

**GÜNEŞ PİLLERİNDE MAKSİMUM GÜÇ NOKTASI
TAKİP SİSTEMLERİ**

Özer YÜMSEK

**YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ŞUBAT 2010
ANKARA**

Özer YÜMSEK tarafından hazırlanan “Güneş pillerinde maksimum güç noktası takip sistemleri” adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Müzeyyen SARITAŞ

Tez Danışmanı, Elektrik Elektronik Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Elektrik Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Y.Doç.Dr. M.Timur AYDEMİR

Elektrik Elektronik Mühendisliği AD., G.Ü.

Prof. Dr. Müzeyyen SARITAŞ

Elektrik Elektronik Mühendisliği AD., G.Ü.

Prof. Dr. Nihal Fatma GÜLER

Elektronik Bilgisayar AD.,G.Ü.

Tarih: 18 / 02 / 2010

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Bilal TOKLU

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Özer YÜMSEK

**GÜNEŞ PİLLERİNDE MAKSİMUM GÜÇ NOKTASI
TAKİP SİSTEMLERİ
(Yüksek Lisans Tezi)**

ÖZER YÜMSEK

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Şubat 2010**

ÖZET

Alternatif enerji kaynaklarının en önemlilerinden birisi güneş enerjisidir. Son zamanlardaki bilimsel çalışmalarla, güneş enerjisinin verimli kullanımında büyük aşamalar kaydedilmiştir. Ancak her bilimsel konuda olduğu gibi bu konu ile ilgili de çözülmesi gereken bazı problemler vardır. Bu problemlerden birisi güneş pilinden elde edilen gücün kullanımının maksimize edilmesi boyutundadır.

Bu çalışmada, güneş pilinde maksimum güç noktasını yakalamak için üç farklı yöntem kullanılmıştır. Bunlar sırasıyla, güneş pilini yüke direk bağlama, kaygılandır&gözetle ve oransal açık devre voltajı metotlarıdır.

Farklı ışıınım ve sıcaklık değerleri için yukarıdaki üç yöntem karşılaştırılmış ve maksimum güç noktası yakalama’da en iyi yöntemin kaygılandır&gözetle yöntemi olduğu görülmüştür.

Simülasyonda, Java dilinde program kullanılmış ve Eclipse IDE yazılım geliştirme ortamı seçilmiştir.

Bilim Kodu : 905.1.035

Anahtar kelimeler : PV, Fotovoltaik, Maksimum güç noktası izleyicisi, Maksimum Güç Noktası, Güneş Pili, DA/DA Dönüştürücü, java.

Sayfa adedi : 144

Tez Yöneticisi : Prof. Dr. Müzeyyen SARITAŞ

**MAXIMUM POWER POINT TRACKING SYSTEMS
IN SOLAR CELLS
(M. Sc. Thesis)**

ÖZER YÜMSEK

**GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
February 2010**

ABSTRACT

The solar energy is one of the most important alternative energy sources. The recent scientific studies has shown that significant progress has been done on the efficient use of the solar energy. However, as in all scientific matters related to this issue, there are still some problems to be solved. One of these problems is to maximize the use of power obtained from the solar cell.

In this study, three different methods were used in order to track the maximum power point of the solar cell. These are, connecting the solar cell directly to the load, perturbing&observing and fractional open circuit voltage, respectively.

These three methods were compared using different temperature and radiation levels and it was observed that perturb&observe is the best method to track the maximum power point.

Java programming language were used in the simulation program and Eclipse IDE was choosen as the development environment.

Scientific Code : 905.1.035

**Key Words : PV, Photovoltaic, Maximum power point tracking,
Maximum Power Point, Solar Cell, DC/DC Converter,
java**

Page Numbers : 144

Adviser : Prof. Dr. Muzeyyen SARITAS

TEŞEKKÜR

Bu tezin hazırlanmasında değerli yardım ve katkılarıyla beni yönlendiren Değerli Hocam Prof. Dr. Müzeyyen SARITAŞ' a, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme ve eşim Züleyha Yıldız YÜMSEK'e, tezin yazılmasındaki katkıları nedeniyle arkadaşım Mesut Cemil İŞLER'e ve mesai arkadaşlarım Salih BOZKURT, Derya ÖZALTAN, Mehmet YAZICI, Birol YILDIRIM'a buradan teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ.....	xiii
ŞEKİLLERİN LİSTESİ.....	xiv
SİMGELER VE KISALTMALAR	xvii
1. GİRİŞ.....	1
2. GÜNEŞ PİLİ KARAKTERİSTİĞİ	7
2.1. Güneş Pili Eşdeğer Devresi.....	7
2.2. Güneş Pili Karakteristiği	8
2.3. Güneş Pili Üretim Teknolojisi.....	10
2.3.1. Kristal silisyum güneş pilleri	10
2.3.2. Ribbon silisyum güneş pilleri.....	10
2.3.3. Polikristal silisyum güneş pilleri.....	11
2.3.4. İnce film güneş piller	11
2.3.5. Amorf silisyum güneş pilleri	11

	Sayfa
2.3.6. Diğer yapılar	11
2.4. Işınım ve Sıcaklığın Etkisi.....	12
2.5. İstenen Güç ile Üretilen Güç Arasındaki Uyumsuzluğun Giderilmesi	15
3. DÖNÜŞTÜRÜCÜ KARAKTERİSTİĞİ VE YÜK EŞLEŞTİRME	20
3.1. Boost Topolojisi	20
3.1.1. Topolojinin genel yapısı	20
3.1.2. Çıkış gerilimi değeri	22
3.1.3. Kapasite ve endüktans değerleri.....	22
3.1.4. Yük direnci değeri	23
3.2. Buck topolojisi	24
3.2.1. Topolojinin genel yapısı	24
3.2.2. Çıkış gerilimi değeri	25
3.2.3. Kapasite ve endüktans değerleri.....	26
3.2.4. Yük direnci değeri	26
3.3. Cuk ve SEPIC Dönüştürücüler	26
3.3.1. Topolojilerin genel yapısı	26
3.3.2. Çıkış gerilimi değeri	28
3.4. Buck – boost dönüştürücü karşılaştırılması	28

	Sayfa
3.5.Yük Eşleştirme	30
4. MGNI TEKNİKLERİ	32
4.1. Giriş	32
4.2. Hill Climbing ve Kaygılandır & Gözetle (K&G)[33]	32
4.3. Artan İletkenlik.....	36
4.4. Sabit Gerilim Metodu – Oransal Açık Devre Gerilimi	39
4.5. Oransal Kısa Devre Akımı Metodu	42
4.6. Bulanık Mantık Kontrol	43
5. SİMÜLASYON YÖNTEMİ VE SONUÇLARI	47
5.1. Simülasyon Ortamı	47
5.2. Geliştirilen Program.....	48
5.3. Karakteristik.....	49
5.3.1. Çalışma noktası yakala.....	49
5.3.2. Karakteristik bilgileri	51
5.4. Maksimum güç yakala	55
5.4.1. Dönüştürücünün belirlenmesi	56
5.4.2. Kullanılabilecek Yöntemler.....	59

	Sayfa
5.4.3. Simülasyonun Çalışması	59
5.4.4. Grafiklerin oluşturulması	68
6. SONUÇ VE ÖNERİLER.....	69
KAYNAKLAR	71
EKLER	75
EK – 1 Load.java kodu	76
EK – 2 Converter.java kodu	78
EK – 3 PvConstant.java kodu	80
EK – 4 PvVariable.java kodu	83
EK – 5 Calc.java kodu	86
EK – 6 ReadFile.java kodu	89
EK – 7 WriteFile.java kodu	91
EK – 8 Graph2.java kodu	93
EK – 9 Graph4.java kodu	96
EK – 10 Menu.java kodu	101
ÖZGEÇMİŞ	144

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 1.1. Ülke Grupları bazında dünya enerji tüketimi (EJ).....	1
Çizelge 1.2. Dünya, OECD ve Türkiye enerji göstergelerinin karşılaştırması [10].....	2
Çizelge 1.3. OECD ve AB ülkelerinde elektrik kurulu gücü (GW - 2005 yılı)[14].....	2
Çizelge 1.4. Türkiye'nin Aylık Ortalama Güneş Enerjisi Potansiyeli(EİE Genel Müdürlüğü).....	5
Çizelge 1.5. Türkiye'nin Yıllık Toplam Güneş Enerjisi Potansiyelinin Bölgelere Göre Dağılımı	6
Çizelge 4.1. Hill Climbing ve K&G metotları	33
Çizelge 4.2. Bulanık kural tabanlı tablo.....	45
Çizelge 5.1. Buck ve boost dönüştürü ile görülen direnç değeri.....	58
Çizelge 5.2. Fazlara göre yöntemlerde elde edilen güçler	67

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. 1990-2003 yılları arasında enerji üretiminin tüketimi karşılama oranı.....	3
Şekil 2.1. Güneş pili eşdeğer devresi.....	7
Şekil 2.2. Güneş pili I-V ve P-V karakteristiği	9
Şekil 2.3. Işınımın I-V ve P-V eğrisine etkisi	13
Şekil 2.4. Hücre sıcaklığının çeşitli parametreler üzerindeki etkisi [7]	14
Şekil 2.5. Sıcaklığın 60 W'lık panel üzerindeki etkisi [6].....	15
Şekil 2.6. PV modülün değişken direnç ile bağlanması.....	16
Şekil 2.7. I-V eğrisinin değişik direnç değerlerinde simülasyonu.....	17
Şekil 2.8. Güneş pili uyumsuzluk karakteristiği [6]	18
Şekil 3.1. Boost topolojisi[18]	20
Şekil 3.2. Boost topolojisi dalga formları	21
Şekil 3.3. Buck topolojisi [5]	24
Şekil 3.4. Buck topolojisi dalga formları	25
Şekil 3.5. Cuk topolojisi.....	27
Şekil 3.6. SEPIC topolojisi	27
Şekil 3.7. Topoloji verimlerinin karşılaştırılması	29
Şekil 3.8. PV tarafından görülen empedans.....	31

Şekil	Sayfa
Şekil 4.1. P&O algoritması akış diyagramı	33
Şekil 4.2. Hill climbing ve K&G MGN'den uzaklaşma hatası	34
Şekil 4.3. K&G Blok Diyagramı	35
Şekil 4.4. DK&G akış diyagramı	36
Şekil 4.5. İki model MGNI (ICb) blok diyagramı	38
Şekil 4.6. Artan iletkenlik akış diyagramı	39
Şekil 4.7. Artan ışıınım altında maksimum güç noktasının konumu	40
Şekil 4.8. Sabit gerilim algoritması blok diyagramı [2]	42
Şekil 4.9. Oransal Kısa Devre Akımı Metodu Blok Diyagramı [2]	43
Şekil 4.10. Bulanık mantık kontrolcü giriş ve çıkış için üyelik fonksiyonu	44
Şekil 5.1. Çalışma noktası yakala giriş ekranı	50
Şekil 5.2. Çalışma noktası yakala sonuç ekranı	51
Şekil 5.3. Karakteristik bilgileri giriş ekranı	52
Şekil 5.4. Karakteristik bilgileri sonuç ekranı	53
Şekil 5.5. Akım gerilim ve güç gerilim grafiği	54
Şekil 5.6. Maksimum güç yakala giriş ekranı	56
Şekil 5.7. Adıma göre kullanılan ışıınım ve sıcaklık değerleri	60
Şekil 5.8. MGNI K&G metodu sonuçları (Yük direnci= 6Ω)	63

Şekil	Sayfa
Şekil 5.9. MGNI Oransal Voc yöntemi sonuçları (Orantı sabiti=0,76 ve yük direnci = 6Ω)	64
Şekil 5.10. Yüke direk bağlı sistem grafikleri (Yük direnci = 6Ω)	66
Şekil 5.11. Direk bağlı sistemde yük-güç grafiği.....	68

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
dk	Dakika
D	Doluluk Oranı
EJ	Exa Joule
f	Frekans
I	Akım
J	Joule
K	Boltzman Sabiti($1,38 \times 10^{-23}$ Joule / ° K)
kJ	Kilo Joule
kW	KiloWatt
kWh	Kilowatt saat
m²	metrekare
MVA	Mega Voltamper
MW	Megawatt
P	Güç
q	Elektron yükü ($1,6 \times 10^{-19}$ Coulomb)
R	Direnç
s	Saniye
T	Sıcaklık
t	Süre
TEP	Ton Eşdeğer Petrol
TWh	Terrawatt saat
V	Voltaj
W	Watt
°C	Santigrat

°F	Fahrenheit
π	pi sayısı (3,14)
Ω	Ohm

Kısaltmalar**Açıklama**

DİE	Devlet İstatistik Enstitüsü
LPG	Sıvılaştırılmış Petrol Gazı
MGN	Maksimum güç noktası
MGNI	Maksimum güç noktası izleyici
OECD	İktisadi Kalkınma ve İşbirliği Örgütü
PV	Fotovoltaik pil
UPA	Uygulama programı arabirimi

1. GİRİŞ

2001 yılında küresel anlamda enerji tüketimi 426,1 EJ olarak hesaplanmıştır (Çizelge 1.1). 2025’de bunun %58’lik bir artışla, 675,3 EJ dolaylarında olması beklenmektedir [10].

Çizelge 1.1. Ülke Grupları bazında dünya enerji tüketimi (EJ)

BÖLGE	1990	2001	2010	2025
Endüstrileşmiş ülkeler	192,8	223,1	253,3	304,1
Doğu Avrupa – Eski Sovyet Cumhuriyetleri	80,5	56,2	73,3	86,8
Gelişmekte olan ülkeler	94,2	146,8	184,3	284,4
Asya	55,4	89,6	116,2	184,2
Orta Doğu	13,8	21,9	26,4	37,9
Afrika	9,8	13,1	15,2	21,1
Orta ve Güney Amerika	15,2	22,0	26,6	41,1
Dünya Toplamı	368,0	426,1	507,0	675,3

Günümüz modern toplumlarında enerjiye olan ihtiyaç, özellikle teknoloji ve endüstriyel üretim süreçlerinde yaşanan hızlı gelişme, hızlı nüfus artışı ve bunlara bağlı olarak değişen yaşam standartlarının bir sonucu olarak, son 20 – 30 yıl içinde büyük bir artış göstermiştir. Sekizinci Beş Yıllık Kalkınma Planı kapsamında Elektrik Enerjisi Özel İhtisas Komisyonu tarafından hazırlanan raporda, 2001 yılında 77 044 000 TEP olan enerji tüketiminin 2020 yılında 217 000 000 TEP değerine ulaşacağı tahmin edilmektedir [11].

Kişi başına düşen toplam birincil enerji arzı açısından, 1,07 TEP/kişi olan Türkiye değerinin, dünya ve OECD değerlerinin altında olduğu Çizelge 1.2’de görülmektedir. Elektrik enerjisi tüketimi dikkate alındığında, bu farkın daha da

açıldığı görülmektedir. Kişi başına düşen elektrik enerjisi tüketimi, dünya ortalaması kişi başına 2,280 kWh ve OECD ortalaması 7,841 kWh iken, bu değer Türkiye’de 1,473 kWh’dır [10].

Çizelge 1.2. Dünya, OECD ve Türkiye enerji göstergelerinin karşılaştırması [10]

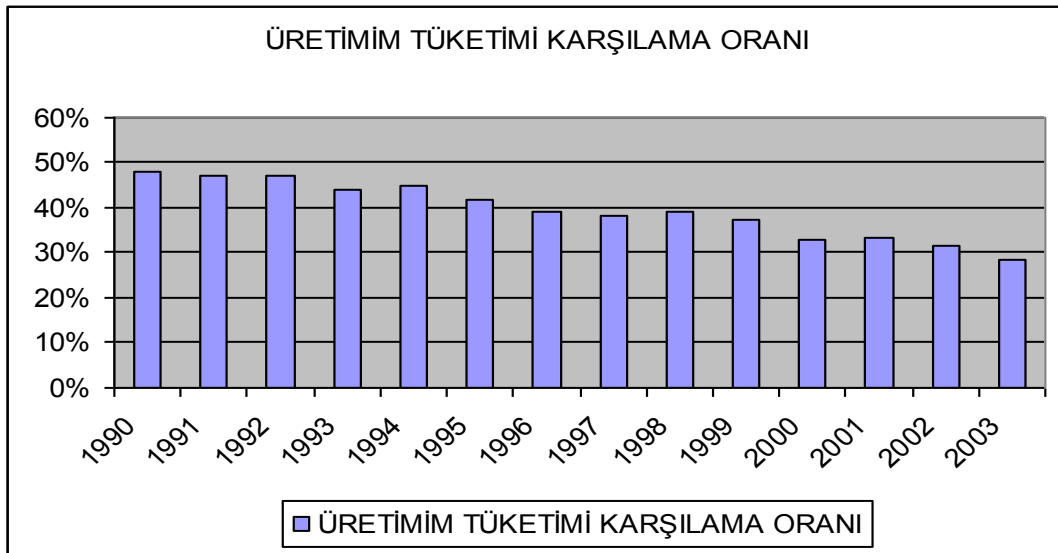
	Kişi başına toplam birincil enerji arzı (TEP/kişi)	Elektrik Tüketimi (TWh)	Kişi başına elektrik tüketimi (kWh/kişi)
Dünya	1,65	13 502,41	2 280
OECD	4,68	8 753,51	7 841
Türkiye	1,07	96,94	1 473

OECD ve AB ülkelerinde kurulu olan elektrik gücü miktarı üretim yöntemine göre Çizelge 1.3’de sunulmuştur.

Çizelge 1.3. OECD ve AB ülkelerinde elektrik kurulu gücü (GW - 2005 yılı)[14]

Ülke	Termik	Nükleer	Hidrolik	Diğer	Toplam
ABD	581,4	97,1	98,6	5,1	784,8
Almanya	79,8	22,3	8,9	2,7	113,6
Finlandiya	10,6	2,6	2,9	0,02	16,1
Fransa	25,5	61,7	25,1	0,3	112,6
İspanya	25,3	7,3	16,6	0,8	50,0
İsveç	6,5	10,1	16,3	0,2	33,0
Japonya	132,9	45,1	43,9	0,5	222,4
Kanada	32,3	10,6	67,0	0,05	109,9
Türkiye	26,6	-	12,9	0,07	39,6

Tüm dünyada bugün halen enerji ihtiyacının büyük bir kısmı, petrol ve kömür gibi yenilenemeyen enerji kaynaklarından sağlanmaktadır. Devlet İstatistik Enstitüsü (DİE) istatistiklerine göre, ülkemizde 2002 yılının Temmuz-Eylül dönemi içinde toplam enerji üretiminin %31,8'i kömür, fueloil, sıvılaştırılmış petrol gazı (LPG) ve motorinden; %39,7'si doğalgazdan; %28,3'ü hidroelektrik santrallerinde sudan ve yalnızca %0,11'i rüzgar ve jeotermal enerji ve diğer enerji türlerinden sağlanmıştır. 1990 yılında yeterli kaynaklara dayalı enerji üretimi, toplam birincil enerji talebinin % 48'ini karşılamış olup, giderek azalan eğilim içerisinde, 1997 yılında % 38'ini, 2003 yılında % 28'ini karşılayabilmiştir [12].



Şekil 1.1. 1990-2003 yılları arasında enerji üretiminin tüketimi karşılama oranı

Fosil yakıtlardaki yoğun kullanım ve tükenme nedeniyle yenilenebilir enerji kaynaklarının önemi artmıştır. Birçok yenilenebilir enerji kaynaklarının arasında, fotovoltaik diziler sulama sistemleri, pil şarjı, hibrid araçlar ve şebeke bağlantılı PV sistemleri gibi uygulamalarda kullanılmaktadırlar [1].

Güneş sistemleri en önemli yenilenebilir enerji kaynaklarından biridir. Gaz, kömür vs. gibi yenilenemeyen kaynaklarla karşılaştırıldığında, güneş enerjisi temiz, tükenmez ve bedavadır.

Ülkemiz, coğrafi konumu nedeniyle sahip olduğu güneş enerjisi potansiyeli açısından birçok ülkeye göre şanslı durumdadır. Devlet Meteoroloji İşleri Genel Müdürlüğünde (DMİ) mevcut bulunan 1966-1982 yıllarında ölçülen güneşlenme süresi ve ışıınım şiddeti verilerinden yararlanarak EİE tarafından yapılan çalışmaya göre Türkiye'nin ortalama yıllık toplam güneşlenme süresi 2640 saat (günlük toplam 7,2 saat), ortalama toplam ışıınım şiddeti 1311 kWh/m²-yıl (günlük toplam 3,6 kWh/m²) olduğu tespit edilmiştir. Aylara göre Türkiye güneş enerji potansiyeli ve güneşlenme süresi değerleri ise Çizelge 1.4'de verilmiştir.

Türkiye'nin en fazla güneş enerjisi alan bölgesi Güney Doğu Anadolu Bölgesi olup, bunu Akdeniz Bölgesi izlemektedir. Güneş enerjisi potansiyeli ve güneşlenme süresi değerlerinin bölgelere göre dağılımı da Çizelge 1.5' de verilmiştir.

Ancak, bu değerlerin, Türkiye'nin gerçek potansiyelinden daha az olduğu, daha sonra yapılan çalışmalar ile anlaşılmıştır. 1992 yılından bu yana EİE ve DMİ, güneş enerjisi değerlerinin daha sağlıklı olarak ölçülmesi amacıyla enerji amaçlı güneş enerjisi ölçümleri almaktadırlar. Devam etmekte olan ölçüm çalışmalarının sonucunda, Türkiye güneş enerjisi potansiyelinin eski değerlerden %20-25 daha fazla çıkması beklenmektedir.

PV üretim sistemlerinin 2 ana problemi bulunmaktadır. Özellikle düşük ışıınım miktarı koşullarında elektrik enerjisi çevrim verimi düşüktür (% 9-17) ve solar diziler tarafından üretilen elektrik gücü miktarı hava koşullarına bağlı olarak sürekli değişmektedir [13].

Çizelge 1.4. Türkiye'nin Aylık Ortalama Güneş Enerjisi Potansiyeli(EİE Genel Müdürlüğü)

AYLAR	AYLIK TOPLAM GÜNEŞ ENERJİSİ		GÜNEŞLENME SÜRESİ
	(Kcal/cm ² -ay)	(KWh/m ² -ay)	(Saat/ay)
OCAK	4,45	51,75	103
ŞUBAT	5,44	63,27	115
MART	8,31	96,65	165
NİSAN	10,51	122,23	197
MAYIS	13,23	153,86	273
HAZİRAN	14,51	168,75	325
TEMMUZ	15,08	175,38	365
AĞUSTOS	13,62	158,4	343
EYLÜL	10,6	123,28	280
EKİM	7,73	89,9	214
KASIM	5,23	60,82	157
ARALIK	4,03	46,87	103
TOPLAM	112,74	1311	2640
ORTALAMA	308,0 cal/cm²-gün	3,6 kWh/m²-gün	7,2 saat/gün

Dahası, solar hücre V-I karakteristiği lineer değildir ve ışınlam ve sıcaklığa bağlı olarak değişiklik göstermektedir. Genel olarak V-I ve P-V eğrilerinde maksimum güç noktası(MGN) adı verilen tek nokta vardır. Bu noktada PV sistem (dizi, dönüştürücü vs.) maksimum verimde çalışmakta ve maksimum çıkış gücünü üretmektedir. MGN noktasının yeri bilinmemektedir fakat arama algoritmaları veya hesaplama modelleri aracılığıyla bulunabilir. Bu nedenle PV dizilerinin maksimum güç noktasında (MGN) çalışmaları için maksimum güç noktası izleme (MGNI) tekniklerine ihtiyaç duyulmaktadır [2].

Çizelge 1.5. Türkiye'nin Yıllık Toplam Güneş Enerjisi Potansiyelinin Bölgelere Göre Dağılımı

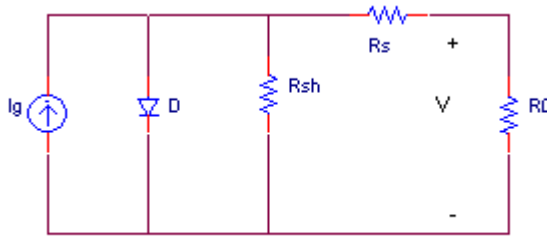
BÖLGE	TOPLAM GÜNEŞ ENERJİSİ (kWh/m²-yıl)	GÜNEŞLENME SÜRESİ (Saat/yıl)
G.DOĞU ANADOLU	1460	2993
AKDENİZ	1390	2956
DOĞU ANADOLU	1365	2664
İÇ ANADOLU	1314	2628
EGE	1304	2738
MARMARA	1168	2409
KARADENİZ	1120	1971

Güneş pillerindeki en önemli problemlerden biri güneşten alınan enerjinin kayıpsız veya az bir kayıpla yüke aktarılmasıdır. Bu tezde bu probleme yönelik çalışmalar yapılmıştır.

2. GÜNEŞ PİLİ KARAKTERİSTİĞİ

2.1. Güneş Pili Eşdeğer Devresi

Güneş pili lineer olmayan bir aygıttır. Eşdeğer devresi Şekil 2.1' de gösterilmiştir [3].



