç performansına etkisi değerlendirilmiştir.



Şekil 1.11: NACA0021 kanat profili ile yapılan iki kanatlı dikey rüzgar türbin testi (Qing'an Li 2016).

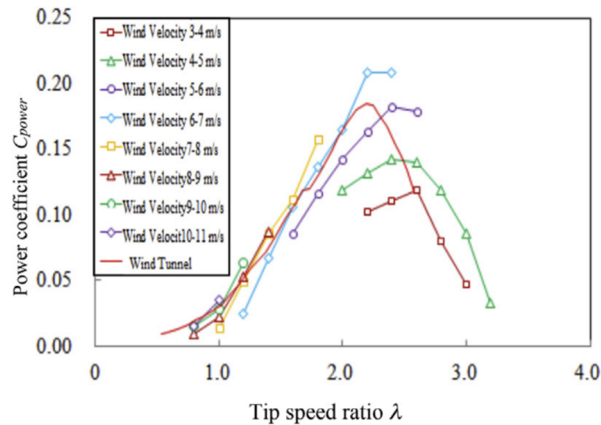
Yapılan rüzgar tüneli ve saha testlerinde kanat uç hız oranı artıçça türbulans hızının azaldığı, pitch açılarının güç katsılları ile ilişkisinde (Şekil 1.12 (a)) 4°, 6° ve 8°lerde en yüksek C_p değlerlerinin, sırasıyla $\lambda=2,13$, $\lambda=2,19$, $\lambda=2,26$ kanat uç hızı oranları için elde edildiği görülmektedir. Güç katsayısının en yüksek değeri pitch açısı 6°, $\lambda=2,19$ ve rüzgar hızı 8.0 m/s iken $c_p = 0,208$ olarak elde edilmiştir.



Şekil 1.12: NACA0021 kanat profiline sahip bir DERT'in (a) farklı kanat pitch açılarında C_p /kanat uç hız oranı değişimi , (b) farklı Re sayılarında C_p /kanat uç hız oranı değişimi (Qing'an Li 2016).

Farklı Reynolds sayısı için yapılan testlerde, yüksek Reynolds sayısının kanatlardaki kaldırma etkisini artırıp yüksek tork üretimi sağladığı ve bununla türbin performansına olumlu katkı sağladığı gözlemlenmiştir (Şekil 1.12(b)).

Farklı rüzgar hızları için türbin başarımı karşılaştırıldığında (Şekil 1.13) ise, aynı kanat uç hız oranında yüksek rüzgar hızının DERT güç performansı açısından olumlu olduğu değerlendirilmiştir. Düşük kanat uç hızı oranlarında rüzgar hızının türbin başarımına etkisinin düşük olduğu, kanat uç hız oranı arttıkça serbest rüzgar hızının etkisinin de arttığı görülmektedir. Bu çalışmada türbin torku tork metre yardımıyla ölçülmüştür.

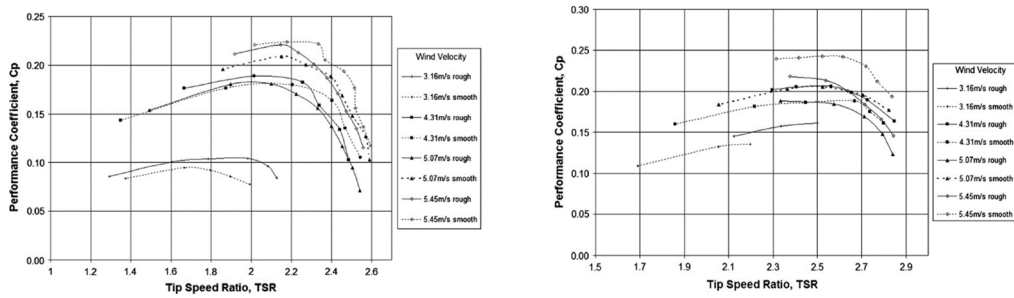


Şekil 1.13: NACA0021 kanat profiline sahip bir DERT'in farklı serbest rüzgar hızları için C_p /kanat uç hız oranı değişimi (Qing'an Li 2016).

Howell vd. (2010) kanat yüzey pürüzlülüğünün türbin başarımına etkisini incelemiştir. Öncelikle kanat yüzeyi pürüzsüz olarak kanat uç hız oranı-güç katsayısı grafikleri 2 kanatlı ve 3 kanatlı olarak oluşturulmuş, daha sonra kanatlar pürüzlü hale getirilip yine kanat uç hız oranı-güç katsayısı grafikleri 2 kanatlı ve 3 kanatlı DERT'in deneysel sonuçları karşılaştırılmıştır.

Bu çalışmada türbin çıkış gücünü ölçmek için basit bir tork fren sistemi kullanılmıştır. Dengeli iki yay arasındaki ayrılmış mesafe değiştirilerek, türbin rotor sürücü şaftına uygulanan tork artırılıp azaltılmıştır. Yay dengesi ve sürücü şaftı yarıçapına uygulanan güçteki farklılıkları kullanarak uygulanan tork hesaplanmış, güç çıkışını hesaplamak için hesaplanan bu tork, optik takometre tarafından her bir dönüşte toplanan veri ile bulunan türbin dönüş hızıyla birleştirilmiştir.

Türbini kontrol etmek için geri beslemeli bir kontrol sistemi kullanılmamıştır, bu yüzden maksimum torku aştıktan sonra türbin özelliklerini ölçmek mümkün değildir (bu nedenle minimum sabit dönme hızı kullanılmıştır). Bunun nedeni bu sınırın ötesinde uygulanan yükteki herhangi küçük bir artışın türbin dönüş hızının ve üretilen kaldırma kuvvetinin düşmesine neden olmasıdır. Yeni oluşan aerodinamik şartlara göre uygulanan torku azaltacak kontrol sistemi yoksa, uygulanan tork deney şartlarına göre uygun olmadığından türbin rotorunun dönüş hızı düşmeye devam edecektir.



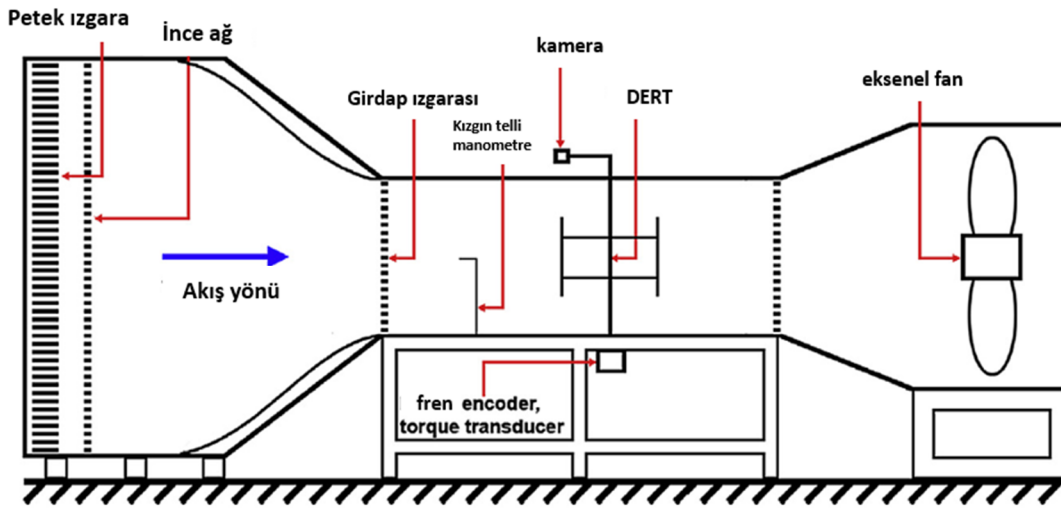
Şekil 1.14: Farklı rüzgar hızlarında pürüzlü ve pürüzsüz kanatlara ait Cp/kanat uç hız oranı (Howell 2010).

Kanat yüzeyi pürüzsüz olarak 2 ve 3 kanatlı kanat uç hız oranı-güç katsayısı grafikleri oluşturulmuş, daha sonra kanatlar pürüzlü hale getirilip yine 2 ve 3 kanatlı kanat uç hız oranı-güç katsayısı grafikleri DERT deneysel sonuçlarından yukarıdaki grafikler

elde edilmiş ve sonuçlar karşılaştırılmıştır. Buna göre düşük Re sayısında pürüzlü kanatlar yüksek güç katsayısı, Yüksek Re sayısında pürüzsüz kanatlar yüksek güç katsayısına ulaşmışlardır. Kanat başına düşen kanat uç hız oranı-güç katsayısı grafiğine göre 2 kanatlı modelde kanat başına daha fazla güç katsayısı elde edilmiştir.

Bu deneyde atalet momenti “spin down” yöntemi ile bulunmuştur. Türbin kanatsız olarak döndürülmüş, en yüksek hızın üzerine çıkarılıp serbest bırakılmış yavaşlaması gözlemlenmiş bu esnada hız ve anlık rotor devri kaydedilmiştir. Buradan da rulman sürtünmeleri ve rüzgar kayıpları hesaplanmış daha sonra kanatlar takılarak güç tekrar hesaplanmış ve bu kanatlar ile hesaplanan güce kayıplarda dahil edilerek türbin gücü bulunmuştur, fakat bu yöntem çok güvenilir değildir.

Howell vd. (2016) dikey rüzgar türbinlerinde katılık katsayısı ile yaptıkları araştırmada iki farklı katılık oranı (kanat 1 modeli için $\sigma=0,26$ $c=0,03m$ ve kanat 2 modeli için $\sigma=0,34$ $c=0,04m$) ile iki farklı, 2 kanatlı DERT tasarlanmış ve 3 farklı rüzgar hızı (6-7-8 m/s) ile güç katsayısı - kanat uç hız oranı grafikleri oluşturulmuştur.



Şekil 1.15: Sheffield üniversitesi rüzgar tüneli tesisi (Howell 2016).

Howell vd. (2016) yaptıkları çalışmada torku doğrudan torkmetre ile ölçmemişlerdir. Tork türbin sistemindeki atalet momenti ile ivmelenmeyi ölçerek hesaplanmıştır. Türbin performansı, önceki çalışmalarında da kullandıkları ‘spin down’ yöntemi ile ölçülmüştür. Bu yöntem, rotorun dönüş hızının değişmesi esnasında açısal hızın optik enkoder ile anlık olarak ölçülmesi ve böylelikle ivmenin hesaplanması temeline dayanır .

Anlık negatif ivmelenme:

$$\xi = \frac{\omega_2 - \omega_1}{t_2 - t_1} \quad (1.12)$$

denkleminde bulunmuştur.

Her iki teste DERT performansını belirlemek için rotor iki kez döndürülmüştür. Rotorun ilk döndürülmesi yatakların tork direncini belirlemek içindir. Tork direncini (T_{res}) hesaplamak için destek kollarının sürüklenme kuvveti, bağlantı bilyaları ve mekanik sürtünmeden kaynaklanan sistem direncini ölçebilmek için rotor kanatsız olarak döndürülmüştür ve tork:

$$T_{res} = I_{rig} \xi \quad (1.13)$$

denkleminde hesaplanmıştır.

İkinci test dönüşü kanat torkundan kaynaklanan DERT'in performansını bulmak için kanatlar rotara takılı olarak yapılmıştır. 7 m/s ve 8 m/s rüzgar hızlarında türbini yavaşlatmak için ilave olarak fren kuvvetine (T_{app}) ihtiyaç duyulmuştur. Kanatların ürettiği aerodinamik tork:

$$T_B = I_{rig} \xi - T_{res} - T_{app} \quad (1.14)$$

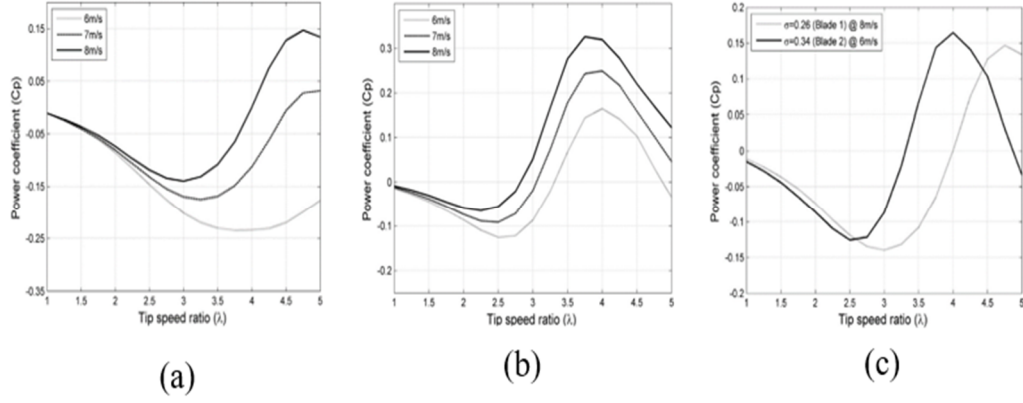
denkleminde elde edilmiştir.

Kanatlardan kaynaklanan DERT'in güç katsayısı spin-down testinden T_B 'yi elde ettikten sonra:

$$C_p = \frac{P_B}{P_w}, P_B = T_B \omega N, P_w = 0.5 \rho A V_w^3 \quad (1.15)$$

eşitlikleriyle bulunmuştur.

Howell vd.(2016) kanat 1 modeli için $\sigma=0,26$ $c=0,03m$ güç katsayısı –kanat uç hız oranı ilişkisi üç rüzgar hızındaki test sonucu grafiği şekil 1.16 (a)'daki gibidir.



Şekil 1.16: (a) Kanat1 modeli için üç rüzgar hızındaki test sonucu güç katsayısı – kanat uç hız oranı, (b) Kanat 2 modeli için üç rüzgar hızındaki test sonucu güç katsayısı –kanat uç hız oranı, (c) Kanat 1 ve 2 modeli için C_p /kanat uç hız oranı (Howell 2016).

DERT'in $\lambda=1$ 'den $\lambda=3,8$ 'e kadar üç rüzgar hızında negatif C_p ürettiğini yani bu bölgelerde DERT'in güç üretmediği görülmüş, 6 m/s rüzgar hızı için tüm kanat uç hız oranında C_p 'nin negatif olduğu, 7 m/s ve $\lambda=4.3$ ve daha büyük değerler için pozitif C_p , 8 m/s ve $\lambda=3.8$ kanat uç hız oranından sonrası için pozitif C_p ve maximum $C_p = 0,15$ tepe noktasına $\lambda=4,75$ değerinde ulaşıldığı gözlemlenmiştir.

Howell vd.(2016) kanat 2 modeli için $\sigma=0,34$ $c=0,04m$ güç katsayısı –kanat uç hız oranı ilişkisi üç rüzgar hızındaki test sonucu grafiği şekil 1.16 (b)'deki gibidir.

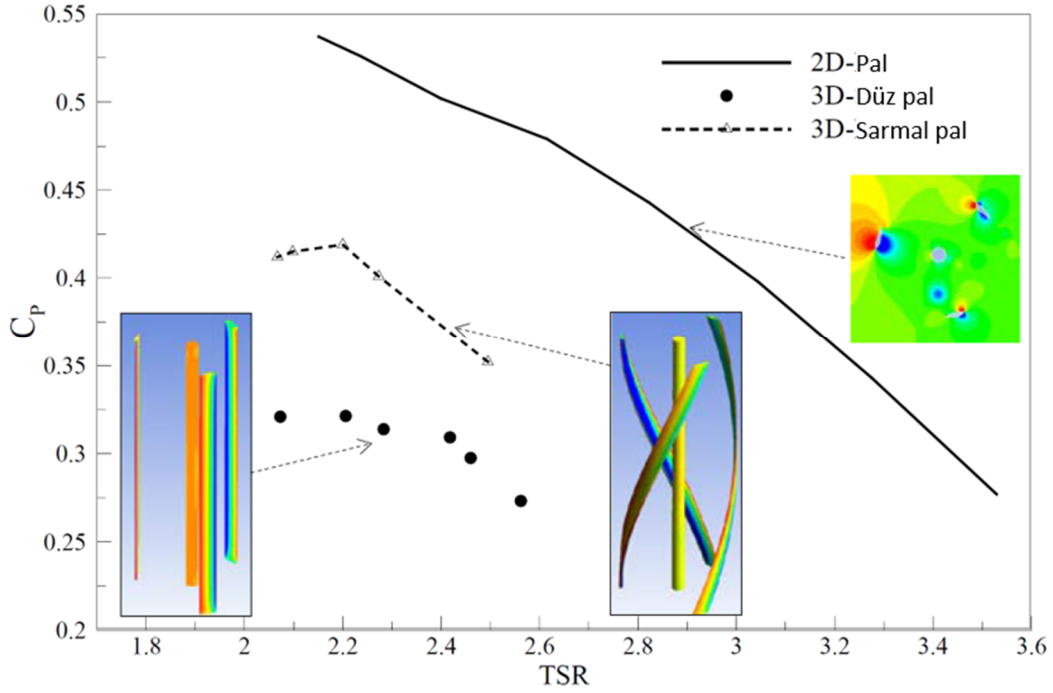
Kanat 2 modeliyle üretilen DERT güç katsayısı-kanat uç hız oranı grafiği bir önceki Kanat 1 modeliyle üretilen DERT güç katsayısı-kanat uç hız oranı grafiği gibi pozitif ve negatif performans bölgeleri içermiştir.

8 m/s rüzgar hızı için $\lambda=1$ 'den $\lambda=2.8$ 'e kadar C_p negatif, en düşük rüzgar hızı olan 6 m/s'de $\lambda=4$ maksimum $C_p = 0.146$ ve $\lambda=2.5$ değeri için $C_p = -0,123$ değeri elde edilmiş, rüzgar hızı 8 m/s'ye arttırıldığında reynold sayısı dolayısıyla $C_p=0,34$ $\lambda=3,75$ 'e kadar olan kısımda artış gözlemlenmiş, ilk testten daha fazla C_p değeri elde edilmiştir.

Şekil 1.16 (c)'de 1 ve 2 nolu test grafikleri karşılaştırılmış, C_p 'nin negatif ve pozitif bölgeleri gösterilmiştir. Negatif bölgelerde güç üretimi olmadığı için bu bölgelere ölü band denir. Yapılan bu testlerde ayrıca DERT'in performansının kanat uç hız oranına

(λ) bağılı olduğu ve iki kanatlı dikey rüzgar türbininin iki farklı katılık oranı ile karşılaştırıldığında katılık oranı yüksek olan DERT'in performansının diğerine göre fazla olduğu sonucuna varılmıştır.

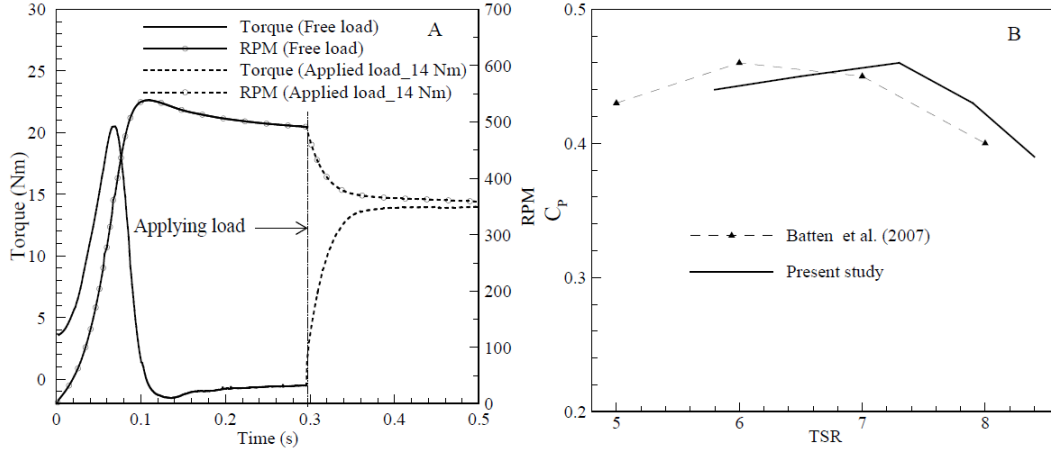
Le vd. (2014) yaptıkları çalışmada denizlerdeki gel-git akımları için düz üç kanatlı DERT ve spiral üç kanatlı DERT testleri 2D ve 3D CFD yöntemiyle karşılaştırılmıştır. Şekil 1.18'de düz ve spiral kanat yapılı DERT C_p /kanat uç hız oranı grafiği gösterilmiştir.



Şekil 1.17: Düz üç kanatlı DERT ve spiral üç kanatlı DERT testleri (Le 2014).

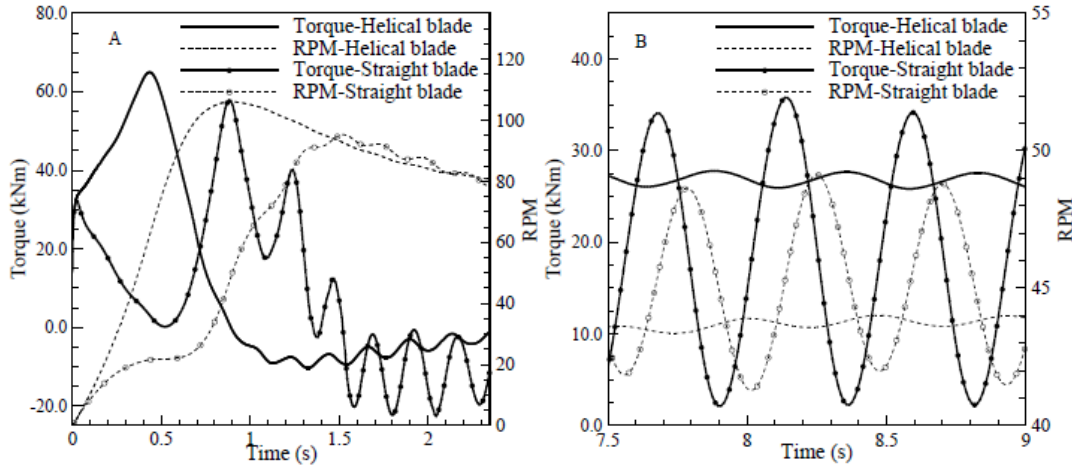
Türbin güç katsayısı iki farklı simülasyon yöntemi ile hesaplanmıştır. İlk olarak diğer sayısal çalışmalardaki yöntemlere benzer kanat uç hız oranı vererek CFD simülasyonu oluşturulmuş. Burada türbin ekseninin dönme hızı kullanıcı tarafından belirtilmiştir.

İkinci yöntem ise belirli bir yük ile akış tahrikli rotor simülasyonu olarak adlandırılan bir CFD simülasyonu kullanılmıştır. Türbin ekseninin dönme hızının sabitlenmediği buna karşın türbin, kanatlerdeki hidrodinamik moment, kanatın atalet momenti ve dönme ekseninde verilen karşı momentin belirli bir hızda döndüğü deneysel bir yaklaşıma benzemektedir.



Şekil 1.18: CFD simülasyonu Tork/zaman ve C_p /kanat uç hız oranı (Le 2014).

Türbin tarafından üretilen anlık güç, türbinin açısal hızı (ω) ve tork'un (T) çarpımına eşittir. Darrieus DERT'de tork ve hız sabit olmadığı için güç sabit değildir. Bu yüzden dönü başına ortalama güç, bu terimlerin ortalama değerlerinin çarpımı olarak hesaplanmıştır.

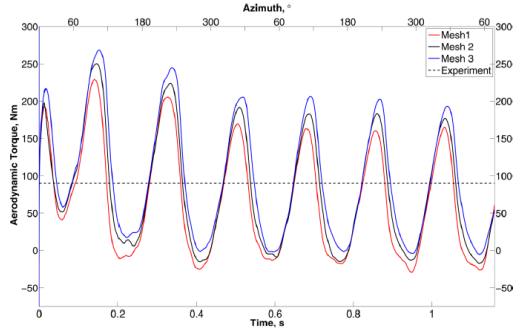


Şekil 1.19: CFD simülasyonu Tork/RPM (Le 2014).

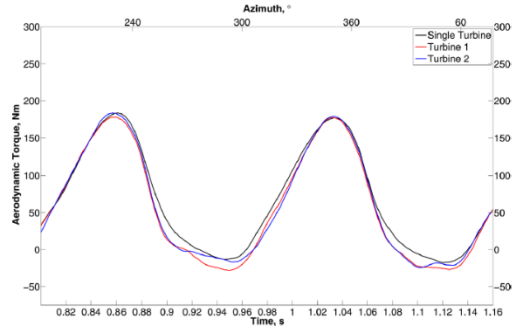
Bu çalışmada, optimal bir tasarım için 3D CFD simülasyonları aracılığıyla, sarmal kanatlı türbin, aynı boyuttaki düz açılı bir türbinden, kendi kendine çalışma yeteneğinde bir gelişme, tork seviyelerinin ve devir sayısının dalgalanmasında en aza indirmeye de dahil bunun yanında güç katsayısının verilen çalışma koşulları altında % 33'ten % 42'ye yükseldiği gözlemlenmiştir.

Korobenko vd. (2013) yaptıkları çalışmada tek DERT ve çoklu konumlandırılmış DERT tork-zaman grafiklerini karşılaştırmışlardır. Darrieus tipi DERT'in, savonius tipi DERT'lerden daha verimli oldukları için darrieus tipi DERT seçilmiş ve deneysel

olarak tek DERT ve çoklu DERT'ler (birbirine ters istikamette dönecek şekilde yan yana konumlandırılmış) 2D, 3D modelleri oluşturulmuş tork-zaman grafikleri karşılaştırılmıştır.



Şekil 1.20: Tekli DERT (Le 2014).



Şekil 1.21: Çoklu DERT (Le 2014).

Yapılan hesaplamalı simülasyon deneylerinde tork-zaman grafiklerinde tekli yerleştirilen DERT ile çoklu yerleştirilmiş DERT'ler arasında tahmini aerodinamik torkta hafif düşüş olduğu sonucuna varılmıştır.

Bahaj vd. (2007) deniz akımlarında kullanılacak türbinlerin deneysel testinde, belirli bir tünel akış hızı ve dönüş hızı için moment (Q) ve itme kuvveti (T) bilgilerini su kanalı içerisinde dönen pervaneye bağlı olan Dinamometreden alarak hesaplamıştır

Yine buna benzer bir çalışmada Rosli vd. (2016) su kanalı içerisindeki turbine bağlı tork ölçer ve kanat uç hız oranı sabit tutmak için fren mekanizması kullanılmıştır.

Yapılan bu testlerden anlaşılacağı üzere rüzgar türbinlerinin önemli karakteristiği kanat uç hız oranı ile güç ve verim ilişkisini veren grafikleridir. Bu grafikler, kanat uç hızı ve rüzgar hızı değerlerinin oranı sabit tutularak rüzgar yükünün doğurduğu torkun ölçülmesi veya bu şartlar altında yapılan matematiksel analizler ile torkun hesaplanması ile elde edilmektedir. Tork ölçümü için en güvenilir yöntem tork ölçer kullanmak olmasına rağmen, literatürde torkun dolaylı olarak hesaplanmasını sağlayan yaklaşımlar da mevcuttur. Ancak rüzgar türbinlerinin ürettiği torkun, motorla sürülen türbin durumu için elektriksel göstergelerden yararlanarak bulunduğu bir çalışmaya ratlanmamıştır.

1.3.2 Rüzgar türbinlerinde kullanılan kontrolcü yöntemleri

Senjyu vd. (2005) Direk tork ölçümünün gerçek ölçekteki türbinlerde uygulanmasının zor olması sebebi ile mevcut sistemlerin farklı alt sistemlerindeki ölçülebilir

büyükükleri kullanarak rüzgar torkunun gerçek zamanlı izlenmesi için gözlemleyeciler önermişlerdir. Bu yöntemler rüzgar yükünden kaynaklanan torkun bulunması için dolaylı yöntemler olarak nitelendirilebilirler. Dolaylı yöntemlerin en yaygını elektrik makinalarının kontrolü için rüzgar torkunun bulunması için tasarlanan gözlemleyecilerdir. Hassas Tork kontrolünün amacı elektrik makinasının veriminin yükseltilmesidir. Bu amaç elektrik makinasının girdisinde oluşan rüzgar yüküne bağı tork değışimleri elektrik makinası içerisinde ölçülen elektriksel değışkenler yardımıyla gözlemlenebilmektedirler.

Farklı kontrol yapılarında bu gözlemleyici yapısının kontrol amaçlı nasıl kullanılacağı Bossanyi vd. (2000) çalışmasında gösterilmiştir. Bu çalışmada değışken hızlarda türbin torku ve dolayısıyla üretilen elektrik enerjisini PI ve PID yöntemi ile geliştirilen gözlemleyici ile kanatların pitch açısını değıştirerek tork ve enerji üretiminde değışken hızlarda daha yumuşak bir geçiş sağlanmıştır.

Laboratuar ölçeğinde iki temel deneysel sistem üzerinde direk tork ölçümü uygulamaları yaygın olarak yapılmaktadır. Bu ölçümler motor ile sürülen türbinler ve akış ile sürülen ve frenleme ile yavaşlatılan türbinlerde kanat uç hız oranına bağı güç değışkenlerinin bulunmasında kullanılmaktadır.