Şekil 2.1. Güneş pili eşdeğer devresi

PV modelin I-V karakteristiğini açıklayan basit denklem aşağıda verilmiştir.

$$I = I_L - I_0 \left(e^{\frac{q(V+IR_s)}{nkT}} - 1 \right) - \frac{V + IR_s}{R_{sh}} \quad (2.1)$$

Burada;

- I Hücre akımı(A)
- I_L, I_g Işık kaynaklı akım(A)
- I_0 Diyot doyma akımı
- q Elektron yükü (1.6×10^{-19} Coulomb)
- K Boltzman Sabiti (J/K)
- T Hücre sıcaklığı (K)
- R_s Hücrenin seri direci
- R_{sh} Hücrenin shunt direci
- V Hücre çıkış voltajıdır. [1]

R_{sh} çok büyük olduğundan, elektriksel modeli basitleştirmek için R_{sh} ihmal edilebilir [2].

$$I = I_L - I_0 \left(e^{\frac{qV+IR_s}{nkT}} - 1 \right) \quad (2.2)$$

$$I_L = I_{sc,1000} \times \frac{G}{G_0} \quad (2.3)$$

$$G_0 = 1000 \text{ W/m}^2 \quad [4] \quad (2.4)$$

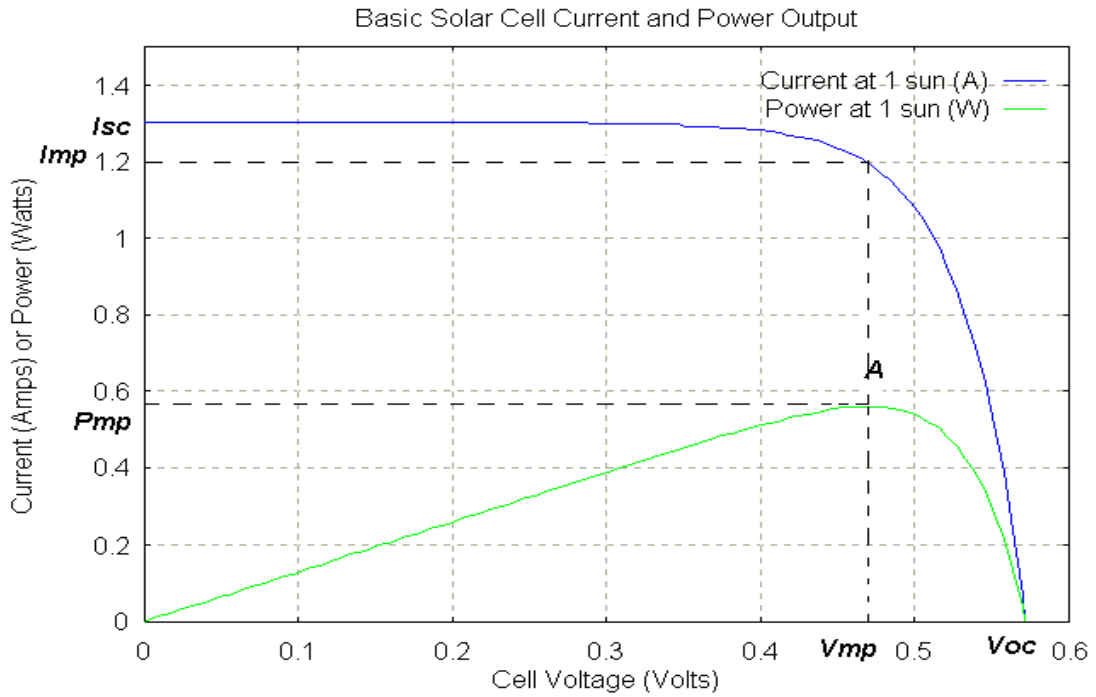
$$I = I_{sc,1000} \times \frac{G}{1000} - I_0 \left(e^{\frac{qV+IR_s}{nkT}} - 1 \right) \quad (2.5)$$

$$I_0 = \frac{I_{sc}}{\frac{qV_{oc}}{e^{nkT}} - 1} \quad [4] \quad (2.6)$$

$$I = \left[I_{sc,1000} \times \frac{G}{1000} \right] - \left[\left(\frac{I_{sc}}{\frac{qV_{oc}}{e^{nkT}} - 1} \right) \left(e^{\frac{qV+IR_s}{nkT}} - 1 \right) \right] \quad (2.7)$$

2.2. Güneş Pili Karakteristiği

Güneş pili için I-V ve P-V karakteristiği Şekil 2.2’de gösterilmiştir.



Şekil 2.2. Güneş pili I-V ve P-V karakteristiği

Şekilde gösterilen performans parametreleri:

Açık Devre Voltajı (Open Circuit Voltage - V_{oc}): Şekil 2.2' de I-V eğrisinin x eksenini kestiği yani akımın sıfır olduğu noktadır. Bu noktada güç sıfıra eşittir.

Kısa Devre Akımı (Short Circuit Current - I_{sc}): Şekil 2.2' de I-V eğrisinin y eksenini kestiği yani voltajın sıfır olduğu noktadır. Bu noktada güç sıfıra eşittir.

Maksimum Güç Noktası (Maximum Power Point – MGN): Gücün maksimuma ulaştığı noktadır. Şekil 2.2' de A noktası ile gösterilmiştir. Bu noktada güç maksimuma ulaşmaktadır.

Maksimum Güç Voltajı (Maximum Power Voltage - V_{mp}): Gücün maksimum olduğu noktadaki voltaj değeridir.

Maksimum Güç Akımı (Maximum Power Current - I_{mp}): Gücün maksimum olduğu noktadaki akım değeridir.

Fill Factor (FF): Maksimum gücün $I_{sc}V_{oc}$ değerine oranıdır. Fill Factor oranı I-V eğrisinin keskinliğini belirlemektedir. İyi PV hücreleri için FF değeri 0,7'den yüksektir. Kalite kontrol testleri için FF değeri önemli bir parametredir.

2.3. Güneş Pili Üretim Teknolojisi

Güneş pili teknolojisi, kullanılan maddeler ve yapım türleri açısından son derece zengindir. Güneş pili yapımı için şu anda kullanılmakta olan bir düzineden fazla maddenin yanı sıra, yüzlerce maddenin de üzerinde çalışılmaktadır. Belli başlı güneş pili türleri aşağıda anlatılmaktadır.

2.3.1. Kristal silisyum güneş pilleri

Silisyum yarı iletken özellikleri tipik olarak gösteren ve güneş pili yapımında en çok kullanılan bir maddedir ve uzun yıllarda bu konumunu koruyacak gibi görünmektedir. Fotovoltaik özellikleri daha üstün olan başka maddeler de olmakla birlikte, silisyum hem teknolojisinin üstünlüğü hem de ekonomik nedenlerle tercih edilmektedir [16].

2.3.2. Ribbon silisyum güneş pilleri

Bu piller, malzeme kaybının azaltılması amacıyla levha halinde silisyum tabakalarından yapılırlar. Efg, Dendritik ağ gibi çeşitli yöntemlerle elde edilen bu piller, halen geliştirme aşamasındadır. Verimleri laboratuvar şartlarında %13 - 14 arasındadır [16].

2.3.3. Polikristal silisyum güneş pilleri

Bu piller de ribbon silisyum teknolojisiyle yapılıp, yapıları polikristal özellik gösterir. Halen laboratuvar aşamasındaki bu pillerin verimleri %10'dur [16].

2.3.4. İnce film güneş piller

Bu teknikte, absorban özelliği daha iyi olan maddeler kullanılarak daha iyi olan maddeler kullanılarak daha az kalınlıkta (tek kristalin 1-500'ü kalınlığında) güneş pilleri yapılır. Örneğin amorf silisyum güneş pillerinin absorpsiyon katsayısı kristal silisyum güneş pillerinin katsayısından daha fazladır. Dalga boyu katsayısı 0,7 mikrondan küçük bir bölgedeki güneş radyasyonu 1 mikron kalınlığında amorf silisyum ile absorblanabilirken, kristal silisyumda ise aynı radyasyonu absorblamak için 500 mikron kalınlıkta malzeme kullanılması gerekmektedir. Bu yüzden amorf yapıli güneş pillerinde daha az malzeme kullanılır ve montaj kolaylığı nedeniyle bir avantaj sağlar [16].

2.3.5. Amorf silisyum güneş pilleri

Amorf silisyum güneş pilleri (a-Si), ince film güneş pili teknolojisinin en önde gelen örneğidir. İlk yapılan a-Si piller Schottky bariyer yapısında iken, daha sonraları p-i-n yapıları geliştirilmiştir. P-i-n yapısındaki pillerin fabrikasyonu kalay oksitle kaplı iletken bir yüzeyin üzerine çöktürme yöntemi ile yapılır, bu yüzeyin arkası daha sonra metalle kaplanır. Verimleri %5 - 8 arasındadır.

Ancak bu piller, kısa zamanda bozulmaya uğrayarak çıkışları azalır [16].

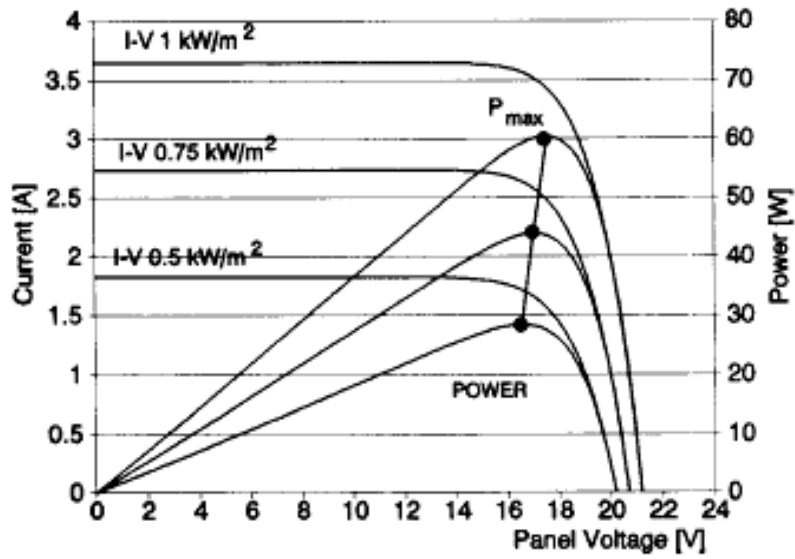
2.3.6. Diğer yapılar

Bakır indiyum diselenit (CuInSe) maddesinden yapılan ve verimleri %13 civarında olan piller halen gelişme aşamasındadır ve daha kararlı çıkışa sahip olduğu için absorban özelliği yüksek, verimleri de %12 civarındadır. Bu güne kadar elde edilen en yüksek verime (%24) galyum arsenitten yapılan piller ulaşmıştır. Bu madde ile çeşitli türde piller elde edilebilmekle birlikte, pahalı olduğu için pillerin, güneş spektrumunun daha büyük bir bölümünden yararlanabilmesi amacı ile denenen bir yöntem ise, birden fazla ince film yapısının üst üste konmasıyla elde edilen çok eklemlili film yapılarıdır.

Bunların dışında, güneş ışınımının yüksek verimli pillerin üzerine optik olarak yoğunlaştıran sistemler üzerinde çalışmalar yapılmaktadır. Bu tür sistemlerde güneşin hareketini izleyen düzeneklerin yanı sıra, güneş ışığını kıran (mercek) ya da yansıtan (ayna) eleman kullanılır [17].

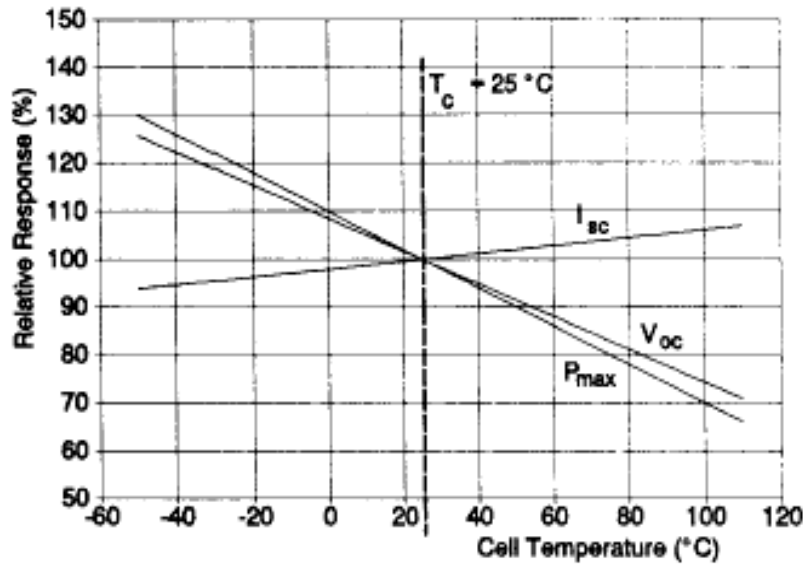
2.4. Işınım ve Sıcaklığın Etkisi

Şekil 2.3'te 60W'lık bir PV panelin 500, 750, 1000 W/m^2 'lik ışınımdaki I-V eğrisi görülmektedir. Işınımın artması ile akımında arttığı görülmektedir. Panelin maksimum güç noktası, ışınım ile orantılı olarak dik bir artış göstermektedir. Hücre sıcaklığı sabit tutulduğunda, ışınımın artmasıyla güç artmakta, ayrıca maksimum gücün elde edildiği voltaj değeri (V_{mpp}) de artmaktadır. Bu durum ışınımın 1 güneş değerinden ($1 \text{ sun} = 1 \text{ kW/m}^2$) daha büyük değerlere kadar sürmektedir. Işınım derecesini etkileyen faktörlerden bazıları: Günlük ve sezonluk güneş yolu, güneş ve dünya arasındaki sis, bulut vs [5].



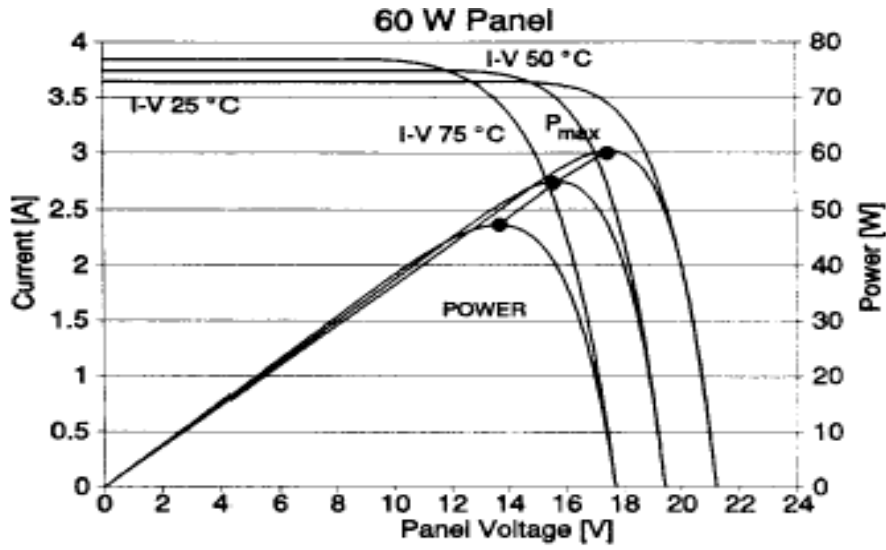
Şekil 2.3. Işınımın I-V ve P-V eğrisine etkisi

PV panelin sıcaklığı; ışınım miktarı, çevre sıcaklığı, panelin soğutma oranı ve panelin yapım şekline bağlıdır. Güney Afrika'nın ılıman bölgelerinde hücre sıcaklığının 1 güneş şartı altında 85°C gibi yüksek bir sıcaklığa kolaylıkla erişebildiği görülmektedir. Şekil 2.4'te Silikon Solar hücrelerde sıcaklığın farklı parametreler üzerinde genel etkileri görülmektedir. Hücre sıcaklığının artışıyla, açık devre voltajında ve maksimum güç miktarında düşüş görülürken, kısa devre akımında artış görülmektedir [6].



Şekil 2.4. Hücre sıcaklığının çeşitli parametreler üzerindeki etkisi [7]

Sıcaklığın Şekil 2.3'te gösterilen 60W'lık panelin I-V eğrisi üzerindeki etkisi Şekil 2.5'te gösterilmiştir. Yüksek hücre sıcaklığının neden olduğu maksimum güçteki ve maksimum güç noktasındaki voltaj değerindeki (V_{mp}) düşüşe dikkat edilmektedir. Yüksek hücre sıcaklığı, PV dizisinden alınan güç üzerinde çok büyük etkiye sahiptir ve Maksimum Güç Noktası Takipçisi (Maximum Power Point Tracker-MGNI) kullanılarak bu güç kayıplarının bazıları giderilebilir [6].



Şekil 2.5. Sıcaklığın 60 W'lık panel üzerindeki etkisi [6]

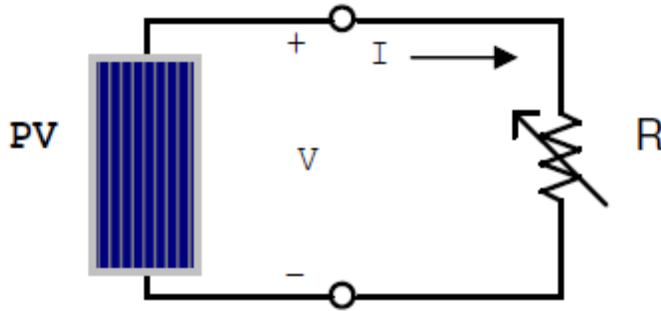
Uluslararası standartlara göre PV panel çıkış gücü 1 kW/m^2 ışınım, $25 \text{ }^\circ\text{C}$ hücre sıcaklığı ve 1,5 air mass şartlarında ölçülmektedir. Her ne kadar 60 W'lık panelin I-V karakteristik eğrisi Şekil 2.3 ve Şekil 2.5'te gösterilmiş olsa da bu güç çıkışları 1 kW/m^2 'lik ışınım ve $25 \text{ }^\circ\text{C}$ hücre sıcaklığında geçerlidir. Bu bilgiye dayanarak; pratikte beklenen PV dizi gücü ile PV dizisinin üreteceği güç arasında uyumsuzluk olacağı açıktır [6].

2.5. İstenen Güç ile Üretilen Güç Arasındaki Uyumsuzluğun Giderilmesi

PV modül yüke direk bağlandığında, PV modüle çalışma noktası akım-gerilim eğrisi ile yükün akım – gerilim ilişkisinin oluşturduğu yük çizgisinin kesiştiği noktada olacaktır.

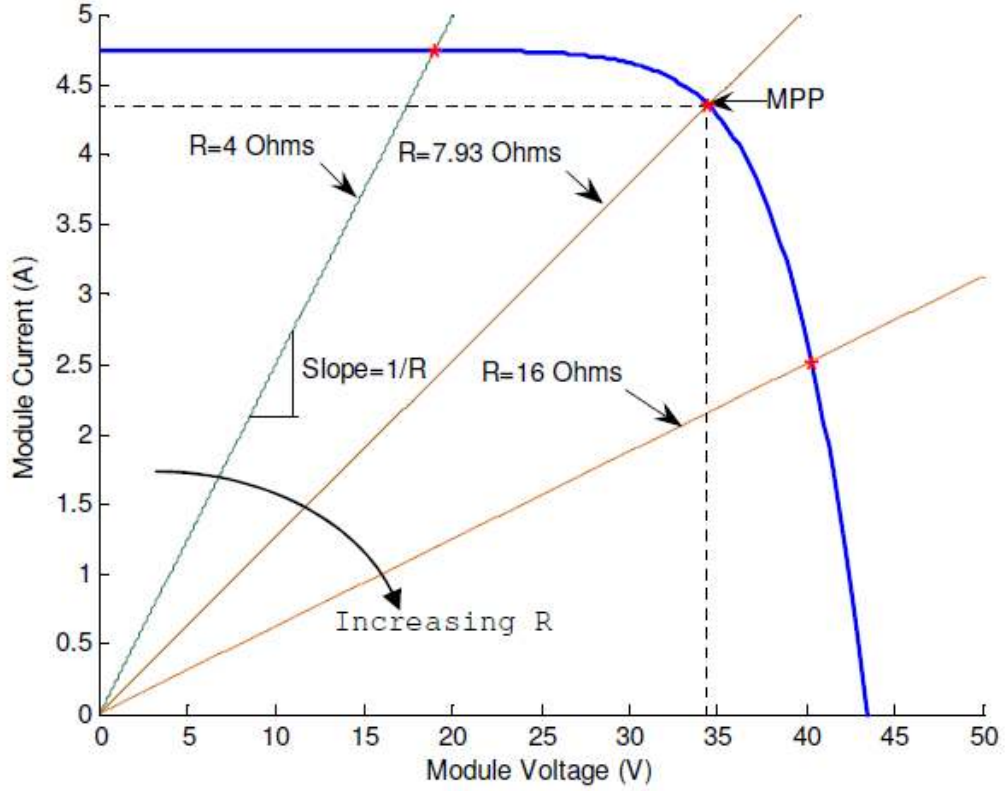
Örnek olarak Şekil 2.6' da dirençsel yükün Şekil 2.7'de gösterildiği gibi $1/R_{\text{load}}$ eğiminde düz bir çizgisi vardır. Diğer bir değişle yükün empedansı PV modülün çalışma koşullarını belirlemektedir. Genel olarak bu çalışma noktası

seyrek olarak PV modülün maksimum güç noktası olmaktadır. Bu nedenle PV modüle maksimum güç üretememektedir [4].



Şekil 2.6. PV modülün değişken direnç ile bağlanması

Yapılan çalışma göstermiştir ki direk bağlı sistem PV kapasitesinin %31'ini kullanabilmektedir [31].



Şekil 2.7. I-V eğrisinin değişik direnç değerlerinde simülasyonu

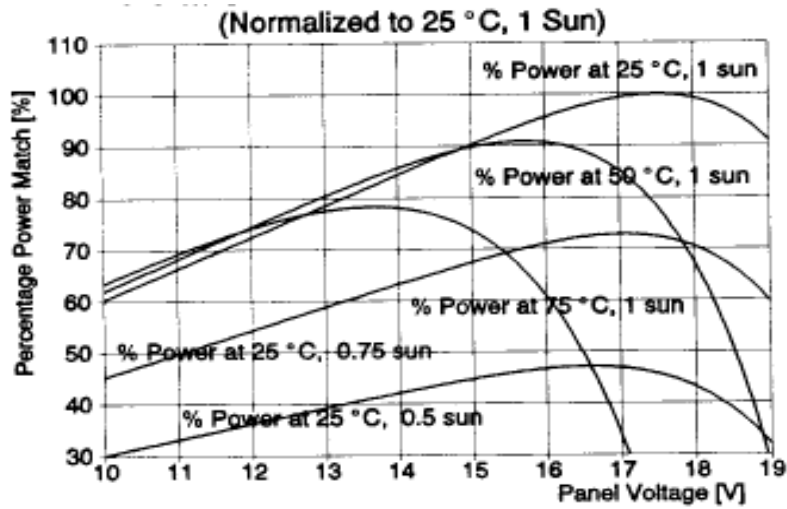
Düşük güç verimli kış ayları boyunca, PV dizisi gücü genellikle tolare edebilecek büyüklüğü düşüktür. PV dizi ile yük arasındaki bu uyumsuzluk PV dizinin daha fazla büyütülmesini gerektirmekte ve bu da genel sistem maliyetini arttırmaktadır. Bu problemi azaltmak için MGNI PV dizisinin maksimum güç noktasında çalışmasını sağlamak için kullanılabilir. MGNI'ler düzgün ayarlandığında PV gücünü %97'den fazla arttırabilirler [21].

Beklenen güç ile üretilen güç arasındaki uyumsuzluk; düşük ışıınım derecesi ve yüksek hücre sıcaklığından kaynaklanmaktadır. PV dizisinin maksimum gücündeki gerilim değeri PV dizisinden alınan gücü etkilediğinden ışıınım derecesi ve hücre sıcaklığı PV dizisinden alınan gücün limit değerini belirlemektedir [6].

Bu nedenle uyumsuzluk 3 parametreden oluşmaktadır:

- Işınım derecesi (Insolation levels)
- Hücre sıcaklığı (Cell temperature)
- Dizi gerilimi (Array voltage)

Şekil 2.8, Şekil 2.3 ve Şekil 2.5'te tanımlanan PV panel uyumsuzluğunu göstermektedir. Sözü edilen 3 parametrede Şekil 2.8'de görülmektedir.



Şekil 2.8. Güneş pili uyumsuzluk karakteristiği [6]

PV güç üzerinde ışıınım derecesi nedeniyle oluşan limitler güneş yolu izleyicisi (Solar Path Tracker) kullanılarak azaltılabilir. Günlük ve sezonluk güneş izlenimi ortalama ışıınım derecesini arttıracak ve sonuçta ortalama günlük uyum limiti artacaktır.

Güneş yolu izlenimi ile PV panel uyum limitinin %47 (0.5 güneş)'den %100 (1 güneş)'e çıktığı Şekil 2.8'de görülmektedir. Hücre ile çevresi arasındaki artan termal iletkenlik nedeniyle oluşan ısı artışı limit değerin azalmasına neden olmaktadır. 75 °C'de %78 olan PV panel uyum limiti 50 °C'de %91'e yükselmektedir.

Daha önce açıkladığımız gibi ışınlm derecesi ve hücre sıcaklığı sadece en iyi uyum limitini belirlemektedir. Asıl uyum PV dizisi gerilimi tarafından belirlenmektedir. Bu nedenle yük ile PV güç kaynağı arasında DC/DC dönüştürücü ile yük uyumu sağlanmalıdır [6].

3. DÖNÜŞTÜRÜCÜ KARAKTERİSTİĞİ VE YÜK EŞLEŞTİRME

Çoğu uygulamalarda, maksimum güç noktası izleyicisi, mevcut maksimum enerjiyi PV dizisinden yüke aktarmak için tavsiye edilir.

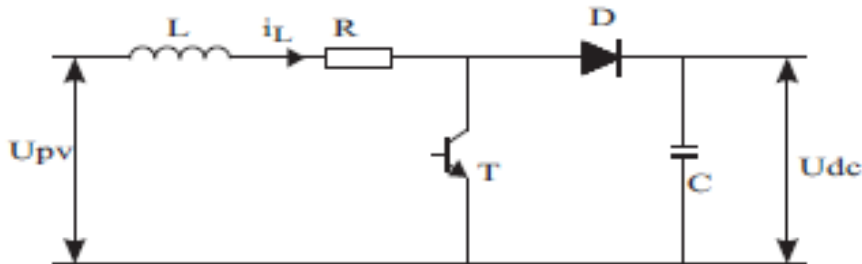
Maksimum güç noktası izleyicisi Buck veya Boost topolojileri üzerine kurulabilir. Buck dönüştürücü genelde düşük çıkış gerilimlerinde, Boost dönüştürücü ise daha yüksek çıkış gerilimlerinde kullanılmaktadır. Çoğu durumlarda Buck dönüştürücü daha yüksek verim sağlamaktadır. Bu iki yapılandırma arasındaki diğer fark ise; Buck dönüştürücüde akım bazen giriş portuna akmamaktadır fakat Boost dönüştürücüde akım sabit bir şekilde giriş portuna akmaktadır.

Buck topolojisinin geleneksel sistemlerdeki yüksek verimine karşın; PV sistemlerde Boost topolojisi daha yüksek uyum göstermektedir. Çünkü Boost dönüştürücü sürekli akım modunda PV hücrelerden mümkün olan en yüksek gücü sağlamaktadır [8].

3.1. Boost Topolojisi

3.1.1. Topolojinin genel yapısı

Boost topolojisinin yapısı Şekil 3.1’de görülmektedir.