Bu yöntemlerden daha yaygın olarak kullanılanı motor ile sürülen deney düzeneğini kullananlardır. Burada üç temel uygulama görölmektedir. Bunlardan ilki sabit devirli kontrolcüsüz motor ile sadece devri ayarlayıp rüzgar yüküne mazur kaldığında motor devrinin değışmediğini kabul eden sistemlerdir . Bu sistemlerde açık olan durum kontrolsüz motorun değışken rüzgar yükü altında ayarlanmış kanat uç hız oranı deđerini sağlayamayacak olmasıdır. İkinci uygulama standart motor yerine servomotor kullanılması ile motor ile sürülen deney düzeneğinin kurulmasıdır. Bu uygulamada, rüzgar yüküne bağı hız değışikliklerinin kapalı çevrim kontrol ile giderilmesi mümkün olmaktadır. Öte yandan hareket kontrol konusundaki çok sayıda çalışma dinamik ve kuvvetli bozucu girdiler söz konusu olduğunda sadece geri beslemenin istenilen hassas hareket kontrol şartını sağlamada yetersiz olduğu bilinmektedir. Bu sebeple bu tipteki deney sistemlerinde ancak ortalama bir kanat uç hız oranı oranın tutturulduğu öngörülmektedir. Üçüncü uygulama tipi ise rüzgar türbinin rüzgar ile döndürölüp hızının bir frenleme sistemi ile kontrol edilmesidir. Bu durumda frenleme sistemlerinin genel olarak sistem üzerinde yarattığı titreşim ve darbeli çalışma durumu sistemin hız kontrolündeki performansını düşürmektedir. Yine ikinci uygulamada ki

duruma benzer şekilde bu sistemlerde de ortalama bir kanat uç hız oranı değerinin ayarlanabileceği fakat bu sistemlerinde ancak ortalama bir kanat uç hız oranı değerini daha büyük hata değerleri sağlayabilecekleri öngörülmektedir. Yapılan çalışmalarda iki sistemde rüzgar yükü altında kanat uç hız oranı şartını sağlayıp sağlamadığı konusunda performans eğrileri verilmemiştir.

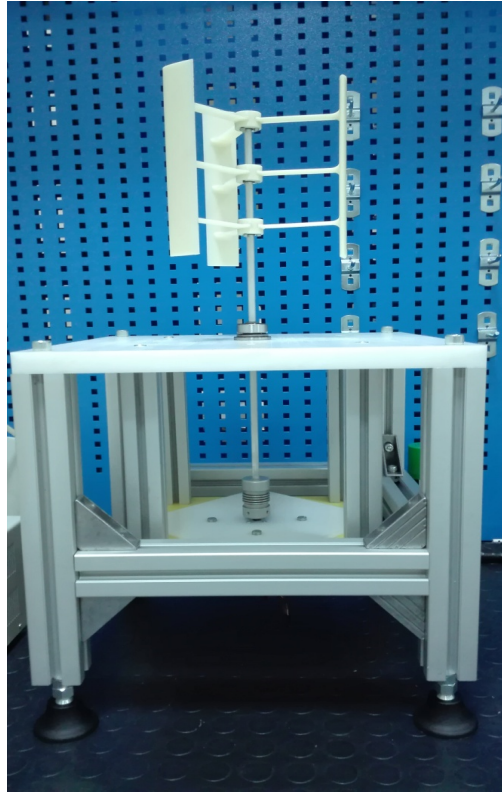
kanat uç hız oranını kontrol etmek için türbin testlerinde kullanmaya planladığımız motor tipi kullanım performansı açısından DC motor olacaktır. Endüstride DC motoru kontrol etmek için bir çok kontrolcü geliştirilmiştir. Litarütüre baktığımızda kapalı çevrim, geri beslemeli, hassas hız kontrolü sistemi olarak encoder ve PID yazılımı geniş ölçüde kullanılmaktadır. Köse vd. (2013) yaptıkları çalışmada PID ve bulanık mantık kontrolü ile DC motor hız kontrol testleri yapmış, bulanık mantık kontrolörde aşırı aşım PID kontrolöre göre fazla, buna nazaran bulanık kontrolörün yükselme zamanı daha az olduğu, oturma zamanı bakımından iki sistem karşılaştırılırsa PID kontrolörün daha kısa sürede oturduğu görülmüştür. Kalıcı durum hatası bakımından iki sistemde kalıcı durum hatası aynı çıkmıştır. Sonuç olarak PID kontrolün motor devirini kontrol etmek için daha verimli olduğu gösterilmiştir. Coşkun ve Terzioğlu (2009) çalışmasında DC motorun hız denetimi sürecini etkin olarak iyileştirmek için PID kontrol tercih etmişlerdir. Bu çalışma için kullanılmış olan FEMSAN firması tarafından üretilen yabancı uyartımlı DC motoru 1,5 KW, 200 V, 7,5 A, 3000 d/dk özelliklerine sahiptir. DC motoru tam yükte ve değişik yüklerde çalıştırılmış ve bu yükler altında hız performans eğrisi üzerinden online hesaplanan kazanç değerleri kullanılarak deneyler gerçekleştirilmiştir. Değişken kazanç parametreleri ile kontrol edilen DC motoru tam yükte ve değişken yüklerde çalıştırılmış ve sonucunda hız kontrolü etkin bir şekilde gerçekleştirildiği görülmüştür. Motorun hızına bağlı olarak hesaplanan kazanç parametreleri ile referans hızdan herhangi bir şekilde bir sapma olduğunda sistemin kısa sürede referans hıza tekrar ulaştığı belirlenmiştir.

Bu çalışmada hassas hız kontrollü motor ile sürülen PID ve PI sistemi geliştirilecektir böylelikle istenilen kanat uç hız oranının sağlanması ile ilgili var olan sistemlerden daha yüksek bir başarımla sağlanması öngörülmektedir. Hassas hız kontrolü için kullanılacak bu sistemler, anlık olarak rüzgar yükünü belirleyeceğinden sistemde ayrıca bir tork sensörü bağlanması gerekmeden tork ve güç değişiklikleri anlık olarak hassas bir şekilde hesaplanabilecek, her iki sistemin sonuçları karşılaştırılıp türbin deneylerindeki en uygun yöntem belirlenecektir.

2. Metaryal ve Metodlar

2.1 Kanat Yapısı

Yapılan ölçümlerde kullanılan NACA 0018 kanat profiline sahip Darrieus tipi türbinlerden birisinin veter uzunluğu 2 cm, kanatların veter ekseninden döndürme eksenine uzaklığı 10 cm, kanatların yerden yüksekliği 20 cm ve kanatların destek kolları 1 cm genişliğinde, 9,5 cm uzunluğunda NACA 0024 tipi olarak, ikinci kanat profili NACA 0018 veter uzunluğu 4 cm, kanatların veter ekseninden döndürme eksenine uzaklığı 10 cm, kanatların yerden yüksekliği 20 cm ve kanatların destek kolları 1 cm genişliğinde, 9,5 cm uzunluğunda NACA 0024 tipi olarak seçilmiş ve 3 boyutlu yazıcı ile AVBS malzemedan üretilmiştir. Üretilen bu kanatlar canon FN 38+encoder 24 vdc motor ile bir rot vasıtasıyla birleştirilmiş, bu tasarım bir sehpa kaidesine montajlanmıştır. Şekil 2-1’de üretilen türbin ve sehpa kaidesi gösterilmiştir.



Şekil 2.1: NACA 0018 profilli türbin modeli ve sehpa kaidesi.

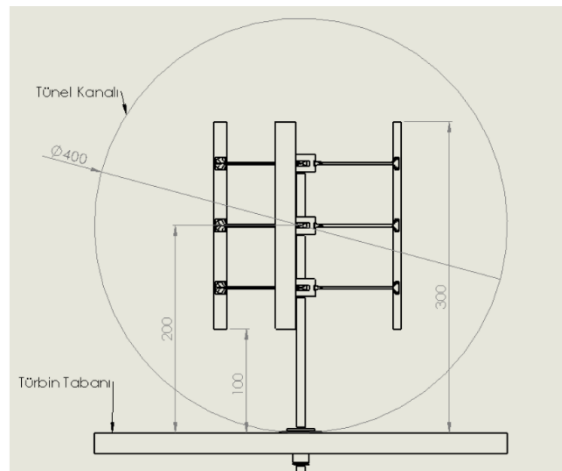
2.2 Rüzgar Test Tüneli

Üretilen bu türbin modeli ve kaidesi İzmir Makine Mühendisleri Odası'na bağlı kalibrasyon merkezinde bulunan şekil 2.2'de gösterilen rüzgar test tüneline yerleştirilmiştir.

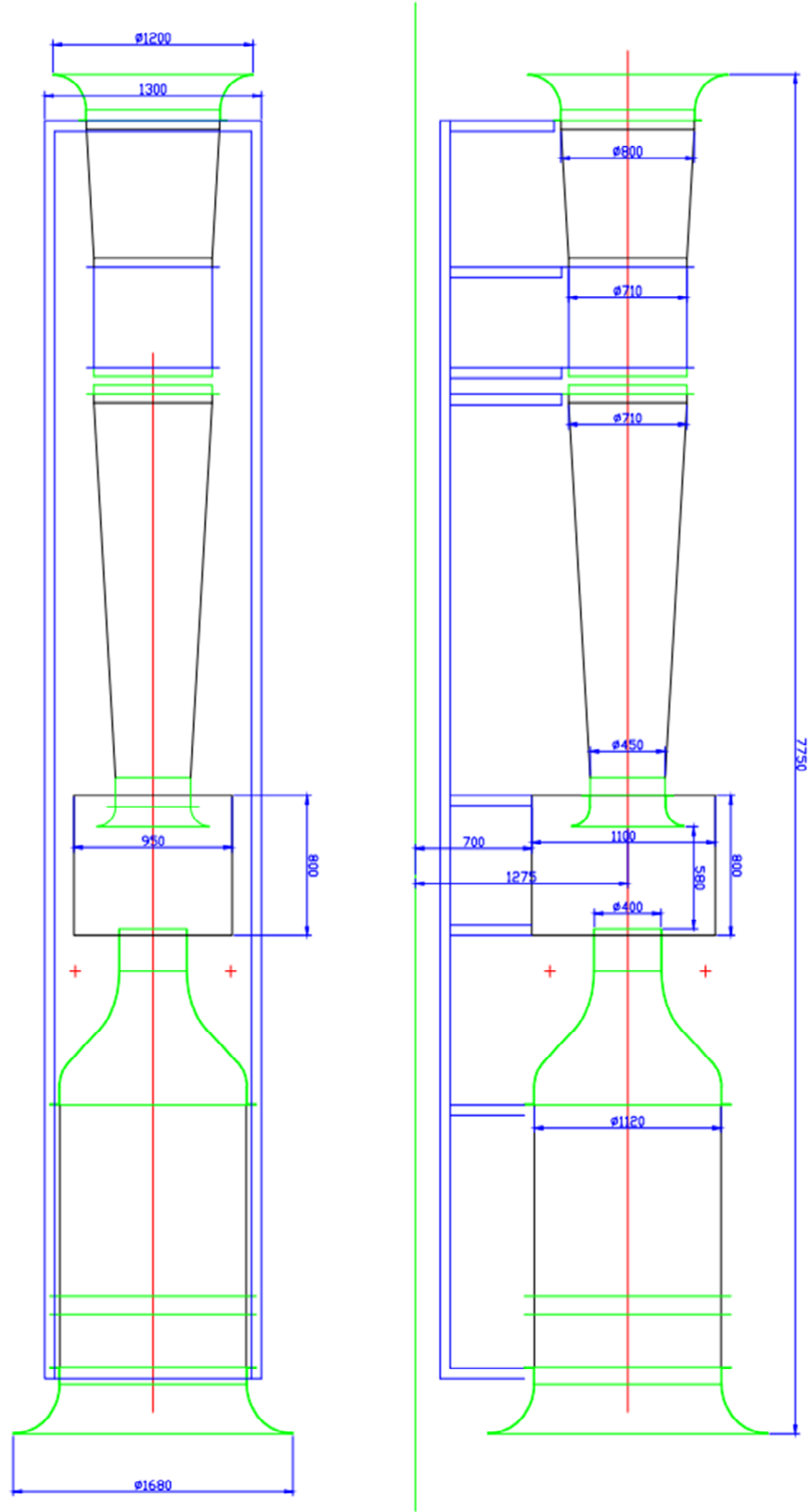


Şekil 2.2: Rüzgar Test Tüneli.

Bu rüzgar tünelinin özellikleri; 0,5 ile 40 m/s ölçüm aralığında çalışan eifel tipi rüzgar tüneli, rüzgar çıkış bölümü çapı: 400 mm, çalışma bölümü uzunluğu: 580 mm, akış hızı : 0.5 m/s ... 40 m/s, fan bağlantısı: 400 Volt / 50 Hz, motor çıkışı: 11 kW, boyutları (BxHxL): 1680 x 2800 x 8000 mm'dir



Şekil 2.3: Darrieus tipi türbinin ölçüleri ve test oryantasyonu.

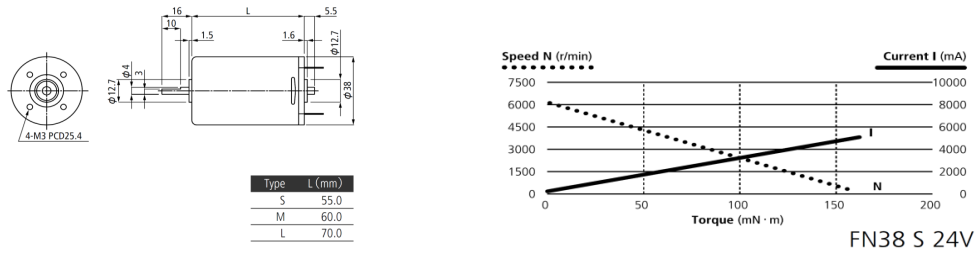


Şekil 2.4: Rüzgar Test Tüneli boyutları.

2.3 FN38 DC Motor ve Encoder

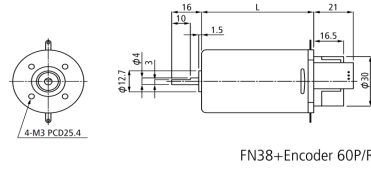
Motor güç bağlantıları HP-250 20A H-Köprü DC Motor Sürücü devresine bağlandı, motor bu sürücü devresi vasıtasıyla sürüldü, bu sürücü devresine harici 0-30 v DC 10 A'lık güç kaynağı ile besleme yapıldı. Şekil 2.4'te canon marka FN-38 motor encodera ait teknik özellikler verilmiştir. Bizim seçtiğimiz motor tipi FN-38 S 24 V DC'dir.

■ FN38 Series



MODEL	Rated Voltage	Rated Power Output	Rated Torque	Rated Speed	Rated Current	No Load Speed	No Load Current	Starting Torque	Torque Constant	Winding Resistance	Winding Inductance	Moment of Inertia	Mechanical Time Constant	Thermal Resistance
	V	W	mN·m(gf·cm)	r/min	mA	r/min	mA	mN·m(gf·cm)	mN·m/(gf·cm/A)	Ω	mH	g·cm²	ms	°C/W
FN38 S	12	13	24.5 (250)	5100	1700	6200	230	137 (1400)	16.7 (170)	1.5	0.7	32	17	20
	24	13	24.5 (250)	5700	840	6100	115	158 (1610)	33.8 (345)	5.0	2.9	32	14	20
FN38 M	12	14	24.5 (250)	5400	1660	6600	205	137 (1400)	16.9 (172)	1.7	0.9	39	23	17
	24	13	24.5 (250)	5150	850	6300	110	132 (1350)	33.1 (338)	5.9	3.7	39	21	17
FN38 L	24	26	58.8 (600)	4200	1600	5000	130	373 (3800)	40.2 (410)	2.7	2.7	117	20	14

■ FN38 + Encoder



MODEL(Phases)	Input Voltage	Number of Pulses	Response Frequency	Encoder Type	Output Level	Index Pulse Width	Logic Width	Duty Ratio	Output Signal Form
	V	P/R	KHz(MAX)						
FP38 (1)	5	60	20	Optical	TTL	---	---	30%~70%	Square Wave
FP38 (2)	5	200	20	Optical	TTL	---	S=90°e±10°e	35%~65%	Square Wave
	5	512	40	Optical	TTL	---	S=90°e±30°e	30%~70%	Square Wave
FP38 (3)	5	1000	50	Optical	TTL	180°e×4	S=90°e±15°e	40%~60%	Square Wave

FP38: FN38+Encoder

Şekil 2.5: Canon FN-38 DC Motor+Encoder Özellikleri (Url3)

2.4 Güç Kaynağı ve Motor Sürücü Kartı

Şekil 2.6 (a)'da TT-Technic marka RXN-3020D modeli güç kaynağı gösterilmiştir, RXN-3020D 30 V gerilime 20 A amkım değerine kadar güç sağlayabilmektedir ve bu değerler seçtiğimiz motor için fazlasıyla yeterlidir.



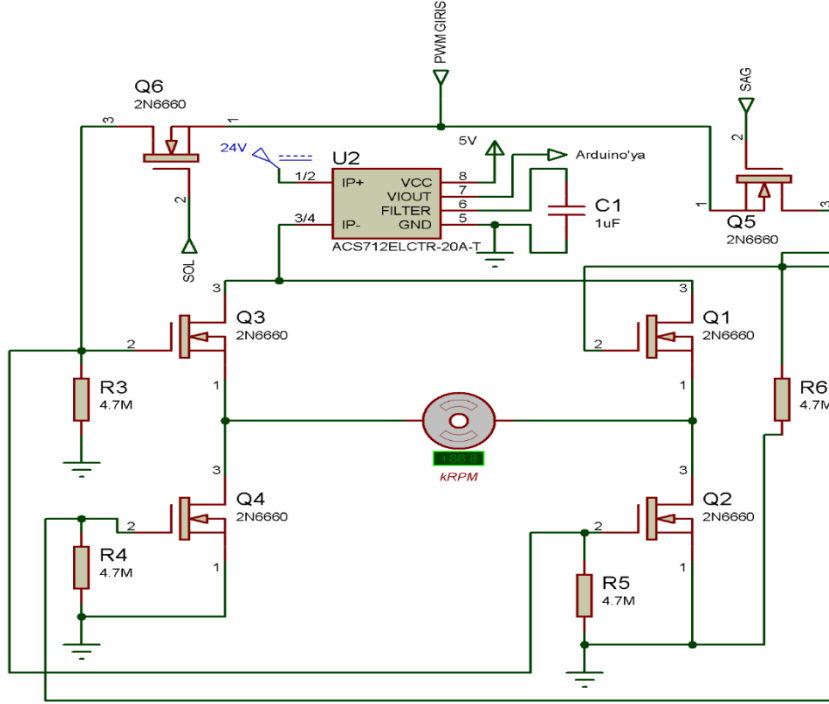
(a)



(b)

Şekil 2.6: (a) TT-Technic marka RXN-3020D modeli güç kaynağı. (b) HP-250 20A H-Köprü DC Motor Sürücü devresi.

Giriş/çıkış birimlerinden dolayı seçtiğimiz motor sürücüsü HP-250 20A H-Köprü DC Motor Sürücü devresi Şekil 2.6 (b) DC motorun yönünü ve hızını kontrol etmek için seçilmiştir. Bu devre aşağıda verilen şekildeki gibi tasarlanmış ve ürün tedarikçisi tarafından bu tasarıma hazır olarak tedarik edilmiştir. Bu motor sürücüsü yapısındaki MOSFET'ler sayesinde yüksek gerilim değerlerine ve yüksek akım değerlerinde dayanabilmektedir.



Şekil 2.7: motor sürücü devresi açık şeması.

Yüksek torka sahip DC motorlar kalkışlarında normal çalışması için gerekli olan akımın 8 ila 10 katı akım çekebilmektedirler. Elektrik terminolojisinde buna demeraj akımı denilmektedir . Kalkıştaki bu akım bir çok motor sürücüsü tasarımının karşılaştığı en büyük problemlerden birisidir. Her elektronik malzemenin kendine has karakteristikleri olduğundan anahtarlama için kullanılan elemanların bu kalkış akımına dayanıklı seçilmesi gerekmektedir. Burada geleneksel BJT transistörler yerine voltaj kontrollü olması ve yüksek gerilim-akım değerlerine karşı olan dayanımı sebebiyle MOSFETler anahtarlama elemanı olarak kullanılmıştır.

Şematik incelenirse sistemi mikrokontrolcü üzerinden aynı anda iki pinle kontrol edilmesi gerektiği görülecektir. Bu pinlerin ilki motora yön bilgisinin verileceği SAĞ ve SOL pinleri diğeri ise sürekli olarak kullanılacak olan ve Darbe Genlik Modülasyonu fonksiyonlarının (PWM) sisteme uygulanmasını sağlayan PWM pinidir. PWM pininden gelen sinyal kare dalgalardan oluşmaktadır. Bu kare dalgaların genişlikleri mikrokontrolcü tarafından değiştirilerek motora her periyodun kaçta kaçında emk uygulanacağı belirlenmektedir. Bu sayede motor –Arduino ile kontrol edildiğinde- 256 farklı hız seviyesinde sürülebilmektedir. Devrede 24 olarak gösterilen kaynak motorları besleyen ana kaynaktır. Bu kaynak ile mosfetlerin “Source” uçları arasına bağlanmış olan ACS 712 sensörü sayesinde motorun kullandığı akım anlık

olarak okunabilmektedir. Bu sensör çıkışında girişe uygulanan sinyal ile orantılı olarak analog sinyal ürettiğinden Arduino üzerinde analog giriş pinlerinden A2'ye bağlanmıştır.

Oluşturulan motor sürücüsü devresinde sisteme toplam 3 giriş (1 PWM 1 YÖN 1 AÇIK-KAPALI) ve 1 çıkış (Akım Sensörü) vardır. Bu sayede Arduino ile motorun hızlı bir şekilde kontrol edilmesi sağlanmış, analog okunan akım sensörü sayesinde de bu konudaki ölçümlerin yüksek hassasiyet ve doğruluk oranıyla hızlı bir şekilde yapılması sağlanmıştır.

Çizelge 2.1: Motor sürücü kartının giriş ve çıkış pinleri.

Pin Grubu	Pin İsmi	Açıklama
CTRL (1)	PWM	Motor hızını ayarlamak için kullanılır, değeri motor çıkışına yansır.
CTRL (2)	DIR	Motor yönünü kontrol etmek için kullanılır. LOW; ileri , HIGH; geri.
CTRL (3)	S/S	Hareketi başlatmak için pine HIGH uygulanır, durdurmak için ise pine LOW uygulanır. Pin LOW olduğu zaman motor fren ile hareketini durdurur.
CTRL (4)	CFB	Akım sensörü çıkışıdır mv/A başına analog çıkış verir. Bu çıkışın voltaj değeri seçilmiş olan akım çözünürlük ayarına göre değişir.
CTRL (5)	VFB	Sürücü besleme voltajı hakkında mv/V başına analog çıkış verir.
CTRL (6)	GND	Referans noktası, toprak.
P_EX (1)	+12V	Shield'leri veya harici devreleri beslemek için en fazla 50 mA voltaj kaynağı çıkışı.

P_EX (2)	GND	Referans noktası, toprak.
P_EX (3)	+5V	Shield'leri veya harici devreleri beslemek için en fazla 50 mA voltaj kaynağı çıkışı.
MOTOR (1)		Motor bağlantı çıkışı.
MOTOR (2)		Motor bağlantı çıkışı.
POWER (1)	V-	Referans noktası, toprak, GND.
POWER (2)	V+	10V-30V arası canlı uç bağlantı girişi .

Arduino PWM çıkışı standart olarak 500 Hz olduğundan, bu frekanstaki kullanım da dikkate alınmıştır.

2.5 Sinyal Genişlik Modülasyonu (PULSE WIDTH MODULATION (PWM))

PWM (sinyal Genişlik Modülasyonu), üretilecek olan darbelerin, genişliklerini kontrol ederek, çıkışta üretilmek istenen analog elektriksel değerin veya sinyalin elde edilmesi tekniğidir. PWM elektrik ve elektronikte birçok alanda, farklı amaçlar için kullanılmaktadır. Telekomünikasyon, güç, voltaj düzenleyiciler, ses üreticileri veya yükselteçler gibi çeşitli uygulama alanlarının yanı sıra Arduino veya elektrik makineleri gibi özel uygulama alanlarında da yer alan bir tekniktir.

Üretilen kare dalga darbe sinyallerinin genişliklerinin ortalaması, çıkışta üretilecek olan analog değerin elde edilmesini sağlar. Şekil 2.8'de kare dalga gösterilmiştir. Bu şekil üzerinden açıklamak istersek;

Ortalama sinyal,

$$\bar{y} = \frac{1}{T} \int_0^T f(t) dt \quad (2.1)$$

$f(t)$: Kare Dalganın Frekansı (Hz)

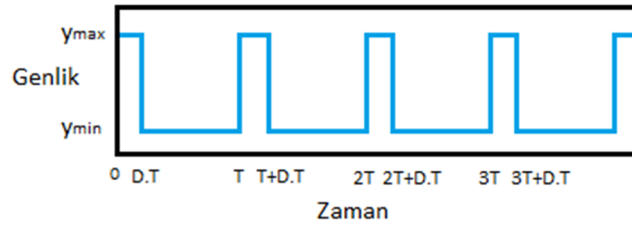
y_{min} : En düşük genliğin değeri

y_{max} : En yüksek genliğin değeri

D : Sinyal Oranı (Duty Cycle)

T : Periyot

t : zaman



Şekil 2.8: Kare dalga.

$f(t)$ kare dalga olduğundan $f(t)$, y_{max} için

$$0 < t < D \cdot T \quad (2.2)$$

ve y_{min} için

$$D \cdot T < t < T \quad (2.3)$$

değerlerini alabilir.

Buradan,

$$\bar{y} = \frac{1}{T} \left(\int_0^{DT} y_{max} dt + \int_{DT}^T y_{min} dt \right) \quad (2.4)$$

$$= \frac{D \cdot T \cdot y_{max} + T(1-D)y_{min}}{T} \quad (2.5)$$

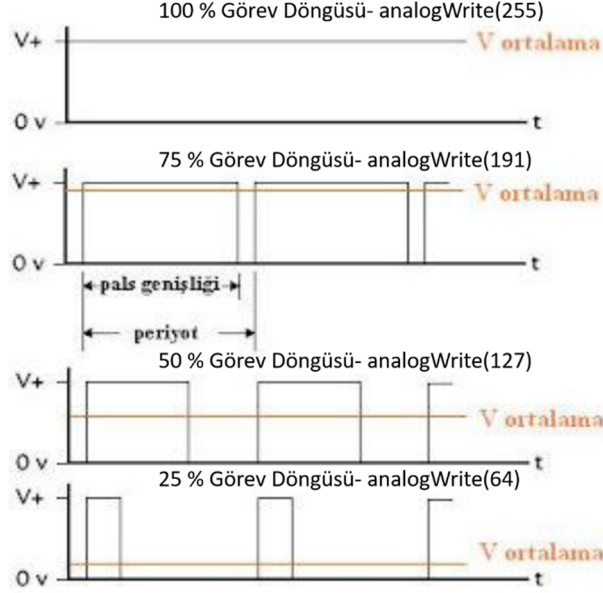
$$= D \cdot y_{max} + (1 - D)y_{min} \quad (2.6)$$

elde edilir.

y_{min} değeri genellikle 0 olarak kullanılır, bu durumda formül

$$y = D \cdot y_{max} \quad (2.7)$$

olarak kullanılır. Görüldüğü gibi elde edilecek ortalama değer direk sinyal oranına (duty cycle) bağlıdır. Şekil 2.9’da sinyal oranının değişimi incelenmiştir.



Şekil 2.9: Sinyal Oranı (Duty Cycle) Arduino PWM çıkışları % 25, % 50, % 75, % 100 olduğunda kara dalga genişlikleri.

PWM ile üretilebilecek çeşitli oranlar Şekil 2.9’da verilmiştir. Burada %25, %50, %75 ve son sıra %100 görev döngüsüne karşılık gelen analog sinyal değeri, genelde 5 volt kullanıldığı için, 5 V giriş değerine karşılık %25 için 1,25 V, %50 için 2,5 V, %75 için 3,75 V ve %100 için 5 V analog çıkış değerleri verilmiştir.

Elbette üretilen sinyalin frekansı da göz ardı edilmemelidir. Çok düşük bir frekans ile üretilen darbe sinyalleri ve bunlarla kontrol edilen bir anahtar ile kontrol edilen örneğin bir lambada (deneyde kullanılan DC motor), lambanın yanma ve sönme zamanları hissedilebilir. Bu durum size ışığın şiddetinin değişikliğinden öte, titretme şeklinde görülecektir. Bunu engellemek için anahtarlama frekansı yükseltilmelidir. Arduino’da PWM çıkışı standart olarak 500 Hz kullanılmaktadır, bu çıkış bu çalışmada kullanılan motor sürücü kartının çalışma frekansı ile uyumludur, ayrıca bu frekans motorun uygun çalışma aralığında seçilmelidir, bu aralık denklem 2.8’deki gibi olmalıdır.

$$2 * \pi * f * L > R \quad (2.8)$$

F = Frekans (Hz)

L = Motor İndüktansı (H)

R = Motor İç Direnci (Ω)

Denklemdede ifade edildiği gibi frekans ile motor indüktansı ve pi sayısının çarpımı motor iç direncinden büyük olmalıdır, 10 kat büyük olması istenen durumdur, denklem frekans ile ayarlandığında, eşitsizlik tersine çıkarsa motorun çalışması fasılalı olacak, daha yüksek frekanslar ise motorun gücünü düşürme, ısı kayıplarını artırma gibi kötü sonuçlar doğuracaktır.

Yaptığımız çalışmada yukarıdaki formülde seçilen DC motorun değerleri yerine konulursa;

$$2*3,14*500*0,0029 > 5,95$$

$$9,106 > 5,95$$

Olarak sonuçlanmış ve motorun PWM ile çalışmasında bahsedilen problemler oluşmamıştır.