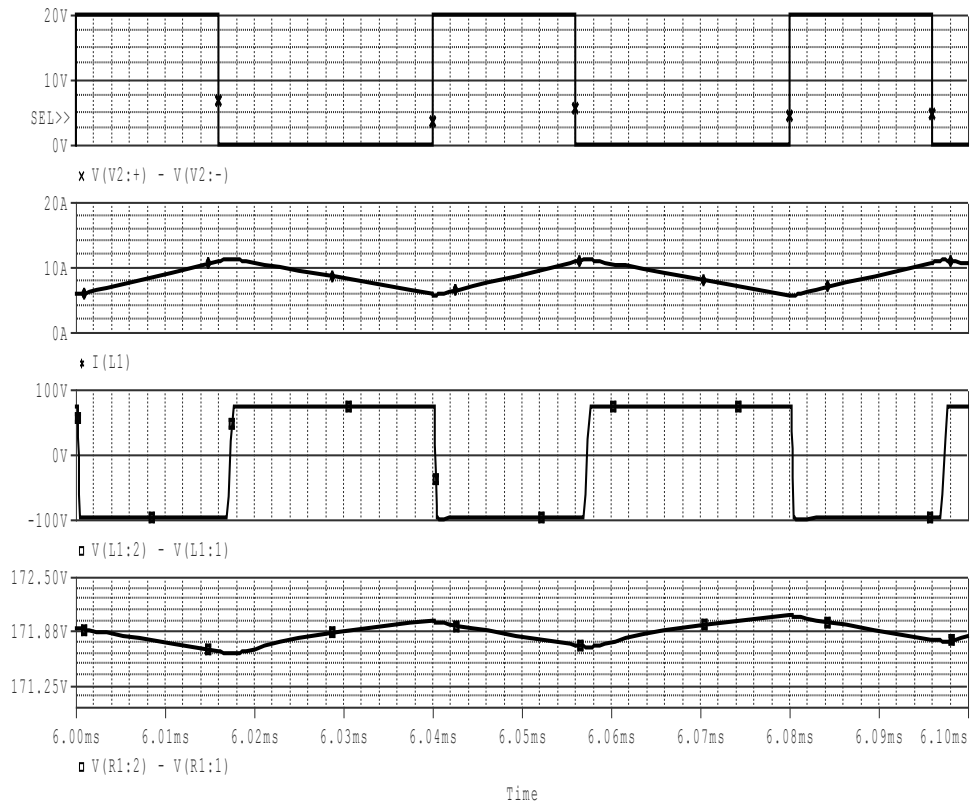


Şekil 3.1. Boost topolojisi[18]

S anahtarı kapalı (iletimde) olduğunda akım L endüktansı üzerinden akmaktadır. Böylece endüktans şarj olmaktadır. Aynı zamanda yük gücünü çıkış kapasitesinden almaktadır. S anahtarı açıldığında akım güç kaynağından endüktans ve diyod üzerinden akacaktır. Böylece C kapasitesi tekrar şarj olacak ve eş zamanlı olarak yükü besleyecektir.

Şekil 8’de tipik dalga formları görülmektedir [8].

- Anahtarlama dalga formları
- Giriş endüktansı üzerindeki akımın dalga formu
- Giriş endüktansı üzerindeki gerilimin dalga formları
- Yük üzerindeki gerilimin dalga formları



Şekil 3.2. Boost topolojisi dalga formları

Yukarıdaki grafikte;

1. grafikte S anahtarının durumu,
2. grafikte L endüktansı üzerindeki akımının değişimi,
3. grafikte L endüktansı üzerindeki gerilim değişimi,
4. grafikte ise yük direnci üzerindeki gerilimin değişimi gösterilmiştir.

Analizi basitleştirmek için;

- Tüm elemanlar ideal,
 - Yük omik,
 - Anahtarlama zamanı elektrik zaman sabitinden daha küçük
- Seçilmiştir.

3.1.2. Çıkış gerilimi değeri

Dönüştürücü durgun durum operasyonunda, endüktans üzerindeki gerilim değeri T periyodu boyunca 0'dır. Çıkış gerilimi denklemi:

$$V_{out} = \frac{V_{in}}{1-D} \quad (3.1)$$

Burada;

V_{out} : Çıkış gerilimi,

V_{in} : Giriş gerilimi,

D : Doluluk oranı (Duty Cycle = $\frac{T_{on}}{T}$)

T_{on} : Anahtarın iletimde olduğu süre [18].

3.1.3. Kapasite ve endüktans değerleri

Gerekli olan maksimum akım ve gerilim dalgalanmalarından, T_{on} ve T_{off} sürelerindeki enerji şarjı ve deşarjı düşünülerek kapasite ve endüktans değerleri aşağıdaki denklemlerle hesaplanabilir.

$$C = \frac{V_{out} * T * D}{2\Delta V_{out} R_{load}} \quad (3.2)$$

$$L = \frac{2C * V_{out} * \Delta V_{out}}{2I_{in} \Delta I_{in}} \quad (3.3)$$

Burada;

C : Çıkış kapasitesi

T : Anahtarlama Periyodu

R_{load} : Yük direnci

ΔV_{out} : Çıkış gerilim dalgalanması

L : Giriş endüktansı

ΔI_{in} : Giriş Akımı Dalgalanması

I_{in} : Giriş akımı [8].

3.1.4. Yük direnci değeri

Giriş ve çıkış gerilimleri arasında doluluk oranına bağlı bir ilişki bulunmaktadır. Giriş ve çıkış arasında güç kaybı olmadığı düşünülürse (POPI tip);

Uygun yük denklemi:

$$R_{load} = \frac{V_{in}}{I_{in}(1-D)^2} \quad [8] \quad (3.4)$$

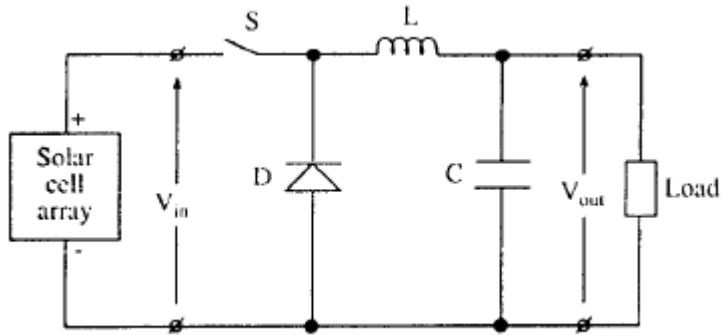
3.2. Buck topolojisi

3.2.1. Topolojinin genel yapısı

Buck dönüştürücü 2 kısımdan oluşmaktadır.

1. Değişken doluluk oranına sahip kare dalga kaynağı
2. 40 dB'lik alçak geçiren filtre (Düşük çıkış dalgalanması için

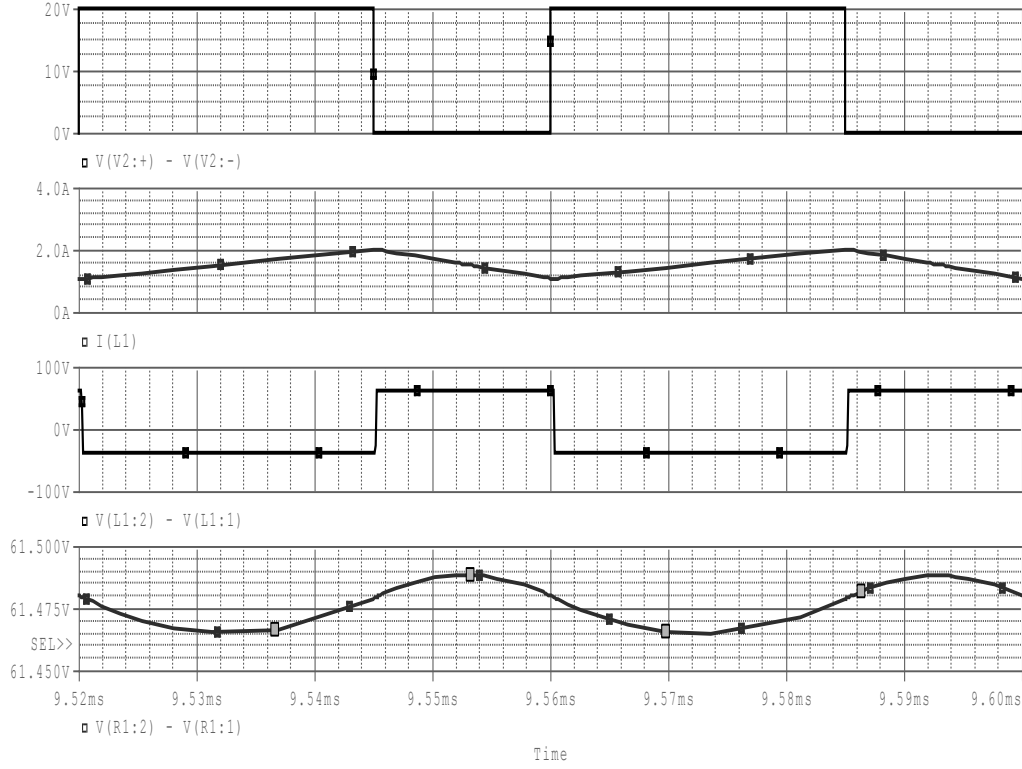
$f_c = \frac{1}{2\pi\sqrt{LC}}$ olarak kabul edilmiştir.)



Şekil 3.3. Buck topolojisi [5]

Bu dönüştürücünün iyi tarafı çıkış gerilimi dalgalanmasının çıkış gerilimine oranla çok düşük olmasıdır. Bu nedenle yük akımı sabite yakındır ve endüktans tarafından üretilen dalgalı akım kapasiteye akmaktadır.

Tipik gerilim ve akım dalga formları Şekil 10'da gösterilmiştir [8].



Şekil 3.4. Buck topolojisi dalga formları

Yukarıdaki diyagramda

1. grafikte S anahtarının durumu,
2. grafikte L endüktansı üzerinden akan akımın değişimi,
3. grafikte endüktans üzerindeki gerilim değişimi,
4. grafikte ise yük direnci üzerindeki gerilim değişimi gösterilmiştir.

3.2.2. Çıkış gerilimi değeri

Endüktans üzerindeki gerilim T periyodu boyunca 0 olduğundan aşağıdaki denklem yazılabilir [5].

$$V_{out} = V_{in} D \quad (3.5)$$

3.2.3. Kapasite ve endüktans değerleri

Herhangi bir zamanda endüktans üzerindeki gerilim değeri aşağıdaki denklem ile hesaplanabilir.[8]

$$V_L = L \frac{di_L}{dt} \quad (3.6)$$

Endüktans ve kapasite değerleri aşağıdaki denklemler ile hesaplanabilir.

$$\Delta I_L = \frac{V_{out}(1-D)T}{L} \quad (3.7)$$

Kapasite değeri ise;

$$C = \frac{T\Delta I_L}{8C\Delta V_{out}} \quad (3.8)$$

3.2.4. Yük direnci değeri

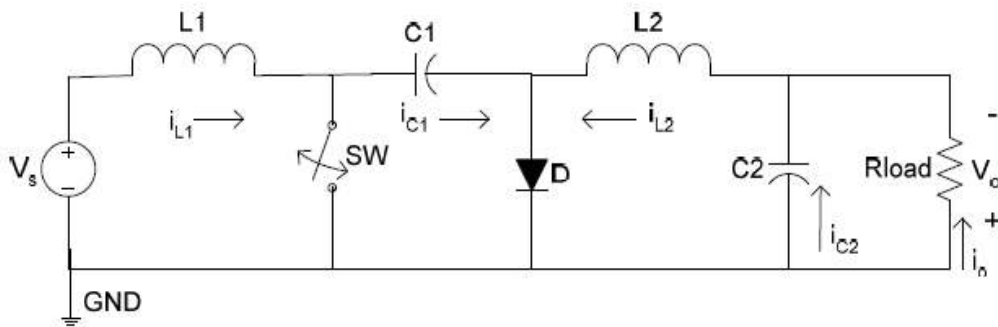
Giriş ve çıkış gerilimleri arasındaki ilişki doluluk oranı ile belirlenmektedir. %100 verim ile dönüştürücünün POPI tipinde olduğu kabul edilirse [8];

$$R_{load} = \frac{V_{in}}{I_{in}} * D \quad (3.9)$$

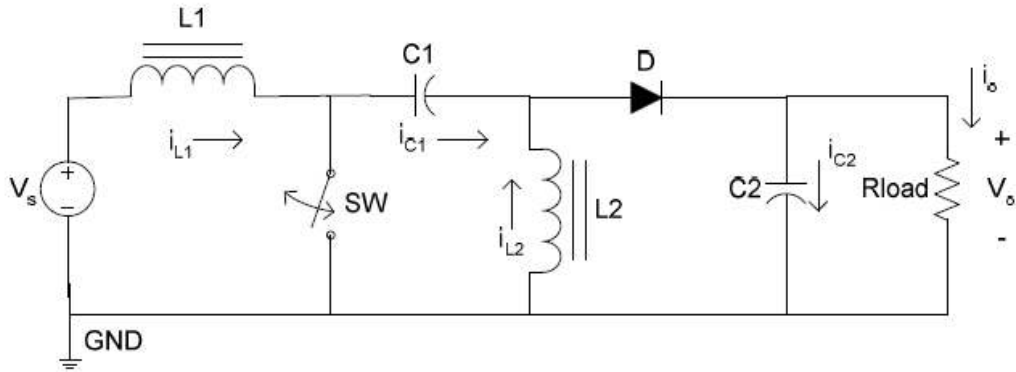
3.3. Cuk ve SEPIC Dönüştürücüler

3.3.1. Topolojilerin genel yapısı

Şekil-3.5 Cuk dönüştürücü basit yapısını göstermektedir. Giriş geriliminden yüksek veya düşük çıkış gerilimi sağlayabilir. Cuk dönüştürücünün değiştirilmiş hali olan SEPIC dönüştürücü de aynı şekilde giriş gerilimini artırıp azaltabilmektedir. Şekil 3.6 SEPIC dönüştürücünün devre diyagramını göstermektedir.



Şekil 3.5. Cuk topolojisi



Şekil 3.6. SEPIC topolojisi

Bu iki topolojinin karakteristikleri birbirinin benzeridir. İkisi de ana enerji deposu olarak kapasite kullanmaktadır. Sonuç olarak giriş akımı sürekli [4]. Devreler düşük anahtarlama kaybı ve yüksek verimliliğe sahiptirler [30]. Bu iki topoloji arasındaki ana farklılık Cuk dönüştürücü giriş geriliminin tersi polaritede çıkış vermesidir. SEPIC dönüştürücü de ise giriş ve çıkış geriliminin polaritesi aynıdır. Bu nedenle SEPIC dönüştürücü bazı

uygulamalarda Cuk dönüştürücü yerine tercih edilmektedir. Ayrıca SEPIC dönüştürücü diyotun bulunduğu yer sayesinde aküden güneş piline akım akışını engellemesi nedeniyle akü şarj sistemlerinde tercih edilebilir. Aynı diyot çıkış akımında yüksek salınıma neden olduğundan dezavantaj oluşturmaktadır. Diğer taraftan Cuk dönüştürücü çıkışta bulunan endüktans sayesinde daha iyi çıkış akımı karakteristiği sağlamaktadır. [4]

3.3.2. Çıkış gerilimi değeri

Cuk ve SEPIC dönüştürücülere ait çıkış gerilimi denklemi aşağıda verilmiştir:

$$\frac{V_o}{V_g} = \frac{D}{1-D} \quad (3.10)$$

D doluluk oranı ile olan ilişki:

- $0 < D < 0,5$ ise çıkış gerilimi giriş geriliminden küçüktür.
- $D = 0,5$ ise çıkış gerilimi giriş gerilimine eşittir.
- $0,5 < D < 1$ ise çıkış gerilimi giriş geriliminden büyüktür [4].

3.4. Buck – boost dönüştürücü karşılaştırılması

Asıl amacın olabilecek en yüksek gücün kaynaktan yüke iletilmesi olduğundan, durgun durumdaki çıkış gerilimi birinci önceliğe sahip değildir. Bu nedenle güneş pilinin çalışma gerilim ve akımının, maksimum akım ve gerilim değerlerine (V_{max} , I_{max}) çok yakın olmalıdır [8].

Güneş pilinden alınabilecek maksimum güç miktarı:

$$P_{max} = V_{max} * I_{max} \quad (3.11)$$

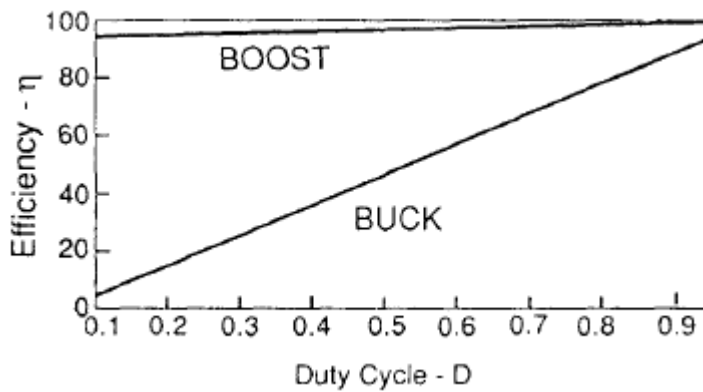
Denklemi ile hesaplanmaktadır. Dönüştürücü tarafından iletilen güç ise:

$$P_{conv} = \frac{1}{T} \int V_{in} I_{in} dt = \frac{1}{T} \left[V_{in} \int_{on} I_{in} dt + V_{in} \int_{off} I_{in} dt \right] \quad (3.12)$$

Ve dönüştürücü verimi ise;

$$\eta_{conv} = P_{conv} / P_{max} \quad (3.13)$$

Şekil 3.7'e göre her iki topolojide de doluluk oranının artışıyla verimin arttığı görülmektedir. Bununla birlikte Buck dönüştürücüdeki verimin artış hızı Boost dönüştürücüde verimin zaten yüksek olması nedeniyle daha yüksektir. Buck dönüştürücü ancak %100 doluluk oranıyla Boost dönüştürücü verimini sağlamaktadır. Sonuç olarak Boost dönüştürücünün güneş pili uygulamalarında daha başarılı olacağı görülmektedir. Buck dönüştürücü ise yüksek doluluk oranında kullanılmalıdır.



Şekil 3.7. Topoloji verimlerinin karşılaştırılması

Boost dönüştürücü giriş gerilimini yükseltirken (step-up) Buck dönüştürücü giriş gerilimini düşürmektedir (Step-down). Giriş geriliminin çıkış geriliminden yüksek olduğu uygulamalarda Buck dönüştürücü kullanılması gerekmektedir.

Boost dönüştürücü kullanımındaki zorluk ise Boost dönüştürücünün kontrolünün zorluğudur. Boost dönüştürücüdeki doluluk oranındaki değişimle verimin çok küçük bir aralıkta değişimidir [8].

Boost dönüştürücü teorik olarak üstün olmakla birlikte, pratikte dönüştürücünün verim avantajları nedeniyle maksimum güç noktasını yakalaması önemsizdir. Bununla birlikte diğer avantajları MGNI için boost topolojisini buck topolojisinden daha iyi bir seçim yapabilir [9].

3.5. Yük Eşleştirme

Bölüm 2.5'te bahsedildiği gibi, PV yüke direk bağlandığında, PV çalışma noktası yük tarafından belirlenmektedir. Yükün empedansı aşağıda tanımlanmıştır:

$$R_{load} = \frac{V_0}{I_0} \quad (3.14)$$

Burada V_0 çıkış gerilimi ve I_0 çıkış akımıdır.

PV için optimal yük değeri ise:

$$R_{opt} = \frac{V_{MPP}}{I_{MPP}} \quad (3.15)$$

Burada V_{MPP} ve I_{MPP} maksimum güç noktasındaki gerilim ve akımdır. R_{load} değeri R_{opt} 'ye eşit olduğunda, maksimum güç transferi gerçekleşmektedir. Bu iki parametre pratikte bağımsızdır ve nadiren eşit olur. MGNI'nin hedefi yükün empedansı ile PV'nin optimal empedansını eşitlemektir [4].

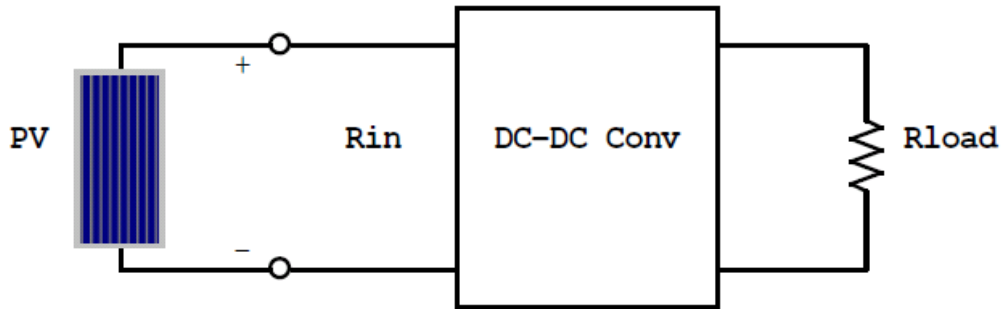
Aşağıdaki yük eşleştirme örneği ideal (Kayıpsız) Cuk dönüştürücü kullanmaktadır.

$$I_s = \frac{D}{1-D} I_0 \quad (3.16)$$

Eşitlik 3.10 ve 3.16'dan;

$$R_{in} = \frac{V_s}{I_s} = \frac{(1-D)^2 V_0}{D^2 I_0} = \frac{(1-D)^2}{D^2} R_{load} \quad (3.17)$$

Şekil 3.8'de görüldüğü gibi, PV tarafından görülen empedans dönüştürücünün girişinde görülen empedansa eşittir (R_{in}). Doluluk oranı (D) değiştirilerek R_{in} değeri R_{opt} değeri ile eşitlenebilir. Böylece yükün empedansı, doluluk oranı ayarlandığı müddetçe herhangi bir değer alabilir.



Şekil 3.8. PV tarafından görülen empedans

4. MGNI TEKNİKLERİ

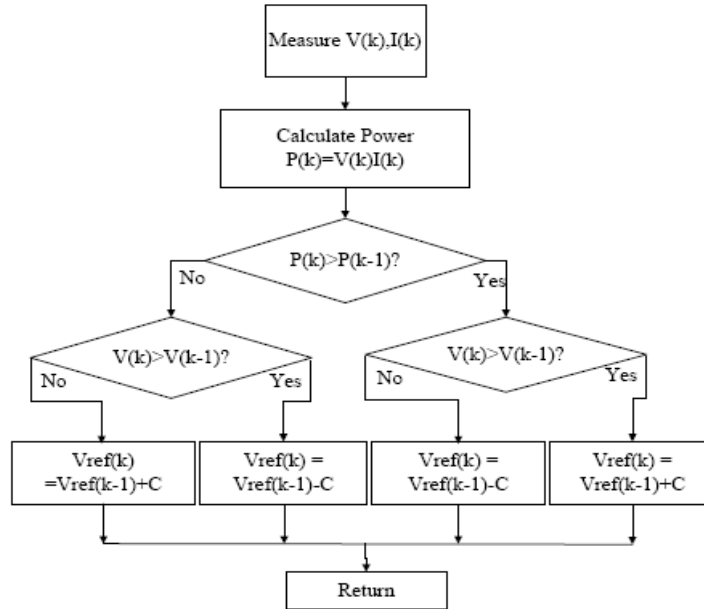
4.1. Giriş

Maksimum güç noktası izleme sistemlerinin kullandığı bir çok yöntem bulunmaktadır. Bu bölümde bu yöntemlere ilişkin incelemeler yapılacaktır.

4.2. Hill Climbing ve Kaygılandır & Gözetle (K&G)[33]

Elde edilen tüm makaleler içerisinde bir çoğu hill climbing ve kaygılandır ve gözetle metotları üzerine yoğunlaşmıştır. Hill Climbing güç çeviricinin doluluk oranında kaygılandırma gerektirirken, K&G metodu PV dizisinin çalışma geriliminde kaygılandırma gerektirir. PV dizisinin güç çeviriciye bağlı olduğu durumda; güç çeviricinin doluluk oranının kaygılandırılması PV dizisi akımını kaygılandıracak ve bu nedenle PV dizi gerilimi kaygılanacaktır. Hill Climbing ve K&G metotları aynı temel metodu tasarlamak için farklı yöntemlerdir [19].

Basitliği nedeniyle K&G metodu en popüler MGNI tekniğidir. Şekil 4.1 K&G metodunun akış diyagramını göstermektedir. Yapılan kaygılandırma sonrasında mevcut güç ölçülür ve geçmiş güç ile karşılaştırılarak güç değişimi(ΔP) hesaplanır. Eğer $\Delta P > 0$ ise uygulama kaygılandırmanın aynı yönünde devam eder. Aksi takdirde uygulama tersi yönde devam eder.[20]



Şekil 4.1. P&O algoritması akış diyagramı

Algoritmalar Çizelge 4.1’de özetlenmiştir. [19]

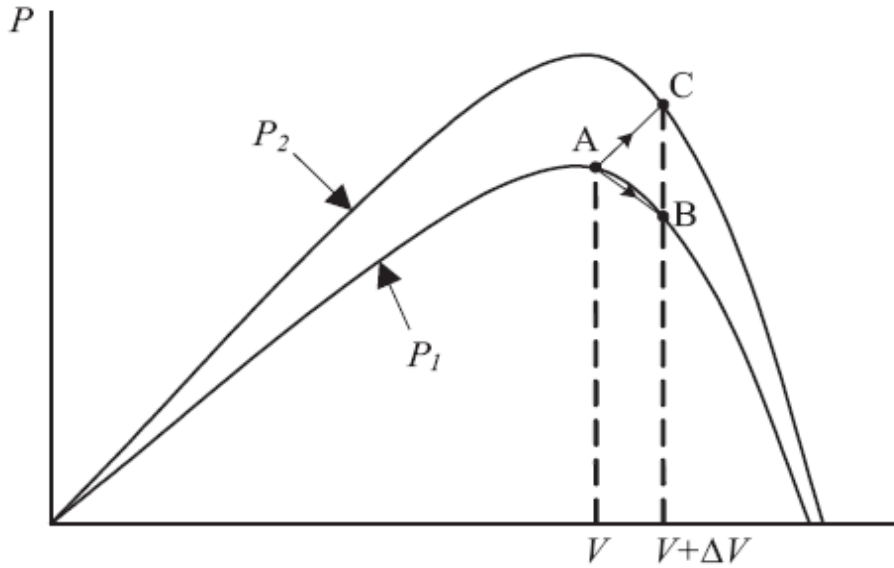
Çizelge 4.1. Hill Climbing ve K&G metotları

Kaygılandırma	Güçteki değişim	Sonraki Kaygılandırma
Pozitif	Artış	Arttır
Pozitif	Azalma	Azalt
Negatif	Artış	Azalt
Negatif	Azalma	Arttır

Maksimum güç noktasına (MGN) ulaşılan kadar işlem tekrarlanmaktadır. Bunun sonrasında sistem MGN çevresinde dalgalanır. Dalgalanma adım büyüklüğünün küçültülmesi ile azaltılabilir. Bununla birlikte adım büyüklüğünün azaltılması MGNİ’nin yavaşlamasına neden olur. Bu tutarsız

durumun çözümü MGN çevresinde küçülen değişken kaygılandırma büyüklüğü kullanmaktır [19].