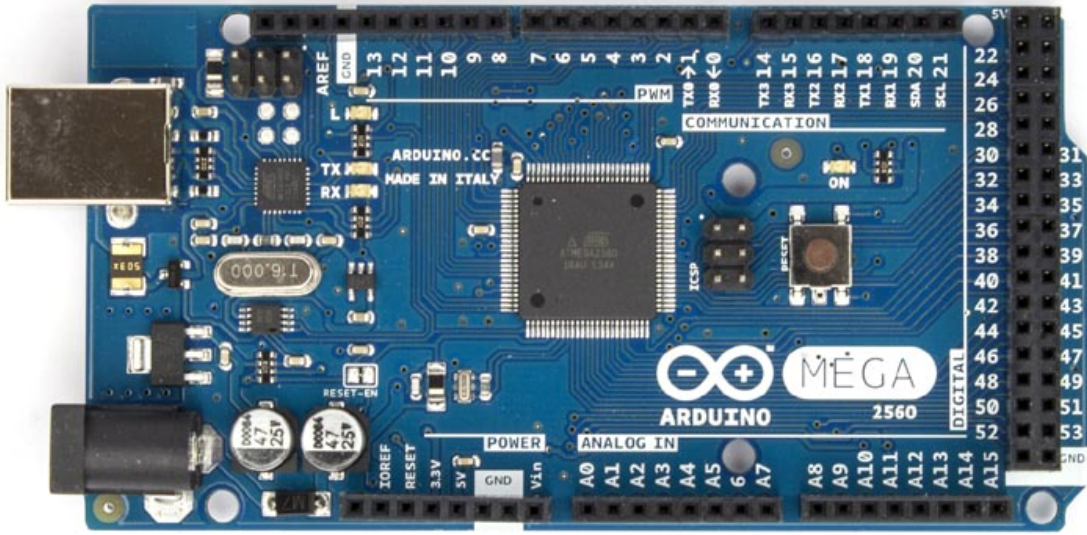
2.6 Arduino Mega 2560 ve PWM uygulaması

Deneyde kullanılan türbin tasarladıktan ve bu tasarıma uygun olan canon marka DC motor yine bu motora uygun olan motor sürücü devresi seçildikten sonra türbin dönüş hızını yani kanat uç hız oranı sabit tutma işlemi için açık kaynak olması, kolay temin edilmesi, fiyatının 45 TL olması, PWM çıkışlarının olması, boyutlarının küçük olması, ortalama bir PC ile sorunsuz haberleşmesi, bir bilgisayara bağlanarak, bir adaptör ile ya da pil ile çalışması gibi özelliklerinden dolayı arduino mega 2560 kartının kullanılmasına karar verilmiştir.

Arduino ile DC motor kontrolü gerçekleştirmek için PWM tekniğini kullanılması gerekiyor. Bu teknik de Arduino'ya gömülen yazılımda yatar. “analogWrite(...)” fonksiyonu ile görev döngüsünün miktarı belirlenir ve kare dalga elde edilir.

Şekil 2.10'da arduino Mega gösterilmiştir. Arduino bir Giriş/Çıkış kartı ve Processing/Wiring dilinin bir uygulamasını içeren geliştirme ortamından oluşan bir fiziksel programlama platformudur. Arduino kartlarının donanımında bir adet Atmel

AVR mikrodenetleyici (ATmega328, ATmega2560, ATmega32u4 gibi) ve programlama ve diğer devrelere bağlantı için gerekli yan elemanlar bulunur. Her Arduino kartında en azından bir 5 voltluk regüle entegresi ve bir 16MHz kristal osilatör (bazılarında seramik rezonatör) vardır. Arduino kartlarında programlama için harici bir programlayıcıya ihtiyaç duyulmaz, çünkü karttaki mikrodenetleyiciye önceden bir bootloader programı yazılıdır. Arduino 'nun temel bileşenleri : Arduino geliştirme ortamı (IDE), Arduino bootloader (Optiboot), Arduino kütüphaneleri, AVR Dude (Arduino üzerindeki mikrodenetleyici programlayan yazılım) ve derleyiciden (AVR-GCC) oluşur.



Şekil 2.10: Arduino Mega 2560.

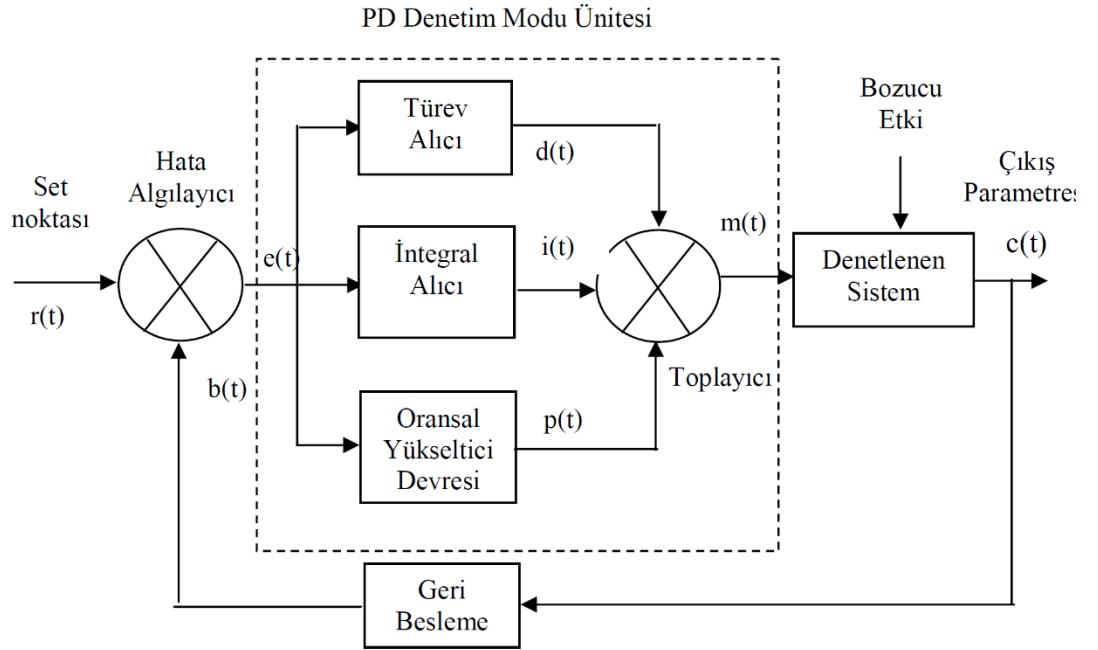
Arduino Mega 2560 ATmega2560 mikrodenetleyici içeren bir Arduino kartıdır. Arduino Mega 2560 'ta 54 tane dijital giriş / çıkış pini vardır. Bunlardan 15 tanesi PWM çıkışı olarak kullanılabilir. Ayrıca 16 adet analog girişi, 4 UART (donanım seri port), bir adet 16 MHz kristal osilatörü, USB bağlantısı, power jakı (2.1mm), ICSP başlığı ve reset butonu bulunmaktadır. Arduino Mega 2560 bir mikrodenetleyiciyi desteklemek için gerekli bileşenlerin hepsini içerir. Arduino Mega 2560 bir bilgisayara bağlanarak, bir adaptör ile ya da pil ile çalıştırılabilir. Arduino Mega, Arduino Duemilanove ya da Diecimila için tasarlanan shieldlerin çoğu ile kullanılabilir.

Arduino programlama dili olarak C++'ı kullanmaktadır, deneyimizde hassas geri beslemeli sistemi oluşturmak için encoderdan aldığımız geri beslemenin üzerine PID işlemini C++ dili ile Arduino kartına yüklememiz gerekmektedir.

2.7 Denetim Yöntemi

2.7.1 Oransal- İntegral-Türevsel (PID) Denetim yöntemi

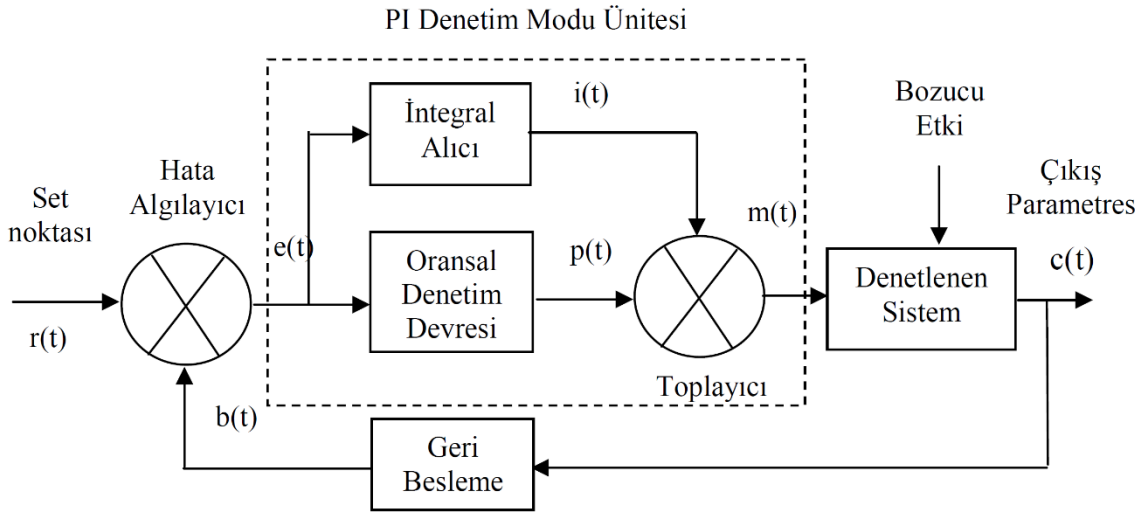
Oransal, integral ve türev terimlerinin bir araya gelmesiyle oluşan PID kontrol daha öncede bahsedildiği gibi basit yapısından dolayı endüstride sıkça kullanılan klasik bir kontrol yöntemidir. PID kontrolör 3-modlu kontrolör olarak da bilinir. İntegral bileşeni büyük yük değişimleri nedeniyle oluşan oransal ofseti (kaymayı) azaltmak ve yok etmek için kullanılır. Türev yöntemi de osilasyon eğilimini azaltır ve hata sinyalini önceden sezen bir etki sağlar. Türev yöntemi özellikle ani yük değişimlerinin olduğu proseslerde çok kullanışlıdır. Diğer bir ifadeyle PID yöntemi PI veya PD denetim yönteminin hatayı kabul edilebilir limitler içerisinde tutamadığı hızlı ve büyük yük değişimleri olan proseslerde kullanılır.



Şekil 2.11: PID prensip şeması.

2.7.2 Oransal-İntegral (PI) Denetim yöntemi

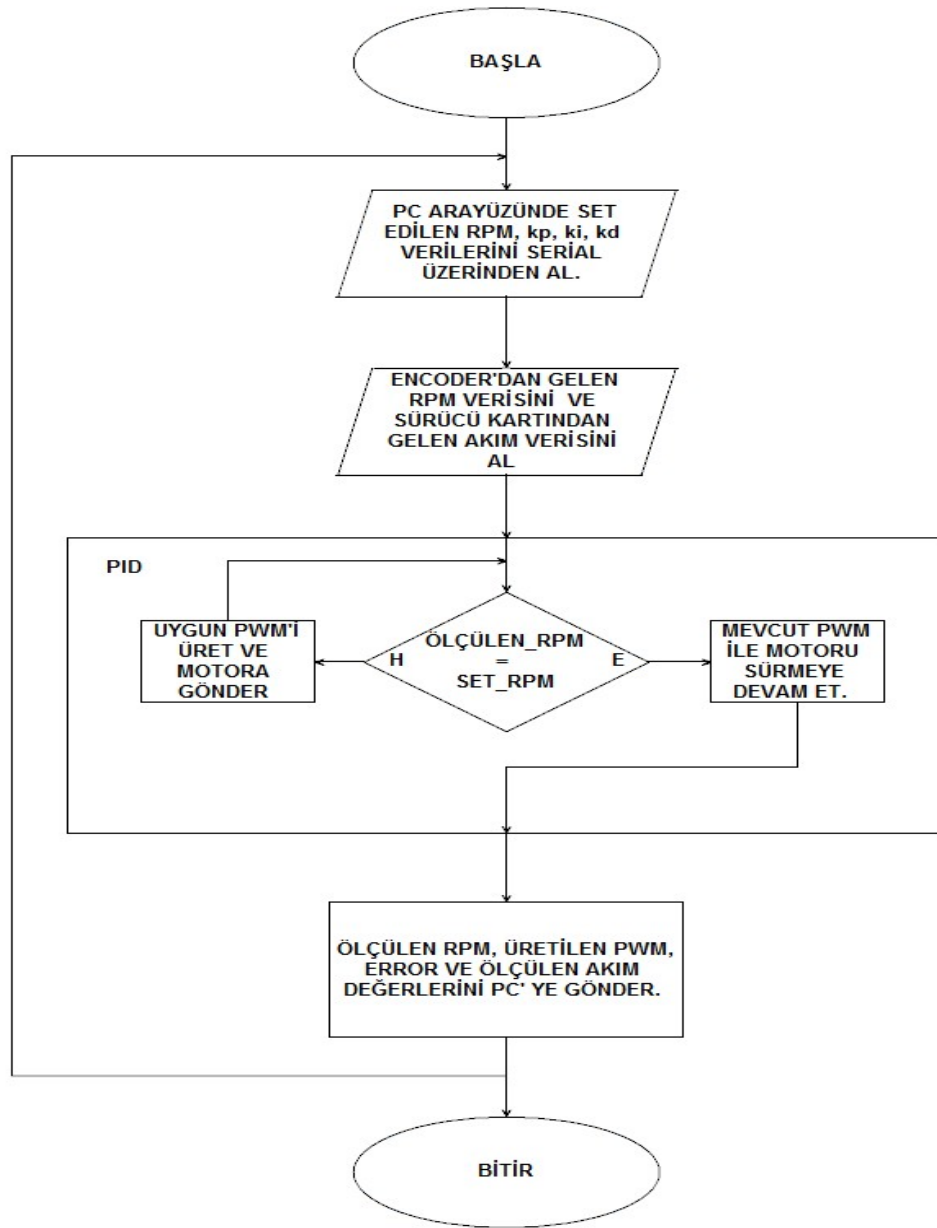
PI denetime başlamadan önce integral denetim yönteminden bahsetmekte fayda vardır. İntegral denetim yöntemi kontrolör çıkışını hata sinyalinin integrali ile orantılı olarak değiştirir. Hata olduğu sürece çıkışta bir değişim meydana getirir. Çoğu sistemde oransal denetim ile başarılabilen denetim derecesinin kabul edilebilir seviyelerde olmadığını belirtmekte fayda vardır. Özellikle birçok sistemde kalıcı durum hatası ya da oransal ofset kabul edilemez. Bu nedenle oransal denetimin en büyük dezavantajı olan kalıcı durum hatasının ortadan kaldırılması ve çıkışta oluşan zayıf denetim sinyallerinin yükseltilmesi gereklidir. İntegral alma yaklaşımı ile kalıcı durum hatası giderilebilir. İntegral etkisi oransal ofseti ortadan kaldıran bir otomatik resetleme sağlar. Bunun yanı sıra yükseltme değeri arttıkça hata da buna bağlı olarak azalacaktır. Bu bağlamda, hata sinyali çok uzun süre ısrarla devam ederse kademeli olarak denetleyici çıkışı yükseltilmelidir. Hata sinyalinin integrali alınıp denetleyici çıkışına integral değeri eklenmelidir. Bu yapılan işlemlerle oransal-integral (PI) denetim elde edilir. Sürekli tip uzun süreli proseslerde ve sapma istenmeyen durumlarda PI veya PID tip denetim seçilebilir.



Şekil 2.12: PI prensip şeması.

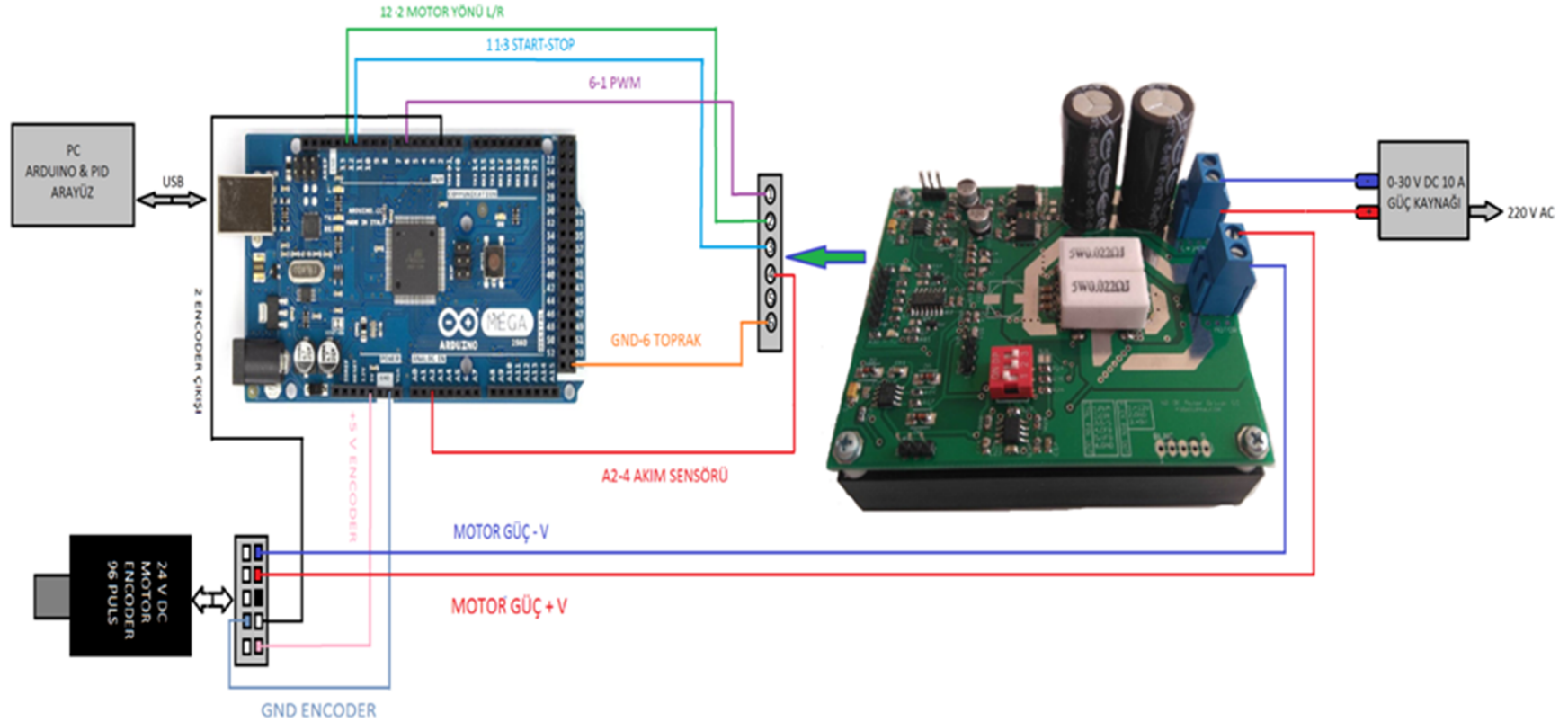
Şekil 2.13'te Arduino mega kartı için yazılan programın akış şeması göstermiştir. Program öncelikle bilgisayar ara yüzünden kontrol edilmesi için microsoft visual studio progmanı ile yazılan arayüz programı ile seri portan haberleştirilmiş ve ara yüzden girilen k_p , k_i ve k_d değerlerini arduino içindeki yazılıma kaydedilmiştir yeni

değerler alındıktan sonra ara yüzden set edilen RPM arduino yazılımına gönderilmiş ve ara yüzden motor aktif edilmesi ile motor seçilen yöne doğru dönmeye başlamıştır, aynı anda encoderdan RPM bilgisi ve sürücü kartından gelen akım bilgileri alınmış, ayarlanan RPM değeri ile encoderdan gelen RPM karşılaştırılmış fark yok ise evet satırından devam ederek aynı PWM ile motor sürülmeye devam edilmiştir, şayet fark var ise hayır satırından devam edilmiş fark için gerekli olan PWM değerine kp, ki, kd oranları uygulanarak yeni PWM ile motor sürülmüş ve istenen PWM'e ulaşılması sağlanmıştır, buradaki kp,ki, ve kd oranları istenen PWM'e ulaşma sırasında aşırı aşım, oturma ve kalıcı durum hatalarını en aza indirecek oranlarda ayarlanmalıdır.



Şekil 2.13: Program Akış Şeması.

2.8 Arduino PID ve Visual studio arayüz programı



Şekil 2.14: Motor Sürücü Devresi Motor+Encoder ve Ardiuno Bağlantı Şeması.

Arduino yazılımı PID ve microsoft visual studio ile yapılan arayüz programı ile haberleşmesi için USB serial kanal kullanımı içermektedir.

Ek-A'da Arduino yazılımı verilmiştir. Bu yazılıma bakacak olursak; öncelikle arduino içerisine güncel <PID_v1.h> PID kütüphane yazılımı gömülmüştür. Kullanıcı arayüzünde istenilen RPM değeri girildikten sonra bu bilgi arduino kartına seri port üzerinden gönderilir. Arduino üzerinde bu RPM değerine karşılık gelen PWM üretilir ve arduinonun 6 nolu çıkışından motor sürücü kartının 1 nolu ucuna çıkış uygulanır. Sürücü kartı bu PWM'e göre akımı ve voltaj ayarlaması yaparak motora belli bir hız sağlar. Bunun sonucunda motor belli bir hızla dönmekte ve encoder bu hıza göre bir dönme pulsü üretmektedir. Kare dalga olarak üretilen bu pulsler arduino kartının 2 nolu girişine uygulanmıştır ve void rpm hesapla alt programı ile ölçülen RPM hesaplanmış olur. Hesaplanan RPM input olarak adlandırılmış ve kullanıcının ayarladığı (set ettiği) RPM ile karşılaştırılmıştır. Aradaki fark değeri hata (error) olarak tanımlanmıştır.

Elde edilen bu hata (error) bilgisi ile kp, ki ve kd değerleri myPID.Compute(); komutu ile kütüphane içerisinde hesaplama işlemine tabi tutulup output değişkenine kaydedilen yeni PWM değeri motor sürücü kartına uygulanarak istenilen RPM'e yakın bir sonuç kalıcı olarak elde edilmiş olur. Sistem kapalı çevrim metoduna PID eklenmesi ile oluşturulmuştur.

Bu koşullar sağlandıktan sonra motorun anlık güç hesaplaması motor sürücü kartının 4 nolu ucundan arduinonun A2 nolu pimine gönderilen akım ve arduino bordunun ürettiği PWM değerlerine karşılık gelen voltajın çarpımı ile bulunmuştur.

$$P=V \times I \quad (2.9)$$

Motorumuzun gücü (P) ve RPM'ini biliyorsak buradan torkuda hesaplayabiliriz. Bir motorun gücü o motorun torkunun açısal hız ile çarpımına eşittir. Buda:

$$P = T \times \omega \quad (2.10)$$

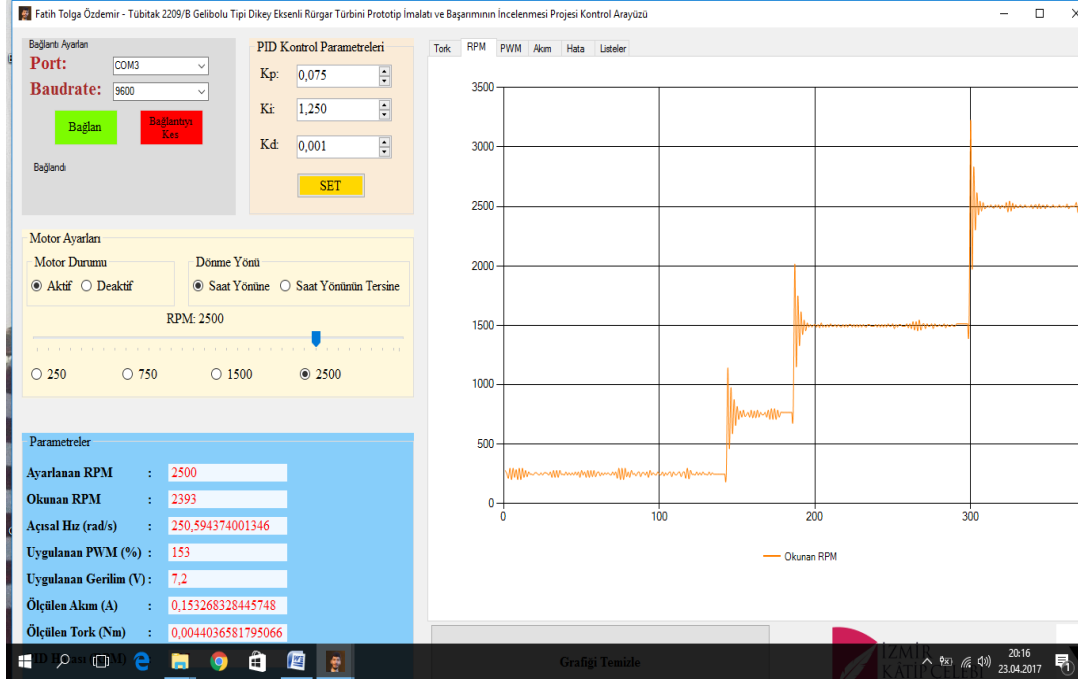
formülü ile bulunur. Açısal hız:

$$\omega = 2 \times \pi \times f \quad (2.11)$$

Buradan

$$\omega = 2 \times \pi \times 1 \text{ saniyedeki tur sayısı (RPM/60)} \quad (2.11-1)$$

formülü ile bulunur. Ayrıca sürücü kartının 3 numarası star-stop girişidir, bu giriş arduinonun 11 numarası ile kontrol edilmiş ve motoru durdurup başlatma komutu buradan sağlanmıştır, sürücü kartının 2 numarası motor yön kontrolü sağ/sol seçimidir, bu giriş arduinonun 12 numarası ile kontrol edilmiş ve motorun sağa ve sola döndürme komutu arayüz programı ile verilebilmektedir. Bu yazılımı Windows yüklü bilgisayar ile kullanılmak üzere laboratuvarında önceden Microsoft Visual Studio 2015 ortamında C# dili kullanılarak tasarlanmış bir arayüz programı kullanılmıştır.



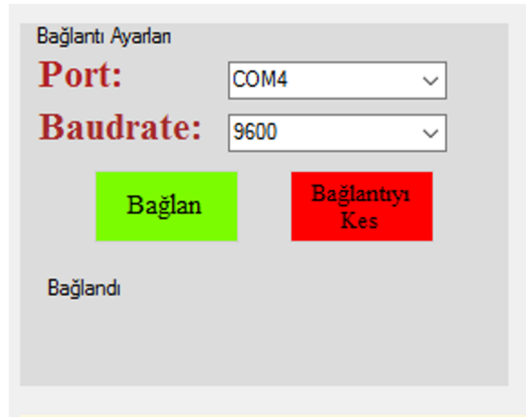
Şekil 2.15: Arayüz Programı.

Bu program verilerin sağlıklı bir şekilde anlık olarak görüntülenmesi ve işlenmesi amacıyla kullanılmaktadır. Verilerin daha sonra yorumlanabilmesi için saklanması oldukça önemlidir.

Programda her bilgi için ayrı grafik penceresi ve tüm bilgilerin anlık olarak gösterildiği listeler bulunmaktadır. Bu bilgilerin kayıt altına alınabilmesi için tüm veriler gerçek zaman değerleri ile birlikte birer listeye eklenmiştir. Bu listelerin veriyi kaydet butonuna basıldığı anda bilgisayarda bulunan “Belgelerim” klasörüne yine kullanıcı tarafından belirlenen isimler ile kaydedilmesi sağlanmıştır. Kaydedilen dosyaların türü neredeyse dünyadaki tüm bilgisayar sistemleri tarafından kullanılabilen bir dosya türü olan “.txt” formatındadır. Arayüz programının genel görünümü Şekil 2.15’te verilmiştir.

Burada ekranın en büyük kısmının grafikleri göstermek için ayrıldığı açıkça görülmektedir. Grafiklerin yorumlanması sayesinde kanat profillerinin performansı hakkında çıkarımlar yapılabilmektedir. Grafik sekmelerinde Tork, RPM, PWM, Akım; Hata ve Listeler bölümleri vardır. Bu alanda alınan tüm verilerin görselleştirilmesi sağlanmıştır. Görselleştirme esnasında ayrı ayrı tüm verilerin farklı renklerdeki grafiklerle gösterilmesi hedeflenmiştir.

Arayüzdeki ilk kısım bağlantı ayarları bölümüdür (Şekil 2.16). Bu bölümde bilgisayarın Arduino ile iletişim kurmasını sağlayan seri port ayarlarının yapılması sağlanmıştır. Port sekmesi o anda Arduino'nun bilgisayarın hangi USB portuna bağlandığını seçmek için, Baudrate sekmesi de bilgisayar ile Arduino'nun hangi hızda haberleşeceklerini belirlemek için kullanılmıştır. Kullanıcı gerekli seçimleri yaptıktan sonra Bağlan sekmesine tıkladığında PC Arduino ile bağlantıyı başlatmakta, veri alıp göndermeye hazır hale gelmektedir. En alttaki metin kısmında kullanıcıya bazı yönergeler gösterilmektedir. Bu yönergeler sayesinde kullanıcının Arduino ile PC arasındaki bağlantıyı sağlıklı bir şekilde yapması sağlanmaktadır.



Şekil 2.16: Bağlantı ayarları bölümü.

Arayüzdeki ikinci kısımda PID kontrolcü için kullanılan parametreleri ayarlamak üzere bir bölüm ayrılmıştır. Bu bölümde ki, kp ve kd değerleri el ile ayarlanabilmektedir. Kullanıcı SET butonuna bastığında bu parametreler seri port üzerinden Arduino 'ya aktarılmaktadır. Kullanıcının bu bölümü kullanarak PID parametrelerini ayarlayabilmesi için önceden Arduino ile seri port bağlantısını ayarları düzgün bir biçimde yapılmış bir şekilde başlatması gerekmektedir.