Hill Climbing ve K&G metotları Şekil 4.2'de gösterildiği gibi hızlı değişen atmosfer koşullarında başarısız olmaktadır. A noktasından başlayarak, eğer atmosfer koşulları sabit kalırsa, PV gerilimi V 'deki ΔV kadar kaygılandırma B noktasına geçirecek ve kaygılandırma güçteki düşüş nedeniyle tersine dönecektir. Bununla birlikte eğer ısıtım artarsa ve bir örnekleme zamanında güç eğrisi P_1 'den P_2 'ye geçerse çalışma noktası A'dan C'ye geçecektir. Bu güçte artış göstermektedir ve kaygılandırma aynı yönde devam edecektir. Bu nedenle çalışma noktası MGN'den uzaklaşacak ve eğer ısıtım düzenli olarak artarsa MGN'den uzaklaşmaya devam edecektir. [2]



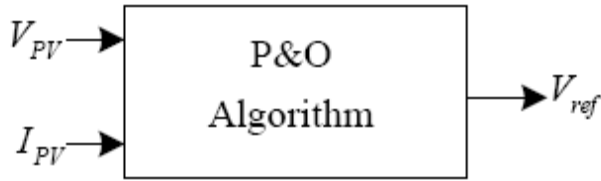
Şekil 4.2. Hill climbing ve K&G MGN'den uzaklaşma hatası

K&G metodu için blok diyagramı Şekil 4.3'te gösterilmiştir.

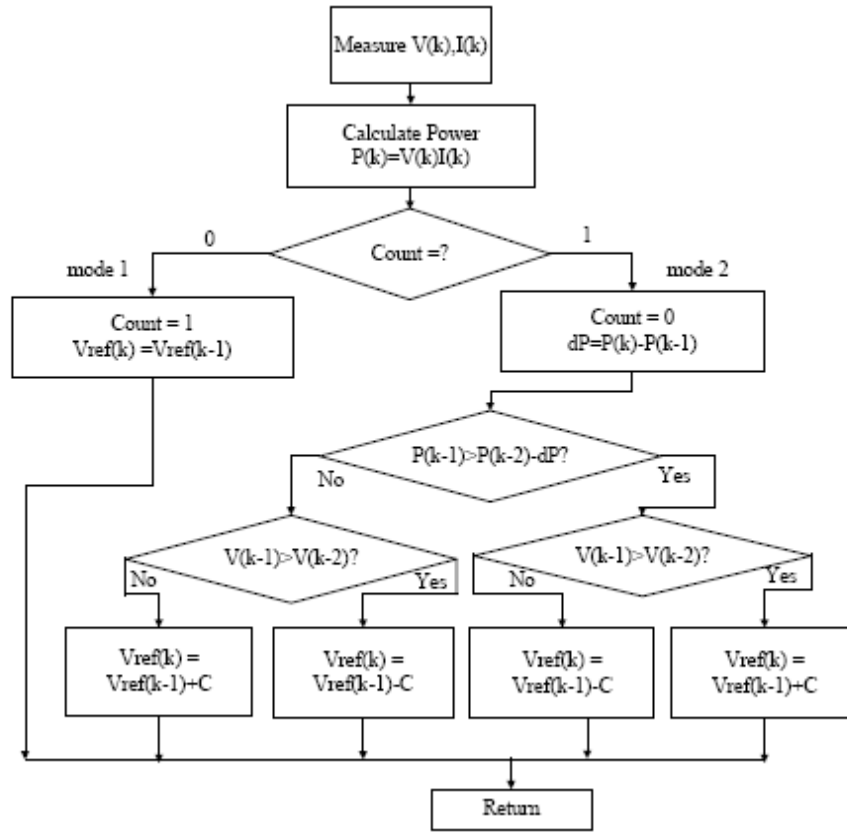
Metotlar hatalı veya yavaş maksimum güç noktası yakalamaya neden olabilir. Bu gibi problemlerin üstesinden gelmek amacıyla, ışıınım ve hava değışimleri nedeniyle oluşan kaygılandırmadaki dalgalandırmaları izole eden Şekil 4.4'teki DK&G akış diyagramı gösterilmiştir.

Bu metot atmosfer koşullarındaki değışim nedeniyle oluşan güç değışimini hesaplamak için her kaygılandırma aşamasına ışıınım değışim tahmini aşaması eklemiştir. [20]

Tüm K&G algoritmaları iki ölçüm gerektirmektedir: PV çıkış gerilimi ve PV çıkış akımı [2]



Şekil 4.3. K&G Blok Diyagramı



Şekil 4.4. DK&G akış diyagramı

4.3. Artan İletkenlik

Artan iletkenlik algoritması kaygılandırma olmaksızın dP/dV hesabıyla PV dizisinin artan iletkenliğini kullanarak K&G algoritmasının kısıtlamalarını aşmaya çalışmaktadır. Bunu maksimum güç noktasındaki $dP/dV=0$ ifadesini kullanarak yapmaktadır. Bu şart ile başlayarak maksimum güç noktasında $dI/dV = -I/V$ olduğunu göstermek mümkündür. Böylece artan iletkenlik maksimum güç noktasının yakalandığını belirleyebilir ve çalışma noktasında uyarıya durdurulabilir. Eğer durum sağlanmamışsa yön dI/dV ve $-I/V$ arasındaki ilişki kullanılarak hesaplanabilir [21].

Artan iletkenlik metodu PV dizi güç eğrisinin eğiminin maksimum güç noktasında sıfır olması, MGN'nin solunda pozitif ve sağında ise negatif olması üzerine kuruludur [19].

$$\begin{aligned} dP/dV &= 0, \text{ maksimum güç noktasında} \\ dP/dV &> 0, \text{ maksimum güç noktasının solunda} \\ dP/dV &< 0, \text{ maksimum güç noktasının sağında} \end{aligned} \quad (4.1)$$

Bu denklemler kullanılarak aşağıdaki çıkarımlarda bulunabilir.

$$\begin{aligned} \Delta I/\Delta V &= -I/V, \text{ maksimum güç noktasında} \\ \Delta I/\Delta V &> -I/V, \text{ maksimum güç noktasının solunda} \\ \Delta I/\Delta V &< -I/V, \text{ maksimum güç noktasının sağında} \end{aligned} \quad (4.2)$$

MGN böylece Şekil 6'daki akış diyagramında görüldüğü gibi anlık iletkenlik (I/V) ile artan iletkenlik ($\Delta I/\Delta V$) karşılaştırılması ile yakalanabilir. V_{ref} PV dizisinin çalışmaya zorlandığı referans gerilimi göstermektedir. Maksimum güç noktasında V_{ref} V_{MPP} 'ye eşittir. MGN'ye bir kez ulaşıldığında PV dizisi çalışmasını atmosferik durumlarda ve MGN'de değişimi gösteren ΔI 'da değişim olasıya kadar bu noktada korumaktadır. Algoritma V_{ref} 'i arttırarak veya azaltarak yeni MGN değerini yakalar.

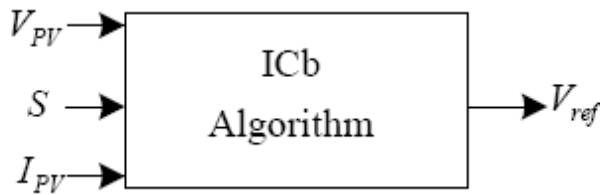
Artım büyüklüğü MGN'nin ne kadar hızlı yakalanacağını belirlemektedir. Hızlı yakalama büyük artım ile sağlanabilir fakat sistem tam olarak MGN'de çalışmayabilir ve MGN çevresinde dalgalanabilir [19].

Artan iletkenlik algoritması yoluyla MGN'ye ne zaman ulaşıldığının bilinmesi teorik olarak mümkündür, böylece kaygılandırmanın ne zaman kesileceği de bilinmektedir. Artan iletkenlik metodu hızlı değişen atmosfer koşullarında iyi performans sağlamaktadır [2].

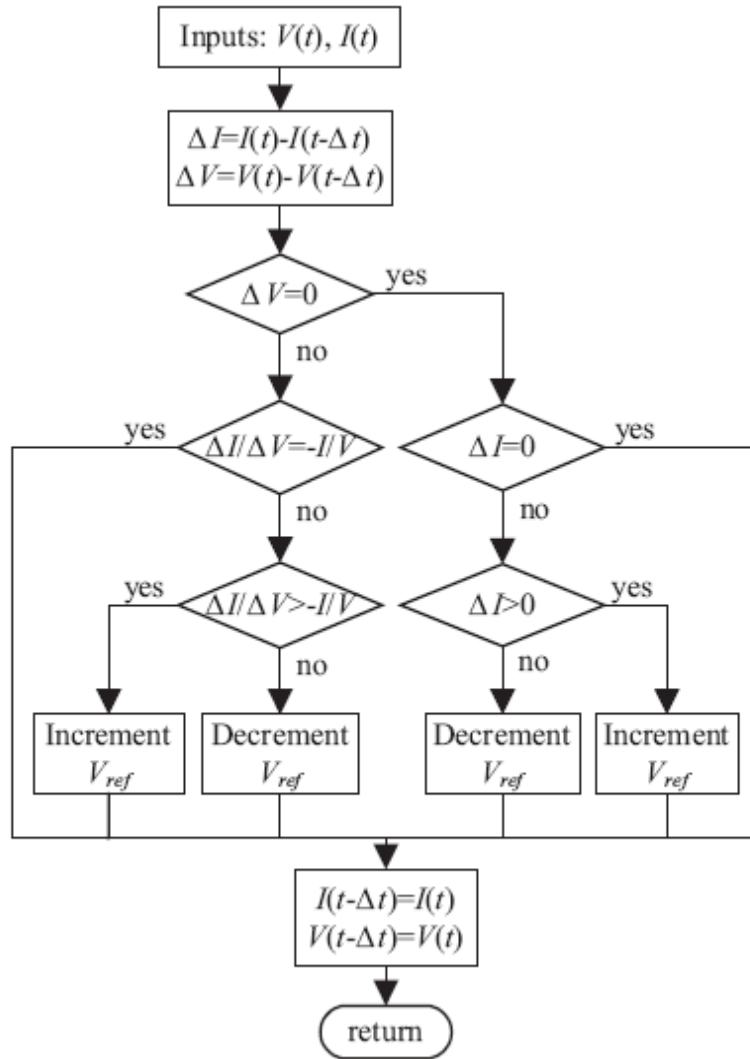
Bu algoritmanın K&G'ya göre avantajı; K&G'nin MGN etrafında dalgalandığı yerde, MGN'ye ulaşıldığının belirlenebilmesidir. Ayrıca artan iletkenlik hızlıca artan ve azalan ışınım şartlarında K&G'den daha yüksek doğrulukla yakalamaktadır. Bu algoritmanın dezavantajı artan karmaşıklığıdır. Bu ölçüm zamanını arttırır ve dizi gerilim ve akımının örnekleme frekansını yavaşlatmaktadır [21].

Anlık PV dizi gerilim ve akımının ölçümü iki algılayıcı gerektirmektedir. IncCond metodu DSP ve mikro kontrolcü kontroller için çok uygundur. Böylece gerilim ve akımın önceki değerleri kolayca tutulabilir ve her döngüde tüm kararlar verilir [19].

İki model MGNI kontrol algoritması(ICb), sabit gerilim ve artan iletkenlik metodlarını birleştirmektedir. Işınım tanımlanmış ışınım derecesinin %30'unun altındayken Sabit gerilim metodu kullanılırken diğer tarafta artan iletkenlik metodu kullanılmaktadır. Böylece bu metod Şekil-4.5'te görüldüğü gibi ek olarak güneş ışınımının (S) ölçümüne ihtiyaç duymaktadır [2].



Şekil 4.5. İki model MGNI (ICb) blok diyagramı



Şekil 4.6. Artan iletkenlik akış diyagramı

4.4. Sabit Gerilim Metodu – Oransal Açık Devre Gerilimi

Sabit gerilim algoritması en basit MGNI kontrol metodudur. PV dizisinin çalışma noktası dizi gerilimi ayarlanarak ve V_{ref} referans geriliminde sabitlenerek maksimum güç noktasına yakın tutulur. V_{ref} değeri PV dizisinin V_{MPP} değerine veya diğer hesaplanan en iyi sabit gerilime eşit ayarlanır [2].

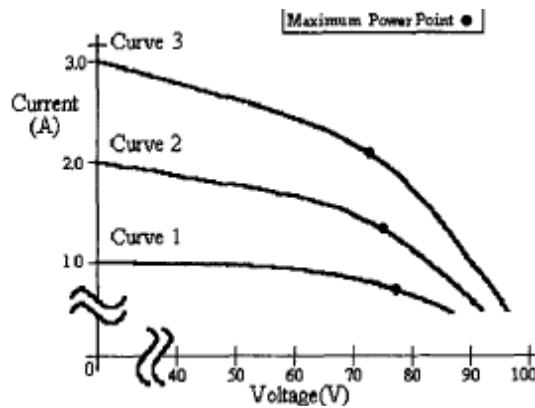
PV dizisinin V_{MPP} ve V_{OC} arasında neredeyse lineer bir ilişki bulunmaktadır.

$$V_{MPP} \approx k_1 * V_{OC} \quad (4.3)$$

Burada k_1 orantı sabitidir. k_1 kullanılan PV dizisinin karakteristiklerine bağımlı olduğundan, farklı ışınım ve sıcaklık derecelerinde belirli PV dizisi için deneysel olarak belirlenmiş V_{MPP} ve V_{OC} değerleriyle önceden hesaplanmış olmalıdır[19]. K_1 faktörünün 0.71 ile 0.78 arasında olduğu söylenilmiştir [1].

V_{MP}/V_{OC} oranı güneş hücre parametrelerine bağımlıdır, fakat genellikle kullanılan değer %76'dır [23].

Bu algoritmada, MGNI Açık devre geriliminin ölçümü amacıyla, PV dizisi akımı anlık olarak sıfıra çeker. Dizin çalışma gerilimi ölçülen bu değer %76'sına çekilir. Bu çalışma noktası belirlenen süre miktarınca kullanılır ve sonrasında döngü tekrarlanır [21].



Şekil 4.7. Artan ışınım altında maksimum güç noktasının konumu

PV panelin düşük ışınlım koşullarında, Sabit gerilim tekniği, K&G metodundan da artan iletkenlik metodundan da daha verimli olduğuna dikkat etmek önemlidir [24].

Bu karakteristik sayesinde sabit gerilim bazen diğer MGNI teknikleriyle birleştirilmektedir.

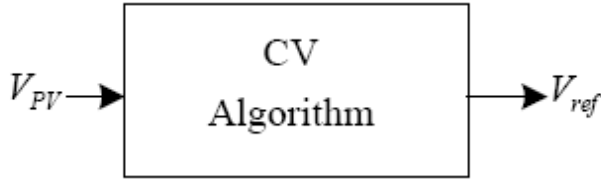
Bu metod, dizi üzerindeki birbirinden ayrı ışınlım ve sıcaklık varyasyonlarının önemsiz ve sabit referans geriliminin doğru MGN için yeterli olduğu yaklaşımını varsaymaktadır. Bu nedenle işlem hiçbir zaman tam olarak MGN'de çalışmayacak ve farklı coğrafik alanlarda farklı bilgi toplanması gerekecektir [2].

Bu algoritmadaki problemi yükün diziye bağlı olmadığı zamanlarda enerjinin israf edilmesidir [21]. Bunu engellemek amacıyla [25] V_{OC} 'nin ölçülebileceği pilot hücre kullanmıştır. Bu pilot hücreler PV dizisinin karakteristik özelliklerini yakın sunacak şekilde dikkatli seçilmelidir [19].

Ayrıca maksimum güç noktası her zaman açık devre geriliminin %76'sında bulunmamaktadır [21].

Eşitlik 4.3 sadece bir yaklaşım olduğundan PV dizisi teknik olarak asla maksimum güç noktasında çalışmayacaktır. PV sistem uygulamasına bağlı olarak bu bazen yeterli olabilir. Orantısal V_{OC} doğru MGNI tekniği olmasa bile, DSP veya mikrokontrolcü gerektirmediğinden uygulanması çok kolay ve basittir. Bununla birlikte [26] numaralı kaynak k_1 'in kısmi bulutlu koşulların varlığında geçerli olmadığına dikkat çekmekte ve PV dizi gerilimine göre k_1 güncellenmesini önermektedir. Bu ise açıkça uygulanma karmaşıklığını arttırmakta ve daha fazla güç kaybına neden olmaktadır [19].

Sabit gerilim metodu hiçbir giriş değerine gereksinim duymamaktadır. Bununla birlikte PV gerilimi V_{PV} 'nin ölçümü Şekil 4.7'de gösterildiği gibi dc/dc çeviricinin'nin doluluk oranının PI düzenleyici tarafından ayarlanması için gereklidir [2].



Şekil 4.8. Sabit gerilim algoritması blok diyagramı [2]

4.5. Oransal Kısa Devre Akımı Metodu

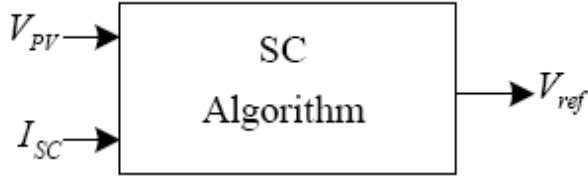
Oransal kısa devre akımı metodu maksimum güç noktasını çalışma akımını (I_{op}) akım kontrollü güç çeviriciye vererek sağlamaktadır. Aslında çeşitli ışınlım derecelerinde (S) maksimum güç çıkışı için gerekli uygun çalışma akımı (I_{op}) kısa devre akımı (I_{sc}) ile orantılıdır.

$$I_{MPP}(S) \approx k_2 * I_{SC}(S) \quad (4.4)$$

Burada k_2 orantı sabitidir. Eşitlik 4.4 I_{sc} 'nin ölçülmesiyle I_{op} 'nin bulunabileceğini göstermektedir. Sıcaklık 0°C ile 60°C arasında değişse bile I_{op} ile I_{sc} arasındaki ilişki halen orantılıdır [2].

K_2 orantı sabitinin 0,78 ile 0,92 arasında olduğu bulunmuştur. Metodun doğruluğu ve yakalama verimliliği k_2 değerinin doğruluğuna ve kısa devre akımının periyodik ölçümüne bağlıdır [1].

Bu nedenle bu kontrol algoritması I_{sc} akımının ölçümüne ihtiyaç duymaktadır [2].



Şekil 4.9. Oransal Kısa Devre Akımı Metodu Blok Diyagramı [2]

Uygulama sırasında I_{sc} ölçümü problemlidir. PV dizisini periyodik olarak kısa devre yapmak için ek anahtar genellikle güç çeviriciye eklenmelidir. Böylece I_{sc} akım sensörü kullanılarak ölçülür. Bu bileşen sayısını ve maliyeti arttırmaktadır. [27] numaralı kaynak, çeviricideki anahtar PV dizisini kısa devre yapmak için kendiliğinden kullanılabileceğinden, boost çevirici kullanmıştır. [32]. kaynakta belirtildiği gibi, I_{sc} hesaplanmasında sadece çıkış gücü azalmaz, bununla birlikte maksimum güç noktası asla kesin olarak yakalanamaz. [19].

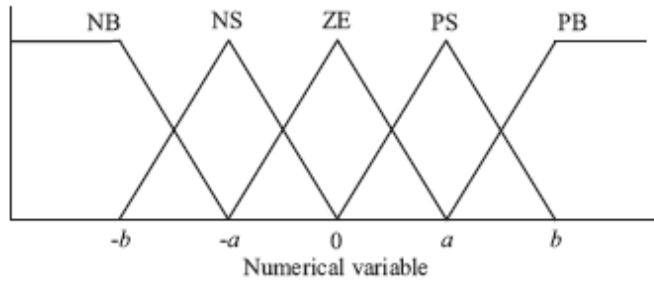
4.6. Bulanık Mantık Kontrol

Son yıllarda MGNI için bulanık mantık kontrolünde mikrokontrolcü kullanımı popüler hale gelmiştir. Bulanık mantık kontrolcülerinin, kesin olmayan girişlerle çalışma, kesin olmayan matematiksel modele ihtiyaç duymama ve lineer olmayan durumlarda çalışmada avantajları bulunmaktadır [19].

Bulanık mantık kontrolcüler, kompleks matematiksel modellerle tanımlanan kontrol problemlerine çözüm sağlamak için kullanılırlar. Bulanık sistemlerin dizaynı nispeten kolaydır ve makul sonuçlar üretebilmektedirler. [28]

Bulanık mantık kontrol genellikle 3 aşama içermektedir: Bulanıklaştırma, kural tabanlı tablo arama ve berraklaştırma. Bulanıklaştırma sırasında,

nümerik giriş değişkenleri Şekil 7'dekine benzer üyelik fonksiyonlarına bağlı olarak sözel değişkenlere çevrilir. Bu örnekte, 6 bulanık derecesi kullanılmıştır: NB(Negatif büyük), NS(Negatif küçük), ZE(Sıfır), PS(Pozitif küçük), PB(Pozitif büyük).



Şekil 4.10. Bulanık mantık kontrolcü giriş ve çıkış için üyelik fonksiyonu

Şekil 4.10'da a ve b nümerik değişkenin değerinin aralığı üzerine kuruludur. Üyelik fonksiyonu, belirli bulanık derecelere daha fazla önem vermek için bazen daha az simetri yapabilir [19].

MGNU bulanık mantık kontrolcü girişleri genellikle hata E ve hatadaki değişim ΔE 'dir. dP/dV maksimum güç noktası sıfır olduğundan, aşağıdaki yaklaşım kullanılabilir.

$$E(n) = \frac{P(n) - P(n-1)}{V(n) - V(n-1)} \quad (4.5)$$

$$\Delta E(n) = E(n) - E(n-1) \quad (4.6)$$

Eşitlik 4.6 çok sık kullanılır. E ve ΔE herhangi bir zamanda hesaplanır ve sözel değişkenlere çevrilir, bulanık mantık kontrolcü Çizelge 4.2'deki gibi bir kural tabanlı tablodan bulunabilecek güç çeviricinin doluluk oranındaki değişim ΔD 'yi çıkış olarak verir.

Çizelge 4.2. Bulanık kural tabanlı tablo

ΔE E	NB	NS	ZE	PS	PB
NB	ZE	ZE	NB	NB	NB
NS	ZE	ZE	NS	NS	NS
ZE	NS	ZE	ZE	ZE	PS
PS	PS	PS	PS	ZE	ZE
PB	PB	PB	PB	ZE	ZE

Sözel değişkenler, kullanıcı bilgisi ve kullanılan güç çeviriciye dayanarak E ve ΔE 'nin değişik kombinasyonlarına göre ΔD 'yi saptamaktadır. Çizelge 4.2 boost çeviriciden yola çıkılarak elde edilmiştir. Örnek olarak eğer çalışma noktası maksimum güç noktasının uzak solundaysa E PB'dir ve ΔE ZE'dir, doluluk oranının büyük miktarda arttırılmalıdır, dolayısıyla ΔD maksimum güç noktasına erişmek için PB olmalıdır.

Berraklaştırma aşamasında, bulanık mantık kontrolcü çıkışı yine Şekil 4.10'daki üyelik fonksiyonları kullanılarak sözel değişkenlerden nümerik değişkenlere çevrilir. Bu güç çeviriciyi maksimum güç noktasına kontrol edecek analog sinyali sağlar.

MGNI bulanık mantık kontrolün faydaları kullanıcının bilgisi, kontrol mühendisinin doğru hata hesaplamasına ve kural tabanlı tabloya çok bağımlı olsa da değişen atmosferik koşullarında iyi çalışmaktadır [19].

[30] numaralı kaynakta, sık sık üyelik fonksiyonlarını ve kural tabanı tablosunu ayarlayan, dolayısıyla en iyi performansı yakalayan uyabilen bulanık mantık kontrol önerilmiştir.[29] numaralı kaynağın deneysel sonuçları

maksimum güç noktasına hızlı yakınsama ve maksimum güç noktası çevresinde minimum dalgalanma göstermiştir [19].

5. SİMÜLASYON YÖNTEMİ VE SONUÇLARI

5.1. Simülasyon Ortamı

Daha önceki uygulamalar incelenmiş ve yapılan araştırmalarda matematiksel modelleme yazılımlarının (MATLAB, Simulink, Labview[32] vb.) kullanıldığı görülmüştür. Matematiksel modelleme yazılımlarının kullanıldığı simülasyon çalışmalarında simülasyon yazılımının çalıştırılacak bilgisayarda yüklü olması gerekliliği, lisans sorunları, modelleme yazılımlarında çıkabilecek sorunlar, işletim sistemi bağımlılığı vb. gibi nedenlerle yazılımın geliştirilmesi düşünülmüştür. Yazılım geliştirme ortamı olarak ücretsiz, açık kaynak ve nesne tabanlı yazılım olan JAVA dili kullanılmıştır.

Java ortamında geliştirilen yazılımın çalıştırılabilmesi için Java Runtime Environment 6.0 bileşeninin yüklü olması gerekmektedir. Günümüzde oluşturulan web sayfalarının birçoğu Java programlama dili üzerinde geliştirilmektedir. Bu nedenle çoğu bilgisayarda bu yazılım yüklü olarak bulunmaktadır. Ayrıca java internet üzerinden <http://java.sun.com> adresinden ücretsiz olarak temin edilebilir.

Java yazılımının tercih edilmesinde en önemli nedenlerden biri işletim sisteminden bağımsız olmasıdır. Java Runtime Environment yazılımının yüklü olduğu her bilgisayarda tez sonucu sunulan program çalıştırılabilir.

Java ortamının kullanılmasının bir diğer nedeni, Java'nın modüler, geliştirilebilir ve nesne tabanlı olmasıdır. Yazılım geliştirilirken modüler yapıda geliştirilmiş ve böylece ek modüllerin eklenmesine, değişiklik yapılmasına ve geliştirilmesine olanak sağlanmıştır. Teknolojik ve bilimsel gelişmeler ışığında güneş pillerinde bulunacak her türlü yenilik, değişik MGNI

yöntemleri, yük eklenmesi vb. gibi eklemeler ve değişiklik program üzerinde yapılabilir.