PID Kontrol Parametreleri

Kp: 1,050

Ki: 1,250

Kd: 0,300

SET

Şekil 2.17: PID parametreleri ayarlama bölümü.

Programın bu bölümünde motor için gerekli ayarlamalar yapılmaktadır. İlk kısımda motorun çalışıp çalışmama durumu ayarlanmaktadır. Ayrıca sonra verilecek olan kodlarda da görülebileceği üzere aktif seçimi programın zamanlayıcısını çalıştırmasına ve periyodik olarak veri alınmasını sağlamaktadır. Dönme yönü kısmından seçilen seçenek sayesinde motorun hangi yönde dönmesi gerektiği bilgisi Arduino'ya gönderilmektedir. Kaydırılabilir bar kısmından motorun hangi hızda döneceği ayarlanmaktadır. Ayrıca en alttaki seçenekler sayesinde motorun dönme hızı hızlı bir şekilde ayarlanabilmektedir.

Motor Ayarları

Motor Durumu

Aktif Deaktif

Dönme Yönü

Saat Yönüne Saat Yönünün Tersine

RPM: 0

250 750 1500 2500

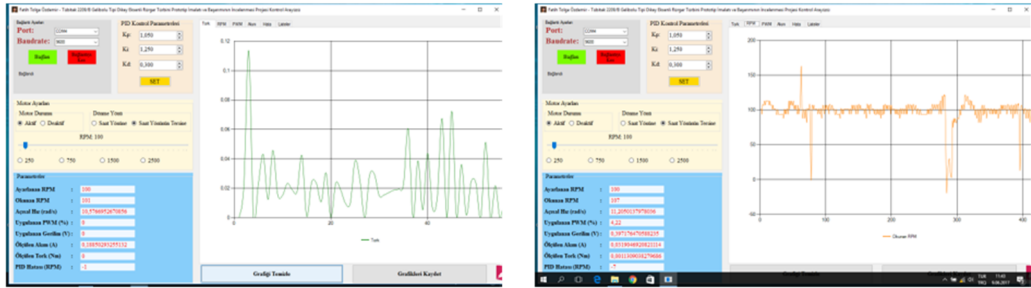
Şekil 2.18: Motor ayarları bölümü.

Programın anlık veri göstergeleri kısmında Arduino'dan alınan ve programın içinde işlenen veriler anlık olarak gösterilmektedir. Bu veriler; ayarlanan RPM, okunan RPM, açısız hız, uygulanan PWM, uygulanan gerilim, ölçülen akım, ölçülen tork ve PID hatası bilgileridir.

Parametreler		
Ayarlanan RPM	:	100
Okunan RPM	:	101
Açısal Hız (rad/s)	:	10,5766952670856
Uygulanan PWM (%)	:	26,09
Uygulanan Gerilim (V)	:	2,45552941176471
Ölçülen Akım (A)	:	0,0475645161290323
Ölçülen Tork (Nm)	:	0,0110427752111438
PID Hatası (RPM)	:	-1

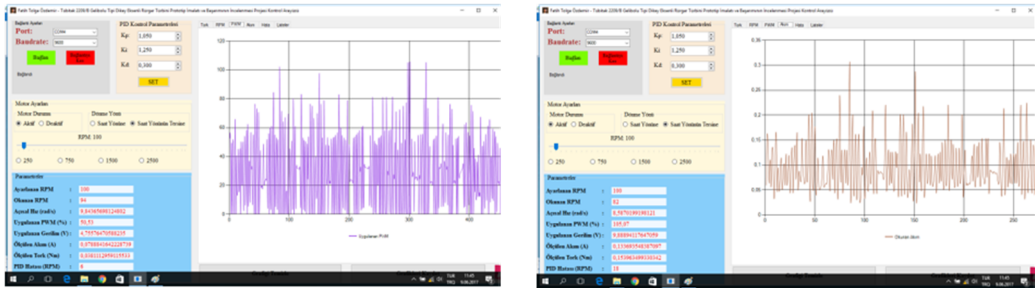
Şekil 2.19: Anlık veri göstergeleri Kısmı.

Kullanıcı grafiklere ilave olarak elde edilen verileri buradan da anlık olarak izleyebilmektedir. Buradaki verilerden okunan RPM, uygulanan PWM ve ölçülen akım verileri Arduino üzerinden alınmakta, ayarlanan RPM verisi programın kendi üzerinden çekilmekte ve diğer veriler de bu verilerin çeşitli hesaplamaları sonucu elde edilmektedir.



(a)

(b)



(c)

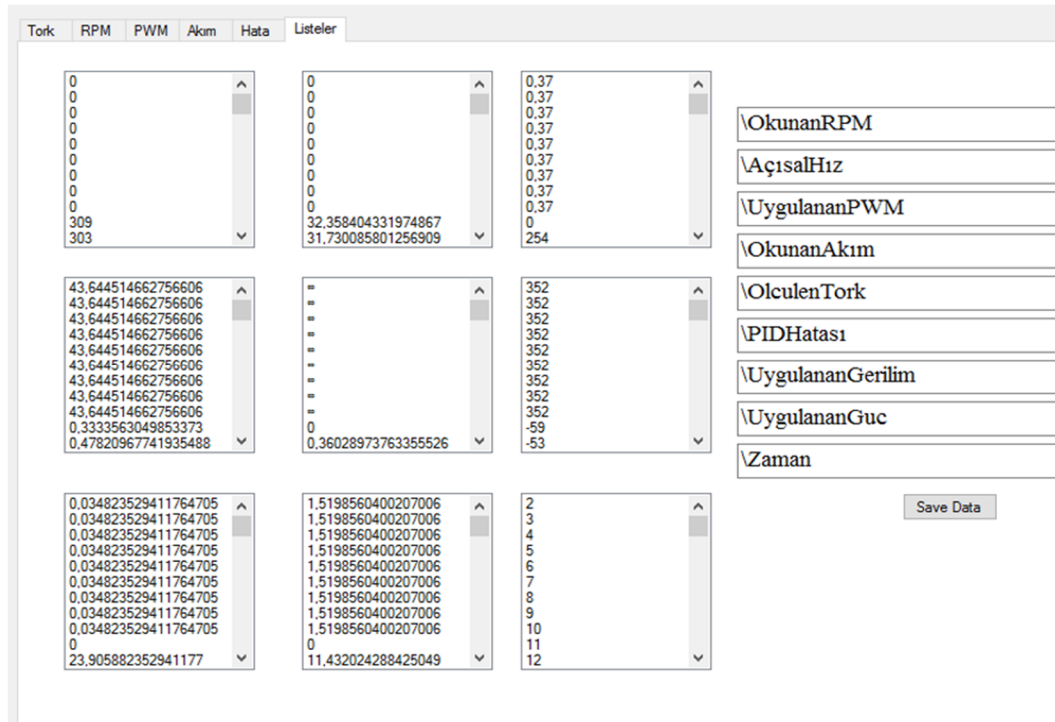
(d)

Şekil 2.20: Veri görselleştirme bölümü.

Bu bölümde daha önce de bahsedildiği gibi elde edilen verilerin görselleştirilmesi işlemi yapılmaktadır. En altta bulunan grafiği temizle butonuna tıkladığında program ekrandan bulunan grafiği temizlemekte ve aynı zamanda zamanlayıcının değerini sıfırlamaktadır. Grafikleri kaydet butonu da grafiklerde elde edilen verilerin kaydedilmesi içindir

Grafiklerin olduğu bölümün en son sekmesinden listeler kısmı bulunmaktadır. Bu kısımda kullanıcının deney boyunca elde ettiği verilerin bir listesi gösterilmektedir. Kullanıcının Save Data butonuna basması durumunda listelerde gösterilmekte olan verilen “Belgelerim” klasörü içerisine butonun üstünde bulunan düzenleme alanlarındaki isimleri ile kaydedilmektedir. Bu işlem sayesinde verilerin daha sonra erişilebilir olması ve daha sonra farklı grafik ortamlarında kullanılması amaçlanmıştır.

Bu arayüz ile elde edilen verilen bu raporun sonuçlar kısmında gösterilmiştir. Grafiklerin çözünürlüklerinin artırılabilmesi adına sonuçlar kısmında verilen grafikler Microsoft Excel ortamında hazırlanmıştır. Excel ortamında grafik hazırlamanın en büyük avantajı grafiklerin Word’e aktarılırken vektörel olarak aktarılması yani boyutunun değiştirilmesi durumunda herhangi veri ve çözünürlük kaybına uğranmamasıdır. Program geliştirilirken kullanılan kodlar aşağıda renklendirilmiş olarak verilmiştir.



Şekil 2.21: Listeler sekmesi.

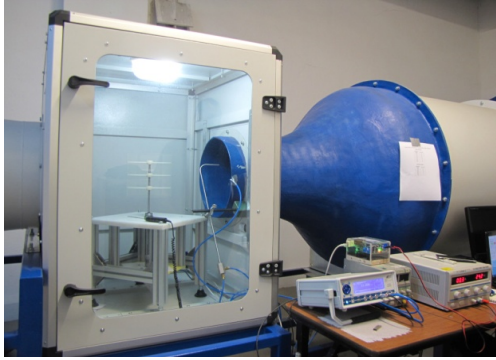
Bu işlemler yazılım sayesinde visual studio arayüz programında grafiksel olarak anlık alınmaktadır. Bu grafikler tork, RPM, akım ve hata olmak üzere dört başlıktadır.

Böylelikle yaptığımız çalışmada rüzgar türbin deneylerinde önemli parametre olan kanat uç hız oranı hassas şekilde kontrol etme imkanı ve yine önemli bir parametre olan torku, torkmetre olmadan ölçme imkanı sağlanmıştır.

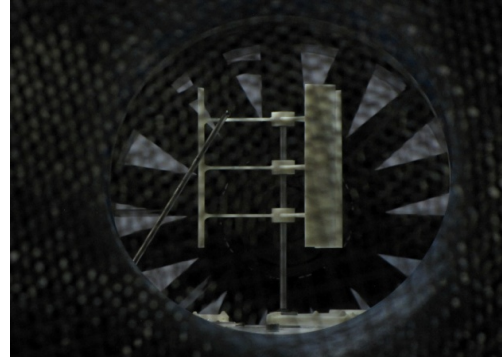
3. BULGULAR VE DEĞERLENDİRME

3.1 Deney Yöntemi

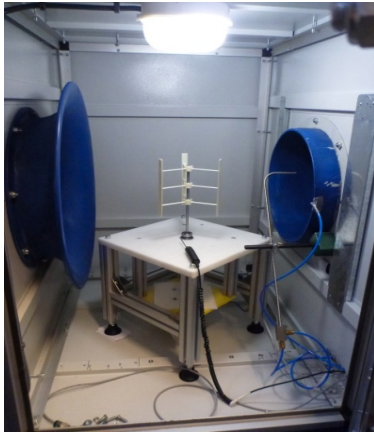
Test düzeneği Şekil 3.1 ve Şekil 3.2’de gösterildiği gibi Makine Mühendisleri odasındaki özellikleri bir önceki bölümde verilen rüzgar tüneline yerleştirilmiştir. Çalışma ekibi test düzeneğindeki bağlantıları yapmış rüzgar türbini ile ilgili ölçüm işlemine başlamadan önce, deney setinin ön çalışma ve boşa çalışma verileri alınmış ve yer emniyeti açısından çalışma ortamı düzenlenmiştir.



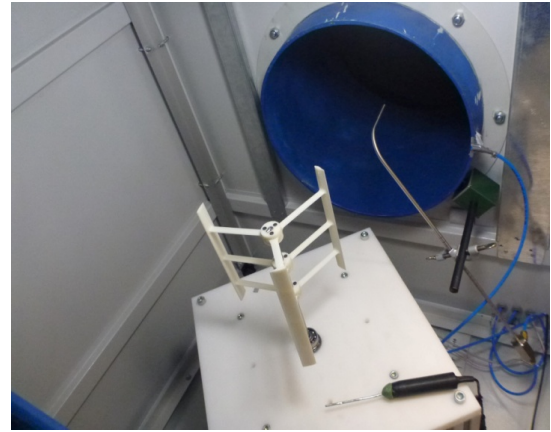
(a)



(b)

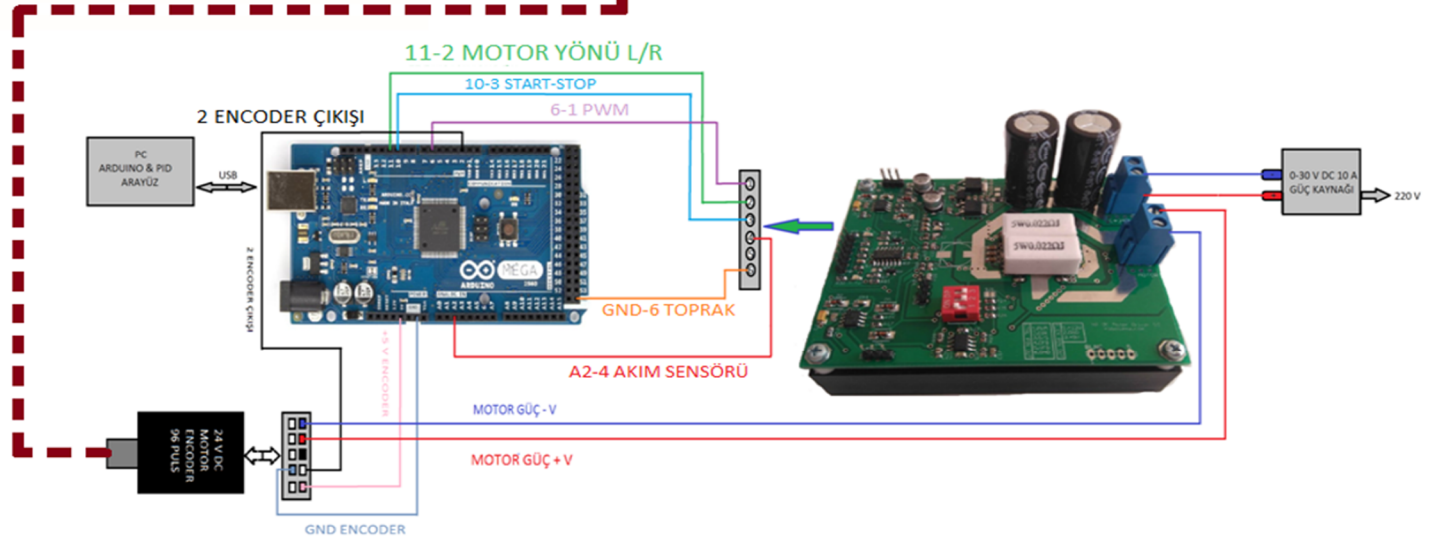
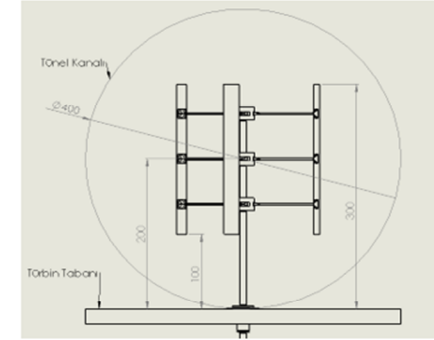
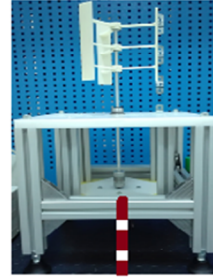


(c)



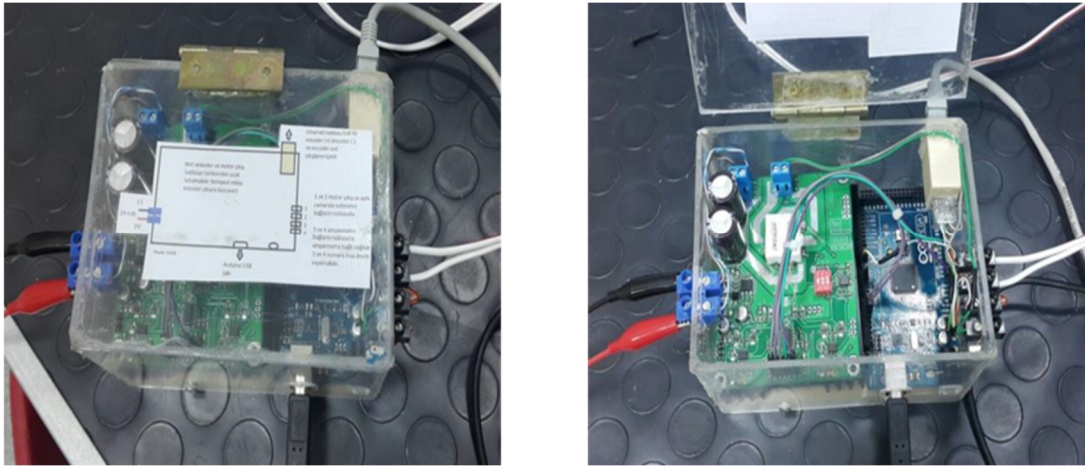
(d)

Şekil 3.1: Makine Mühendisleri Odası KALMEM Rüzgar Tüneli.



Şekil 3.2: Deney Setinin Prensi Şeması.

Test esnasında olası kazaları önlemek için tüm elektronik donanımlar için kutu hazırlanmıştır. Ayrıca tünel içindeki test kabini içerisinde türbin harici ekipman bulundurulmadığından motor ve enkoder kabloları 5 metre uzatılmıştır. Kutu ve kablolar şekil 3.3'te verilmiştir. Çalışma ekibine önceden planlanan Çizelge 3.1 ve Çizelge 3.4 deki tablo verileri işlem sırası ve test esnasındaki görevleri hatırlatılmış herhangi bir acil durumda hareket tarzı odada bulunan tüm personele anlatılmıştır buradaki acil durum rüzgar türbin protatipinin yapısal olarak bozulup rüzgar türbini fanına doğru emilmesi durumunda makine veya personele zarar vermesi olarak değerlendirilmiştir, bu durumda acil durdurma düğmesi tüm personele gösterilmiştir.



Şekil 3.3: Çalışma Ortamındaki karışıklığı önlemek için elektronik donanımlar kutu içerisine yerleştirilmiştir.

Deney odasının hava sıcaklığı klimalar vasıtası ile 24 C°'ye ayarlanmıştır. Konum olarak deniz seviyesinde bulunduğu için basıncı 1 atm olarak alınmış ve havanın yoğunluğu (ρ) bu şartlar altında 1,200 kg/m³ olarak belirlenmiştir. Deneye başlamadan önce ön hazırlık olarak kanat uç hız oranı ve Re sayıları için türbin çalışma aralıklarını belirleyen çizelgeler oluşturulmuştur. Çizelge 3.1'de farklı açısal hız ve serbest rüzgar hızlarına karşılık gelen kanat uç hız oranı değerleri görülmektedir. Kullanılan motorun limitleri dahilinde, farklı çalışma koşulları altında ölçüm yapılabilen kanat uç hız oranı değerleri tabloda yeşil renkli gösterilmiştir. Çizelgede görüldüğü gibi en düşük kanat uç hız oranı değeri 0.05 en yüksek kanat uç hız oranı değeri ise 7.33 olarak gerçekleşmiştir. Bu aralık Darrieus tipi rüzgar türbinlerini incelemek için uygun bir aralıktır.

Çizelge 3.1: Kanat uç hız oranı Çalışma Sahası.

		V_{∞} m/s										
RPM	ω RAD	2,00	3,00	5,00	7,00	8,00	10,00	11,00	14,00	15,00	17,00	20,00
100,00	10,47	0,52	0,35	0,21	0,15	0,13	0,10	0,10	0,07	0,07	0,06	0,05
200,00	20,93	1,05	0,70	0,42	0,30	0,26	0,21	0,19	0,15	0,14	0,12	0,10
300,00	31,40	1,57	1,05	0,63	0,45	0,39	0,31	0,29	0,22	0,21	0,18	0,16
400,00	41,87	2,09	1,40	0,84	0,60	0,52	0,42	0,38	0,30	0,28	0,25	0,21
500,00	52,33	2,62	1,74	1,05	0,75	0,65	0,52	0,48	0,37	0,35	0,31	0,26
600,00	62,80	3,14	2,09	1,26	0,90	0,79	0,63	0,57	0,45	0,42	0,37	0,31
700,00	73,27	3,66	2,44	1,47	1,05	0,92	0,73	0,67	0,52	0,49	0,43	0,37
800,00	83,73	4,19	2,79	1,67	1,20	1,05	0,84	0,76	0,60	0,56	0,49	0,42
900,00	94,20	4,71	3,14	1,88	1,35	1,18	0,94	0,86	0,67	0,63	0,55	0,47
1000,00	104,67	5,23	3,49	2,09	1,50	1,31	1,05	0,95	0,75	0,70	0,62	0,52
1100,00	115,13	5,76	3,84	2,30	1,64	1,44	1,15	1,05	0,82	0,77	0,68	0,58
1200,00	125,60	6,28	4,19	2,51	1,79	1,57	1,26	1,14	0,90	0,84	0,74	0,63
1300,00	136,07	6,80	4,54	2,72	1,94	1,70	1,36	1,24	0,97	0,91	0,80	0,68
1400,00	146,53	7,33	4,88	2,93	2,09	1,83	1,47	1,33	1,05	0,98	0,86	0,73
1500,00	157,00	7,85	5,23	3,14	2,24	1,96	1,57	1,43	1,12	1,05	0,92	0,79
1600,00	167,47	8,37	5,58	3,35	2,39	2,09	1,67	1,52	1,20	1,12	0,99	0,84
1700,00	177,93	8,90	5,93	3,56	2,54	2,22	1,78	1,62	1,27	1,19	1,05	0,89
1800,00	188,40	9,42	6,28	3,77	2,69	2,36	1,88	1,71	1,35	1,26	1,11	0,94
1900,00	198,87	9,94	6,63	3,98	2,84	2,49	1,99	1,81	1,42	1,33	1,17	0,99
2000,00	209,33	10,47	6,98	4,19	2,99	2,62	2,09	1,90	1,50	1,40	1,23	1,05
2100,00	219,80	10,99	7,33	4,40	3,14	2,75	2,20	2,00	1,57	1,47	1,29	1,10
2200,00	230,27	11,51	7,68	4,61	3,29	2,88	2,30	2,09	1,64	1,54	1,35	1,15
2300,00	240,73	12,04	8,02	4,81	3,44	3,01	2,41	2,19	1,72	1,60	1,42	1,20
2400,00	251,20	12,56	8,37	5,02	3,59	3,14	2,51	2,28	1,79	1,67	1,48	1,26
2500,00	261,67	13,08	8,72	5,23	3,74	3,27	2,62	2,38	1,87	1,74	1,54	1,31

Test işlemlerinde iki farklı veter uzunluğuna sahip kanat yapısı kullanılmıştır. Bu kanat yapıları için Re sayısı;

$$R_e = \frac{\rho CU}{\mu} \quad (1.3)$$

formülünden hesaplanmıştır. Burada, $\rho = 1,2 \text{ kg/m}^3$, dinamik viskozite $\mu = 0,00001846 \text{ N m}^2/\text{s}$, yarı çap $r = 0,1 \text{ m}$ veter uzunluğu $C = 0,02 \text{ m}$ ve $C = 0,04 \text{ m}$ değerleri için kanadın çizgisel hızı (U) ve Re sayısı değerleri Çizelge 3.2 ve Çizelge 3.3 'te verilmiştir, bu çizelgeler birinci bölümde de bahsedildiği gibi akış karakteristiğini belirlemek için oluşturulmuştur.

Çizelge 3.2: 100-1200 RPM Aralığında çizgisel Hız Re Sayısı.

RPM	100	200	300	400	500	600	700	800	900	1000	1100	1200
ω (rad/s)	10,47	20,93	31,40	41,87	52,33	62,80	73,27	83,73	94,20	104,67	115,13	125,60
U (m/s)	1,04	2,09	3,14	4,18	5,23	6,28	7,32	8,37	9,42	10,46	11,51	12,56
Re (C=0,02 m)	1352	2717	4082	5434	6800	8165	9517	10882	12247	13599	14964	16329
Re (C=0,04m)	2704	5434	8165	10869	13599	16329	19034	21764	24494	27198	29928	32659

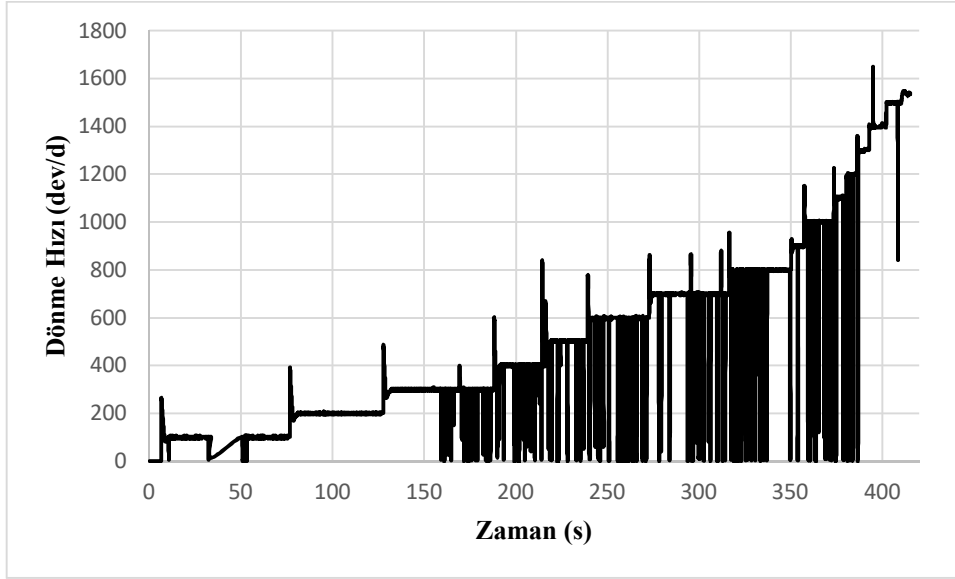
Çizelge 3.3: 1300-2100 RPM Aralığında çizgisel Hız Re Sayısı.

RPM	1300	1400	1500	1600	1700	1800	1900	2000	2100
ω (rad/s)	136,07	146,53	157,00	167,47	177,93	188,40	198,87	209,33	219,80
U (m/s)	13,60	14,65	15,70	16,74	17,79	18,84	19,88	20,93	21,98
Re (C=0,02 m)	17681	19047	20412	21764	23129	24494	25846	27211	28576
Re (C=0,04m)	35363	38093	40823	43528	46258	48988	51692	54423	57153

Yukarıdaki tablolar test için kullanılan veter uzunluğu 2 cm ve 4 cm olan türbinler için çalışma aralığını belirlemek için yapılmıştır. Veter uzunluğu 0,02 m olan kanat yapısında Re sayısı 1352 ile başlamış değerlendirme aşaması 1400 dev/d kadar yapılabilmiş ve Re aralığı 1400 dev/d'da 19047 Re'ye karşılık gelmiştir. Veter uzunluğu 0,04 m olan ikinci kanat yapısında bu aralık 2704 ile 38093 Re sayısına karşılık gelmiştir. Büyük kanatta Re sayısı iki kat artmasına rağmen istenilen Re sayısı aralığına tam ulaşamamış düşük Re sayılarında kalınmıştır. Şekil 3.32'de düşük Re sayısından kaynaklanan negatif bölgelerin fazlalığı gösterilmiş çok az aralıkta pozitif değer elde edilmiştir.

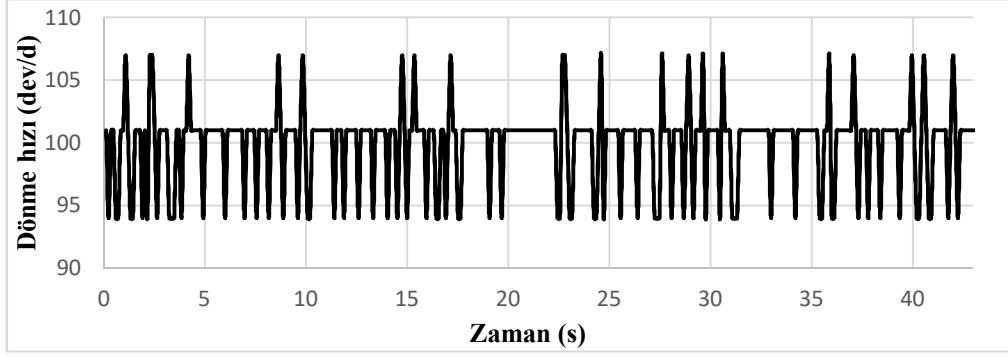
3.2 Veri İşleme

Test işlemine veter uzunluğu 2 cm olan türbin (küçük kanat) için PID ve PI kontrolcüsü uygulanarak öncelikle motor sürücü arayüz programdan devir sayısı 100 dev/d ayarlandı ve devir sayısının kararlı hale gelmesi beklendi, sonrasında benzer işlem devir sayısı 100 dev/d'dan 100'er dev/d aralıklarla kademeli olarak artırılmıştır. Her bir kademede devir sayısının kararlı duruma gelmesi beklenmiş sistemin çalışması incelenmiş ve arayüz programı sayesinde RPM, güç, tork, motorun anlık olarak çektiği akım ve voltaj değerleri alınmıştır. Arayüz programından alınan veriler excel programı ile grafiğe dönüştürülmüş arduino ve arayüz programı arasındaki zaman uyumsuzluğundan kaynaklandığı düşünülen okuma hataları tespit edilmiştir. Grafik değerlendirme aşamasında buradaki bariz okuma hataları manuel düzeltilerek grafikler temizlenmiştir. Şekil 3.4'de okuma hatası içeren dönüş hızı verilerine bir örnek görülmektedir.