Java ortamının bir diğer getirisi ise, mevcut olan çeşitli uygulama UPA'ları (Örnek olarak bulanık mantık vb.) kullanılarak daha karmaşık yapıdaki kontrol yöntemler kolaylıkla eklenebilir.

Yazılım geliştirmesi Eclipse IDE kullanılarak yapılmıştır. Eclipse yazılımı birçok programlama dilini destekleyen yazılım geliştirme ortamıdır. Özellikle Java ortamında yazılım geliştirmede en çok kullanılan yazılım geliştirme ortamlarından biridir.

5.2. Geliştirilen Program

Java ortamında Eclipse üzerinde geliştirilen yazılım jar uzantılı çalıştırılabilir dosya olarak dışa aktarılmıştır. Bu dosya ile program çalıştırılmaktadır.

Programın çalıştırıldığında Şekil-1.1'de görülen güneş piline ait akım-gerilim ve güç-gerilim grafiklerinin çiziminin yapıldığı ekran açılmaktadır. Başlangıç ekranı olarak bu ekran tercih edilmiştir.

Program 4 ana menüden oluşmaktadır:

1. *Karakteristik*: Güneş piline ait karakteristik eğrilerin, maksimum güç değerinin elde edildiği noktaya ait bilgilerin ve yük değerine bağlı olarak akım ve güç değerlerinin bulunabildiği bölümün yer aldığı menüdür. Menü içeriğinde "Çalışma Noktası Yakala" ve "Karakteristik Bilgileri" alt menüleri yer almaktadır.
2. *Maksimum güç*: Güneş pilinin maksimum güç noktasında çalıştırılmasına yönelik yöntemlerin simülasyonunun yapıldığı menüdür. Menü içeriğinde "Maksimum Güç Yakala" alt menüsü yer almaktadır.

3. *Yardım*: Programın versiyon bilgisi, tez ile ilgili bilgiler ve teşekkür bilgisinin yer aldığı menü bölümüdür. Menü içeriğinde “Hakkında” ve “Teşekkür” alt menüleri yer almaktadır.

4. *Çıkış*: Programdan çıkış için kullanılacak menü bölümüdür.

Yazılımda kullanılan alanların hepsi değiştirilebilir alanlardır. Katalog bilgileri, ideallik faktörü ve seri iç direnç değerleri bilinen diğer güneş pili modülleri için de kullanılabilir.

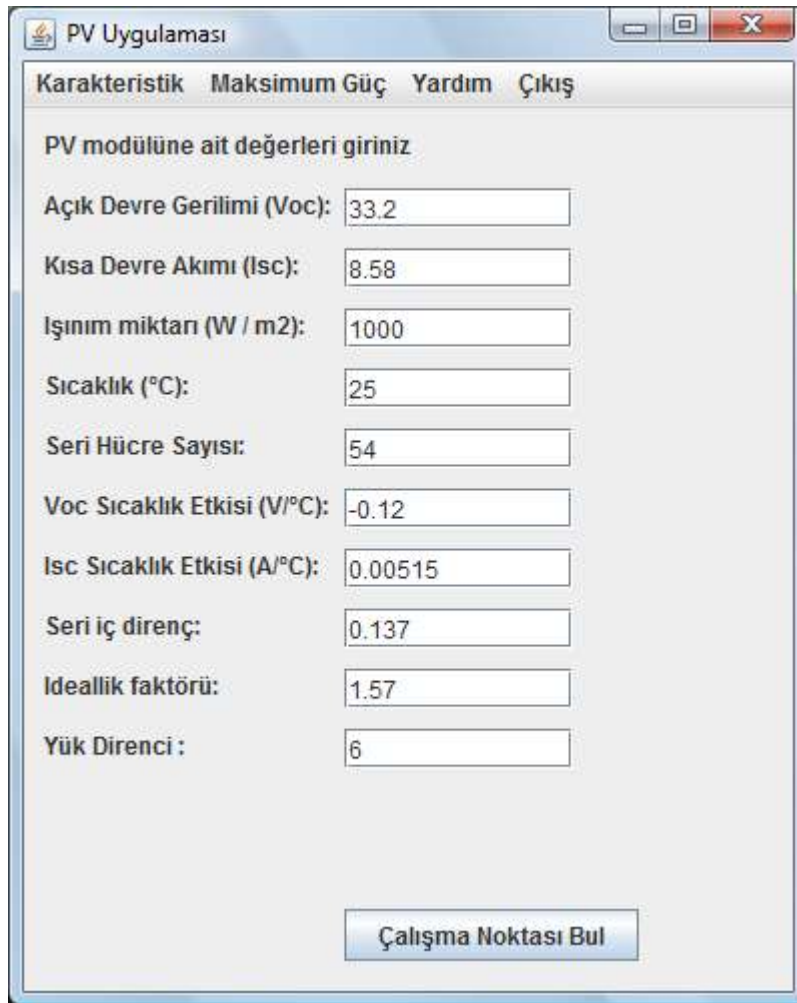
5.3. Karakteristik

Güneş piline ait karakteristik bilgiler elde edilebilmektedir.

5.3.1. Çalışma noktası yakala

Bağlanılacak bir yük değerine göre çalışma noktası hesaplanmaktadır. Ekranda katalog’dan elde edilebilecek verilerin girişi istenmektedir. Geçerli olarak tanımlanmış veriler Kyocera firmasına ait 210 W’lık KD-210 GX-LPU modülüne ait bilgilerdir.

Şekil 5.1’de gösterilen çalışma noktası yakala ekranındaki açık devre gerilimi, kısa devre akımı, seri hücre sayısı, Voc sıcaklık etkisi, Isc sıcaklık etkisi değerleri katalog’dan elde edilen verilerdir. Işınım miktarı ve sıcaklık değerleri çalışma anındaki değerlerdir. Seri iç direnç ve ideallik faktörü değerleri güneş piline ve ışınım, sıcaklığa bağlı olarak değişebilecek değerlerdir. Bu değerler sözü edilen güneş pili modülü için hesaplanmış ve seri iç direnç değeri 0,137 ve ideallik faktörü 1,57 alınmıştır. R_s seri iç direnç değeri sıcaklık ve ışınım gibi faktörlerden etkilenmektedir. Bu nedenle R_s direnci sabit kabul edilerek simülasyon ortamına eklenmiştir. Yük direnci olarak geçerli verilen değer 6 Ω ’dur.



PV Uygulaması

Karakteristik Maksimum Güç Yardım Çıkış

PV modülüne ait değerleri giriniz

Açık Devre Gerilimi (Voc): 33.2

Kısa Devre Akımı (Isc): 8.58

Işınım miktarı (W / m2): 1000

Sıcaklık (°C): 25

Seri Hücre Sayısı: 54

Voc Sıcaklık Etkisi (V/°C): -0.12

Isc Sıcaklık Etkisi (A/°C): 0.00515

Seri iç direnç: 0.137

İdeallik faktörü: 1.57

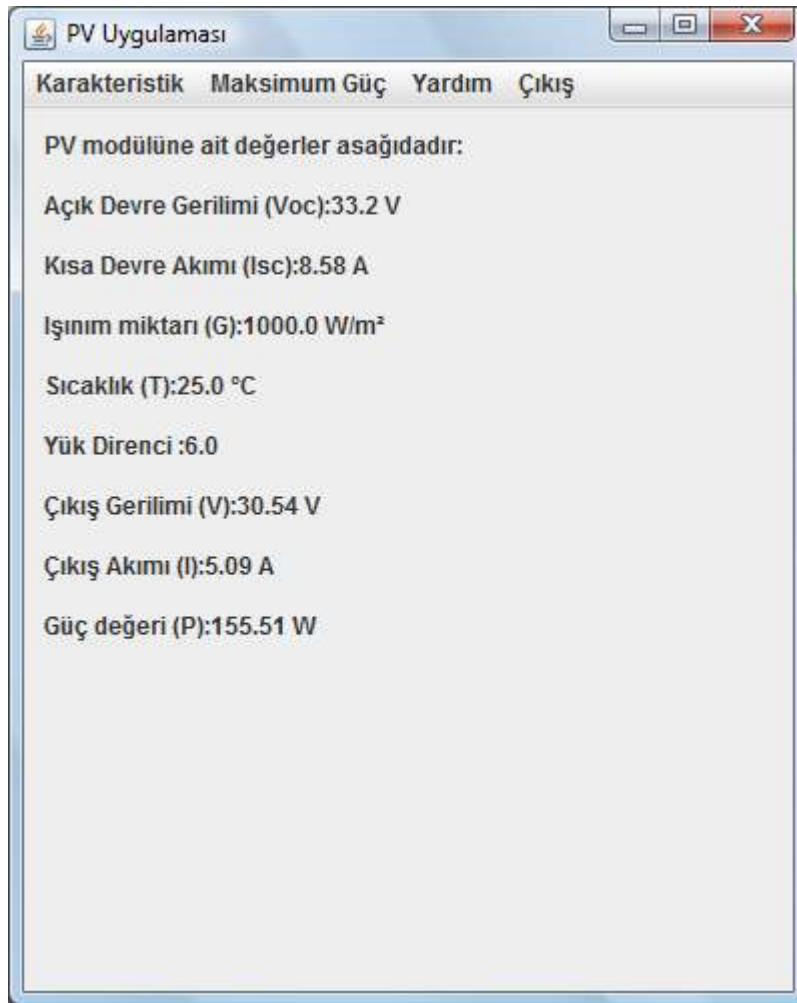
Yük Direnci : 6

Çalışma Noktası Bul

Şekil 5.1. Çalışma noktası yakala giriş ekranı

Çalışma noktası bul butonuna basıldığı belirtilen yükte çalışmada elde edilecek gerilim, akım ve güç değerleri sonuç olarak dönmektedir.

Bu yazılım modülü sayesinde kullanılacak yük ile elde edilebilecek güç miktarları bulunabilmektedir.



Şekil 5.2. Çalışma noktası yakala sonuç ekranı

5.3.2. Karakteristik bilgileri

Güneş piline ait karakteristik eğrilerin sonucunun alındığı menüdür. Şekil 5.3'te görüldüğü gibi çalışma noktası yakala alt menüsünde bulunan giriş değerleri yük direnci dışında bulunmaktadır.

PV Uygulaması

Karakteristik Maksimum Güç Yardım Çıkış

PV modülüne ait değerleri giriniz

Açık Devre Gerilimi (Voc): 33.2

Kısa Devre Akımı (Isc): 8.58

Işınım miktarı (W / m2): 1000

Sıcaklık (°C): 25

Seri Hücre Sayısı: 54

Voc Sıcaklık Etkisi (V/°C): -0.12

Isc Sıcaklık Etkisi (A/°C): 0.00515

İdeallik faktörü: 1.57

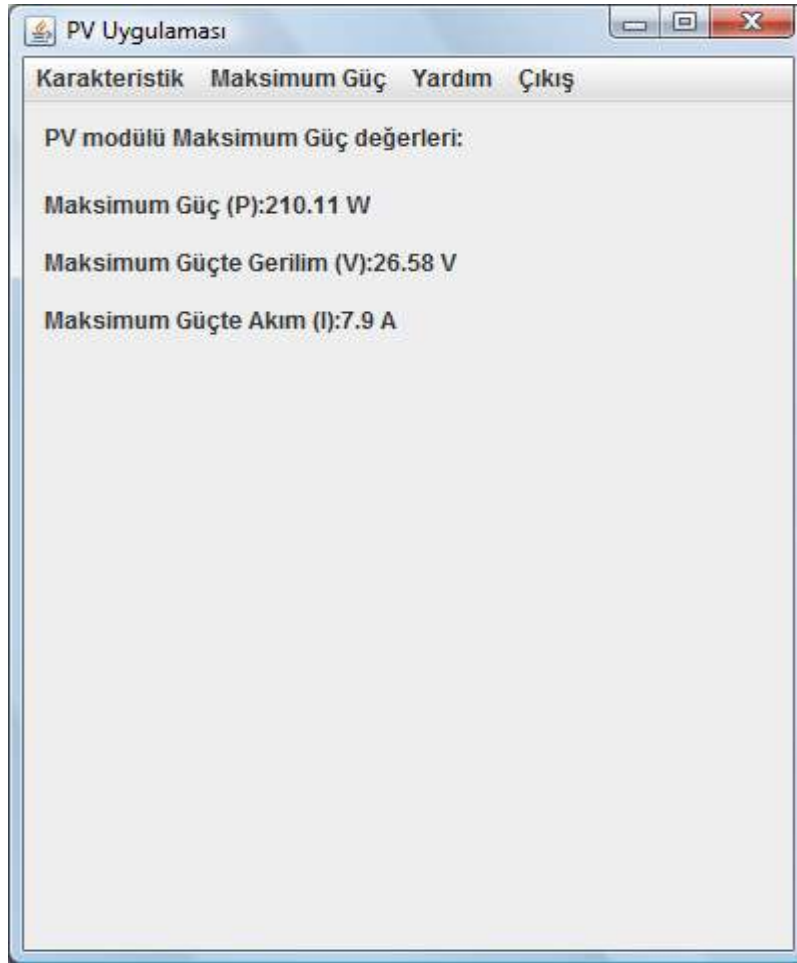
Seri iç direnç: 0.137

Karakteristik Göster

Şekil 5.3. Karakteristik bilgileri giriş ekranı

Belirtilen güneş piline ait girilen geçerli değerlerin birçoğu katalog bilgilerinden alındığı yukarıda belirtilmişti. İç direnç hesabı Eşitlik 2.8 kullanılarak elde edilebilir. İdeallik faktörü değeri katalog'da verilen güç-gerilim ve akım-gerilim grafiklerine uygun olacak seçilecektir. İdeallik faktörü değeri 1 ile 2 arasında olmaktadır. 1 değeri ideal güneş pilini gösterirken 2 değeri malzeme ve eklem kalitesinin düşüklüğünü göstermektedir. Yapılan çalışma sonucunda ideallik faktörü değeri 1,57 olarak alınmıştır.

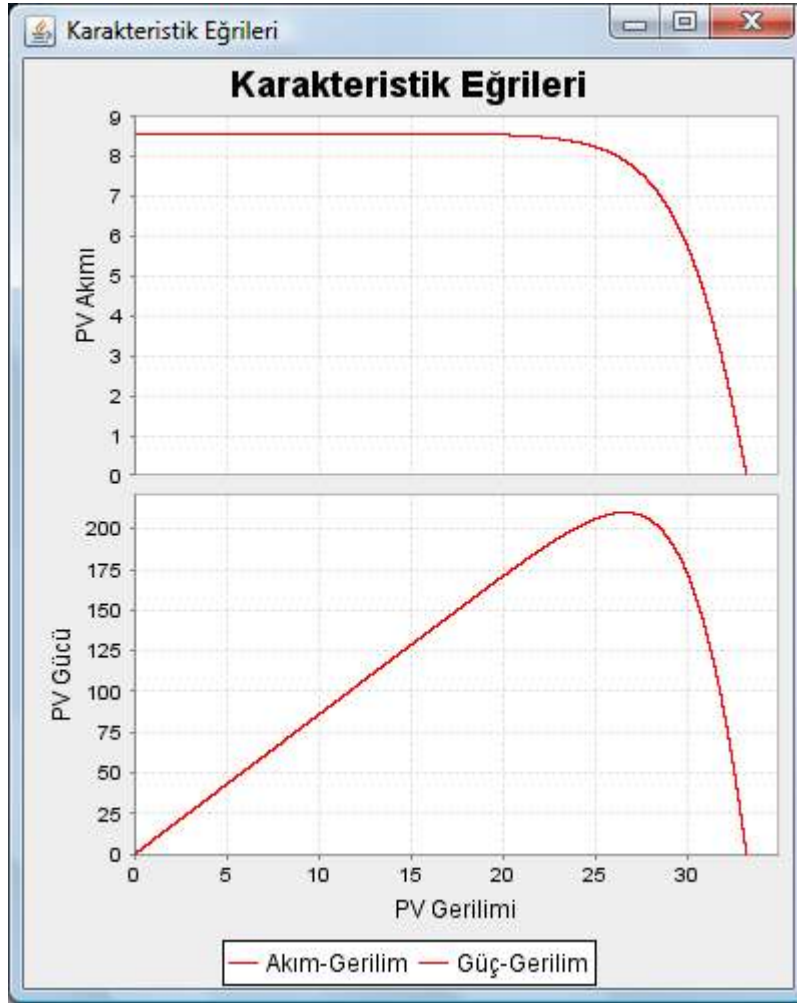
Karakteristik göster butonuna basıldığında; verilen katalog bilgilerine uygun olarak maksimum güç noktasına ait bilgilerin gösterildiği ekran görüntüsü ve gerilim, akım ve güce ait grafikler gösterilmektedir (Şekil-5.4, Şekil-5.5).



Şekil 5.4. Karakteristik bilgileri sonuç ekranı

Karakteristik bilgilerinin elde edilmesinde ve çiziminde eşitlik 2.7 kullanılmıştır.

Katalog bilgilerinde gösterilen maksimum güç değeri 210 (± 5) W'tır. Simülasyon sonucunda 210.11W güç elde edilmiştir (Şekil 5.5).



Şekil 5.5. Akım gerilim ve güç gerilim grafiği

Katalog üzerinde yer alan 800 W/m^2 / 50°C değerleri için verilen güç değeri 149 W 'tır. Simülasyon bu sonuçlar ile tekrarlandığında maksimum güç değeri 146.61 W bulunmaktadır. Bu değer farklılıkları R_s seri iç direncinin sabit kabul edilmesi, R_{sh} paralel iç direncin göz ardı edilmesi ve ideallik faktöründen kaynaklanmaktadır.

Bulunan değerlere bakıldığında simülasyon sonuçlarının gerçeğe yakın olduğu görülmektedir.

Ayrıca yapılan simülasyondaki tüm adımlarda elde edilen gerilim, akım, güç ve o noktadaki direnç değerleri VIP.csv dosyası olarak dışarı verilmektedir.

5.4. Maksimum güç yakala

Güneş pilinin maksimum güç noktasında çalıştırılmasına yönelik olarak hazırlanmış ekrandır. Ekran açılışında daha önceki bölümlerde olduğu gibi güneş piline ait karakteristik değerlerin girilmesinin istendiği Şekil 5.6'da görülen maksimum güç noktası yakala giriş ekranı gelmektedir.

Giriş ekranında; karakteristik bilgileri ekranında bulunan açık devre gerilimi, kısa devre akımı, seri hücre sayısı, V_{oc} ve I_{sc} sıcaklık etkisi, ideallik faktörü ve seri iç direnç değerlerinin yanı sıra yük direnci ve kullanılacak yöntem alanları yer almaktadır. Yük direnci alanına güneş piline bağlanacak olan yük direnci değeri girilecektir. Kullanılacak yöntem alanı radyo buton kullanılarak oluşturulmuştur. Burada bulunan yöntemlerden biri seçilerek yöntem seçilmiş olacaktır. Varsayılan olarak yöntem K&G'dir. Kullanılacak yöntemin altında bulunan orantı sabiti alanı kullanılacak yöntem olarak Oransal V_{oc} seçildiğinde kullanılan bir alandır. Bu alan varsayılan olarak, kullanılan güneş pili dizisine uygun olarak en uygun katsayı seçilmelidir. Varsayılan değerler Kyocera firmasına ait 210 W'lık KD-210 GX-LPU modülü için hesaplanarak giriş yapılmıştır.

Bu alanların dışında hesaplamalar için gerekli olan ışıınım ve sıcaklık değerleri, sabit olmadığından ve değişiklik göstereceğinden aynı dizinde bulunacak harici bir dosya aracılığı ile alınmıştır. Bu dosyalar;

- Işıınım değeri için G.csv
- Sıcaklık değeri için T.csv'dir.

Işınım ve sıcaklık değerleri için oluşturulacak csv dosyaları her bir satıra o adımda bulunacak sıcaklık ve ışınım değerleri yazılarak elde edilir. Dosya formatının csv seçilmesinin nedeni MS Excel programında da değerler girilerek kaydedilebilir olmasıdır. Ayrıca bu tip dosyalar metin dosyası editörleri kullanılarak değiştirilebilirler.

Şekil 5.6. Maksimum güç yakala giriş ekranı

5.4.1. Dönüştürücünün belirlenmesi

Dönüştürücülerin farklarının belirlenmesi amacıyla, güneş pili tarafından görülen yük değerleri incelenecektir.

3.5 Yük eşleştirme bölümünde gösterilen yöntem kullanılarak buck ve boost dönüştürücü için güneş pili tarafından görülen yük değerleri aşağıdaki şekilde bulunmuştur:

Boost dönüştürücü için;

Akım ve gerilim denklemleri aşağıdaki gibidir:

$$V_0 = \frac{1}{1-D} V_{in} \quad (5.1)$$

$$I_0 = (1-D) I_{in} \quad (5.2)$$

Güneş pili tarafından görülen yük değeri ise:

$$R_{in} = \frac{V_s}{I_s} = \frac{1}{1-D} \frac{V_0}{I_0} = \frac{1}{(1-D)^2} R_{LOAD} \quad (5.3)$$

Buck dönüştürücü için ise;

Akım ve gerilim denklemleri aşağıdaki gibidir:

$$V_0 = D * V_{in} \quad (5.4)$$

$$I_0 = \frac{1}{D} I_{in} \quad (5.5)$$

Güneş pili tarafından görülen yük değeri ise:

$$R_{in} = \frac{V_s}{I_s} = D^2 \frac{V_0}{I_0} = D^2 R_{LOAD} \quad (5.6)$$

Dönüştürücülerin karakteristiklerine göre doluluk oranı değeri 0 ile 1 değerleri arasında değişmektedir. Buna göre eşitlik 5.3 ve eşitlik 5.6 kullanılarak 6Ω'luk direnç değerine göre güneş pili tarafından görülecek direnç değerleri aşağıdaki tabloda gösterilmiştir.

Çizelge 5.1. Buck ve boost dönüştürü ile görülen direnç değeri

Yük direnci (Ω)	Doluluk oranı	Buck dönüştürücü Görülen direnç (Ω)	Boost dönüştürücü Görülen direnç (Ω)
10,00	0,00	0,00	10,00
10,00	0,10	0,10	12,34
10,00	0,40	1,60	27,77
10,00	0,70	4,90	1 111,11
10,00	1,00	10,00	Çok büyük

Çizelge 5.1'de görülen değerlere göre yük değeri; buck dönüştürücü ile yük direncinin altında, boost dönüştürücü ile ise yük direncinin üzerinde olmaktadır.

Bölüm 2.5'te belirtilen R_{opt} direnç değeri düşünüldüğünde;

Buck dönüştürücü kullanılması için $R_{opt} < R_{LOAD}$,

Boost dönüştürücü kullanılması için ise $R_{opt} > R_{LOAD}$ olmalıdır.

Yapılan simülasyonun geniş alanda kullanılabilmesi amacıyla simülasyonda Cuk dönüştürücü kullanılmıştır. Cuk dönüştürücü kullanıldığında güneş pili tarafından görülen yük direnci değeri eşitlik 3.7 kullanılarak hesaplandığında 0 ile sonsuz değerleri arasında bulunabilmektedir.

5.4.2. Kullanılabilecek Yöntemler

Maksimum güç yakala giriş ekranından görüldüğü gibi simülasyonda 3 adet MGNI yöntemi kullanılabilmektedir:

- K&G
- Oransal V_{oc}
- Yüke direk bağlı

K&G

Şekil 4.1'de gösterildiği gücün ve gerilimin değerlerine göre yöntem çalışmaktadır. Bu yöntemde maksimum güç noktasına ulaşılmaya çalışılmaktadır.

Oransal V_{oc}

Yöntem belirlenen k değeri ile eşitlik 4.3'deki çalışma geriliminin yakalanması üzerine kuruludur.

Yüke direk bağlı

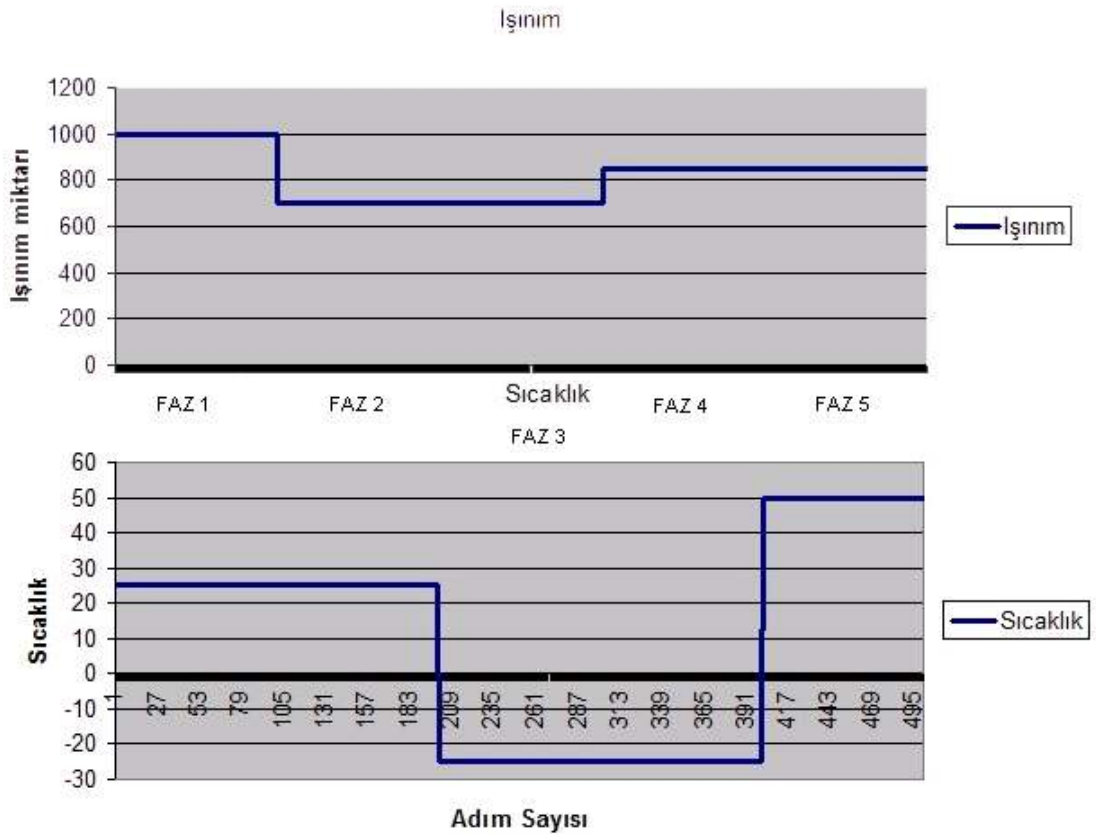
Yöntemde bağlanılan yük değerine göre güneş pilinin çalışma noktası bulunmaktadır.