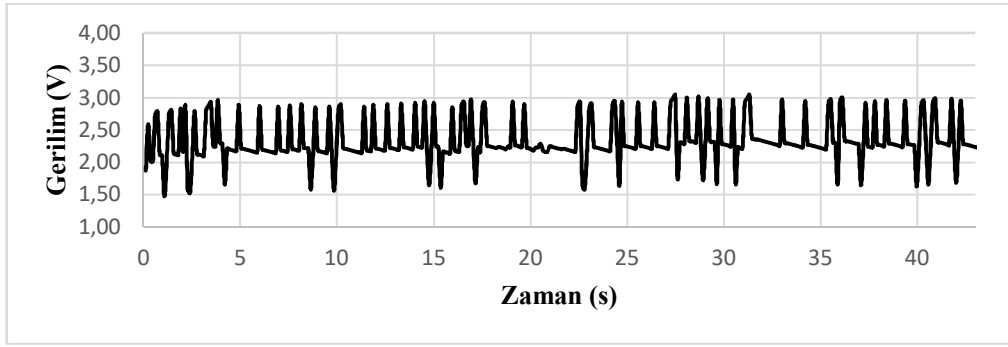


Şekil 3.4: Örnek okuma hatası içeren RPM grafiği.

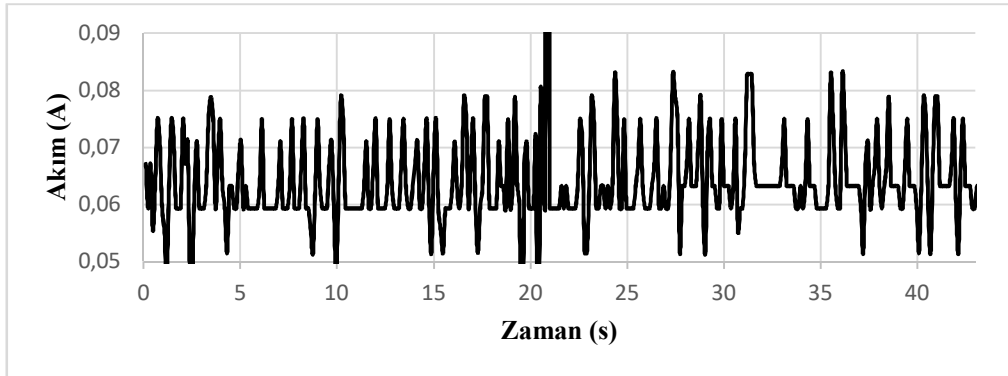
Arduino yazılımında 120 milisaniyede bir veri alınmaktadır. 1000 milisaniye 1 saniyeye eşit olduğu için $1000/120=0,12$ oranı bulunmuş, bu oran veri sayısı ile çarpılarak saniye değeri hesaplanmış ve yatay grafikte veri miktarı saniye cinsinden zamana dönüştürülerek gösterilmiştir.



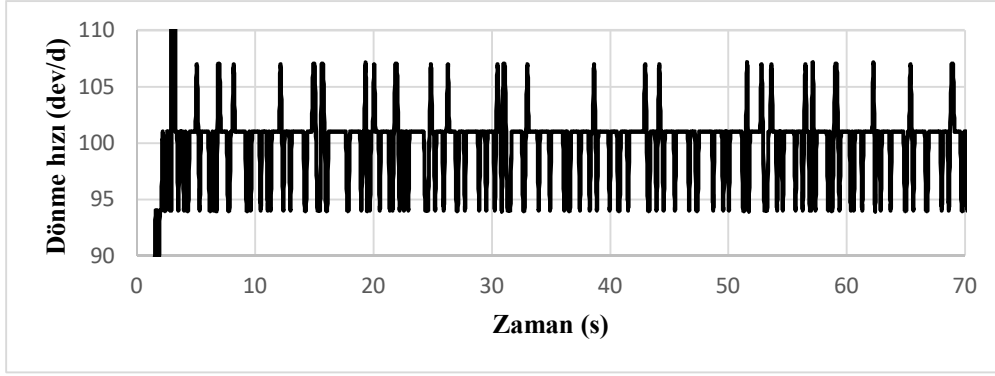
Şekil 3.5: Yüksüz durum için PID kontrolcü ile 100 dev/d'ya sabitlenen dönme hızının zamanla değişimi.



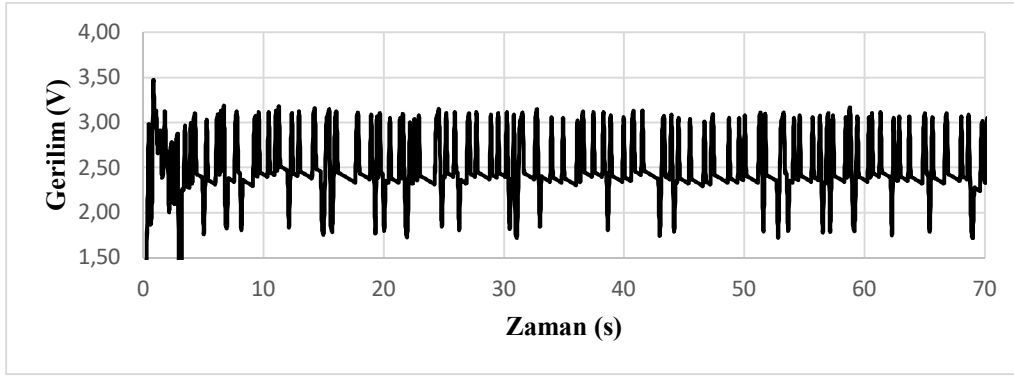
Şekil 3.6: Yüksüz durum için PID kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için gerilimin zamanla değişimi.



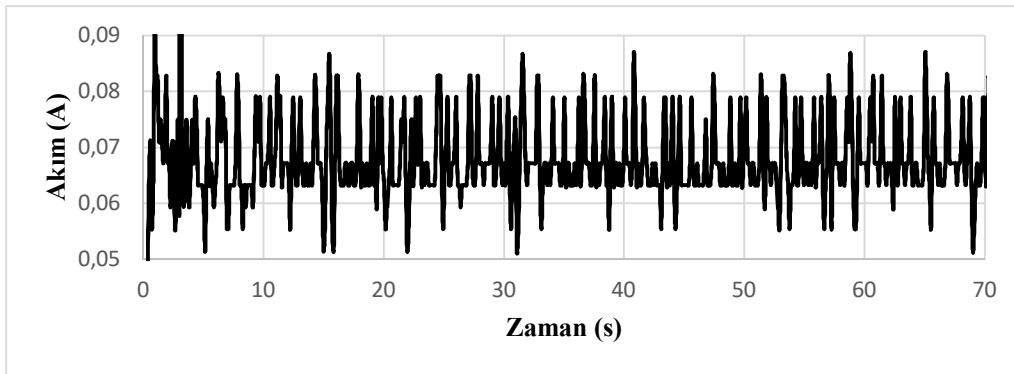
Şekil 3.7: Yüksüz durum için PID kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için akımın zamanla değişimi.



Şekil 3.8: Yüksüz durum için PI kontrolcü ile 100 dev/d'ya sabitlenen dönme hızının zamanla değişimi.



Şekil 3.9: Yüksüz durum için PI kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için gerilimin zamanla değişimi.



Şekil 3.10: Yüksüz durum için PI kontrolcü ile 100 dev/d'ya sabitlenen dönme hızı için akımın zamanla değişimi.

Şekil 3.5'te PID 100 dev/d değerleri elde edilmiş, PID test işlemine ait akım ve gerilim değerleri şekil 3.6 ve şekil 3.7'de elde edilmiştir.

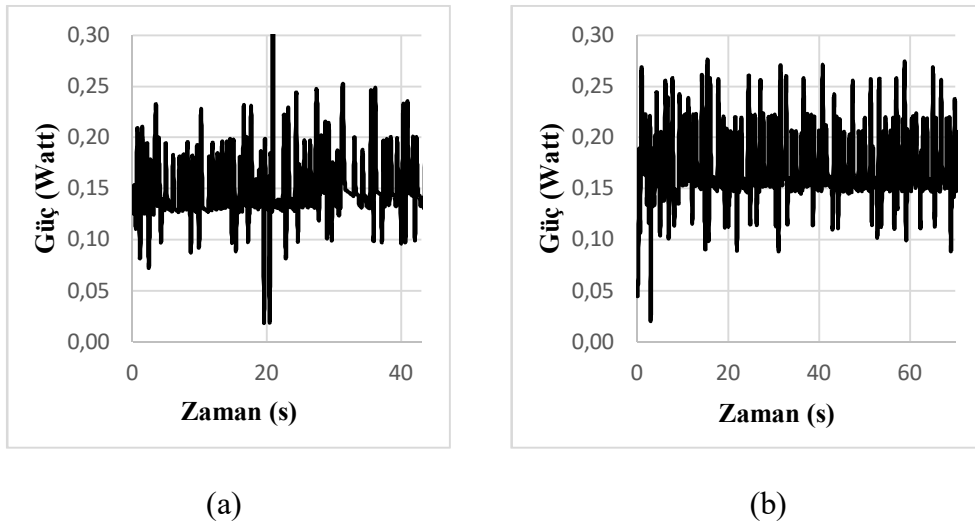
Şekil 3.8'de PI 100 dev/d değerleri elde edilmiş, PI test işlemine ait akım ve gerilim değerleri şekil 3.9 ve şekil 3.10'da elde edilmiştir.

Grafikteki veriler test işleminin 100 dev/d bölümünden örnek olarak alınmıştır. Test işleminin tamamı 100 dev/d'dan 1400 dev/d'ya kadar devam etmektedir. Her iki kontrolcüde (PID ve PI) motorun dönme hızına karşılık motorun harcadığı gerilim ve akım değerleri alınmıştır. Bir elektriksel makinanın (bu deneyde DC motor) gücü, harcadığı akım ve gerilimin çarpımıyla elde edilir, denklem:

$$\text{Güç(Watt)} = \text{Gerilim (V)} * \text{Akım (A)} \quad (2.9)$$

denklemden hesaplanır. Test düzeneyinde akım doğrudan motor sürücü kartının 4 nolu pininden Arduino kartının A2 nolu girişine uygulanması ile akım değeri yazılıma aktarılır, gerilim değeri ise PWM değerinin oranı olarak yazılım tarafından hesaplanır, tam PWM çıkışı 255 değeri için besleme geriliminin tamamı 24 volta karşılık gelmektedir ve bu oran doğrusaldır, motora gönderilen PWM değerine karşılık hesaplanan voltaj hesaplandıktan sonra denklem 3.1 ile güç hesaplaması yazılım tarafından yapılır.

Şekil 3.11 (a)'da 100 dev/d için PID güç değerleri, (b)'de 100 dev/d için PI güç değerleri gösterilmiştir.



Şekil 3.11: (a) PID ve (b) PI kontrolcülerinde 100 dev/d için yazılım tarafından hesaplanan güç değerleri.

Test düzeneyinde kullanılan motorun gücü (P) değeri elde edilmiş ve devir hızı'nda Şekil 3.5 ve şekil 3.8'de hesaplanmıştı, devir hızı ve gücü bilinen bir DC motorun torku :

$$T = \frac{P}{\omega} \quad (3.1)$$

formülü ile bulunmuştur.

T : Tork

P : Güç

ω : Açısal hız'dır.

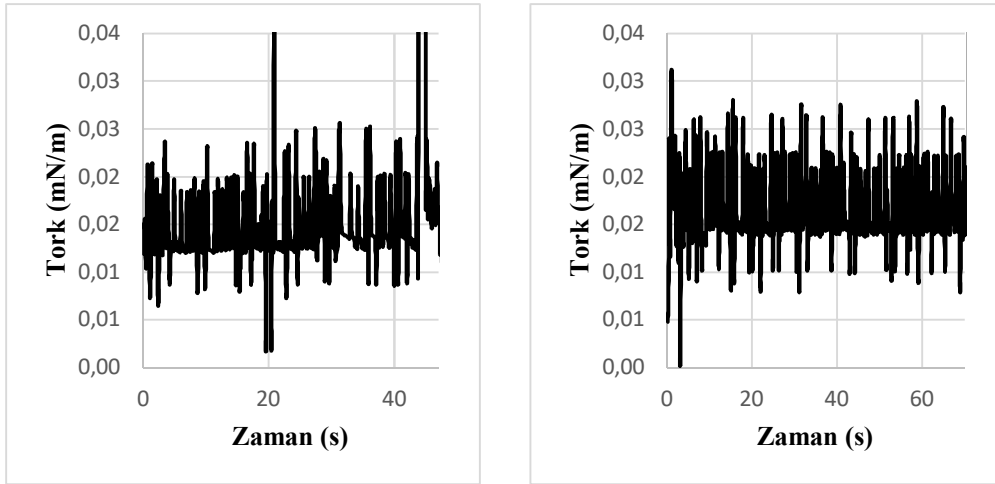
Açısal hız:

$$\omega = 2 \times \pi \times f \quad (2.11)$$

Buradan

$$\omega = 2 \times \pi \times 1 \text{ saniyedeki tur sayısı (RPM/60)} \quad (2.11-1)$$

formülü ile açısal hız bulunmuş, arduino yazılımı dönü sayısını, akım ve gerilim değerlerini, arayüz yazılımına iletmış, arayüz yazılımı bu değerlerden, güç ve tork bilgilerini elde etmiştir.



(a)

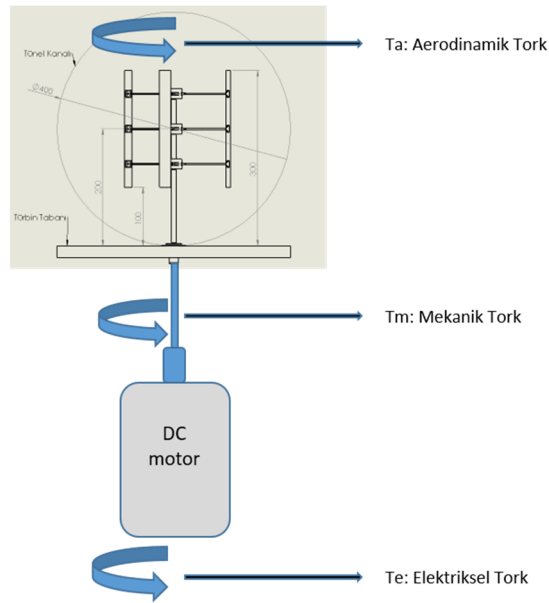
(b)

Şekil 3.12: (a) PID ve (b) PI kontrolcülerinde 100 dev/d için yazılım tarafından hesaplanan Tork değerleri.

Kurgulanan tork ölçüm işleminin işleyişi denklem 3.2’de verilmiştir.

$$\text{Mekanik Güç Farkı} = \text{Elektriksel Güç Farkı} \quad T \times \omega = V \times I \quad (3.2)$$

Formüldeki T değeri torku, ω değeri açısal hızı, V değeri gerilimi ve I değeri akımı ifade eder. Test esnasında kontrolcü yardımıyla türbini sabit bir devirde döndürürken rüzgar verildiğinde, rüzgar o anda türbin kanatlarının davranışına göre tork değerine pozitif veya negatif etki yapar. Tork ölçümü türbinin boşta dönme elektriksel güç tüketimi değerinden rüzgar yükü altındaki elektriksel güç tüketiminin çıkartılıp, bu değer test esnasındaki sabit açısal hıza bölünmüştür. Türbin tüm koşullar altında sabit devirde olduğundan sistemin ataletinden kaynaklı tork üretimi bu farka dahil değildir. Böylece bu farkla yalnızca kanatların ve dolayısı ile türbinin ürettiği tork ölçülebilmektedir, ölçüm işlemi arayüz programı sayesinde anlık tork değeri ile kaydedilerek grafikler oluşturulmuştur. Şekil 3.13’te sistem üzerindeki Tork bölgeleri gösterilmiştir.



Şekil 3.13: T_e : Elektriksel Tork, T_m : Mekanik Tork, T_a : Aerodinamik Tork oluşum bölgeleri.

Kurulan sistemde dönme hızı sabit olduğu için elektriksel tork (T_e) değerinin, mekaniksel tork (T_m) değerine eşit ve zıt yönde olduğu kabul edilmiştir. Aerodinamik tork (T_a) ise kanatları etkileyen rüzgar hızına bağlı olarak değişmektedir. Rüzgar hızı sıfır ise T_a sıfırdır.

$$V_{\infty} = 0 \Rightarrow T_a = 0 \quad (3.3)$$

Deneyde öncelikle her iki kontrolcü ile sıfır rüzgar hızı altındaki türbin verileri kaydedilmiş, deneyin ikinci aşaması ise farklı rüzgar hızlarında türbin verileri kaydedilmiştir. İkinci aşamada rüzgar hızı sıfırdan farklı olduğu için T_a değeri sıfırdan farklı olmuştur. T_e değeri T_a ve T_m değerlerinin toplamına eşittir.

$$T_e = T_a - T_m \quad (3.4)$$

Sisteme uygulanan rüzgar yükü, T_a 'yı artı ya da eksi yönde oluşturur.

$$T_a < T_m \Rightarrow T_e < 0 \Rightarrow \text{sisteme iş giriyor} \quad (3.5)$$

$$T_a > T_m \Rightarrow T_e > 0 \Rightarrow \text{sistemden iş çıkıyor} \quad (3.6)$$

Deneyin PID ve PI kontrolcülere altındaki verileri bölüm 3.3'te incelenmiş rüzgar yükü altında PID kontrolcünün daha kararlı çalıştığı gözlenmiştir bir sonraki bölüm 3.4'te PID kontrolcü kullanılmıştır. Deneyin birinci aşamasındaki alınan tork değerleri ikinci aşamasındaki tork değerlerinden çıkarıldığında T_a üzerinde oluşan yani kanatlar üzerindeki rüzgar hızından oluşan tork hesaplanmıştır. Tork sayesinde hesaplanan türbin gücü, uygulanan kinetik rüzgar gücüne bölünmesi ile güç katsayısı (C_p) elde edilmiştir. Güç katsayısı/kanat uç hız oranı grafikleri ile türbin karakterizasyonu bölüm 3.4'te incelenmiştir.

3.3 Farklı Denetim Yöntemlerinden Alınan Sonuçların Karşılaştırılması

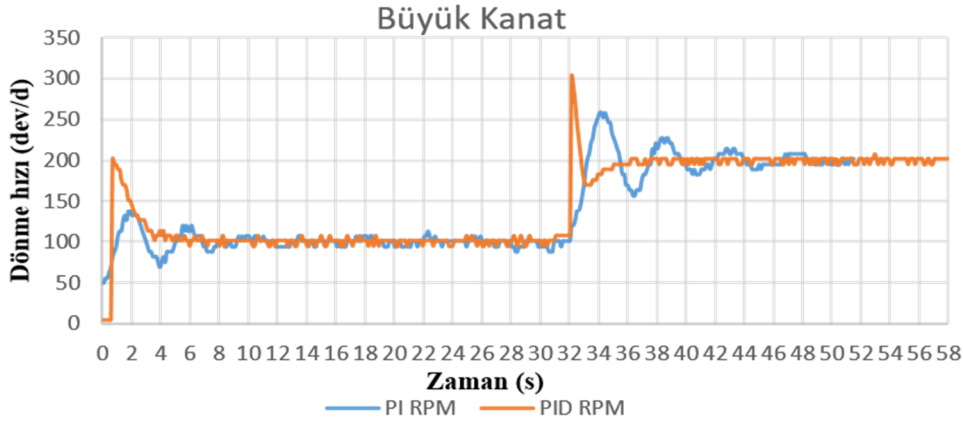
Deney işlem sırası Çizelge 3.4'te verilen plana göre yapılmıştır. Bu planda sabit rüzgar hızında farklı devirlerde büyük kanat PI ve PID küçük kanat PID kontrolcülü deneyler yapılmış ve sabit devirde değişken rüzgar hızlarıyla PID kontrolcülü deneyler yapılmıştır. Sabit rüzgar değişken devirde büyük kanat ile kontrolcü karşılaştırmaları yapılmış PID kontrolcünün PI kontrolcüsüne göre daha iyi sonuçlar verdiği grafiklerle ortaya çıkmış, ikinci adımda türbin karakterizasyonu aşamasında PID kontrolcü kullanılmıştır.

Çizelge 3.4: Deney Uygulama Planı.

Sabit rüzgar değişken RPM(100-1600 RPM)					Sabit RPM değişken rüzgar (2-20 m/s)				
Büyük kanat (4 cm)			Küçük kanat (2 cm)		Büyük kanat (4 cm)			Küçük kanat (2 cm)	
	PI	PID	PI	PID		PI	PID	PI	PID
BOŞTA	x	x	x	x	RPM				
3 m/s	x	x		x	300				x
5 m/s	x	x		x	600				x
7 m/s	x				900				x
8 m/s		x		x	1200				x
10 m/s	x				1400				x
11 m/s		x		x	1500				x
14 m/s		x		x	1600				x
15 m/s	x				Yüksek hız 1000		x		
					Yüksek hız 1500		x		
					Yüksek hız 2000		x		

PID kontrolcü uygulamalarında kullanılan kontrolcülerin katsayılarının belirlenmesi en büyük zorluklardan birini oluşturmaktadır. Önelikle kontrolcü katsayılarını belirlemek için yaygın bir yöntem olan Ziegler-Nichols metodu kullanılmış fakat gerek okuma hatalarından gerekse sistemin cevap gecikmesinden bu katsayıların başarılı olmadığı görülmüştür. Bu durumda sistemin istenilen en iyi performansı sergilediği değerler el ile ayarlanmıştır. En iyi değerler oransal katsayının (P) 1,050, integral katsayısının (I) 1,250 ve türev katsayısının (D) 0,300 olduğu durumlarda gerçekleşmiştir. Bu katsayılar PID kontrolcüsü için kullanılmıştır, PI kontrolcüsü için D katsayısı 0 olarak atanmıştır. PID ve PI kontrolcülerini rüzgar altında test etmeden önce her iki tipteki türbinin (veter 2 ve 4 cm) sıfır rüzgar altındaki çalışma durumları yani boştaki çalışmaları test edilerek kontrolcülerin devir sayısını sabitleme başarısını incelenmiştir.

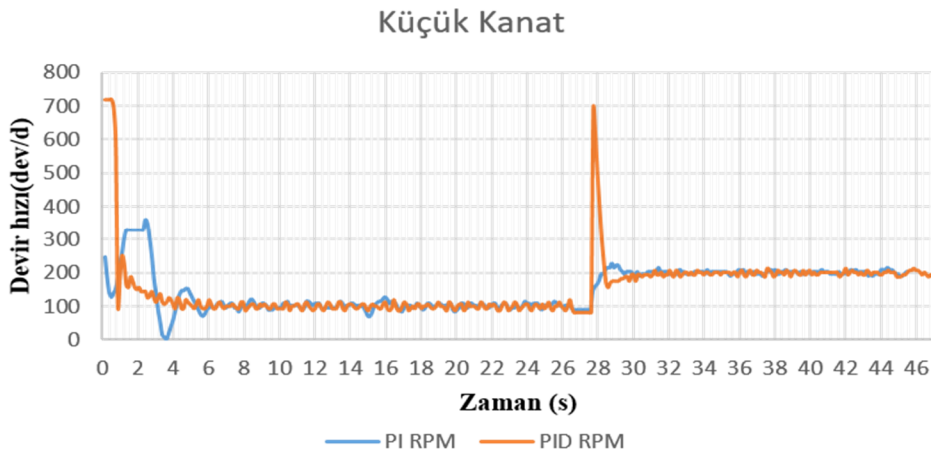
Şekil 3.14 grafiğinde referans dönüş hızı 100 ve 200 dev/d'ya ayarlandığı durumlar görülmektedir. Ayarlanan 200 dev/d'ya karşı elde edilen sonuçlar Çizelge 3.5'te verilmiştir, PID yönteminde aşırı aşım PI yöntemine göre fazla olsada oturma zamanı yani ayarlanan değere gelme zamamı daha azdır.



Şekil 3.14: Boşta Büyük Kanat PID ve PI dev/d verileri.

Çizelge 3.5 : Büyük kanat PID ve PI değerlendirmesi.

	PID	PI
Aşırı aşım	%100	%50
Oturma zamanı	4,3 sn	10,25 sn
Kalıcı durum hatası	7 rpm	7 rpm



Şekil 3.15: Boşta Küçük Kanat PID ve dev/d verileri.

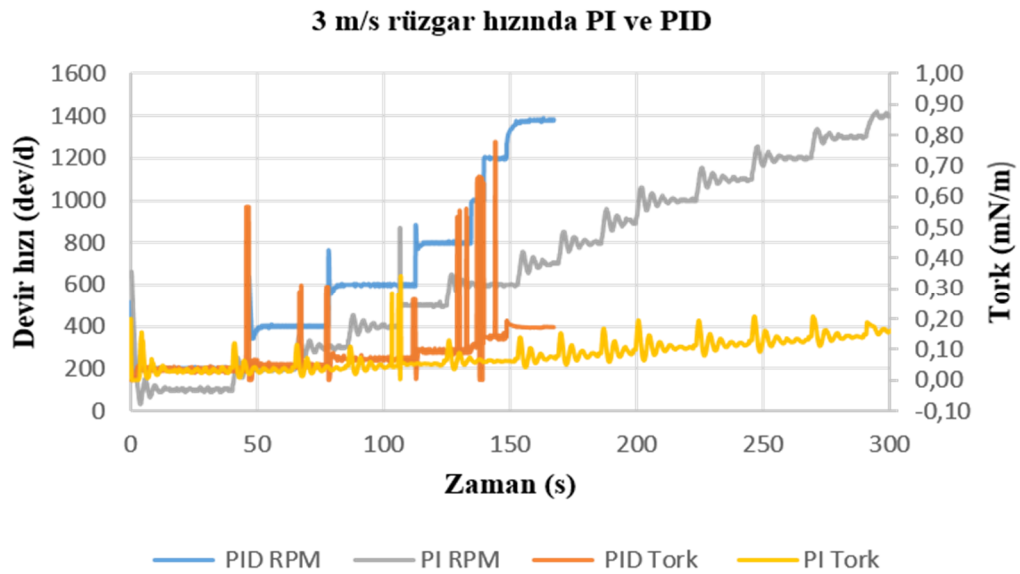
Şekil 3.15'te referans dönme hızı 100 ve 200 dev/d olarak ayarlanmıştır. Ayarlanan 200 RPM'e karşı elde edilen sonuçlar Çizelge 3.6'da verilmiştir, pid yönteminde aşırı aşım pi yöntemine göre fazla olsada oturma zamanı yani ayarlanan değere gelme zamanı daha azdır.

Çizelge 3.6: Küçük kanat PID ve PI değerlendirmesi.

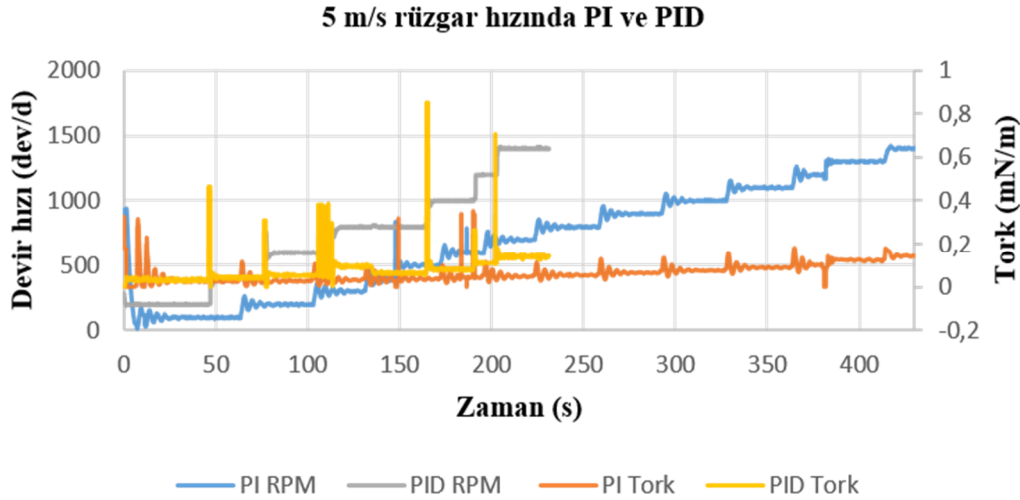
	PID	PI
Aşırı aşım	%600	%27
Oturma zamanı	2,25 sn	3,5 sn
Kalıcı durum hatası	7 rpm	7 rpm

Büyük ve küçük kanat arasındaki aşırı aşım ve oturma zamanı farkları motor üzerindeki kanatların yaratmış olduğu ataletten kaynaklanmaktadır. PID ve PI arasındaki farka bakacak olursak ayarlanan değere daha kısa sürede gelmesi bakımından PID kontrolcününün PI kontrolcüye göre daha üstün olduğu görülmektedir.

a. Sabit rüzgar hızında PID ve PI kontrolcülü büyük kanat tork RPM grafikleri



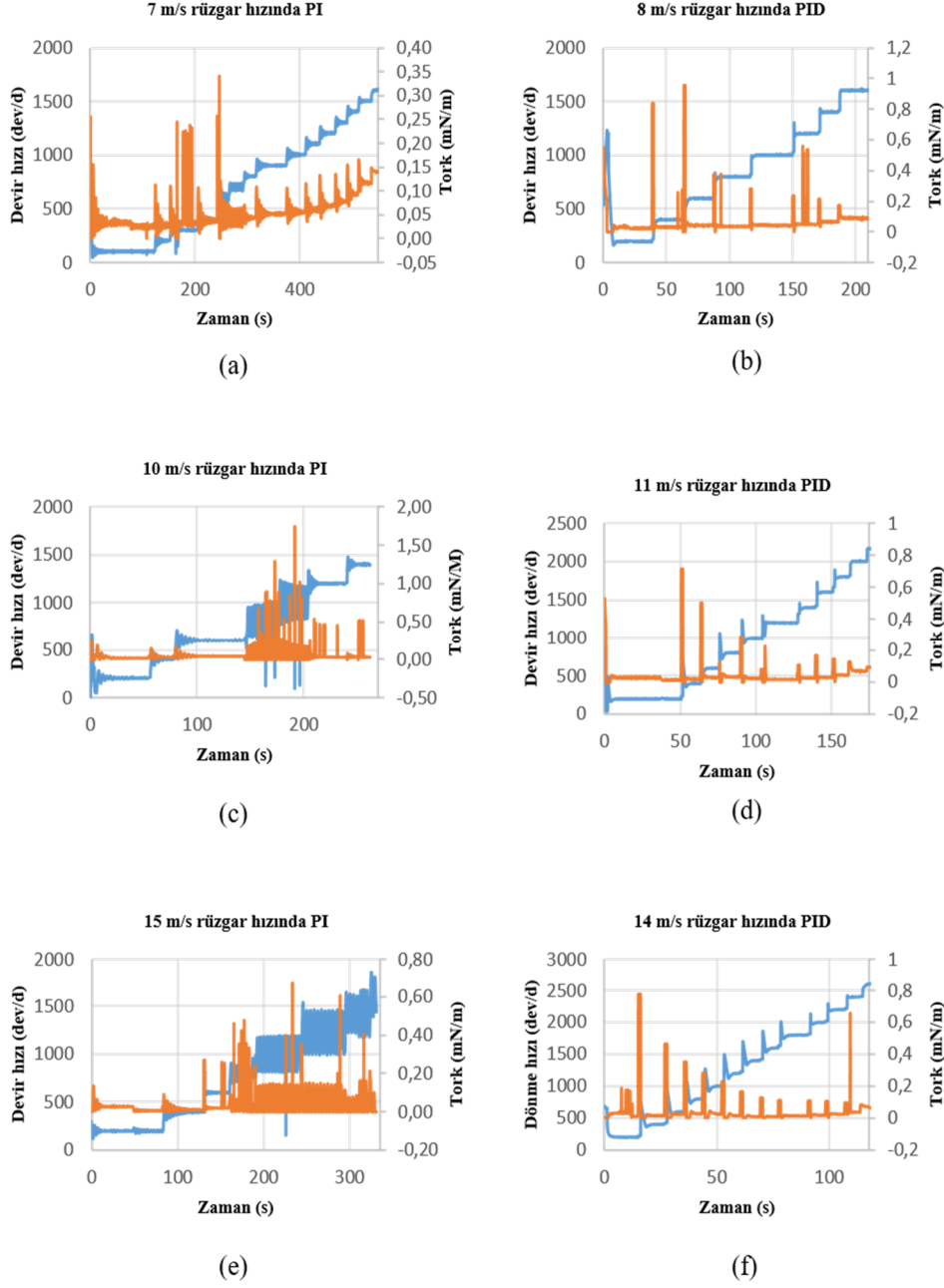
Şekil 3.16: 3 m/s rüzgar hızında PID ve PI kontrollü Tork- dev/d verileri.