5.4.3. Simülasyonun Çalışması

Maksimum güç yakala butonuna basıldığında giriş ekranında bulunan değerler atanmaktadır. Bu işlem sonrasında ışıınım ve sıcaklık değerleri dizilere aktarılmaktadır. Işıınım ve sıcaklık değerleri dizilere aktarılırken simülasyonun çalıştırılacağı adım sayısı da belirlenmektedir. Simülasyonda kullanılacak adım sayısı ışıınım değerlerinin bulunduğu G.csv dosyasındaki

adım sayısı ile belirlenmektedir. Bu aşamada G.csv ve T.csv dosyalarında bulunan adım sayısının eşit olmasına dikkat edilmelidir.

Şekil 5.7 simülasyon ortamında kullanılan adıma göre ışıınım ve sıcaklık değerlerini göstermektedir. Faz 1’de ışıınım ve sıcaklık değerleri 1000 W/m^2 25°C ’dir. Faz 2’ye geçişte ışıınım değeri 700 W/m^2 ’ye düşürülmüş ve sıcaklık değıştirilmemiştir. Faz 3’te bu kez ışıınım değeri sabit tutulmuş ve sıcaklık değeri -25°C ’ye düşürülmüştür. Faz 4’ten itibaren yapılan düşüşler yükseltilmeye başlanmış, ışıınım arttırılmış ve faz 5’te sıcaklık 50°C ’ye yükseltılmıştir.



Şekil 5.7. Adıma göre kullanılan ışıınım ve sıcaklık değeri

Belirlenen bu ışınlm ve sıcaklık deęerleri alındıktan sonra simölasyonun adımları başlamaktadır.

Başlangıçta dönüştürücü doluluk oranı 0,5 olarak belirlenmiştir. Bu deęerin seçilmesinin nedeni başlangıçta yükün direk baęlı olduęu gibi davranmasını sağlamaktır.

Her adımda belirlenen ışınlm, sıcaklık ve sıcaklık etkisi deęerlerine baęlı olan kısa devre akımı ve açık devre gerilimleri tekrar hesaplanmaktadır. Kısa devre akımı ve açık devre gerilimi hesaplanmasından sonra dönüştürücü doluluk oranı ve direnç deęerleri ile güneş pilinin çalışma noktası hesaplanmaktadır. Bu noktanın belirlenmesi sonucunda güneş pili tarafından dönüştürücüye ve dolayısıyla yüke aktarılan gerilim ve akım deęerleri bulunmaktadır.

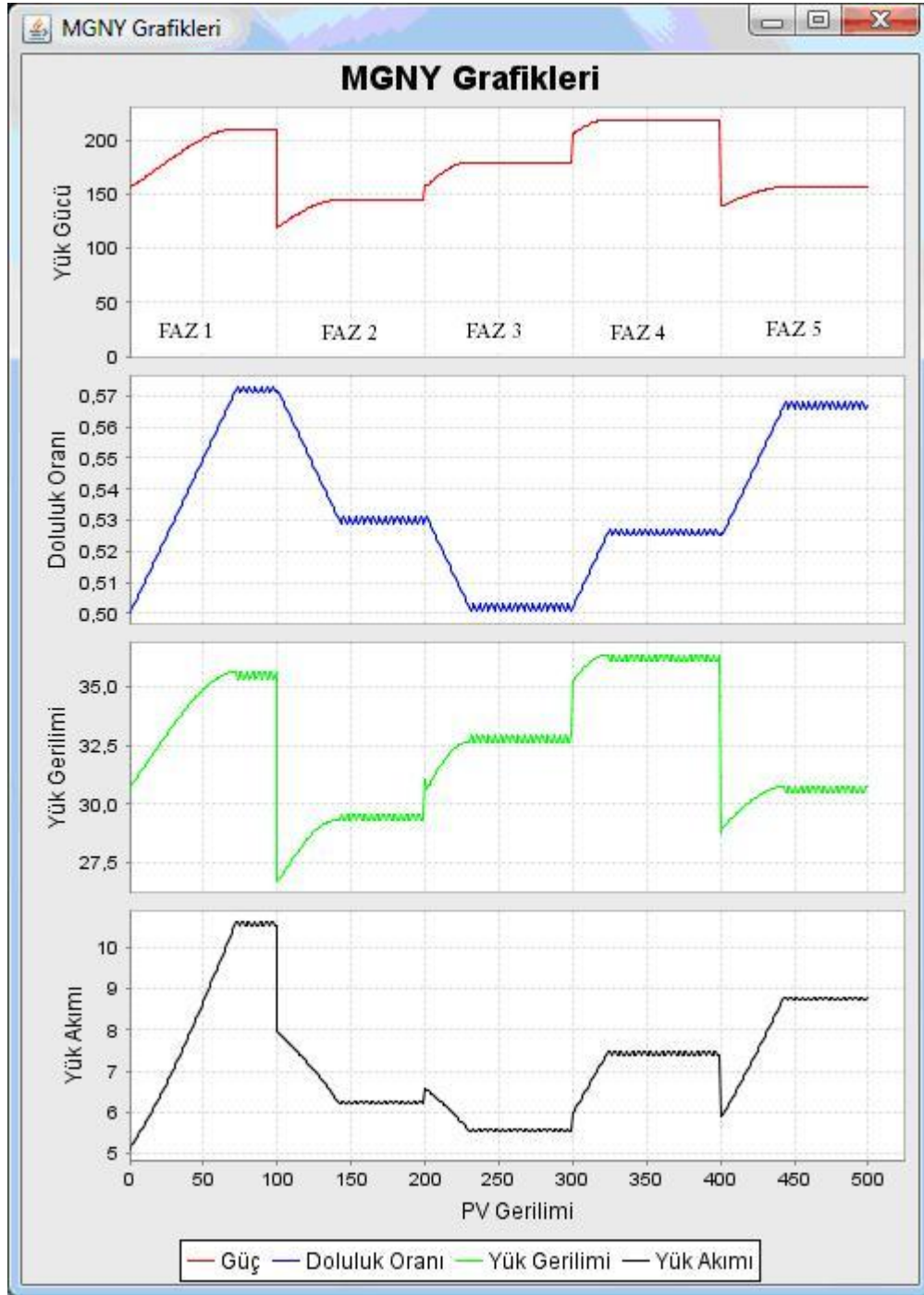
Elde edilen akım ve gerilim deęerleri kullanılarak seçilen yöntemle baęlı olarak dönüştürücü doluluk oranı belirlenmektedir.

Elde edilen akım, gerilim, güç ve dönüştürücü doluluk oranı kaydedilerek bu deęerler ile simölasyon sonuçları Şekil 5.8, Şekil 5.9, Şekil 5.10'daki gibi grafiksel olarak gösterilmiştir. Yük direnci olarak 6 Ω 'luk direnç kullanılmıştır.

Şekil 5.10'da görüldüğü gibi, yük direk baęlandığında faz 1'de 155W civarında güç elde edilmiştir. Bununla birlikte K&G ile orantısal V_{oc} metodlarında 155W ile başlayan güç deęeri yükselmiştir. Başlangıçta 155W güç elde edilmesinin nedeni başlangıçta doluluk oranının 0,5 alınmasıdır. K&G metodunda 70 adım sonrasında 210W güce ulaşılmışken, oransal V_{oc} yönteminde 80 adım sonrasında oransal deęer yakalanmış ve 210W elde edilmiştir. Işınlımın düşmesiyle faz 2'de MGNI yöntemleri benzer tepkiler vermiş ve yakın adım sayısı ile 150W'ın biraz altında bir güç elde etmişlerdir.

Direk baėlı sistemde 140 W g retebilmiřtir. Faz 3 K&G ile oransal V_{oc} arasındaki farklılıėı daha net gstermektedir. Faz 3'te oransal V_{oc} yntemi 150W deėerinde g saėlarken, K&G metodu 20 adım kadar sonrasında maksimum g yakalamıř ve 180W civarında g retmiřtir. Faz 3'te yukarıda belirtildiėi gibi sıcaklık deėeri 25 °C'den -25°C'ye dřrlmřtir.

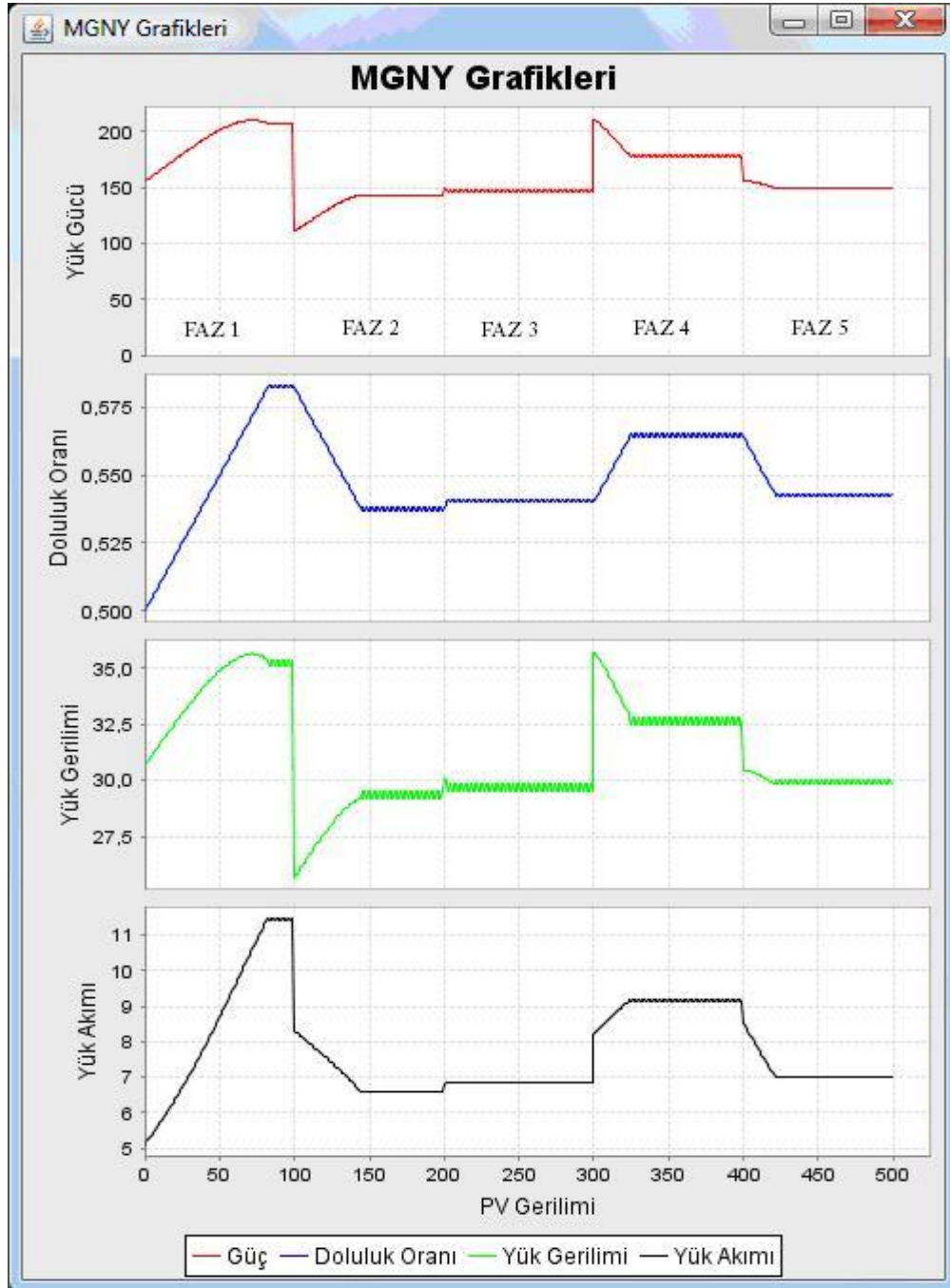
Gte artıř beklenmesine raėmen oransal V_{oc} yntemi bunu saėlayamamıřtır. Direk baėlı sistem ancak 180W'a ıkabilmiřtir.



Şekil 5.8. MGNI K&G metodu sonuçları (Yük direnci= 6Ω)

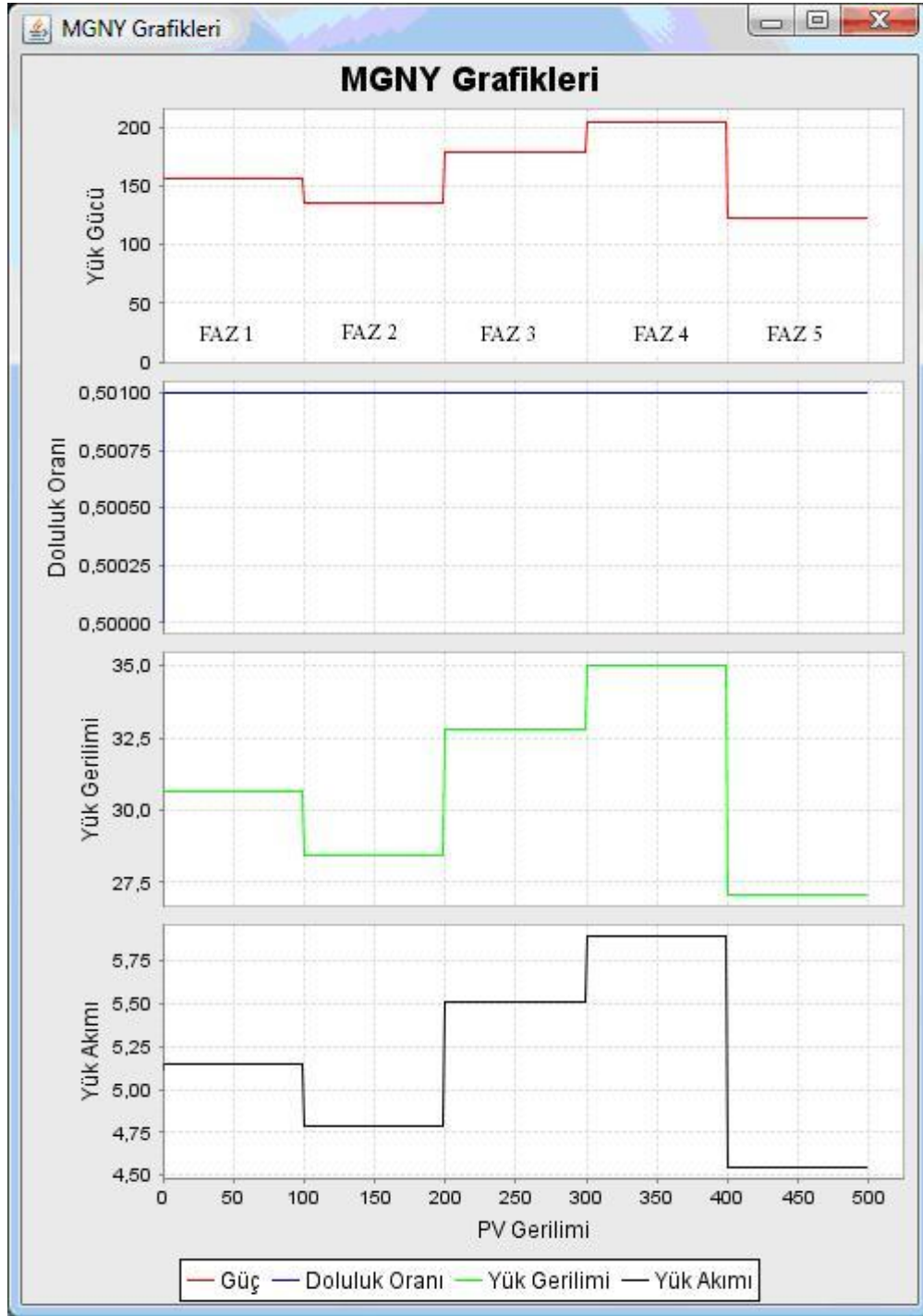
Faz 3'ten faz 4'e geçişte K&G güçte artışını devam ettirmiş 210W'a yakın güç değerine ulaşmış oransal V_{oc} 'de artış göstermiş fakat 175W civarında

kalmıştır. Faz 4'te direk bağlı sistem maksimuma yakın güç üretebilirken oransal V_{oc} metodu bu güç değerini yakalayamamıştır.



Şekil 5.9. MGNI Oransal Voc yöntemi sonuçları (Orantı sabiti=0,76 ve yük direnci = 6Ω)

Faz 5'te sıcaklık artışı gerçekleşmiş ve güçte düşüş beklenilmiştir. Direk bağlı sistem 120W civarına düşmüş, oransal V_{oc} çalışma noktasından uygun oransal gerilim değerine 10 adımda gelerek 150W değerinde sabitlenmiştir.



Şekil 5.10. Yüke direk bağlı sistem grafikleri (Yük direnci = 6Ω)

Buna karşılık K&G metodu 60W'a yakın bir güç düşüşünün ardından 155W'a değerine çıkarak bu değerde sabitlenmiştir.

Şekil 5.8 incelendiğinde, faz 3'te doluluk oranı düşmüş ve 0,5 değerine yaklaşmış olmasına rağmen güç artmış ve 180W civarında güç yakalanmıştır. Doluluk oranının 0,5'e yakın olması faz 3'teki ışınlam ve sıcaklık değerleri için uygun direnç değeri yük direnci olan 6Ω 'a yakın olduğunu göstermektedir. Bu nedenle direk bağlı sistemde faz 3'te elde edilen güç miktarı K&G metodu ile birbirine çok yakındır.

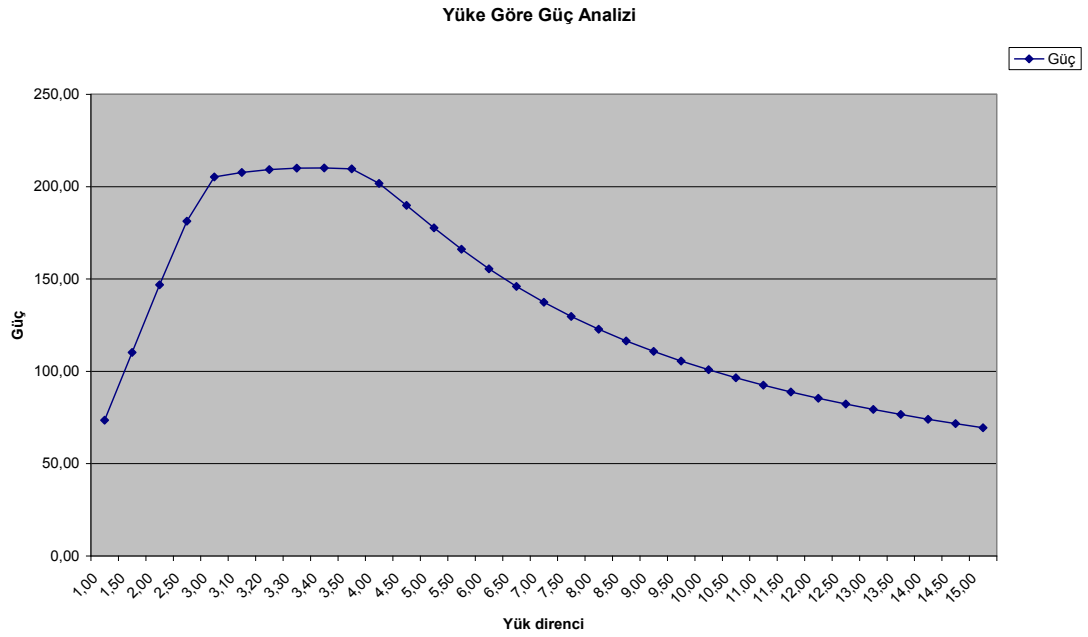
Fazlara göre ışınlam ve sıcaklık ile yöntemlere göre üretilen güçler Çizelge 5.2'deki gösterilmiştir.

Çizelge 5.2. Fazlara göre yöntemlerde elde edilen güçler

	İşinım	Sıcaklık	Yöntem	Güç(W)
Faz 1	1000	25	Direk bağlı	155
	1000	25	K&G	210
	1000	25	Oransal V	205
Faz 2	700	25	Direk bağlı	140
	700	25	K&G	145
	700	25	Oransal V	145
Faz 3	700	-25	Direk bağlı	180
	700	-25	K&G	180
	700	-25	Oransal V	150
Faz 4	850	-25	Direk bağlı	205
	850	-25	K&G	210
	850	-25	Oransal V	180
Faz 5	850	50	Direk bağlı	120
	850	50	K&G	155
	850	50	Oransal V	150

Yük direk bağlanarak yapılan çalışma sonucunda yüke göre güç değerleri hesaplanmış ve Şekil 6.1'deki yük-güç grafiği elde edilmiştir. Grafikte görülen maksimum güç noktası, 1000 W/m^2 ışınlam değerinde ve 25°C sıcaklıkta elde edilmiştir. Grafikten görüleceği gibi, verilen ışınlam ve sıcaklık değerlerinde,

yük direnci 3Ω ile $3,5\Omega$ arasında olduğunda kabul edilebilecek seviyede bir güç elde edilmektedir.



Şekil 5.11. Direk bağlı sistemde yük-güç grafiği

5.4.4. Grafiklerin oluşturulması

Grafiklerin oluşturulmasında JFreeChart UPA'si kullanılmıştır. Bu UPA internet üzerinden ücretsiz olarak indirilebilmektedir.

6. SONUÇ VE ÖNERİLER

Bu çalışmada, güneş pilinin çalışabileceği maksimum güç noktasına (MGN), yüke direk bağlı, kaygılandır&gözete ve oransal V_{oc} olmak üzere 3 farklı maksimum güç noktası izleyici yöntem kullanılmıştır. Farklı ışıınım ve sıcaklık değerleri için 5 farklı faz göz önüne alınarak optimum yük değeri elde edilmiş ve maksimum güç noktası yakalanmıştır.

Yapılan simülasyon sonucunda elde edilen grafikler analiz edildiğinde K&G yönteminin tüm fazlarda en yüksek güç değerini yakaladığı gözlemlenmiştir. Bununla birlikte oransal V_{oc} metodunun ışıınım ve sıcaklık değerleri ile oransal sabitin (k_1) uygun olduğu koşullarda kabul edilebilir seviyede güç ürettiği görülmüştür. Yükün direk bağlı olduğu sistemde ise ışıınım ve sıcaklık değerlerine göre optimum direnç değerlerinin uzağında güçte hayli düşme olduğu gözlemlenmiştir.

Simülasyonu yapılan yöntemlerden K&G yönteminin kullanımı önerilmektedir.

Geliştirilen yazılım, Java ortamında geliştirilmesi sayesinde kaynak kodu üzerinde değişiklik yapılarak kullanılan denklemler iyileştirilebilir ve yazılım geliştirilebilir. Özellikle R_{shunt} direnci simülasyon üzerine eklenerek bu direncin etkisinde incelenebilir.

Ayrıca Java için üretilen değişik yöntemlere ait UPA'lar kullanılarak yeni yöntemler simülasyon ortamına eklenebilir. Bu işlem için kullanılabilecek yöntemlerden biri bulanık mantık yöntemidir. Bu yöneme ait UPA JFuzzyLogic'tir ve <http://jfuzzylogic.sourceforge.net/> sitesinden indirilebilir.

Simülasyon ortamı, dirençsel yükler için geliştirilmiş ve simüle edilmiştir. Geliştirilen yazılım, yapılacak eklemelerle, diğer yüklerle kullanılabilir duruma getirilebilir.

Geliştirilen yazılımın arayüz sayfaları geliştirilerek kolaylıkla web ortamına aktarılabilir ve daha geniş kitleler tarafından kullanılması sağlanabilir.

KAYNAKLAR

1. Azab, M. , "A new Maximum Power Point Tracking for Photovoltaic Systems", ***Proceeding of World Academy of Science Engineering and technology***, 34: 571-574 (2008).
2. Faranda, R. , Leva, S. , "Energy comparison of MPPT techniques for PV Systems", ***Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century***, Pittsburgh, 1-6 (2008).
3. Katan, R.E. , Agelidis V.G. ve Nayer C.V. ; "Performans Analysis of A Solar Water Pumping System" , ***IEEE***, 1: 81-87 (1996).
4. Oi, A. , "Design and simulation of photovoltaic water pumping system", Yüksek Lisans, ***Faculty of California Polytechnic State University***, San Luis Obispo, 1-113 (2005).
5. Patel, M.R. , "Wind and Solar Power Systems, 1 ed.", ***CRC Pres***, New York,10-360 (1999).
6. Enslin J.H. ve Snyman D.B. ; "Simplified Feed-Forward Control of The Maximum Power Point in PV Installation", ***IEEE***, 1: 548-553 (1992).
7. Hua C. ve Shen C. , "Control of DC/DC Converters for Solar Energy System with Maximum Power Tracking", ***IEEE IECON 97***, 2: 827-832 (1997).
8. Glasner, I. ve Appelbaum, J. , "Advantage of Boost vs. Buck Topology for Maximum Power Point Tracker in Photovoltaic Systems", ***IEEE***, Israil, 355-358 (1996).
9. Walker G.R. , "Evaluating MPPT Converter topologies using a matlab pv model"; ***Journal of Electrical & Electronics Engineering***, 21: 49-55 (2001).
10. International Energy Agency, "Key World Energy Istatics", ***IEA***, 6-8 (2005).