Şekil 3.17: 5 m/s rüzgar hızında PID ve PI kontrollü Tork- dev/d verileri.

Şekil 3.16’da 3 m/s rüzgar hızı için dönme hızı tork grafikleri PID ve PI kontrolcüler için karşılaştırılmıştır. PID kontrolcünün 200, 400, 600, 800, 1000, 1200 ve 1400 dev/d seviyelerini PI kontrolcüsündeki aynı dev/d seviyelerine göre daha kararlı ve ayarlanan dev/d ‘ya ulaşmada oturma zamanı bakımından daha kısa sürede ulaştığı görülmektedir. Aynı grafikteki tork değerlerine bakacak olursak PID kontrolcüsünde dev/d değerini korumak için kontrolcünün anlık olarak PI kontrolcüsüne göre daha yüksek değerlere ulaşmıştır. 200 dev/d ‘dan 400 dev/d geçişi sırasında PID torku 0,60 mN/m ‘ye anlık olarak ulaşmış PI kontrolcüsünde 300 dev/d’den 400 dev/d geçişi sırasında 0,10 mN/m ‘ye anlık olarak ulaşmıştır.

Şekil 3-17’de 5 m/s Serbest rüzgar hızı için PID ve PI kontrolcü dev/d ve tork değerleri karşılaştırılmıştır. Değerlerin Şekil 3-16’daki 3 m/sn PID ve PI kontrolcü grafiği ile benzerlik gösterdiği görülmüştür. Bu durum deneyin tekrarlanabilir olduğunu göstermiştir. Şekil 3-17’de PID kontrolcüsü Şekil 3-16’da olduğu gibi daha kararlı çalıştığı görülmüştür.

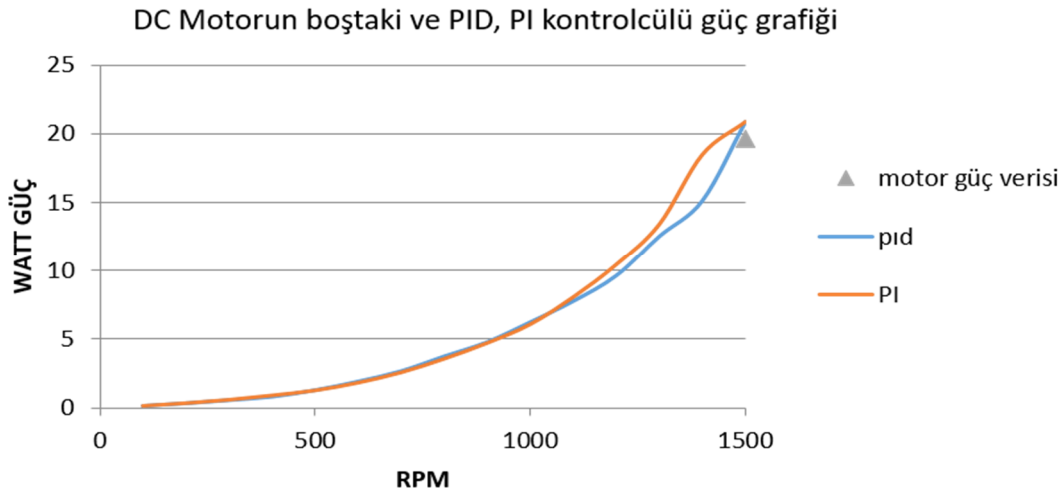


Şekil 3.18: (a) 7 m/s PI, (b) 8 m/s PID, (c) 10 m/s PI, (d) 11 m/s PID, (e) 15 m/s PID, (f) 14 m/s PID Kontrolcülü Tork- dev/d verileri.

Şekil 3.18’de 7 m/s, 10 m/s ve 15 m/s rüzgar hızlarında PI kontrolcüsü kullanılmıştır, 7 m/s PI dev/d -Tork grafiğinde 800 – 1000 dev/d arası PI kontrolcününün işlevsiz hale geldiği, 15 m/s rüzgar hızı için PI dev/d-Tork grafiğinde ise 600 RPM’den sonrası için işlevsiz hale geldiği gözlemlenmiştir.

8 m/s, 11 m/s ve 14 m/s rüzgar hızlarında ise PID kontrolcüsü kullanılmış bu grafiklerde ayarlanan dev/d değerlerinin kontrolcü tarafından kararlı bir şekilde sağlandığı görülmüştür.

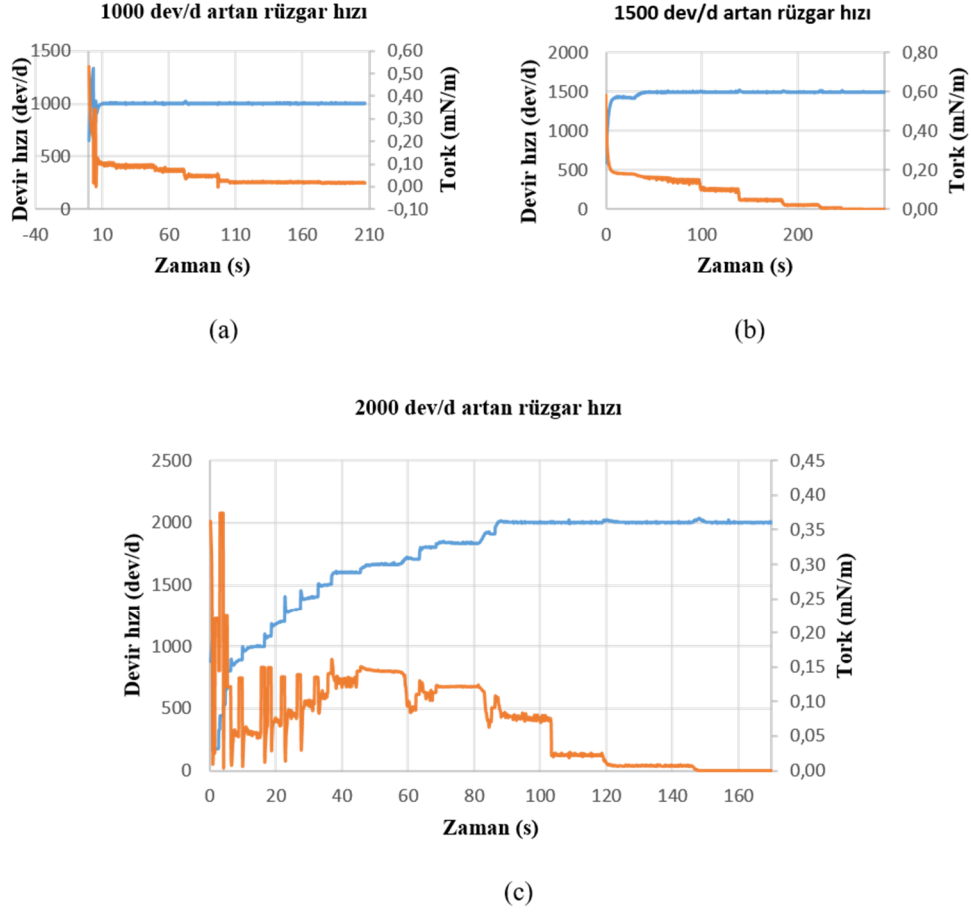
Bu sonuçlardan sonra PI kontrolcü kullanılması durumunda sistem rüzgar yükü olmadan alınan verileri ile rüzgar yükü altındaki (3 m/s, 5 m/s, 7 m/s, 10 m/s, 15 m/s rüzgar hızındaki) veriler değerlendirildiğinde rüzgar türbin testi için istenen RPM’i sabit tutma, ortalama oturma zamanı değerleri açısından uygulanabilir bir yöntem olmadığı, PID kontrolcüsünün ise rüzgar yükü olmadan alınan veriler ve rüzgar yükü altındaki (3 m/s, 5 m/s, 8 m/s, 11 m/s, 14 m/s rüzgar hızındaki) verilerinde istenen dev/d’yı 600 RPM’e kadar %4 600 RPM’den 1400 RPM’e kadar %2 hata payı ile sabit tuttuğunu, ortalama oturma zamanının 4 saniye olduğu sonucuna ulaşılmıştır. Testin ikinci aşaması olan türbin karakterizasyonu için kullanılacak yöntemin bu deney seti için PID kontrolcüsü olması gerektiği sonucuna varılmıştır.



Şekil 3.19: DC motor güç verileri.

Şekil 3.19’da güç-dev/d grafiği DC motorun katalog verisinden hesaplanmıştır, bu veri motora 24 volt DC akım uygulanıp harici bir frenle istenen RPM değerindeki motorun harcadığı güç olarak elde edilmiştir. Ayrıca PID ve PI kontrollündeki motorun güç grafiği bu tezde yapılan deneyler sonucu oluşturulmuş ve bu üç veri karşılaştırılmıştır. Bu tezde 1460 RPM’e motora takılan türbinden dolayı 255 PWM’de ulaşılmıştır, bu PWM değeri 24 volta karşılık gelmektedir. Grafik DC motorun şekil 2.5’te verilen üretici güç verileri ile tezdeki güç verileri 20 Watt’ta 24 volt seviyesinde 1460 RPM’de 19.8 Watt’lık güçte birleştiğini göstermiş ve çalışmada yapılan deney sonuçlarının DC motor katalog verileri ile yakın sonuçlar verdiğini desteklemiştir

b. Sabit RPM artan hızdaki PID kontrolü ile dev/d- tork verileri.



Şekil 3.20: (a) 1000 dev/d , (b) 1500 dev/d , (c) 2000 dev/d artan rüzgar hızında tork verileri.

Türbin karakterizasyon deneyine geçmeden önce 3 boyutlu yazıcıda yapılan Dairrus tipi dikey rüzgar türbinin rüzgar tüneline uygulanan rüzgar hızlarına verdiği tepkiyi net görebilmek ve yapılan bu modelin çalıştığından emin olabilmek için Şekil 3.20’de ayrıntıları verilen yüksek devir sayılarında serbest rüzgar hızı artırılarak PID kontrolcünün tepkisi incelenmiştir. Şekil 3.20 (a) testinde 1000 RPM sabit tutularak rüzgar hızı sırasıyla 14- 15- 16- 17- 18- 20 m/s olarak uygulanmıştır. Grafikte bu hızlara karşılık gelen tork değerleri 0,1- 0,07- 0,05- 0,03- 0,02-0,01 Nm olduğu gözlenmiştir. Rüzgar hızı artıkça türbin rüzgardan dönme kuvveti almış motorun 1000

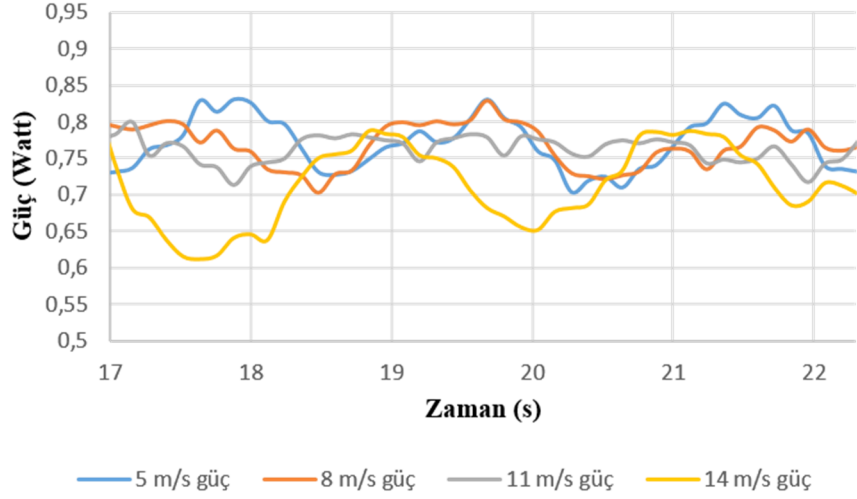
RPM’de sabit tutulması için kullanılan voltaj ve akım dolayısıyla tork azalmıştır. Bu deney 1500 RPM için (Şekil 3.20 (b)) tekrarlanmış sonuçlar bir önceki deneye benzer olduğu görülmüştür. Deneyde kullandığımız türbin motorunun torkunu hangi rüzgar hızında sıfırlayacağı sorusuna cevaplamak ve türbinin maksimum RPM değerini ulaştırmak için şekil 3.20 (c)’deki deney yapılmıştır. Motor ile yüksüz çalışmalarda 1460 RPM’e kadar çıkabildiğimiz için 2000 devire rüzgar tünelineki kontrollü verilen rüzgar sayesinde kademeli olarak çıkmıştır. 1500 RPM’e kadar tork grafiği artan seyir izlemiştir. 1500 devirden sonra kademeli olarak 14 m/s, 16 m/s, 18 m/s rüzgar hızı verilmiş RPM 2000 devire rüzgar sayesinde ulaşırken tork değeri düşüş eğilimine geçmiştir. 2000 RPM’e ulaşıldıktan sonra RPM PID kontrolcü sayesinde sabit tutulabilmiş ve kademeli olarak 20 m/s, 21 m/s, 22 m/s, 24 m/s rüzgar hızları uygulanmış tork değerinin 24 m/s rüzgar hızında 0’a geldiği gözlemlenmiştir. Bu noktada türbin 2000 RPM’i rüzgar sayesinde sağlamış motor tork üretmemiştir. Bu deney yapılan test düzeneyinin rüzgar tüneline uygulanan rüzgarlara net tepkiler verdiğini göstermiş ve bir sonraki aşama olan türbin karakterizasyonu için uygulanabilir olduğunu göstermiştir.

3.4 Türbin Karakterizasyonu

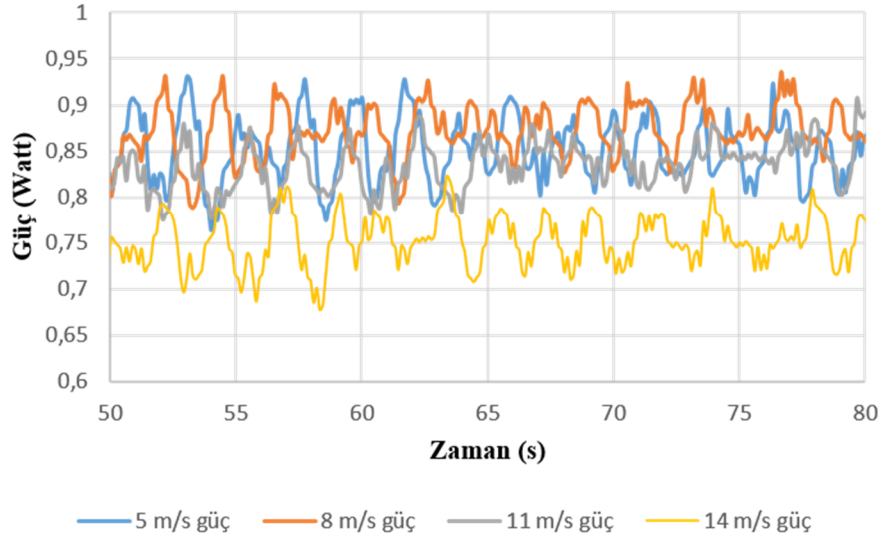
3.4.1 Küçük kanat türbin karakterizasyonu

Veri işleme ile ilgili bölümde anlatılan yöntem kullanılarak farklı çalışma koşulları altında türbin verileri incelenmiş ve türbin karakterizasyonu yapılmıştır.

300 RPM sabit devirde dönen darrieus tipi türbinin değişken serbest akış hızlarındaki ilgili verileri Şekil 3.21’de verilmiştir.



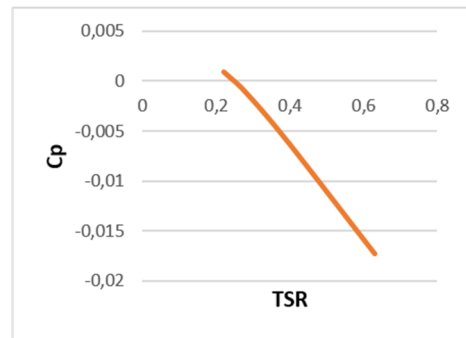
(a)



(b)



(c)



(d)

Şekil 3.21: 300 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.

300 RPM için birinci grafikte (Şekil 3.21(a)) türbin, aynı çalışma ortamında farklı zamanlarda birden fazla çalıştırılıp boşta çalışma koşullarında elde edilen veriler görülmektedir. Bu verilerin aynı trendi takip ettiği grafikte gözlemlenmiş ve deneyin tekrarlanabilir olduğu teyit edilmiştir.

İkinci grafikte (Şekil 3.21(b)) ise yine sabit devirde, türbin rüzgar yükü altındayken motorun harcadığı güç verisi elde edilmiştir. Boşta ve yük altındaki verilerin ortalama değeri alınıp daha sonra C_p hesaplamalarında türbin üzerindeki net gücü bulmak için bu grafikteki değerler kullanılmıştır.

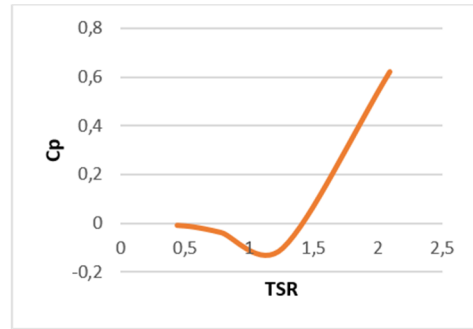
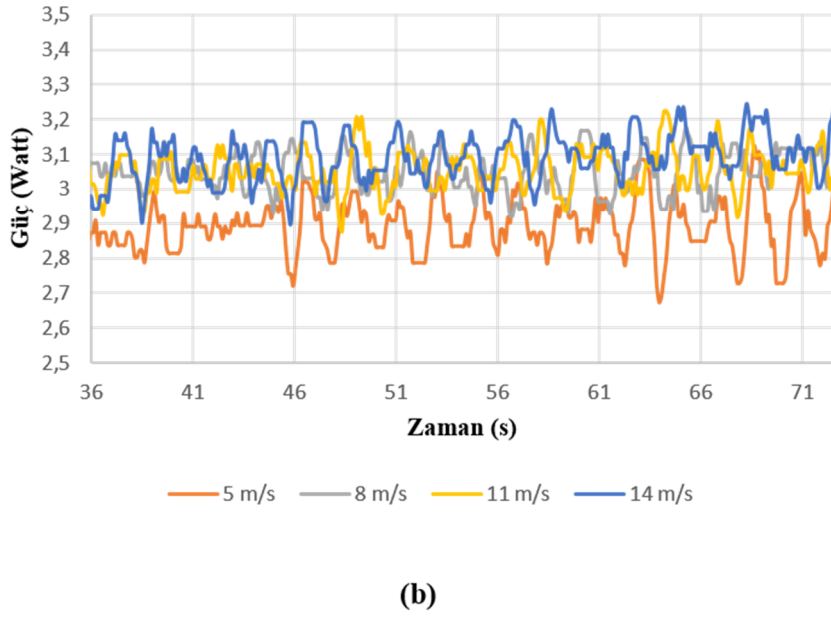
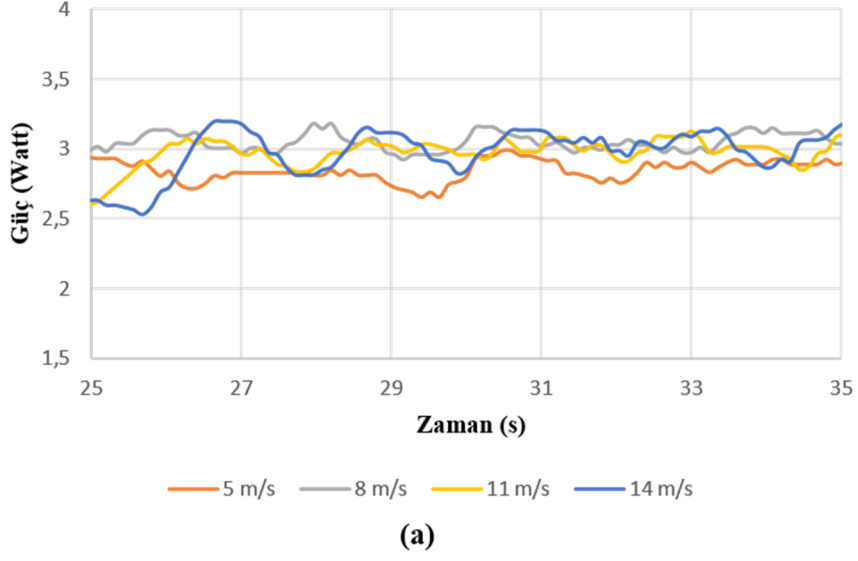
Üçüncü grafik (Şekil 3.21 (c)) ise değişik serbest akış hızlarında türbinin ne kadar net güç oluşturduğunu gözlemlemek için oluşturulmuştur. Grafik yapısı itibariyle üçüncü grafiği teyit eder niteliktedir. NACA 0018 kanat profili ve 300 RPM sabit devir hızına sahip darrieus tipi türbinin yaklaşık 7 m/s serbest akış hızından itibaren pozitif güç üretemediği gözlemlenmiştir.

Dördüncü grafik (Şekil 3.21(d)) ise üzere türbinin performans direk performans göstergesi olan C_p /kanat uç hız oranı grafiğidir. Bu grafikte C_p değerinin yükselmesi türbinin daha verimli bölgede çalıştığının göstergesidir. Bu grafikten 0,22 kanat uç hız oranı değerinde türbinin pozitif güç ürettiği görülmektedir. Yani türbindeki kanatlar gelen rüzgarla birlikte güç üreterek pozitif net güç oluşturduğu belirlenmiştir. 0,22 ve 0,28 kanat uç hız oranı arasında kalan bölgede C_p değeri negatif değere geçtiği görülmektedir. Türbinin ilgili kanat uç hız oranı değerinden itibaren güç üretmeyip aksine türbinin dönmesine direnç oluşturduğu, bu nedenle motorun daha fazla elektrik enerjisi tüketmesi gerektiği anlamına gelmektedir.

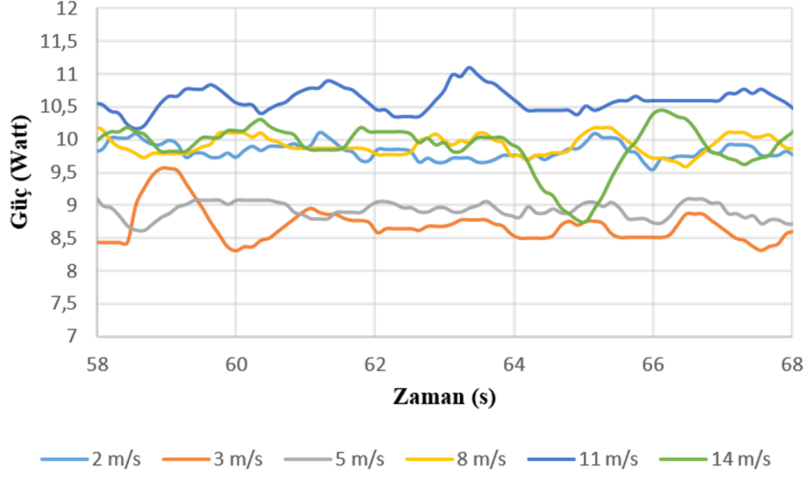
600 RPM için grafikte (Şekil 3.22 (a)) türbin, aynı çalışma ortamında farklı zamanlarda birden fazla çalıştırılıp boşta çalışma koşullarında güç tüketimi verileri elde edilmiştir. Tüm verilerin belirli bir trendi takip ettiği teyit edilmiş ve testin tekrarlanabilirliği onaylanmıştır.

Üçüncü grafikte ise (Şekil 3.22 (c)) C_p /kanat uç hız oranı grafiğinde C_p değerinin 2 kanat uç hız oranı için oldukça yüksek artış yaptığı görülmektedir. Yaklaşık 0,2-1,3 kanat uç hız oranı aralığında türbinin negatif güç ürettiği görülmektedir.

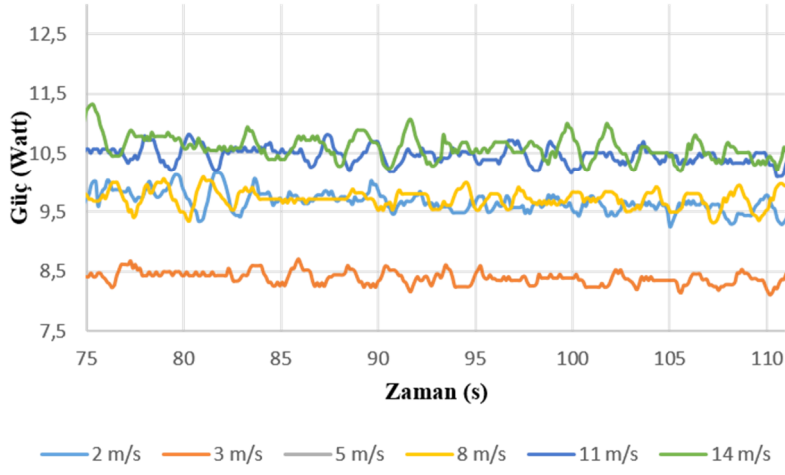
Dördüncü grafikte (Şekil 3.22 (d)) ise türbinin ürettiği net gücün yaklaşık 11 m/s hızına kadar negatif bölgede olduğu, ancak 11 m/s serbest akış hızından itibaren türbin kanatlarının güç üretmeye başladığı gözlemlenmiştir



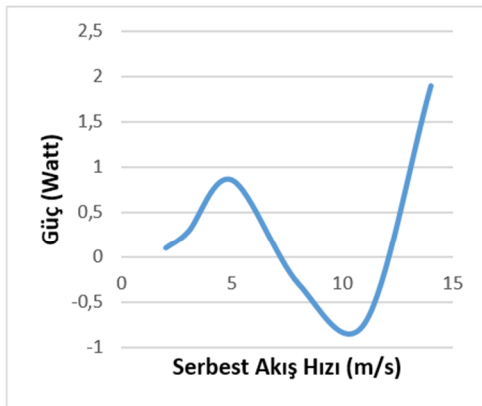
Şekil 3.22: 600 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.



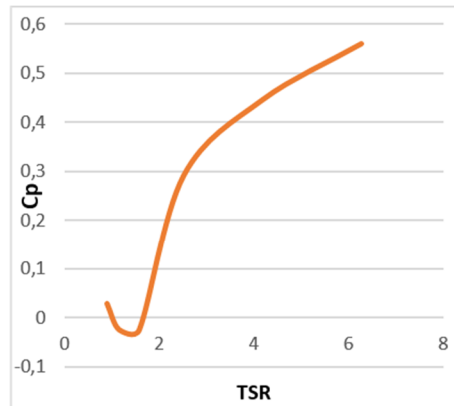
(a)



(b)



(c)



(d)

Şekil 3.23: 1200 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.

1200 RPM için grafikte (Şekil 3.23 (a)) yine türbinin boşta çalışma gücü tüketimi birbiriyle kıyaslanarak ve birden fazla ölçümün yapıldığı bölgeden seçilmiştir.

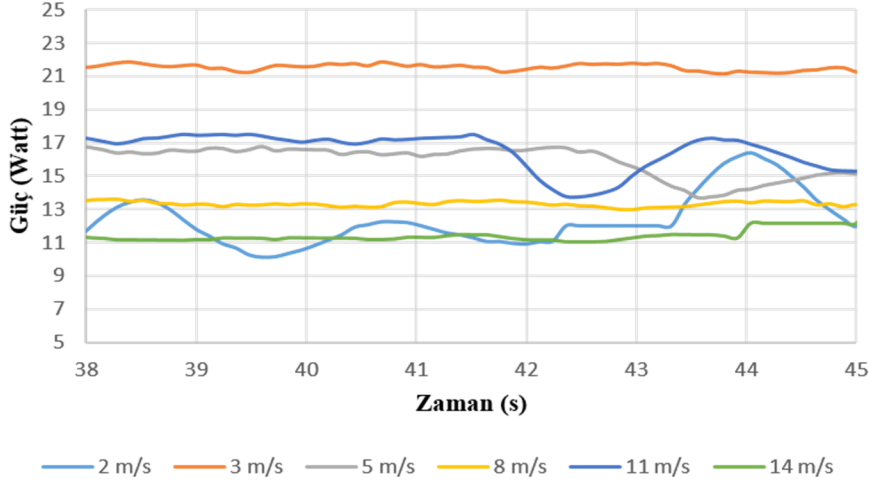
İkinci grafik (Şekil 3.23 (b)) C_p /kanat uç hız oranı grafiğinde C_p değerinin oluşturulması ve türbinin rüzgar yükü altındaki davranışını incelemek amacıyla oluşturulmuştur.

Üçüncü grafik (Şekil 3.23 (c)) C_p /kanat uç hız oranı grafiği olmakla birlikte türbinin verimli çalışma aralığının belirlenmesinde en önemli grafiklerdir. Bu grafiğe göre yüksek kanat uç hız oranlarında (Yüksek devir, düşük serbest akış hızı) türbin daha verimli davranmaktadır. Dördüncü grafikte (Şekil 3.23 (d)) ise türbinin ilgili test seti için farklı serbest akış hızlarında kanatların ürettiği net gücü göstermektedir. Bu grafiğe göre türbin, yalnızca 8 m/s ve 11 m/s'nin biraz üstündeki serbest akış hızlarında negatif güç üretmiştir.

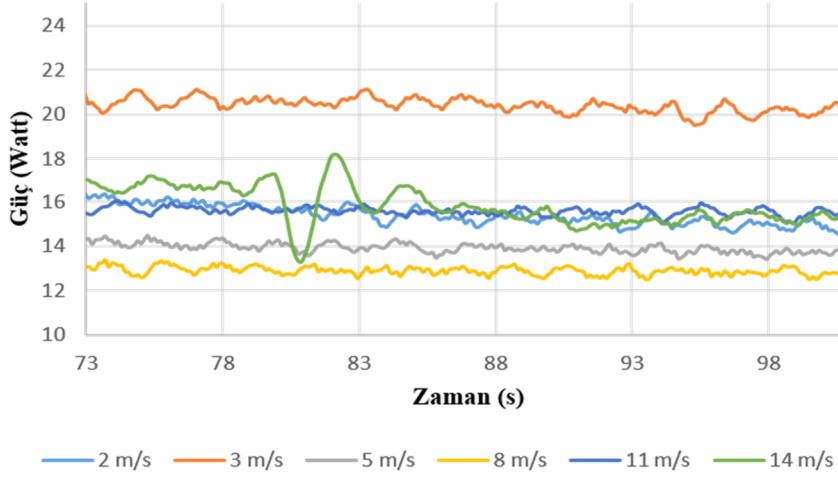
1500 RPM için (Şekil 3.24 (a)) grafikte türbinin yük altında olmadan boşta çalışma karakteristiği elde edilmiştir. Yeşil renkli veri diğerlerinden farklı bir trend izlemektedir. Türbinin boşta çalıştığı güç değeri birden fazla verinin kümelendiği bölgeden seçilmiştir.

Şekil 3.24 (b) ise türbinin değişik rüzgar yükleri altındaki cevabını görebilmek amacıyla hazırlanmıştır. Buradan gelecek her veri setinin ortalama değeri yine C_p /kanat uç hız oranı grafiğinde C_p hesaplaması için kullanılmıştır. Grafikte görüldüğü üzere 8 m/s serbest akış hızına kadar motor gücü tüketimi düşmüş ve 8 m/s akış hızı üzerindeki hızlarda tekrar yükselmiştir.

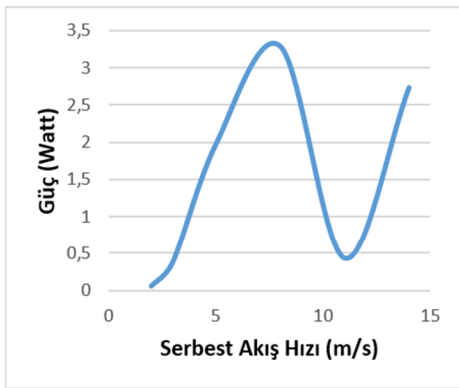
Şekil 3.24 (b) ise 1500 RPM sabit devirli türbinin verimli çalışma aralığını gösteren C_p /kanat uç hız oranı grafiğidir. Bu dönüş hızında alınan veriler hiçbir noktada negatif güç üretmemiştir. Buradan yapılan çıkarımla türbinin yüksek devirlerde daha verimli ve pozitif güç üreterek çalıştığı söylenebilir. Ayrıca 1500 RPM sabit devir için türbinin en verimli ve en çok güç üreten bölgesi 8 m/s olarak görülmektedir.



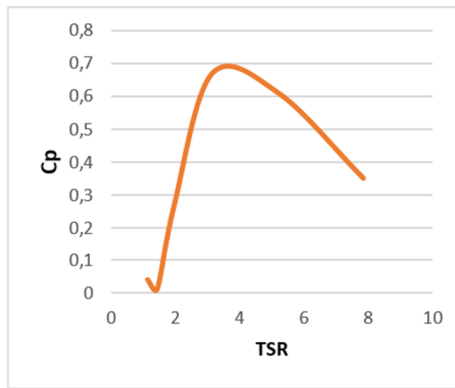
(a)



(b)



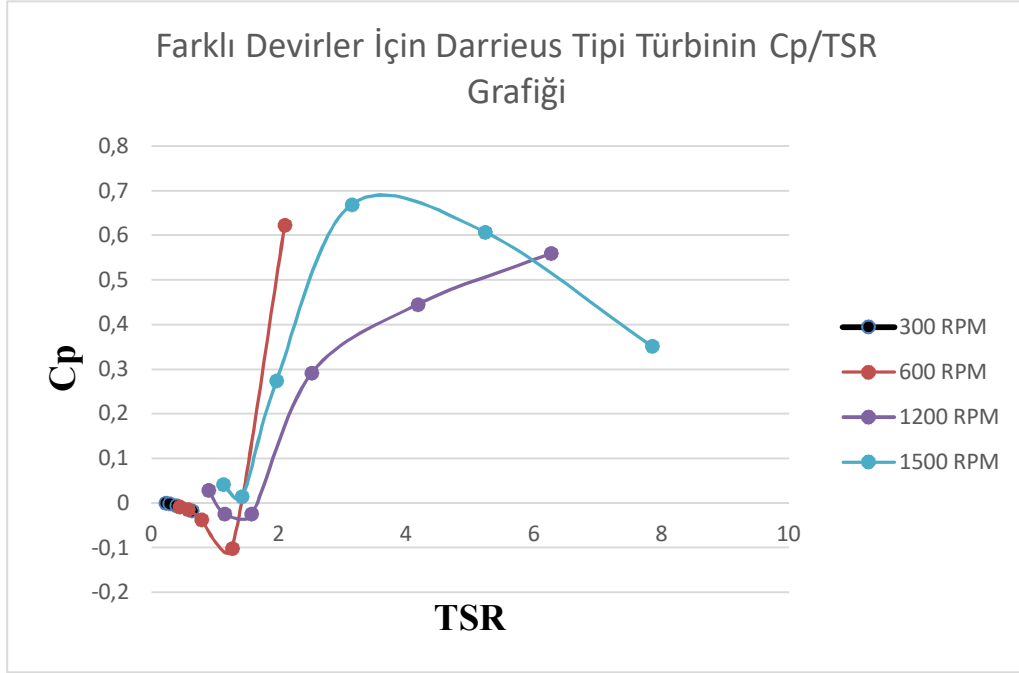
(c)



(d)

Şekil 3.24: 1500 RPM Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.

Şekil 3.24 (c) ise yük altındaki türbin kanatlarının ne kadar net güç oluşturduğu görülmektedir. Bu veri seti için hiçbir noktada negatif güç üretilmezken maksimum güç 8 m/s serbest akış hızında ölçülmüştür. Ayrıca 8 m/s hızdan 11 m/s hıza geçişte net güçte azalma, 11 m/s'den 14 m/s'ye geçişte net gücün arttığı gözlemlenmiştir. 2 cm kanat kirişi uzunluklu 1500 RPM ve 11 m/s serbest akış hızlarının oluşturduğu reynolds sayısında türbin stall bölgesine girdiği görülmektedir.

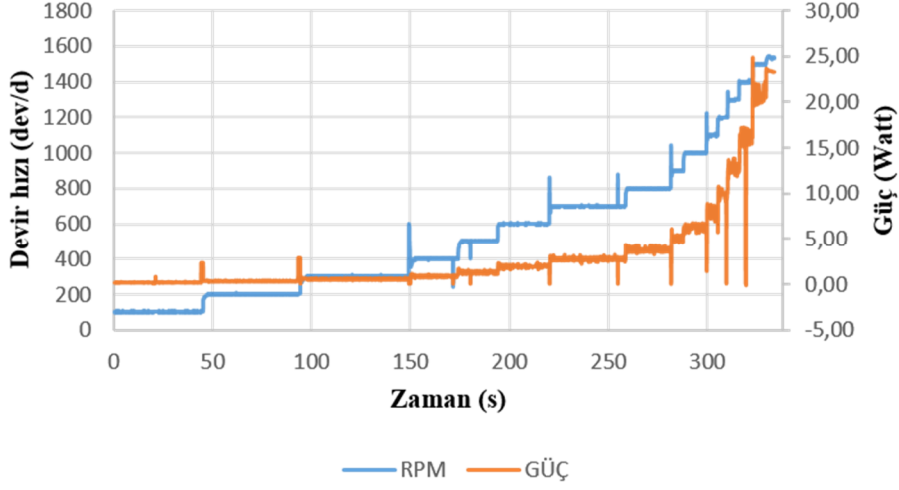


Şekil 3.25: Sabit Devirde Dönen Darrieus Tipi Türbinin Değişken Serbest Akış Hızlarındaki Grafikleri.

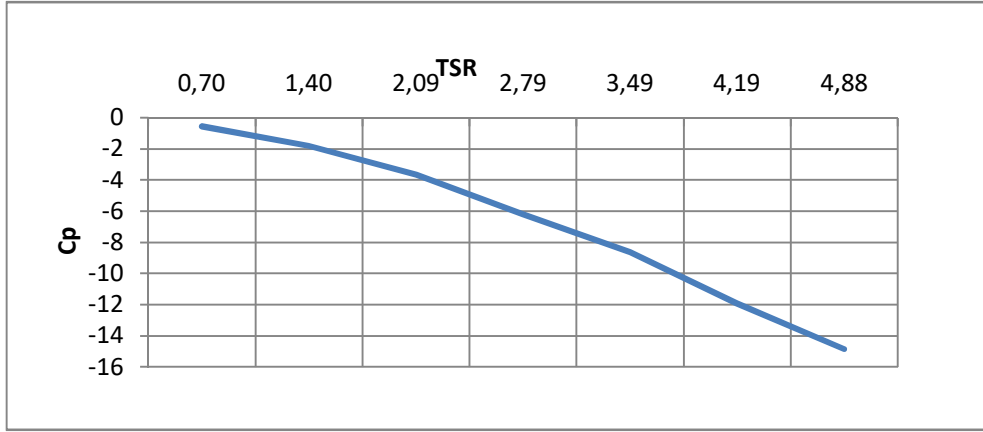
Şekil 3.25'te küçük kanat (veter 2 cm) ile PID kontrolcülü deneylerin Cp/kanat uç hız oranı grafiği birleştirilmiş olarak verilmiştir.

3.4.2 Büyük kanat türbin karakterizasyonu

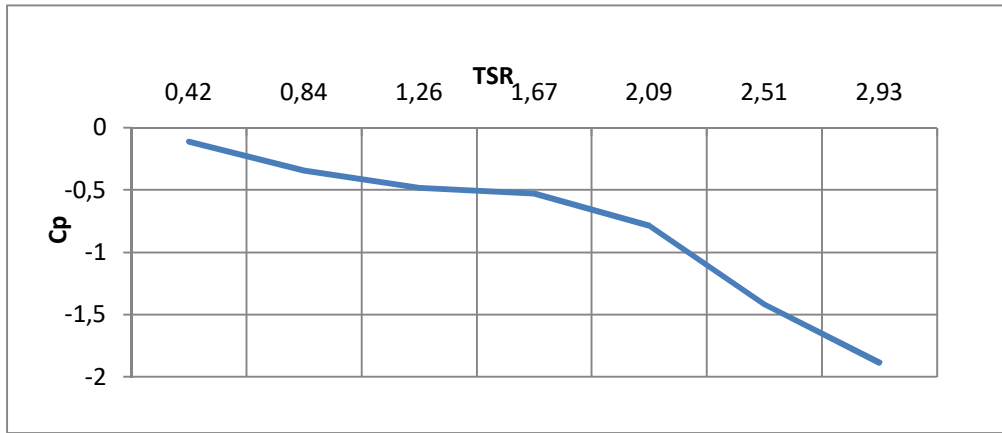
Darrieus türbinin küçük kanatlarını söküp büyük kanatların montajı yapıldıktan sonra boşta çalışma verileri alındı. Şekil 3.26'da büyük kanatlı Darrieus'un boştaki güç verileri gösterilmiştir. Bu veriler ile rüzgar hızı altındaki güç verileri karşılaştırılıp net güç verisi ulunmuş bu net güç verisi rüzgarın kinetik gücüne bölünmesiyle Cp verileri elde edilmiştir. Daha sonra Re değerleri ile Cp'lerin grafikleri elde edilmiştir.



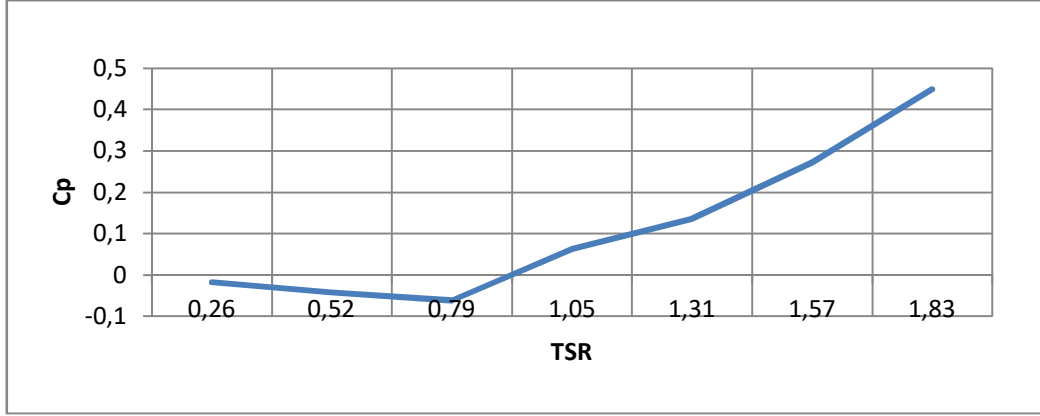
Şekil 3.26: Farklı RPM değerlerinde Darrieus Boşta Çalışma Güç Tüketimi.



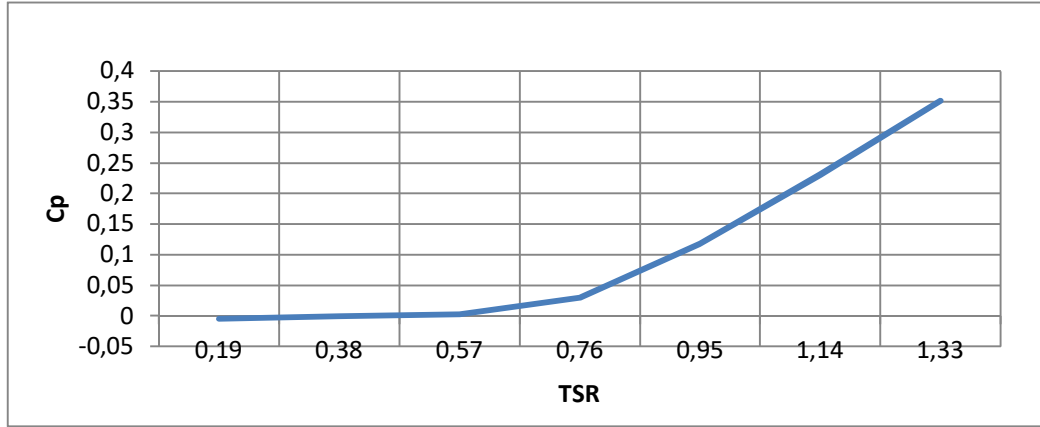
Şekil 3.27: 3 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.



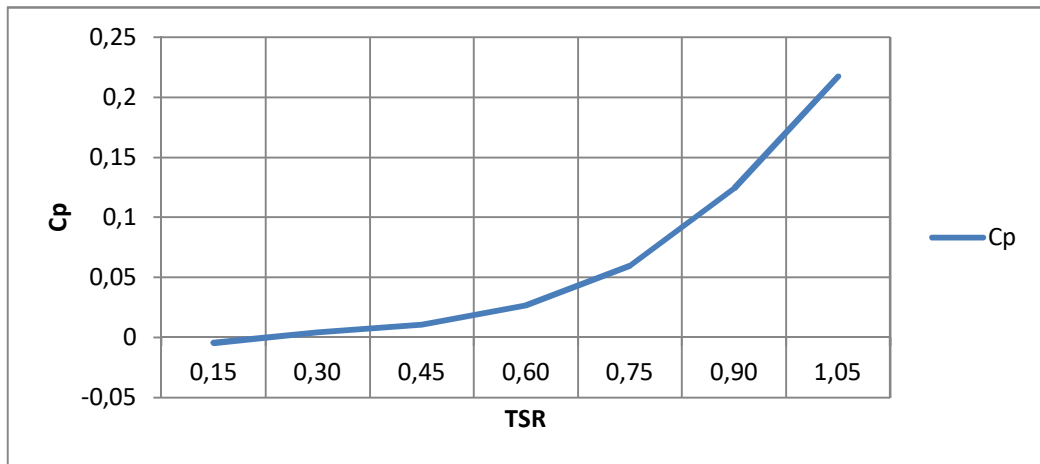
Şekil 3.28: 5 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.



Şekil 3.29: 8 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.



Şekil 3.30: 11 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.



Şekil 3.31: 14 m/s Sabit Akış Hızlarında Farklı RPM Devirde Dönen Darrieus Tipi Türbinin Grafiği.

Büyük kanat karakterizasyonu C_p /kanat uç hız oranı grafiği için PID kontrolcü ile sürülen büyük kanat darrieus türbinin boştaki çalışma güç tüketimi 100 RPM'den 1400 RPM'e kadar her RPM kademeki gücün ortalaması hesaplanmıştır. 3 m/s, 5 m/s, 8 m/s, 11 m/s ve 14 m/s rüzgar hızları için test işlemleri yapılmış her hız testindeki 100 RPM'den 1400 RPM'e kadar 200 RPM aralıkla her kademedeki rüzgar hızı altındaki gücün ortalaması alınmıştır ve net gücü bulmak için boştaki güçten rüzgar yükü altındaki hesaplanan güç çıkarılmış ve net güç bulunmuştur. Elde edilen net güç rüzgar testinde uygulanan rüzgarın kinetik gücüne bölünerek C_p değeri elde edilmiştir.

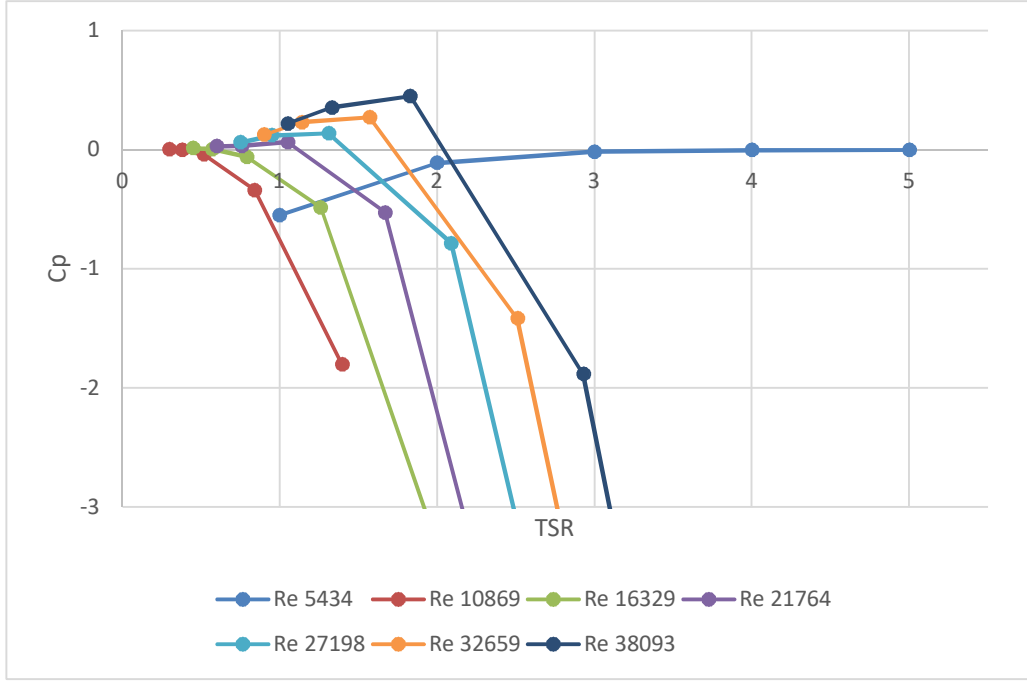
Şekil 3.27 ve Şekil 3.28'de C_p negatif yöndedir. Bu bölgelerde türbin güç üretememektedir.

Şekil 3.29 8 m/s'deki C_p 'yi göstermektedir. 8 m/s rüzgar hızında 0,79 kanat uç hız oranından sonra C_p pozitif değere geçmiş ve bu bölgeden sonra kanat uç hız oranı ile birlikte C_p 'de artmıştır. C_p maksimum değerini 0,45 olarak yaklaşık 1,8 kanat uç hız oranında ulaşmıştır. Bu bölgeden türbin güç üretmiştir.

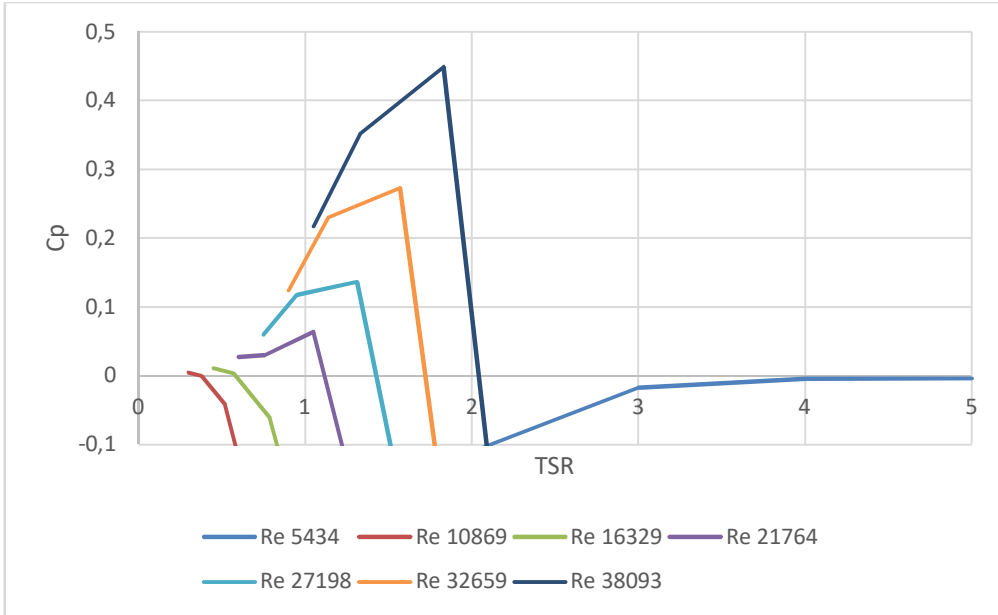
Şekil 3.30 11 m/s'deki C_p 'yi göstermektedir. Bu grafikte C_p ,0,5 kanat uç hız oranından sonra pozitif bölgeye geçmiş ve maksimum 0,35 C_p 'ye 1,3 kanat uç hız oranı değerinde ulaşmıştır. Pozitif olan bu bölgede türbin güç üretmiştir.

Şekil 3.31'de 14 m/s'deki C_p 'yi göstermektedir. Bu grafikte C_p ,0,15 kanat uç hız oranından sonra pozitif bölgeye geçmiştir. Maksimum değeri 1,02 kanat uç hız oranında 0,22 C_p olarak ölçülmüştür.

Şekil 3.32 (a)'da Re sayısına göre oluşturulan büyük kanat pıd kontrolcü darrieus Re sayısının C_p /kanat uç hız oranı gösterilmiştir. C_p 'nin pozitif bölgesini netleştirmek için (b) grafiği yapılmıştır. Bu grafiğe göre C_p 'nin pozitif bölgesine Re 38093, Re 32659, Re 27198, Re 21764 grafiklerinin geçtiği görülmüştür. Bu bölgeler için türbinin enerji üretimi yaptığı sonucuna varılmıştır.



(a)



(b)

Şekil 3.32: Büyük Kanat PID kontrolcü Darrieus Re sayısının Cp/kanat uç hız oranı verileri.

4. SONUÇ

4.1 Sonuç ve Değerlendirme

Litaratürdeki benzer çalışmalarda motor sürücülü veya akış sürücülü yöntemler kullanılmış iki durum için de kanat uç hız oranını sabit tutmak için fren mekanizması kullanılmış ayrıca sistemde torku ölçmek için tork metre kullanılmıştır. Bu çalışmada araştırmacılar fren mekanizmasının türbin üzerinde titreşime neden olduğunu ve bu titreşimin ölçüm kalitesini düşürdüğünden bahsetmişlerdir. Sistemde kullanıldığı bahsedilen tork metrenin fiyatının 1400 dolar olması yapılan bu tezin çıkış noktası olmuştur.

Rüzgar türbinlerinde deneysel tork ölçümü için kontrol algoritmalarının karşılaştırılması tezinin deneysel kısmı mekanik ve elektronik yapılarla iki bölümden oluşmaktadır. Mekanik kısım Darrieus 3 kanatlı rüzgar türbini ve kaidesi, elektronik kısım ise motor, encoder, motor sürücü, arduino mega 2560, arduino yazılımı ve visual studio ile yazılan ara yüz programıdır. Bu tezin deney kısmından sonuç alınabilmesi için bahsedilen bu iki kısmın birbiri ile uyumlu çalışması gerekmektedir. Deney aşamasına geçilmeden önce 2 farklı güçte motor ve 2 farklı tipte kanat denenmiş, türbin ölçeğine göre rüzgar hızlarına cevap veren motor seti seçilmiştir.

Deneysel ilk önce farklı kontrol algoritmaları PID ve PI karşılaştırılması yapılmış PID algoritmasının PI algoritmasına göre daha başarılı olduğu sonucuna varılmıştır. Bu sonuçtan sonra ikinci sayfaya geçilmiş ve türbin karakterizasyonu için PID kontrolcüsü ile veter uzunluğu 2 cm olan (küçük kanat) ve veter uzunluğu 4 cm olan (büyük kanat) darrieus türbin için sabit RPM değişken rüzgar hızı, sabit rüzgar hızı farklı RPM için türbini kontrol eden DC motorun RPM, akım, gerilim, tork, PWM bilgileri anlık olarak gözlenmiş bu veriler txt formatında kaydedilmiş ve bu veriler excel programı ile grafiklere dönüştürülmüşlerdir. Bu işlem sırasında arayüz ve arduino programı arasındaki senkronizasyon hatasından kaynaklanan bariz okuma hataları manuel olarak düzeltilmiştir.

Güç verimliliği (C_p) türbin gücünün rüzgar gücüne bölünmesi ile elde edilir. Akışkan ve bu akışkan içerisinde hareket eden cismin neden olduğu akışın karakteristliğini belirleyen Re sayısı ve kanat uç hız oranı verileri rüzgar türbin karakterizasyonunda kullanılan temel parametrelerdir. Deneyde alınan verilerden akım ve gerilimin çarpımı ile güç hesaplanmış, güç ve rpm değerleri ile tork hesaplanmıştır. Elde edilen bu veriler ile tezin kurgusu olan deneysel türbin testlerinde tork metre olmadan tork ölçümü yapabilme işlemi gerçekleşmiş ve ikinci aşamada C_p /kanat uç hız oranı grafikleri ile türbin karakteristikleri incelenmiştir.

Türbin karakterizasyonu için C_p /kanat uç hız oranı grafikleri önceden planlanan RPM ve rüzgar hızlarında oluşturulmuştur. C_p değerinin kanat uç hız oranına göre negatif ve pozitif noktaları elde edilmiştir fakat pozitiften negatif bölgeye geçiş noktaları kesin olarak belirlenmemiştir. İlerideki çalışmalarda geçiş bölgeleri arasında daha fazla veri alınarak grafik çözünürlüğü artırılabilir. Bir sonraki çalışma için yapılan bu tezdeki deneyler önemli bir veri niteliğindedir.