11. T.C. Başbakanlık Devlet Planlama Teşkilatı Müsteşarlığı, "Sekizinci Beş Yıllık Kalkınma Planı: Elektrik Enerjisi Özel İhtisas Komisyonu Raporu", **DPT 2569, Ankara**, 15-240 (2001).
12. Özemre, A. Y. , 29 Ocak 2002, "Türkiye'nin Enerji Açısından Trajik Durumu", **Zaman Gazetesi**, (29.01.2002).
13. İnternet: Elektrik İşleri Etüt İdaresi "Güneş Enerjisi Çalışmaları" http://www.eie.gov.tr/turkce/YEK/gunes/gunes_index.html (16.08.2009).
14. İstanbul Teknik Üniversitesi, "Türkiye'de enerji ve geleceği", **İTÜ**, 10-180 (2007).
15. Kenneth, Z. , "Basic Photovoltaic Principles And Methods", **Van Nosrand Reinhold Company**, New York, 30-250 (1984).
16. Partain, L.D. , "Solar Cells and Their Applications" , **A Wiley-Interscience Publication**, California, 50-550 (1995).
17. Miloseviç, M. , Andersson, G. , Grabic, S. , "Decoupling current control and maximum power point control in small power network with photovoltaic source", Power Systems Conference and Exposition, **PSCE '06 2006 IEEE PES**, Atlanta, 1005-1011 (2006).
18. Esram, T. , Chapman, P.L. , "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques", **IEEE Energy conversion**, 22: 439-449 (2007).
19. Yafaoui, A. , Wu, B. , Cheung, R. , "Implementation of Maximum Power Point Tracking Algorithm for Residential Photovoltaic Systems", **2nd Canadian Solar Buildings Conference**, Calgary, 1-6 (2007).
20. Hohm, D.P. , Ropp, M.E., "Comparative Study of Maximum Power Point Tracking Algorithms Using an Experimental, Programmable, Maximum Power Point Tracking Test Bed", **IEEE Photovoltaic Specialists Conference**, AK, USA, 1699 – 1702 (2000).
21. Swiegers, W. , Enslin, J. , "An Integrated Maximum Power Point Tracker For Photovoltaic Panels", **IEEE International Symposium on Industrial Electronic Proceedings of the 1998**, Pretoria, 1: 40-44 (1998)

22. Yu, G.J. , Jung, Y.S. , Choi, J.Y. , Choy, I. , Song, J.H. ve Kim, G.S. , “A Novel Two-Mode MPPT Control Algorithm Based on Comparative Study of Existing Algorithms”, **Proc. Photovoltaic Specialists Conference**,1531-1534.(2002)
23. Hart, G.W. , Branz, H.M. , Cox, C.H. , "Experimental tests of openloop maximum-power-point tracking techniques", **Science Direct**, 13: 185-195 (1984).
24. Bekker, B. ve Beukes, H.J. , "Finding an optimal PV panel maximum power point tracking method", **7th AFRICON Conference**, Afrika, 2: 1125-1129 (2004).
25. Yuvarajan, S. ve Xu, S. , "Photovoltaic power converter with a simple maximum-power-point-tracker", **ISCAS 2003**, Fargo, ND, USA, 3: 399-402 (2003).
26. Wilamowski, B.M. , Li, X. , “Fuzzy System Based Maximum Power Point Tracking for PV Systems”, **IECON 02**,Boise, ID, USA, 4: 3280-3284 (2002).
27. Senjyu, T. , Uezato, K. "Maximum power point tracker using fuzzy control for photovoltaic arrays", **Proc. IEEE International Conf. On Ind. Technol.** , 143-147 (1994).
28. Patcharaprakiti, N. , Premrudeepreechacharn, S. , "Maximum power point tracking using adaptive fuzzy logic control for grid-connected photovoltaic system", **IEEE Power Eng. Society Winter Meeting**, 1: 372-377 (2002).
29. Rashid, M.H. , “Power Electronics - Circuits, Devices, and Applications 3rd Edition”, **Prentice Hall**, 200-800(2003).
30. Hussein, K.H. ,Muta, I. , Hoshino, T. , Osakada, M. , “Maximum Photovoltaic Power Tracking: an Algorithm for Rapidly Changing Atmospheric Conditions”, **IEE Proceedings – Generation, Transmission and Distribution**, 142: 59-64 (1995).
31. Koutroulis, E. ,Kalaitzakis, K. , Voulgaris, N.C. , "Development of a microcontroller-based, photovoltaic maximum power point tracking control system", **IEEE Trans. Power Electron.**, Chania, 16: 46-54 (2001).

32. Kangal, H. , “Fotovoltaik sistem analizi ve Labview tabanlı MPPT simülasyonu”, Yüksek Lisans, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 1-130 (2008).
33. Nakir, İ. , “Fotovoltaik güneş panellerinde GTS ve MGTS kullanarak verimliliğin artırılması”, Yüksek Lisans, **Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü**, San Luis Obispo, 1-113 (2005).

EKLER

EK – 1 Load.java kodu

Load.java

```
package model;
public class Load {
    double r;
    double p;
    double v;
    double i;
    public double getR() {
        return r;
    }
    public void setR(double r) {
        this.r = r;
    }
    public double getP() {
        return p;
    }
    public void setP(double p) {
        this.p = p;
    }
    public double getV() {
        return v;
    }
    public void setV(double v) {
        this.v = v;
    }
    public double getI() {
        return i;
    }
}
```

EK – 1 (Devam) Load.java kodu

```
public void setI(double i) {  
    this.i = i;  
}  
}
```

EK – 2 Converter.java kodu

Converter.java

```
package model;

public class Converter {
    double d;
    boolean yon;
    boolean po;
    boolean fra;
    public double getFrak() {
        return fra;
    }
    public void setFrak(double fra) {
        this.fra = fra;
    }
    double fra;
    boolean direct;
    public double getD() {
        return d;
    }
    public void setD(double d) {
        this.d = d;
    }
    public boolean isYon() {
        return yon;
    }
    public void setYon(boolean yon) {
        this.yon = yon;
    }
    public boolean isPo() {
```


EK – 2 (Devam) Converter.java kodu

```
        return po;
    }
    public void setPo(boolean po) {
        this.po = po;
    }
    public boolean isFra() {
        return fra;
    }
    public void setFra(boolean fra) {
        this.fra = fra;
    }
    public boolean isDirect() {
        return direct;
    }
    public void setDirect(boolean direct) {
        this.direct = direct;
    }
}
```

EK – 3 PvConstant.java kodu

PvConstant.java

```
package model;

public class PvConstant {
    double isc;
    double voc;
    double n;
    double tempCoefficientVoc;
    double tempCoefficientIsc;
    double rs;
    double vRs;
    double ideal;
    public double getIsc() {
        return isc;
    }
    public void setIsc(double isc) {
        this.isc = isc;
    }
    public double getVoc() {
        return voc;
    }
    public void setVoc(double voc) {
        this.voc = voc;
    }
    public double getN() {
        return n;
    }
    public void setN(double n) {
        this.n = n;
    }
}
```

EK – 3 (Devam) PvConstant.java kodu

```
}  
public double getTempCoefficientVoc() {  
    return tempCoefficientVoc;  
}  
public void setTempCoefficientVoc(double tempCoefficientVoc) {  
    this.tempCoefficientVoc = tempCoefficientVoc;  
}  
public double getTempCoefficientIsc() {  
    return tempCoefficientIsc;  
}  
public void setTempCoefficientIsc(double tempCoefficientIsc) {  
    this.tempCoefficientIsc = tempCoefficientIsc;  
}  
public double getRs() {  
    return rs;  
}  
public void setRs(double rs) {  
    this.rs = rs;  
}  
public double getVRs() {  
    return vRs;  
}  
public void setVRs(double rs) {  
    vRs = rs;  
}  
public double getIdeal() {  
    return ideal;  
}  
public void setIdeal(double ideal) {
```

EK – 3 (Devam) PvConstant.java kodu

```
        this.ideal = ideal;  
    }  
}
```

EK – 4 PvVariable.java kodu

PvVariable.java

```
package model;

public class PvVariable {
    double v;
    double i;
    double p;
    double pOld;
    double g;
    double t;
    double tempVoc;
    double tempIsc;
    double pTotal;
    int step;

    public double getV() {
        return v;
    }

    public void setV(double v) {
        this.v = v;
    }

    public double getI() {
        return i;
    }

    public void setI(double i) {
        this.i = i;
    }

    public double getP() {
        return p;
    }
}
```

EK – 4 (Devam) PvVariable.java kodu

```
public void setP(double p) {
    this.p = p;
}
public double getpOld() {
    return pOld;
}
public void setpOld(double pOld) {
    this.pOld = pOld;
}
public double getG() {
    return g;
}
public void setG(double g) {
    this.g = g;
}
public double getT() {
    return t;
}
public void setT(double t) {
    this.t = t;
}
public double getTempVoc() {
    return tempVoc;
}
public void setTempVoc(double tempVoc) {
    this.tempVoc = tempVoc;
}
public double getTemplsc() {
    return templsc;
}
```

EK – 4 (Devam) PvVariable.java kodu

```
}  
public void setTemplsc(double templsc) {  
    this.templsc = templsc;  
}  
public double getpTotal() {  
    return pTotal;  
}  
public void setpTotal(double pTotal) {  
    this.pTotal = pTotal;  
}  
public int getStep() {  
    return step;  
}  
public void setStep(int step) {  
    this.step = step;  
}  
}
```

EK – 5 Calc.java kodu

Calc.java

```
package controller;
import org.apache.log4j.Logger;
import model.Converter;
import model.Load;
import model.PvConstant;
import model.PvVariable;
public class Calc {
    static Logger logger = Logger.getLogger(Calc.class);
    public PvVariable workPoint(PvConstant pvConstant,PvVariable
pvVariable,Load load,Converter converter){
        double rLoad = 0;
        rLoad = ((Math.pow((1-converter.getD()), 2) / Math.pow(converter.getD(), 2))
* load.getR());
        rLoad = rLoad + pvConstant.getRs();
        double rCalc = 0;
        double v = 0;
        while(Math.round(rLoad * 100) > Math.round(rCalc * 100) &&
v<pvVariable.getTempVoc()){
            pvVariable.setV(v);
            pvVariable = calculateCurrent(pvConstant,pvVariable);
            rCalc = pvVariable.getV() / pvVariable.getI();
            v+=0.001;
        }
        return pvVariable;
    }
    public double cnvVoltage(double v,double d){
        double ret=0;
```


EK – 5 (Devam) Calc.java kodu

```

ret= ((d / (1 - d)) * v);
return ret;
}

public double cnvCurrent(double i,double d){
double ret=0;
ret= (((1 - d) / d) * i);
return ret;
}

public PvVariable calculateCurrent(PvConstant pvConstant, PvVariable
pvVariable){
double q = 1.602e-19;
double k = 1.381e-23;
double voc = pvVariable.getTempVoc() / pvConstant.getN();
double isc=pvVariable.getTempIsc();
double v = pvVariable.getV() / pvConstant.getN();
double il = isc * pvVariable.getG() / 1000;
double io = isc / (Math.exp((q * voc)/(pvConstant.getIdeal() * k *
(pvVariable.getT() + 273 ))) - 1);
double iloss = io * (Math.exp((q * v)/(pvConstant.getIdeal() * k *
(pvVariable.getT() + 273 ))) - 1);
double i = (il - iloss);
pvVariable.setI(i);
return pvVariable;
}

public PvVariable tempVocIsc(PvConstant pvConstant,PvVariable
pvVariable){
pvVariable.setTempVoc(pvConstant.getVoc() +
(pvConstant.getTempCoefficientVoc() * (pvVariable.getT() - 25))
);

```

EK – 5 (Devam) Calc.java kodu

```
pvVariable.setTempIsc(pvConstant.getIsc() +  
(pvConstant.getTempCoefficientIsc() * (pvVariable.getT() - 25)));  
return pvVariable;  
}  
}
```

EK – 6 ReadFile.java kodu

ReadFile.java

```
package controller;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import org.apache.log4j.Logger;
public class ReadFile {
    static Logger logger = Logger.getLogger(ReadFile.class);
    public ArrayList<String> read(String fileName) {
        ArrayList<String> file = new ArrayList<String>();
        BufferedReader br = null;
        try {
            br = new BufferedReader(new FileReader(fileName));
        } catch (FileNotFoundException e) {
            logger.error(e);
            JFrame frame = null;
            JOptionPane.showMessageDialog(frame,
                "Dosya Bulunamadi.Dosya adi:"+fileName,
                "Hata",
                JOptionPane.ERROR_MESSAGE);
        }
        String line;
        try {
            while ((line = br.readLine()) != null) {
```

EK – 6 (Devam) ReadFile.java kodu

```
file.add(line);
}
} catch (IOException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Dosya okunurken hata oluřtu.",
"Hata",
JOptionPane.ERROR_MESSAGE);
} finally{
try {
br.close();
} catch (IOException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Dosya okuma kapanırken hata oluřtu.",
"Hata",
JOptionPane.ERROR_MESSAGE);
}
}
return file;
}
}
```

EK – 7 WriteFile.java kodu

WriteFile.java

```
package controller;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import org.apache.log4j.Logger;
public class WriteFile {
    static Logger logger = Logger.getLogger(WriteFile.class);
    public void writeFile(ArrayList<double[]> allV, ArrayList<double[]> alPV,String
    filename) {
        BufferedWriter bufferedWriter = null;
        try {
            bufferedWriter = new BufferedWriter(new FileWriter(filename));
        } catch (IOException e) {
            logger.error(e);
            JFrame frame = null;
            JOptionPane.showMessageDialog(frame,
            "Dosyaya yazma başlatılırken hata oluştu. Dosyanın yazılacağı klasörün
            varlığından emin olunuz.",
            "Hata",
            JOptionPane.ERROR_MESSAGE);
        }
        try {
            bufferedWriter.write("V;I;P;R");
            bufferedWriter.newLine();
```

EK – 7 (Devam) WriteFile.java kodu

```

for (int i = 0; i < allV.size(); i++) {
String line = String.valueOf(allV.get(i)[0]).replace(".", ",") + ";"
+ String.valueOf(allV.get(i)[1]).replace(".", ",") + ";"
+ String.valueOf(alPV.get(i)[1]).replace(".", ",") + ";"
+ String.valueOf(allV.get(i)[0] / allV.get(i)[1]).replace(".", ",");
bufferedWriter.write(line);
bufferedWriter.newLine();
}
} catch (Exception e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Dosyaya yazılırken hata oluştu.",
"Hata",
JOptionPane.ERROR_MESSAGE);
} finally{
try {
bufferedWriter.close();
} catch (IOException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Dosya yazma kapatılırken hata oluştu.",
"Hata",
JOptionPane.ERROR_MESSAGE);
}
}
}
}

```

EK – 8 Graph2.java kodu

Graph2.java

```
package view;
import java.awt.Color;
import java.util.ArrayList;
import javax.swing.JFrame;
import org.apache.log4j.Logger;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.AxisLocation;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CombinedDomainXYPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.plot.XYPlot;
import org.jfree.chart.renderer.xy.StandardXYItemRenderer;
import org.jfree.chart.renderer.xy.XYItemRenderer;
import org.jfree.data.xy.XYDataset;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
public class Graph2 extends JFrame {
    private static final long serialVersionUID = 1L;
    static Logger logger = Logger.getLogger(Graph2.class);
    public Graph2(final String title, ArrayList<double[]> alIV, ArrayList<double[]>
alPV) {
        super(title);
        final JFreeChart chart = createCombinedChart(title,alIV,alPV);
        final ChartPanel panel = new ChartPanel(chart, true, true, true, false, true);
        panel.setPreferredSize(new java.awt.Dimension(500, 270));
        setContentPane(panel);}
```

EK – 8 (Devam) Graph2.java kodu

```
private JFreeChart createCombinedChart(String title, ArrayList<double[]>
allV, ArrayList<double[]> alPV) {
// create subplot 1...
final XYDataset data1 = createDataset1(allV);
final XYItemRenderer renderer1 = new StandardXYItemRenderer();
renderer1.setSeriesPaint(0, Color.red);
final NumberAxis rangeAxis1 = new NumberAxis("PV Akımı");
final XYPlot subplot1 = new XYPlot(data1, null, rangeAxis1, renderer1);
subplot1.setRangeAxisLocation(AxisLocation.BOTTOM_OR_LEFT);
// create subplot 2...
final XYDataset data2 = createDataset2(alPV);
final XYItemRenderer renderer2 = new StandardXYItemRenderer();
renderer2.setSeriesPaint(0, Color.red);
final NumberAxis rangeAxis2 = new NumberAxis("PV Gücü");
rangeAxis2.setAutoRangeIncludesZero(false);
final XYPlot subplot2 = new XYPlot(data2, null, rangeAxis2, renderer2);
subplot2.setRangeAxisLocation(AxisLocation.TOP_OR_LEFT);
// parent plot...
final CombinedDomainXYPlot plot = new CombinedDomainXYPlot(new
NumberAxis("PV Gerilimi"));
plot.setGap(10.0);
// add the subplots...
plot.add(subplot1, 1);
plot.add(subplot2, 1);
plot.setOrientation(PlotOrientation.VERTICAL);
// return a new chart containing the overlaid plot...
return new JFreeChart(title,
JFreeChart.DEFAULT_TITLE_FONT, plot, true);
}
```


EK – 8 (Devam) Graph2.java kodu

```
private XYDataset createDataset1(ArrayList<double[]> allV) {  
    // create dataset 1...  
    final XYSeries series1 = new XYSeries("Akım-Gerilim");  
    for (int i=0; i<allV.size(); i++){  
        if(allV.get(i)[1]>=0){  
            series1.add(allV.get(i)[0], allV.get(i)[1]);  
        }  
    }  
    return new XYSeriesCollection(series1);  
}  
  
private XYDataset createDataset2(ArrayList<double[]> alPV) {  
    // create dataset 2...  
    final XYSeries series2 = new XYSeries("Güç-Gerilim");  
    for (int i=0; i<alPV.size(); i++){  
        if(alPV.get(i)[1]>=0){  
            series2.add(alPV.get(i)[0], alPV.get(i)[1]);  
        }  
    }  
    return new XYSeriesCollection(series2);  
}  
}
```

EK – 9 Graph4.java kodu

Graph4.java

```
package view;
import java.awt.Color;
import java.util.ArrayList;
import javax.swing.JFrame;
import org.apache.log4j.Logger;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.AxisLocation;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.CombinedDomainXYPlot;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.plot.XYPlot;
import org.jfree.chart.renderer.xy.StandardXYItemRenderer;
import org.jfree.chart.renderer.xy.XYItemRenderer;
import org.jfree.data.xy.XYDataset;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
public class Graph4 extends JFrame {
private static final long serialVersionUID = 1L;
static Logger logger = Logger.getLogger(Graph4.class);
public Graph4(final String title, ArrayList<double[]> pLoad,
ArrayList<double[]> d, ArrayList<double[]> vLoad, ArrayList<double[]> iLoad)
{
super(title);
final JFreeChart chart = createCombinedChart(title,pLoad,d,vLoad,iLoad);
final ChartPanel panel = new ChartPanel(chart, true, true, true, false, true);
panel.setPreferredSize(new java.awt.Dimension(500, 270));
```

EK – 9 (Devam) Graph4.java kodu

```

setContentPane(panel);
}

private JFreeChart createCombinedChart(String title, ArrayList<double[]>
pLoad, ArrayList<double[]> d, ArrayList<double[]> vLoad,
ArrayList<double[]> iLoad) {
// create subplot 1...
final XYDataset data1 = createDataset1(pLoad);
final XYItemRenderer renderer1 = new StandardXYItemRenderer();
renderer1.setSeriesPaint(0, Color.red);
renderer1.setBaseItemLabelsVisible(true);
final NumberAxis rangeAxis1 = new NumberAxis("Yük Gücü");
final XYPlot subplot1 = new XYPlot(data1, null, rangeAxis1, renderer1);
subplot1.setRangeAxisLocation(AxisLocation.BOTTOM_OR_LEFT);
// create subplot 2...
final XYDataset data2 = createDataset2(d);
final XYItemRenderer renderer2 = new StandardXYItemRenderer();
renderer2.setSeriesPaint(0, Color.blue);
renderer2.setBaseItemLabelsVisible(true);
final NumberAxis rangeAxis2 = new NumberAxis("Doluluk Oranı");
rangeAxis2.getLabel();
rangeAxis2.setVisible(true);
rangeAxis2.setAutoRangeIncludesZero(false);
final XYPlot subplot2 = new XYPlot(data2, null, rangeAxis2, renderer2);
subplot2.setRangeAxisLocation(AxisLocation.TOP_OR_LEFT);
// create subplot 3...
final XYDataset data3 = createDataset3(vLoad);
final XYItemRenderer renderer3 = new StandardXYItemRenderer();
renderer3.setSeriesPaint(0, Color.green);
renderer3.setBaseItemLabelsVisible(true);

```

EK – 9 (Devam) Graph4.java kodu

```
final NumberAxis rangeAxis3 = new NumberAxis("Yük Gerilimi");
rangeAxis3.setAutoRangeIncludesZero(false);
final XYPlot subplot3 = new XYPlot(data3, null, rangeAxis3, renderer3);
subplot3.setRangeAxisLocation(AxisLocation.TOP_OR_LEFT);
// create subplot 4...
final XYDataset data4 = createDataset4(iLoad);
final XYItemRenderer renderer4 = new StandardXYItemRenderer();
renderer4.setSeriesPaint(0, Color.black);
renderer4.setBaseItemLabelsVisible(true);
final NumberAxis rangeAxis4 = new NumberAxis("Yük Akımı");
rangeAxis4.setAutoRangeIncludesZero(false);
final XYPlot subplot4 = new XYPlot(data4, null, rangeAxis4, renderer4);
subplot4.setRangeAxisLocation(AxisLocation.TOP_OR_LEFT);
// parent plot...
final CombinedDomainXYPlot plot = new CombinedDomainXYPlot(new
NumberAxis("PV Gerilimi"));
plot.setGap(10.0);
// add the subplots...
plot.add(subplot1, 1);
plot.add(subplot2, 1);
plot.add(subplot3, 1);
plot.add(subplot4, 1);
plot.setOrientation(PlotOrientation.VERTICAL);
// return a new chart containing the overlaid plot...
return new JFreeChart(title,
JFreeChart.DEFAULT_TITLE_FONT, plot, true);
}
private XYDataset createDataset1(ArrayList<double[]> pLoad) {
// create dataset 1...
```

EK – 9 (Devam) Graph4.java kodu

```

final XYSeries series1 = new XYSeries("Güç");
for (int i=0; i<pLoad.size(); i++){
    if(pLoad.get(i)[1]>=0){
        series1.add(pLoad.get(i)[0], pLoad.get(i)[1]);
    }
}
return new XYSeriesCollection(series1);
}

private XYDataset createDataset2(ArrayList<double[]> d) {
    // create dataset 2...
    final XYSeries series2 = new XYSeries("Doluluk Oranı");
    for (int i=0; i<d.size(); i++){
        if(d.get(i)[1]>=0){
            series2.add(d.get(i)[0], d.get(i)[1]);
        }
    }
    return new XYSeriesCollection(series2);
}

private XYDataset createDataset3(ArrayList<double[]> vLoad) {
    // create dataset 2...
    final XYSeries series3 = new XYSeries("Yük Gerilimi");
    for (int i=0; i<vLoad.size(); i++){
        if(vLoad.get(i)[1]>=0){
            series3.add(vLoad.get(i)[0], vLoad.get(i)[1]);
        }
    }
    return new XYSeriesCollection(series3);
}

private XYDataset createDataset4(ArrayList<double[]> iLoad) {

```

EK – 9 (Devam) Graph4.java kodu

```
// create dataset 2...
```

```
final XYSeries series4 = new XYSeries("Yük Akımı");
```

```
for (int i=0; i<iLoad.size(); i++){
```

```
if(iLoad.get(i)[1]>=0){
```

```
series4.add(iLoad.get(i)[0], iLoad.get(i)[1]);}}
```

```
return new XYSeriesCollection(series4);}}
```

EK – 10 Menu.java kodu

Menu.java

```
package view;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
import javax.swing.JTextField;
import javax.swing.SpringLayout;
import model.Converter;
import model.Load;
import model.PvConstant;
import model.PvVariable;
import org.apache.log4j.Logger;
import org.jfree.ui.RefineryUtilities;
import controller.Calc;
import controller.ReadFile;
import controller.WriteFile;
public class Menu extends JFrame {
    static Logger logger = Logger.getLogger(Menu.class);
    private static final long serialVersionUID = 1L;
```

EK – 10 (Devam) Menu.java kodu

```

SpringLayout layout = new SpringLayout();
//Menu tanımlarının yapıldığı kısım
JMenuBar menuBar = new JMenuBar();
JMenu mCharacter = new JMenu("Karakteristik");
JMenu mMaximumPower = new JMenu("Maksimum Güç");
JMenu mHelp = new JMenu("Yardım");
JMenu mExit = new JMenu("Çıkış");
//Alt Menü tanımlarının yapıldığı kısım
JMenuItem smCalcWorkPoint = new JMenuItem("Çalışma Noktası Yakala");
JMenuItem smDrawCh = new JMenuItem("Karakteristik Bilgileri");
JMenuItem smCatchMaxPoint = new JMenuItem("Maksimum Güç Yakala");
JMenuItem smThank = new JMenuItem("Teşekkür");
JMenuItem smAbout = new JMenuItem("Hakkında");
JMenuItem smExit = new JMenuItem("Programı Kapat");
//Buton ve Alanlarının tanımlaması
JTextField voc = new JTextField(10); //Hepsi
JTextField isc = new JTextField(10);
JTextField g = new JTextField(10);
JTextField t = new JTextField(10);
JTextField n = new JTextField(10);
JTextField tcVoc = new JTextField(10);
JTextField tcIsc = new JTextField(10);
JTextField rs = new JTextField(10);
JTextField r = new JTextField(10);
JTextField ideal = new JTextField(10);
JRadioButton po = new JRadioButton("P&O");
JRadioButton fra = new JRadioButton("Oransal Voc");
JTextField frak = new JTextField(10);
JRadioButton direct = new JRadioButton("Yüke Bağlı");

```


EK – 10 (Devam) Menu.java kodu

```
JLabel baslik = new JLabel("PV modülüne ait değerleri giriniz");
JLabel lvoc = new JLabel("Açık Devre Gerilimi (V):");
JLabel lisc = new JLabel("Kısa Devre Akımı(A):");
JLabel ln = new JLabel("Seri Hücre Sayısı:");
JLabel ltcVoc = new JLabel("Voc Sıcaklık Etkisi (V/"+(char)176+"C:");
JLabel ltclsc = new JLabel("Isc Sıcaklık Etkisi (A/"+(char)176+"C:");
JLabel lr = new JLabel("Yük Direnci:");
JLabel lRs = new JLabel("Seri iç direnç:");
JLabel lideal = new JLabel("İdeallik faktörü:");
JLabel lmethod = new JLabel("Kullanılacak yöntem:");
JLabel lg = new JLabel("Işınım Miktarı (W / m2):");
JLabel lt = new JLabel("Sıcaklık ("+(char)176+"C:");
JLabel lv = new JLabel("");
JLabel li = new JLabel("");
JLabel lpmp = new JLabel("");
JLabel limp = new JLabel("");
JLabel lvmp = new JLabel("");
JLabel lthank = new JLabel("<html>Maddi ve manevi desteklerinden  
dolayı;<br>"+
"<ul><li>Hocam Prof. Dr. Müzeyyen SARITAŞ'a"+
"<li>Anne ve babama"+
"<li>Esim Züleyha Yıldız YUMSEK'e"+
"<li>Sevgili arkadaşım Mesut Cemil ISLER'e"+
"<li>Mesai arkadaşlarım "+
"<ul><li>Derya OZALTAN"+
"<li>Bilgi Deniz BASLIOGLU"+
"<li>Serez Demir UNAL"+
"<li>Derya KUM"+
"<li>Salih BOZKURT"+
```