İlerideki çalışmalar için elde edilen C_p /kanat uç hız oranı grafikleri önemli bir veri olarak kullanılacağını inanmakla birlikte, sistemin kontrolcü bölümündeki visual studio ile yazılan arayüz programı, geliştirilip okuma hataları minimuma indirilebilir ayrıca mevcut sistemde arduino mega kartının işlemci hızı ve motor+encoder sistem gereksimleri nedeniyle 96 pulslü bir encoder kullanılmıştır. Kurulan sistemin daha hassas çalışması için 400 pulslü encoder ve bu veriyi işleme hızına sahip işlemci kartı sisteme entegre edilebilir.

KAYNAKLAR

- A.S. Bahaj** (2007) “Experimental verifications of numerical predictions for the hydrodynamic performance of horizontal axis marine current turbines” *Renewable Energy* 32 (2007) 2479–2490
- D. D. Chao** (2007) ”Computational Aerodynamic Analysis of a Blunt Trailing-edge Airfoil Modification to the NREL Phase VI Rotor WIND ENERGY” *Wind Energ.* 2007; 10:529–550
- Daniel B. Araya John O. Dabiri** (2015) “A comparison of wake measurements in motor-driven and flow-driven turbine experiments” DOI 10.1007/s00348-015-2022-7
- Earl P. N. Duque** (2010) “Navier-Stokes and Comprehensive Analysis Performance Predictions of the NREL Phase VI Experiment” DOI: 10.1115/1.1624088
- Ferreira** (2008) “Visualization by PIV of dynamic stall on a vertical axis wind Turbine” DOI 10.1007/s00348-008-0543-z
- Fatih Köse** (2013) “PID ve Bulanık Mantık ile DC Motorun Gerçek Zamanda STM32F407 Tabanlı Hız Kontrolü” *Otomatik Kontrol Ulusal Toplantısı, TOK2013*, 26-28 Eylül 2013, Malatya
- İsmail Coşkun** (2007)” Hiz Performans Eğrisi Kullanılarak Kazanç (PID) Parametrelerinin Belirlenmesi” ISSN 1302-6178
- Jonathan M. Edwards** (2010) “Novel Experimental Power Curve Determination and Computational Methods for the Performance Analysis of Vertical Axis Wind Turbines” DOI: 10.1115/1.4006196
- Korobenko** (2014)” Aerodynamic Simulation of Vertical-Axis Wind Turbines” DOI: 10.1115/1.4024415
- Gabriele Bedon, Marco Raciti Castelli, Ernesto Benini** (2013) ”Experimental Tests of a Vertical-Axis Wind Turbine with Twisted Blades” *World Academy of Science, Engineering and Technology* 78 2013
- O Eboibi, LAM Danao, RJ Howell** (2016) “Experimental investigation of the influence of solidity on the performance and flow field aerodynamics of vertical axis wind turbines at low Reynolds numbers” *Renewable Energy* 92 (2016) 474-483
- Q Li, T Maeda, Y Kamada, J Murata** (2016) “Study on power performance for straight-bladed VAWT by field and wind tunnel test” *Renewable Energy* 90 (2016) 291-300

- Prakash** (2014) “Tuning of PID Controller by Ziegler-Nichols Algorithm for Position Control of DC Motor” ISSN 2348 - 7968
- Robert Howell** (2009) “Wind tunnel and numerical study of a small vertical axis wind turbine” doi:10.1016/j.renene.2009.07.025
- R. Rosli** (2015) “Experimental investigations of the Hydro-Spinna turbine performance” Renewable Energy 99 (2016) 1227e1234
- Toygun Dağdevir** (2015) “Darrieus Dikey Eksenli Rüzgar Türbininin Sayısal ve Deneysel Araştırılması, Tasarlanması Ve Üretilmesi” Yüksek Lisans Tezi
- Tomonobu Senjyu** (2008) “Sensor-less maximum power point tracking control for wind generation system with squirrel cage induction generator” Renewable Energy 34 (2009) 994–999
- Tomonobu Senjyu** (2005) “Wind velocity and rotor position sensorless maximum power point tracking control for wind generation system” Renewable Energy 31 (2006) 1764–1775
- Tuyen Quang Le,** (2014) “Flow-driven rotor simulation of vertical axis tidal turbines: A comparison of helical and straight blades”
http://dx.doi.org/10.2478/IJNAOE-2013-0177
- Url-1** <<https://www.google.com.tr/=wind+turbine+farm>>, erişim tarihi 28.05.2017
- Url-2** <<https://yandex.com.tr/gorsel/searcht=betzlimitisimage>>, erişim tarihi 28.05.2017
- Url-3** <<http://www.dynetics.eu/media/1085/canondcmicromotorse.pdf>>, erişim tarihi 28.05.2017

EKLER

EK A: Arduino kontrolcü yazılım kodları.

EK B: Ara yüz yazılım kodları.

EK A: Arduino kontrolcü yazılım kodları.

Arduino'ya yazılan kod aşağıda yanlarında açıklamalarıyla verilmiştir.

```
//***** DEĞİŞKENLERİN TANIMLANDIĞI BÖLÜM
*****

#include <PID_v1.h> // "PID_v1.h" kütüphane dosyasını çağır.

int t1, t2; // t1 ve t2 periyod değerlerinin saklanacağı değişkenleri oluştur.

// 4400 -- error

// 4500 -- rpm oku

// 4600 -- akım oku

// 4700 -- pwm oku

int PWMdeger; // PC arayüz programından gönderilen bilgiyi "PWMdeger"
değişkenine kaydedeceğiz.

int j; // PC arayüz programına gönderilecek olan "akım" bilgisini "j" değişkene
kaydedeceğiz.

int s; // Anlık olarak ölçülen akım bilgisini "s" değişkene kaydedeceğiz.

int rpm = 0; // ölçülen RPM değerini ilk önce "rpm" değişkenine kaydedeceğiz.

double Input; // Ölçülen RPM değerini daha sonra "Input" değişkenine kaydedeceğiz.

double Output; // PWM sinyalindeki modülasyon değerini (0-255) bu değişkene
kaydedeceğiz.

double Setpoint; // PC arayüz programında set edilen ve Arduino Board'a gönderilen
RPM bilgisini "Setpoint" değişkenine kaydedeceğiz.

const int analogin = A2; // Arduino kartının A2 girişine "analogin" adını ver.

long count = 0; // "count" isimli sayaç değişkeni tanımla ve başlangıçta "0" (sıfır)
değerini ata.

int err; // set edilen RPM ile ölçülen RPM arasındaki farkı hata bilgisi olarak "err"
değişkenine kaydedeceğiz.
```



```

double k; // PC arayüz programında set edilen RPM değerinin "k" değişkenine
kaydedeceğiz.

double kp = 0.075; // PID hesaplamasında kullanılacak olan (P)roportional
değişkenini tanımla ve "0.075" başlangıç değerini veriyoruz.

double ki = 1.250; // PID hesaplamasında kullanılacak olan (I)ntegral geğişkenini
tanımla ve "1.25" başlangıç değerini veriyoruz.

double kd = 0.200; // PID hesaplamasında kullanılacak olan (D)erivative geğişkenini
tanımla ve "0.200" başlangıç değerini veriyoruz.

String gonderi; // PC arayüz programına gönderilecek olan paket veriyi "gonderi"
değişkenine kaydedeceğiz.

String a = "a"; // PC arayüz programına gönderilecek olan paket veride her bir veri
arasına "a" karakteri koyacağız.

PID myPID(&Input, &Output, &Setpoint, kp, ki, kd, DIRECT); // PID hesaplama
işleminde kullanılacak olan verilerin gösterildiği bölüm. (PID_v1.h kütüphanesinde
istenen veriler.)

//*****
*****

void setup() { // ***** SETUP İŞLEMLERİ BÖLÜMÜ *****

Serial.begin(9600); // Bilgisayar ile USB haberleşmeyi 9600 baud hızında başlat.

pinMode(6, OUTPUT); // Arduino Board'un 6 nolu pinini Motor sürücü kartına
gönderilen PWM çıkışı olarak tanımla.

pinMode(11, OUTPUT); // Arduino Board'un 11 nolu pinini Motor sürücü kartına
gönderilen Start-Stop çıkışı olarak tanımla.

pinMode(12, OUTPUT); // Arduino Board'un 12 nolu pinini Motor sürücü kartına
gönderilen CW-CCW yön bilgisi çıkışı olarak tanımla.

pinMode(A2, INPUT); // Arduino Board'un A2 nolu pinini Motor sürücü kartından
gelen akım değeri (0-5v arası analog bilgi) bilgisinin okunması için giriş olarak
tanımla.

pinMode(2, INPUT); // Arduino Board'un 2 nolu pinini encoderdan gelen rpm
bilgisinin okunması için giriş olarak tanımla.

```

```

myPID.SetMode(AUTOMATIC); // Sistem AUTOMATIC PID hesaplama modunda
olsun. (PID_v1.h kütüphanesinde tanımlı.)

}

//*****

void loop () // ***** ANA PROGRAM BÖLÜMÜ *****

{

t1 = millis(); // program başlangıcında ilk süreyi milisaniye cinsinden t1 değişkenine
kaydet.

Setpoint = k; // PC arayüz programında tanımlanan ve "k" değişkenine kaydedilen
"set RPM" değerini "Setpoint" değişkeni içerisine kaydet.

rpmhesapla(); // rpmhesapla alt programına git.

Input = rpm; // rpmhesapla() alt programında ölçülen "rpm" bilgisini "Input"
değişkenine kaydet.

myPID.SetTunings(kp, ki, kd); // set edilen kp, ki, ve kd değerlerini PID hesaplama
işleminde kullan. (PID_v1.h kütüphanesinde istenen veriler.)

myPID.Compute(); // PID hesaplama işlemini gerçekleştir. (PID_v1.h
kütüphanesindeki işlem sırasını uygula.)

if (Output > 256) { //PWM Modülasyonu 255 in üstüne çıkmasın diye

Output = 254; //maksimum 255 değerini almasını sağla.

}

analogWrite(6, Output); // Arduino Board'tan Motor Sürücü Kartına PWM analog
çıkışını başlat. ("Output" olarak tanımlanmış PWM modülasyon bilgisini kullan.)

err = Setpoint - Input; // set edilen RPM ile ölçülen RPM arasındaki farkı "hata"
olarak "err" değişkenine kaydet.

for (int y = 0; y < 20; y++) { // Motor Sürücü Kartından gelen analog akım değeri
bilgisini

s = s + analogRead(A2); // her 1ms lik gecikme ile 20 sefer oku ve bütün değerleri
toplayarak "s" değişkeninin içine kaydet.

```

```

delay(1); // Bu değeri daha sonra 20 ye bölerek "ortalama akım" bilgisini oluştur ve
bunu "j" değişkenine kaydet.

} // Burada amaç anlık akım bilgisini kullanmaktansa 20 adet alınan akım bilgisinin
ortalamasını kullanmaktır.

j = s / 20; //

s = 0; // Bir sonraki loop işleminde kullanmak için "s" değişkenini sıfırla.

gonderi = rpm + a + Output + a + j + a + err + a; // Bütün değerleri yanyana birleştir
(aralarına 'a' karakteri koy) ve "gonderi" string değişkenine kaydet.

Serial.println(gonderi); // ve bu bilgiyi Serial üzerinden PC arayüz programına
gönder.

t2 = millis(); // bu ana kadar geçen süreyi milisaniye cinsinden t2 değişkenine kaydet.
delay(120 - (t2 - t1)); // her bir loop döngüsünü tam 120 ms'de tamamlamak için t1
ile t2 arasındaki fark kadar süreyi

} // 120 ten çıkar ve gerçek olarak 120 ms lik toplam loop süresini sağlamış ol.

//*****

void sayma () { // ***** YAZILIMSAL OLARAK COUNTER
OLUŞTULAN ALT PROGRAM *****

count++; // bu alt program her çalıştığında "count" değişkenini 1 artır.

} // ve geldiğin yere geri dön.

//*****

void rpmhesapla() { // ***** ÖLÇÜLEN RPM DEĞERİ HESAPLAMASININ
YAPILDIĞI ALT PROGRAM *****

attachInterrupt(0, sayma, FALLING); // her bir "düşen kenar" tetikleme geldiğinde
interrupt (kesme) işlemi yap ve bu esnada "sayma" alt programını çalıştır.

delay(100); // bu işlemi gerçekleştirmesi için 100 ms kadar bekleme yap.

detachInterrupt(0); // interrupt(0) (kesme) işlemini resetle.

rpm = count * 10 * 60 / 95; // RPM değerini hesapla ve "rpm" değişkenine kaydet.

count = 0; // bir sonraki hesaplama için "count" değişkenini sıfırla.

```

```

} // ve geldiğin yere geri dön.

//void serialEvent() { //
***** PC'DEN GELEN BİLGİLERİN AYRIŞTIRILDIĞI BÖLÜM *****
if (Serial.available()) { // PC arayüz programından karta bilgi gönderilip
gönderilmediğini kontrol et.

PWMdeger = Serial.parseInt(); // Eğer gelen bilgi tamsayı ise bu sayıyı "PWMdeğer"
değişkenine kaydet.

}

if (PWMdeger > 0 && PWMdeger < 3251) // PC arayüz programından karta 1 ile
3250 arası bilgi gönderilmiş ise,

{ // bu değeri RPM değeri olarak algıla

k = PWMdeger; // ve bu değeri "k" değişkenine kaydet.

}

if (PWMdeger == 4000) // eğer bu değer 4000 ise,

{ // PC arayüz programından "Motor Aktif" bilgisi gönderilmiştir.

digitalWrite(11, HIGH); // bu durumda Motor Sürücü Kartına "Start" sinyalini
gönder.

}

if (PWMdeger == 4100) // eğer bu değer 4100 ise,

{ // PC arayüz programından "Motor Pasif" bilgisi gönderilmiştir.

digitalWrite(11, LOW); // bu durumda Motor Sürücü Kartına "Stop" sinyalini
gönder.

}

if (PWMdeger == 4200) // eğer bu değer 4200 ise,

{ // PC arayüz programından "CCW (Ters Saat Yönü)" bilgisi gönderilmiştir.

digitalWrite(12, HIGH); // bu durumda Motor Sürücü Kartına "Geri Yönde Dön"
sinyalini gönder.

}

```

```
if (PWMdeger == 4300) // eğer bu değer 4300 ise,
{
    // PC arayüz programından "CW (Saat Yönü)" bilgisi gönderilmiştir.
    digitalWrite(12, LOW); // bu durumda Motor Sürücü Kartına "İleri Yönde Dön"
    sinyalinini gönder.
}

if (PWMdeger >= 7000 && PWMdeger < 9000) { // eğer bu değer 7000 ile 8999
    arasında ise,
    kp = (PWMdeger - 7000) / 1000; // PC arayüz programında set edilen "kp" bilgisi
    tamsayıya çevrilip gönderilmiş demektir.
    // bu bilgiyi normal "kp" değerine çevir ve "kp" değişkenine kaydet.
}

if (PWMdeger >= 9000 && PWMdeger < 11000) { // eğer bu değer 9000 ile 10999
    arasında ise,
    ki = (PWMdeger - 9000) / 1000; // PC arayüz programında set edilen "ki" bilgisi
    tamsayıya çevrilip gönderilmiş demektir.
    // bu bilgiyi normal "ki" değerine çevir ve "ki" değişkenine kaydet.
}

if (PWMdeger >= 11000 && PWMdeger < 13000) { // eğer bu sayı 11000 ile 12999
    arasında ise,
    kd = (PWMdeger - 11000) / 1000; // PC arayüz programında set edilen "kd" bilgisi
    tamsayıya çevrilip gönderilmiş demektir.
    // bu bilgiyi normal "ki" değerine çevir ve "ki" değişkenine kaydet.}
```

EK B : Ara yüz programı yazılım kodları.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Threading;
using System.Text;
using System.Collections.Generic;
using System.IO;
namespace TubitakV2
{
    public partial class Form1 : Form
    {
        // Programda kullanıcak değişkenler oluşturuluyor
        int ki, kp, kd,okunanrpm,pidhatasi,o=0,m,i;
        long t;
        double rpminradian,uygulanangerilim,okunanakim,olculentork,l,
pwmokunan,guc;
        string
yol,filename1,filename2,filename3,filename4,filename5,filename6,filename7,filena
me8,filename9;
        char[] separatechars = { 'a', 'b', 'c', 'd' };
        double[] deger;
        // Aşağıdaki yöntem ile ekranda yer alan grafiklerin dosyaya
        kaydedilmesi sağlanmıştır.
        private void button6_Click(object sender, EventArgs e)
        {
            string mydoCpath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
            this.torkchart.SaveImage(mydoCpath +
"tork.png",System.Windows.Forms.DataVisualization.Charting.ChartImageF
ormat.Png);
            this.rpmchart.SaveImage(mydoCpath +
"rpm.png",System.Windows.Forms.DataVisualization.Charting.ChartImageF
ormat.Png);
            this.pwmchart.SaveImage(mydoCpath + "pwm.png",
System.Windows.Forms.DataVisualization.Charting.ChartImageFormat.Png)
;
            this.akimchart.SaveImage(mydoCpath + "AKIM.png",
System.Windows.Forms.DataVisualization.Charting.ChartImageFormat.Png)
;
        }
    }
}
```

```

        this.errorchart.SaveImage(mydoCpath + "Hata.png",
System.Windows.Forms.DataVisualization.Charting.ChartImageFormat.Png)
;
}
double a1, a2, a3, a4;
string text;
public Form1()
{
    InitializeComponent();
}
// Program ilk açıldığında yapılacaklar aşağıdaki yöntem içinde
verilmiştir.
private void Form1_Load(object sender, EventArgs e)
{
    string[] ports = SerialPort.GetPortNames();
    foreach (string port in ports)
    {
        comboBox1.Items.Add(port); // Port isimlerini combobox1'de
gösteriyoruz.
        comboBox1.SelectedIndex = 0;
    }
    comboBox2.Items.Add("2400"); // Baudrate'leri kendimiz combobox2'ye
giriyoruz.
    comboBox2.Items.Add("4800");
    comboBox2.Items.Add("9600");
    comboBox2.Items.Add("19200");
    comboBox2.Items.Add("115200");
    comboBox2.SelectedIndex = 2;
    label3.Text = "Bağlantı Ayarlarını Yapınız!";
    numericUpDown1.Value = Convert.ToDecimal( 0.075);
    numericUpDown2.Value = Convert.ToDecimal(1.25);
    numericUpDown3.Value = Convert.ToDecimal(0.001);
}
// Komutları Arduino'ya göndermek için kullanılan fonksiyon hazırlanıyor.
private void komutgonder(int komutnumara)
{
    try
    {
        serialPort1.WriteLine(Convert.ToString(komutnumara));
    }
    catch
    {
        MessageBox.Show("Seri Port Ayarları Doğru Yapılmadı ya da Seri
Port Meşgul!");
    }
}
// trackbar kaydırıldığında Arduino'ya gönderilecek veri hazırlanıyor.
private void trackBar1_Scroll(object sender, EventArgs e)
{
    komutgonder(trackBar1.Value);
}

```

```

        radioButton5.Checked = false;
        radioButton6.Checked = false;
        radioButton7.Checked = false;
        radioButton8.Checked = false;
        label9.Text = "RPM: " + Convert.ToString(trackBar1.Value);
    }

    // Çeşitli radiobuttonların sistemde hangi işi yapacağı tanımlanıyor.
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton5.Checked == true)
    {
        komutgonder(250);
        trackBar1.Value = 250;
        label9.Text = "RPM: " + Convert.ToString(trackBar1.Value);
    }
}
private void radioButton6_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton6.Checked == true)
    {
        komutgonder(750);
        trackBar1.Value = 750;
        label9.Text = "RPM: " + Convert.ToString(trackBar1.Value);
    }
}
private void radioButton7_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton7.Checked == true)
    {
        komutgonder(1500);
        trackBar1.Value = 1500;
        label9.Text = "RPM: " + Convert.ToString(trackBar1.Value);
    }
}
private void radioButton8_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton8.Checked == true)
    {
        komutgonder(2500);
        trackBar1.Value = 2500;
        label9.Text = "RPM: " + Convert.ToString(trackBar1.Value);
    }
}

    // Ekranı temizleme butonu ayarlanıyor.
private void button5_Click(object sender, EventArgs e)
{
    torkchart.Series["Tork"].Points.Clear();
    rpmchart.Series["Okunan RPM"].Points.Clear();
    pwmchart.Series["Uygulanan PWM"].Points.Clear();
    akimchart.Series["Okunan Akım"].Points.Clear();
}

```



```

        errorchart.Series["Hata"].Points.Clear();
        t = 0;
    }
    // Program kapanırken neler yapılacağı açıklanıyor.
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    timer1.Stop();
    komutgonder(4100);
    serialPort1.Close();
}
    // Elde edilen verilerin kaydeilmesi işlemi aşağıdaki fonksiyon ile
    yapılıyor.
private void button3_Click(object sender, EventArgs e)
{
    string mydoCpath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments)
;
    filename1 = textBox1.Text;
    filename2 = textBox2.Text;
    filename3 = textBox3.Text;
    filename4 = textBox4.Text;
    filename5 = textBox5.Text;
    filename6 = textBox6.Text;
    filename7 = textBox7.Text;
    filename8 = textBox8.Text;
    filename9 = textBox9.Text;
    using (StreamWriter outfile = new StreamWriter(mydoCpath +
@filename1 + ".txt"))
    {
        for (i = 0; i < listBox1.Items.Count; i++)
        {
            outfile.WriteLine(listBox1.Items[i]);
        }
    }
    using (StreamWriter outfile = new StreamWriter(mydoCpath +
@filename2 + ".txt"))
    {
        for (i = 0; i < listBox2.Items.Count; i++)
        {
            outfile.WriteLine(listBox2.Items[i]);
        }
    }
    using (StreamWriter outfile = new StreamWriter(mydoCpath + @filename3 +
".txt"))
    {
        for (i = 0; i < listBox3.Items.Count; i++)
        {
            outfile.WriteLine(listBox3.Items[i]);
        }
    }
}

```

```

        using (StreamWriter outfile = new StreamWriter(mydoCpath + @filename4 +
".txt"))
        {
            for (i = 0; i < listBox4.Items.Count; i++)
            {
                outfile.WriteLine(listBox4.Items[i]);
            }
        }
        using (StreamWriter outfile = new StreamWriter(mydoCpath + @filename5 +
".txt"))
        {
            for (i = 0; i < listBox5.Items.Count; i++)
            {
                outfile.WriteLine(listBox5.Items[i]);
            }
        }
        using (StreamWriter outfile = new StreamWriter(mydoCpath +
@filename6+".txt"))
        {
            for (i = 0; i < listBox6.Items.Count; i++)
            {
                outfile.WriteLine(listBox6.Items[i]);
            }
        }
        using (StreamWriter outfile = new StreamWriter(mydoCpath + @filename7 +
".txt"))
        {
            for (i = 0; i < listBox7.Items.Count; i++)
            {
                outfile.WriteLine(listBox7.Items[i]);
            }
        }
        using (StreamWriter outfile = new StreamWriter(mydoCpath + @filename8 +
".txt"))
        {
            for (i = 0; i < listBox8.Items.Count; i++)
            {
                outfile.WriteLine(listBox8.Items[i]);
            }
        }
        using (StreamWriter outfile = new StreamWriter(mydoCpath + @filename9 +
".txt"))
        {
            for (i = 0; i < listBox9.Items.Count; i++)
            {
                outfile.WriteLine(listBox9.Items[i]);
            }
        }
    }
}
private void radioButton1_CheckedChanged(object sender, EventArgs e)

```

```

{
    if (radioButton1.Checked == true)
    {
        radioButton2.Checked = false;
        komutgonder(4000);
        timer1.Start();
    }
}private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked)
    {
        if (timer1.Enabled)
        {
            radioButton1.Checked = false;
            timer1.Stop();
            for (int i = trackBar1.Value; i > 1; i--)
            {
                trackBar1.Value = i-1;
                komutgonder(i);
                Thread.Sleep(1);
                label9.Text = "RPM: " +
Convert.ToString(trackBar1.Value);
            }
            komutgonder(4100);
        }
    }
}
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton3.Checked == true)
    {
        if (timer1.Enabled == true)
        {
            komutgonder(4200);
        }
    }
}
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton4.Checked == true)
    {
        komutgonder(4300);
    }
}
// Seri porta bağlanma butonu oluşturuluyor.
private void button1_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == false)
    {

```

```

serialPort1.PortName = comboBox1.Text;
serialPort1.BaudRate = Convert.ToInt32(comboBox2.Text);
serialPort1.Open();
    radioButton2.Checked = true;
    komutgonder(4100);
    radioButton4.Checked = true;
    komutgonder(4300);
    if (serialPort1.IsOpen == true)
    {
        label3.Text = "Bağlandı";
    }
    else
    {
        label3.Text = "Hata!";
    }
}
else
{
    label3.Text = "Seri Port Zaten Bağlı, Ayarları Değiştirmek İçin
Bağlantıyı Kesiniz!";
}
}
// Her bir zaman darbesinde periyodik olarak yapılan işlemler aşağıda
tanımlanıyor.
private void timer1_Tick(object sender, EventArgs e)
{
    try
    {
        //if (o < 1) { Thread.Sleep(155); }
        //o++;
        if (serialPort1.IsOpen == true)
        {
            lblayarlananrpm.Text =
Convert.ToString(trackBar1.Value);
            t = t + 1;
            if (serialPort1.BytesToRead > 0)
            {
                text = serialPort1.ReadExisting();
                string[] words = text.Split(separatchars);
                foreach (string s in words)
                {
                    m++;
                    if (m == 1) { a1 = Convert.ToDouble(s); }
                    if (m == 2) { a2 = Convert.ToDouble(s); }
                    if (m == 3) { a3 = Convert.ToDouble(s); }
                }
                if (m == 4) { a4 = Convert.ToDouble(s); }
            }
        }
        m = 0;
    }
}

```

```

    }
    okunanrpm = Convert.ToInt32(a1);
    lblokunanrpm.Text = Convert.ToString(okunanrpm);
    listBox1.Items.Add(okunanrpm);
    rpminradian = okunanrpm * 2 * Math.PI / 60;
    lblacisalhiz.Text = Convert.ToString(rpminradian);
    listBox2.Items.Add(rpminradian);
    pwmokunan = Convert.ToDouble(a2); // Çeşitli
hesaplama işlemleri yapılıyor.
    pwmokunan = pwmokunan / 100;
    lbluygulananpwm.Text =
Convert.ToString(pwmokunan);
    listBox3.Items.Add(pwmokunan);
    uygulanangerilim = 24 * pwmokunan / 255;
    lbluygulanangerilim.Text =
Convert.ToString(uygulanangerilim);
    listBox7.Items.Add(uygulanangerilim);
    okunanakim = Convert.ToDouble(a3);
    okunanakim = 5 * okunanakim / 1023 * 0.801 +
0.0045;
    lblolculenakim.Text = Convert.ToString(okunanakim);
    listBox4.Items.Add(okunanakim);
    guc = okunanakim * uygulanangerilim;
    listBox8.Items.Add(guc);
    olculentork = uygulanangerilim * okunanakim /
rpminradian;
    lbltork.Text = Convert.ToString(olculentork);
    listBox5.Items.Add(olculentork);
    pidhatasi = Convert.ToInt32(a4);
    lblpidhata.Text = Convert.ToString(pidhatasi);
    listBox6.Items.Add(pidhatasi);
    listBox9.Items.Add(t);
    try
    {
        if (olculentork <= 100 && okunanrpm <=
10000 && pwmokunan < 300 && okunanakim < 20 && pidhatasi < 10000)
        {
            this.torkchart.Series["Tork"].Points.AddXY(t,
olculentork);
            this.rpmchart.Series["Okunan RPM"].Points.AddXY(t, okunanrpm);
            this.pwmchart.Series["Uygulanan PWM"].Points.AddXY(t, pwmokunan);
            this.akimchart.Series["Okunan Akım"].Points.AddXY(t, okunanakim);
            this.errorchart.Series["Hata"].Points.AddXY(t,
pidhatasi);
        }
    }
    catch
    {
        Thread.Sleep(1);
    }
}

```

```

    }
}
catch
{
    MessageBox.Show("Hata");
    Thread.Sleep(1);
}
}

// Seri port bağlantısını kesmek için kullanılan fonksiyon.
private void button2_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen == true)
    {
        serialPort1.Close();
        label3.Text = "Seri Port Kapandı";
    }
    else
    {
        label3.Text = "Seri Port Zaten Kapalı";
    }
}

// PID parametrelerini gönderme butonu ayarlanıyor.
private void button4_Click(object sender, EventArgs e)
{
    kp = Convert.ToInt32( numericUpDown1.Value * 1000 + 7000);
    ki = Convert.ToInt32(numericUpDown2.Value * 1000 + 9000);
    kd = Convert.ToInt32(numericUpDown3.Value * 1000 + 11000);
    komutgonder(kp);
    Thread.Sleep(1);
    komutgonder(ki);
    Thread.Sleep(1);
    komutgonder(kd);
}
double bilgioku()
{
    if (serialPort1.IsOpen == true)
    {
        try
        {
            l = Convert.ToDouble(serialPort1.ReadLine());
        }
        catch
        {
            l = 0;
        }
    }
}
return l;
}
}
}
}

```

ÖZGEÇMİŞ



Ad-Soyad : Haydar ÜLKER
Doğum Tarihi ve Yeri : Aydın- 08.03.1982
E-posta : haydar_ulker@hotmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2007, Anadolu Üniversitesi, İktisat Fakültesi

MESLEKİ DENEYİM VE ÖDÜLLER:

-