EK – 10 (Devam) Menu.java kodu

```
"<li>Mehmet YAZICI"+
"<li>Birol YILDIRIM" +
"<li>Selim KARA'ya</ul></ul>" +
"Teşekkürü bir borç bilirim.<br><br>" +
"<font color=blue>Özer YÜMSEK</font></html>"
);
JLabel labout = new JLabel("<html><font color=red>2009-2010 yılı Yüksek
Lisans Tezi</font><br>" +
"<br>Gazi üniversitesi<br>" +
"<br>Fen Bilimleri Enstitüsü<br>" +
"<br>Elektrik Elektronik Mühendisliği Anabilim Dalı<br>" +
"<br><font color=red>Danışman:</font> Prof. Dr. Müzeyyen SARITAŞ<br>" +
"<br><font color=red>Konu:</font> Güneş pilinin Maksimum Güç noktasında
Çalıştırılması<br>" +
"<br><font color=red>Versiyon:</font> 6.0"+
"<br>Bilgi ve iletişim için;<br>" +
"<br><font color=blue>Özer YÜMSEK<br>" +
"<br>E-mail: ozeryumsek@yahoo.com</font>");
final JLabel Ifra = new JLabel("Orantı sabiti:");
ButtonGroup bg = new ButtonGroup();
public Menu() {
RefineryUtilities.positionFrameOnScreen(this, 0.01, 0.01);
//Akim bulma Menu'su
mCharacter.add(smCalcWorkPoint);
smCalcWorkPoint.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent event) {
calcWorkPoint();
}
});
});
```

EK – 10 (Devam) Menu.java kodu

//Karakteristik Menu

```
mCharacter.add(smDrawCh);
smDrawCh.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        drawChart();
    }
});
```

//MPPT Menu

```
mMaximumPower.add(smCatchMaxPoint);
smCatchMaxPoint.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        mppt();
    }
});
```

//Yardim Menu

```
mHelp.add(smAbout);
smAbout.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        about();
    }
});
```

```
mHelp.add(smThank);
smThank.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        thank();
    }
});
```

//Cikis Menu

```
mExit.add(smExit);
```

```

EK – 10 (Devam) Menu.java kodu
smExit.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent event) {
System.exit(0);
}
});
//Ana Menu tanımları
menuBar.add(mCharacter);
menuBar.add(mMaximumPower);
menuBar.add(mHelp);
menuBar.add(mExit);
}
protected void mppt() {
getContentPane().setLayout(layout);
JButton wp = new JButton("Maksimum Güç Yakala");
bg.add(po);
bg.add(fra);
bg.add(direct);
baslik.setText("PV modülüne ait değerleri giriniz");
lvoc.setText("Açık Devre gerilimi (Voc):");
lisc.setText("Kısa Devre Akımı (Isc):");
lg.setText("Işınım miktarı (W / m2):");
lt.setText("Sıcaklık ("+(char)176+"C):");
ln.setText("Seri Hücre Sayısı:");
lr.setText("Yük Direnci :");
lv.setText("Çıkış gerilimi (V):");
li.setText("Çıkış Akımı (I):");
getContentPane().setVisible(false);
getContentPane().removeAll();
getContentPane().add(baslik);

```

EK – 10 (Devam) Menu.java kodu

```
getContentPane().add(lvoc);
getContentPane().add(voc);
voc.setText("33.2");
getContentPane().add(lisc);
getContentPane().add(isc);
isc.setText("8.58");
getContentPane().add(ln);
getContentPane().add(n);
n.setText("54");
getContentPane().add(ltcVoc);
getContentPane().add(tcVoc);
tcVoc.setText("-0.12");
getContentPane().add(ltclsc);
getContentPane().add(tclsc);
tclsc.setText("0.00515");
getContentPane().add(lRs);
getContentPane().add(rs);
rs.setText("0.137");
getContentPane().add(lideal);
getContentPane().add(ideal);
ideal.setText("1.57");
getContentPane().add(lr);
getContentPane().add(r);
r.setText("6");
getContentPane().add(lmethod);
getContentPane().add(po);
po.setSelected(true);
getContentPane().add(fra);
getContentPane().add(direct);
```

EK – 10 (Devam) Menu.java kodu

```
getContentPane().add(lfra);
```

```
lfra.setVisible(false);
```

```
getContentPane().add(frak);
```

```
frak.setText("0.76");
```

```
frak.setVisible(false);
```

```
getContentPane().add(wp);
```

```
//Yerlerinin belirlenmesi
```

```
layout.putConstraint(SpringLayout.WEST,          baslik,          10,
```

```
SpringLayout.WEST,getContentPane());
```

```
layout.putConstraint(SpringLayout.NORTH,         baslik,          10,
```

```
SpringLayout.NORTH,getContentPane());
```

```
layout.putConstraint(SpringLayout.WEST,          lvoc,           10,
```

```
SpringLayout.WEST,getContentPane()
```

```
);
```

```
layout.putConstraint(SpringLayout.NORTH,         lvoc,           40,
```

```
SpringLayout.NORTH,getContentPane());
```

```
layout.putConstraint(SpringLayout.WEST,          voc,            160,
```

```
SpringLayout.WEST,getContentPane());
```

```
layout.putConstraint(SpringLayout.NORTH,         voc,            40,
```

```
SpringLayout.NORTH,getContentPane());
```

```
layout.putConstraint(SpringLayout.WEST,          lisc,            10,
```

```
SpringLayout.WEST,getContentPane());
```

```
layout.putConstraint(SpringLayout.NORTH,         lisc,           70,
```

```
SpringLayout.NORTH,getContentPane());
```

```
layout.putConstraint(SpringLayout.WEST,          isc,            160,
```

```
SpringLayout.WEST,getContentPane());
```

```
layout.putConstraint(SpringLayout.NORTH,         isc,           70,
```

```
SpringLayout.NORTH,getContentPane()
```

```
);
```

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.WEST,	In,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	In,	100,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	n,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	n,	100,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ltcVoc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ltcVoc,	130,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	tcVoc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	tcVoc,	130,
SpringLayout.NORTH,getContentPane()		
);		
layout.putConstraint(SpringLayout.WEST,	ltclsc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ltclsc,	160,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	tclsc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	tclsc,	160,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lr,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lr,	190,
SpringLayout.NORTH,getContentPane());		

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.WEST,	r,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	r,	190,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	IRs,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	IRs,	220,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	rs,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	rs,	220,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lideal,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lideal,	250,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ideal,	160,
SpringLayout.WEST,getContentPane()		
);		
layout.putConstraint(SpringLayout.NORTH,	ideal,	250,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lmethod,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lmethod,	280,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	po,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	po,	280,
SpringLayout.NORTH,getContentPane());		

EK – 10 (Devam) Menu.java kodu

```

layout.putConstraint(SpringLayout.WEST,          fra,          160,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,         fra,          340,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          lfra,         10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,         lfra,         370,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          frak,         160,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,         frak,         370,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          direct,       160,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,         direct,       310,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          wp,          160,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,         wp,          400,
SpringLayout.NORTH,getContentPane());
wp.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent event) {
mpptOut();
}
});
fra.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent event) {
lfra.setVisible(true);
frak.setVisible(true);

```

EK – 10 (Devam) Menu.java kodu

```

}
});
direct.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent event) {
lfra.setVisible(false);
frak.setVisible(false);
}
});
po.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent event) {
lfra.setVisible(false);
frak.setVisible(false);
}
});
getContentPane().setVisible(true);
}
protected void mpptOut() {
PvConstant pvConstant = new PvConstant();
PvVariable pvVariable = new PvVariable();
Calc calc = new Calc();
Load load = new Load();
Converter converter = new Converter();
ArrayList<String> g =new ArrayList<String>();
ArrayList<String> t =new ArrayList<String>();
ReadFile rf = new ReadFile();
ArrayList<double[]> pLoad = new ArrayList<double[]>();
ArrayList<double[]> converterD = new ArrayList<double[]>();
ArrayList<double[]> vLoad = new ArrayList<double[]>();
ArrayList<double[]> iLoad = new ArrayList<double[]>();

```

EK – 10 (Devam) Menu.java kodu

```

g = rf.read("G.csv");
t = rf.read("T.csv");
int stepSize = g.size();
pvVariable.setStep(stepSize);
pvVariable.setpTotal(0);
try {
pvConstant.setVoc(Double.parseDouble(voc.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Voc değeri 33.2 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setVoc(33.2);
}
try {
pvConstant.setIsc(Double.parseDouble(isc.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Isc değeri 8.58 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setIsc(8.58);
}
try {
pvConstant.setN(Double.parseDouble(n.getText()));

```

EK – 10 (Devam) Menu.java kodu

```

} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "N değeri 54 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);
    pvConstant.setN(54);
}
try {
    pvConstant.setTempCoefficientVoc(Double.parseDouble(tcVoc.getText()));
} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "Voc Sıcaklık Etkisi değeri -0.12 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);
    pvConstant.setTempCoefficientVoc(-0.12);
}
try {
    pvConstant.setTempCoefficientIsc(Double.parseDouble(tclsc.getText()));
} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "Isc Sıcaklık Etkisi değeri 0.00515 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);

```

EK – 10 (Devam) Menu.java kodu

```

pvConstant.setTempCoefficientIsc(0.00515);
}
try {
pvConstant.setRs(Double.parseDouble(rs.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Seri iç direnç değeri 0.165 verilmiştir.",
"Uyarı",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setRs(6);
}
try {
pvConstant.setIdeal(Double.parseDouble(ideal.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"İdeallik faktörü değeri 0.57 verilmiştir.",
"Uyarı",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setIdeal(0.57);
}
try {
load.setR(Double.parseDouble(r.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;

```

```

EK – 10 (Devam) Menu.java kodu
OptionPane.showMessageDialog(frame,
"Yük Direnci değeri 6 verilmiştir.",
"Uyarı",
OptionPane.INFORMATION_MESSAGE);
load.setR(6);
}
try {
converter.setFrak(Double.parseDouble(frak.getText()));
} catch (NumberFormatException e) {
logger.error(e);
converter.setFrak(0.76);
}
converter.setDirect(direct.isSelected());
converter.setFra(fra.isSelected());
converter.setPo(po.isSelected());
converter.setD(0.5);
converter.setYon(true);
for(int step=0;step<stepSize;step++){
double[] p;
double[] cnvD;
double[] v;
double[] i;
p = new double[2];
cnvD = new double[2];
v = new double[2];
i = new double[2];
p[0]= step;
cnvD[0]=step;
v[0]=step;

```

EK – 10 (Devam) Menu.java kodu

```

i[0]=step;
pvVariable.setG(Double.parseDouble(g.get(step)));
String tcr;
try {
tcr = t.get(step);
} catch (Exception e) {
tcr = "25";
}
pvVariable.setT(Double.parseDouble(tcr));
calc.tempVocIsc(pvConstant, pvVariable);
double workV = pvConstant.getVoc() * converter.getFrak();
cnvD[1] = converter.getD();
calc.workPoint(pvConstant, pvVariable, load, converter);
pvConstant.setVRs(pvVariable.getI() * pvConstant.getRs());
pvVariable.setV(pvVariable.getV() - pvConstant.getVRs());
pvVariable.setP(pvVariable.getV() * pvVariable.getI());
if(step>0){
if(converter.isPo()){
if(pvVariable.getP()>pvVariable.getpOld() && converter.isYon()){
converter.setD(converter.getD() + 0.001);
pvVariable.setpOld(pvVariable.getP());
//converter.setYon(true);
}else if(pvVariable.getP() < pvVariable.getpOld() && converter.isYon()){
converter.setD(converter.getD() - 0.001);
pvVariable.setpOld(pvVariable.getP());
converter.setYon(false);
}else if(pvVariable.getP()>pvVariable.getpOld() && !converter.isYon()){
converter.setD(converter.getD() - 0.001);
pvVariable.setpOld(pvVariable.getP());

```

EK – 10 (Devam) Menu.java kodu

```
//converter.setYon(false);
}else if(pvVariable.getP() < pvVariable.getpOld() && !converter.isYon()){
converter.setD(converter.getD() + 0.001);
pvVariable.setpOld(pvVariable.getP());
converter.setYon(true);
}else{
pvVariable.setpOld(pvVariable.getP());
if(converter.isYon()){
converter.setD(converter.getD() + 0.001);
}else{
converter.setD(converter.getD() - 0.001);
}
}
}else if(converter.isFra()){
if(pvVariable.getV() < workV ){
converter.setD(converter.getD() - 0.001);
pvVariable.setpOld(pvVariable.getP());
}else if(pvVariable.getV() >= workV ){
converter.setD(converter.getD() + 0.001);
pvVariable.setpOld(pvVariable.getP());
}
}else {
pvVariable.setpOld(pvVariable.getP());
}
pvVariable.setpTotal(pvVariable.getpTotal() + pvVariable.getP());
}else{
pvVariable.setpOld(pvVariable.getP());
converter.setD(converter.getD() + 0.001);
pvVariable.setpTotal(pvVariable.getpTotal() + pvVariable.getP());
```


EK – 10 (Devam) Menu.java kodu

```

}
load.setV(calc.cnvVoltage(pvVariable.getV(), converter.getD()));
load.setI(calc.cnvVoltage(pvVariable.getI(), converter.getD()));
p[1]=pvVariable.getP();
v[1]=load.getV();
i[1]=load.getI();
pLoad.add(p);
converterD.add(cnvD);
vLoad.add(v);
iLoad.add(i);
}
Graph4          Chart          =          new          Graph4("MGNI
Grafikleri",pLoad,converterD,vLoad,iLoad);
Chart.pack();
RefineryUtilities.positionFrameOnScreen(Chart, 0.55, 0.01);
Chart.setSize(500, 700);
Chart.setVisible(true);
}
protected void drawChart() {
JButton wp = new JButton("Karakteristik Göster");
//Content Pane aktarımı

baslik.setText("PV modülüne ait değerleri giriniz");
lvoc.setText("Açık Devre Gerilimi (Voc):");
lisc.setText("Kısa Devre Akımı (Isc):");
lg.setText("Işınım miktarı (W / m2):");
lt.setText("Sıcaklık ("+(char)176+"C):");
ln.setText("Seri Hücre Sayısı:");
SpringLayout layout = new SpringLayout();

```

```
EK – 10 (Devam) Menu.java kodu
getContentPane().setLayout(layout);
getContentPane().setVisible(false);
getContentPane().removeAll();
getContentPane().add(baslik);
getContentPane().add(lvoc);
getContentPane().add(voc);
voc.setText("33.2");
getContentPane().add(lisc);
getContentPane().add(isc);
isc.setText("8.58");
getContentPane().add(lg);
getContentPane().add(g);
g.setText("1000");
getContentPane().add(lt);
getContentPane().add(t);
t.setText("25");
getContentPane().add(ln);
getContentPane().add(n);
n.setText("54");
getContentPane().add(ltcVoc);
getContentPane().add(tcVoc);
tcVoc.setText("-0.12");
getContentPane().add(ltclsc);
getContentPane().add(tclsc);
tclsc.setText("0.00515");
getContentPane().add(lideal);
getContentPane().add(ideal);
ideal.setText("1.57");
getContentPane().add(lRs);
```

EK – 10 (Devam) Menu.java kodu

```
getContentPane().add(rs);
```

```
rs.setText("0.137");
```

```
getContentPane().add(wp);
```

```
//Yerlerinin belirlenmesi
```

layout.putConstraint(SpringLayout.WEST,	baslik,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	baslik,	10,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lvoc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lvoc,	40,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	voc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	voc,	40,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lisc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lisc,	70,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	isc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	isc,	70,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lg,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lg,	100,
SpringLayout.NORTH,getContentPane()		
);		

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.WEST,	g,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	g,	100,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lt,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lt,	130,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	t,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	t,	130,
SpringLayout.NORTH,getContentPane()		
);		
layout.putConstraint(SpringLayout.WEST,	ln,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ln,	160,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	n,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	n,	160,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ltcVoc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ltcVoc,	190,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	tcVoc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	tcVoc,	190,
SpringLayout.NORTH,getContentPane());		

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.WEST,	ltclsc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ltclsc,	220,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	tcclsc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	tcclsc,	220,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lideal,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lideal,	250,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ideal,	160,
SpringLayout.WEST,getContentPane()		
);		
layout.putConstraint(SpringLayout.NORTH,	ideal,	250,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	IRs,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	IRs,	280,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	rs,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	rs,	280,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	wp,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	wp,	400,
SpringLayout.NORTH,getContentPane());		

EK – 10 (Devam) Menu.java kodu

```

wp.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {
        drawChartOut();
    }
});
getContentPane().setVisible(true);
}

protected void drawChartOut() {
    ArrayList<double[]> allV = new ArrayList<double[]>();
    ArrayList<double[]> alPV = new ArrayList<double[]>();
    PvConstant pvConstant = new PvConstant();
    PvVariable pvVariable = new PvVariable();
    Calc calc = new Calc();
    WriteFile wr = new WriteFile();
    try {
        pvConstant.setVoc(Double.parseDouble(voc.getText()));
    } catch (NumberFormatException e) {
        logger.error(e);
        JFrame frame = null;
        JOptionPane.showMessageDialog(frame,
            "Voc değeri 33.2 verilmiştir.",
            "Uyari",
            JOptionPane.INFORMATION_MESSAGE);
        pvConstant.setVoc(33.2);
    }
    try {
        pvConstant.setIsc(Double.parseDouble(isc.getText()));
    } catch (NumberFormatException e) {
        logger.error(e);
    }
}

```

EK – 10 (Devam) Menu.java kodu

```

JFrame frame = null;
JOptionPane.showMessageDialog(frame,
    "Isc değeri 8.58 verilmiştir.",
    "Uyari",
    JOptionPane.INFORMATION_MESSAGE);
pvConstant.setIsc(8.58);
}
try {
    pvConstant.setN(Double.parseDouble(n.getText()));
} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "N değeri 54 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);
    pvConstant.setN(54);
}
try {
    pvConstant.setTempCoefficientVoc(Double.parseDouble(tcVoc.getText()));
} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "Voc Sıcaklık Etkisi değeri -0.12 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);
    pvConstant.setTempCoefficientVoc(-0.12);
}

```

EK – 10 (Devam) Menu.java kodu

```
try {
pvConstant.setTempCoefficientIsc(Double.parseDouble(tclsc.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Isc Sıcaklık Etkisi değeri 0.00515 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setTempCoefficientIsc(0.00515);
}
try {
pvVariable.setG(Double.parseDouble(g.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"G değeri 1000 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvVariable.setG(1000);
}
try {
pvVariable.setT(Double.parseDouble(t.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"T değeri 25 verilmiştir.",
```


EK – 10 (Devam) Menu.java kodu

```

"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvVariable.setT(25);
}
try {
pvConstant.setIdeal(Double.parseDouble(ideal.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Ideallik faktörü değeri 0.57 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setIdeal(0.57);
}
try {
pvConstant.setRs(Double.parseDouble(rs.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Seri iç direnç değeri 0.165 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setRs(25);
}
calc.tempVocIsc(pvConstant, pvVariable);
double Vmp = 0, Imp = 0, Pmp=0;
for(double v=0;v<pvVariable.getTempVoc();v+=0.001){

```

EK – 10 (Devam) Menu.java kodu

```
double[] iv;
double[] pv;
iv = new double[2];
pv = new double[2];
pvVariable.setV(v);
pvVariable = calc.calculateCurrent(pvConstant, pvVariable);
pvConstant.setVRs(pvVariable.getI() * pvConstant.getRs());
pvVariable.setV(pvVariable.getV() - pvConstant.getVRs());
pvVariable.setP(pvVariable.getV() * pvVariable.getI());
if(pvVariable.getP()>Pmp){
    Pmp = pvVariable.getP();
    Imp = pvVariable.getI();
    Vmp = pvVariable.getV();
}
if(pvVariable.getP()>=0){
    iv[0]=pvVariable.getV();
    pv[0]=pvVariable.getV();
    iv[1]=pvVariable.getI();
    pv[1]=pvVariable.getP();
    allV.add(iv);
    alPV.add(pv);
}
}

//Grafik alani
Graph2 c1 = new Graph2("Karakteristik Eğrileri",allV,alPV);
c1.pack();
RefineryUtilities.positionFrameOnScreen(c1, 0.55, 0.01);
c1.setVisible(true);
c1.setSize(400, 500);
```

EK – 10 (Devam) Menu.java kodu

```
wr.writeFile(alIV,alPV,"VIP.csv");
```

```
//Sonuc Bilgileri
```

```
baslik.setText("PV modülü Maksimum Güç değerleri:");
```

```
lpmp.setText("Maksimum Güç (P):"+Round(Pmp,2)+" W");
```

```
limp.setText("Maksimum Güçte Akım (I):"+Round(Imp,2)+" A");
```

```
lvmp.setText("Maksimum Güçte Gerilim (V):"+Round(Vmp,2)+" V");
```

```
//ContentPane'e eklenmesi
```

```
SpringLayout layout = new SpringLayout();
```

```
getContentPane().setLayout(layout);
```

```
getContentPane().setVisible(false);
```

```
getContentPane().removeAll();
```

```
getContentPane().add(baslik);
```

```
getContentPane().add(lpmp);
```

```
getContentPane().add(lvmp);
```

```
getContentPane().add(limp);
```

```
//Yerlerinin belirlenmesi
```

```
layout.putConstraint(SpringLayout.WEST,          baslik,          10,
SpringLayout.WEST,getContentPane());
```

```
layout.putConstraint(SpringLayout.NORTH,         baslik,          10,
SpringLayout.NORTH,getContentPane());
```

```
layout.putConstraint(SpringLayout.WEST,          lpmp,          10,
SpringLayout.WEST,getContentPane());
```

```
layout.putConstraint(SpringLayout.NORTH,         lpmp,          45,
SpringLayout.NORTH,getContentPane());
```

```
layout.putConstraint(SpringLayout.WEST,          lvmp,          10,
SpringLayout.WEST,getContentPane()
```

```
);
```

```
layout.putConstraint(SpringLayout.NORTH,         lvmp,          75,
SpringLayout.NORTH,getContentPane());
```

EK – 10 (Devam) Menu.java kodu

```

layout.putConstraint(SpringLayout.WEST,      limp,      10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,      limp,      105,
SpringLayout.NORTH,getContentPane());
getContentPane().setVisible(true);
}

protected void calcWorkPoint() {
JButton wp = new JButton("Çalışma Noktası Bul");
baslik.setText("PV modülüne ait değerleri giriniz");
lvoc.setText("Açık Devre Gerilimi (Voc:");
lisc.setText("Kısa Devre Akımı (Isc:");
lg.setText("Işınım miktarı (W / m2:");
lt.setText("Sıcaklık ("+(char)176+"C:");
ln.setText("Seri Hücre Sayısı:");
lr.setText("Yük Direnci :");
SpringLayout layout = new SpringLayout();
getContentPane().setLayout(layout);
getContentPane().setVisible(false);
getContentPane().removeAll();
getContentPane().add(baslik);
getContentPane().add(lvoc);
getContentPane().add(voc);
voc.setText("33.2");
getContentPane().add(lisc);
getContentPane().add(isc);
isc.setText("8.58");
getContentPane().add(lg);
getContentPane().add(g);
g.setText("1000");

```

EK – 10 (Devam) Menu.java kodu

```

getContentPane().add(lt);
getContentPane().add(t);
t.setText("25");
getContentPane().add(ln);
getContentPane().add(n);
n.setText("54");
getContentPane().add(ltcVoc);
getContentPane().add(tcVoc);
tcVoc.setText("-0.12");
getContentPane().add(ltclsc);
getContentPane().add(tclsc);
tclsc.setText("0.00515");
getContentPane().add(lRs);
getContentPane().add(rs);
rs.setText("0.137");
getContentPane().add(lideal);
getContentPane().add(ideal);
ideal.setText("1.57");
getContentPane().add(lr);
getContentPane().add(r);
r.setText("6");
getContentPane().add(wp);
//Yerlerinin belirlenmesi
layout.putConstraint(SpringLayout.WEST,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,
SpringLayout.WEST,getContentPane());

```

baslik, 10,

baslik, 10,

lvoc, 10,

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.NORTH,	lvoc,	40,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	voc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	voc,	40,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lisc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lisc,	70,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	isc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	isc,	70,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lg,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lg,	100,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	g,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	g,	100,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lt,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lt,	130,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	t,	160,
SpringLayout.WEST,getContentPane()		
);		

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.NORTH,	t,	130,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ln,	10,
SpringLayout.WEST,getContentPane()		
);		
layout.putConstraint(SpringLayout.NORTH,	ln,	160,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	n,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	n,	160,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ltcVoc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ltcVoc,	190,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	tcVoc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	tcVoc,	190,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ltclsc,	10,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ltclsc,	220,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	tclsc,	160,
SpringLayout.WEST,getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	tclsc,	220,
SpringLayout.NORTH,getContentPane());		
layout.putConstraint(SpringLayout.WEST,	IRs,	10,
SpringLayout.WEST,getContentPane());		

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.NORTH,	IRs,	250,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	rs,	160,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	rs,	250,
SpringLayout.NORTH, getContentPane()		
);		
layout.putConstraint(SpringLayout.WEST,	lideal,	10,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lideal,	280,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ideal,	160,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ideal,	280,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lr,	10,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lr,	310,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	r,	160,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	r,	310,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	wp,	160,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	wp,	400,
SpringLayout.NORTH, getContentPane());		
wp.addActionListener(new ActionListener() {		
public void actionPerformed(ActionEvent event) {		


```

EK – 10 (Devam) Menu.java kodu
workPointOut();
}
});
getContentPane().setVisible(true);
}
protected void workPointOut() {
    PvConstant pvConstant = new PvConstant();
    PvVariable pvVariable = new PvVariable();
    Load load = new Load();
    Converter converter = new Converter();
    Calc calc = new Calc();
    try {
        pvConstant.setVoc(Double.parseDouble(voc.getText()));
    } catch (NumberFormatException e) {
        logger.error(e);
        JFrame frame = null;
        JOptionPane.showMessageDialog(frame,
            "Voc değeri 33.2 verilmiştir.",
            "Uyari",
            JOptionPane.INFORMATION_MESSAGE);
        pvConstant.setVoc(33.2);
    }
    try {
        pvConstant.setIsc(Double.parseDouble(isc.getText()));
    } catch (NumberFormatException e) {
        logger.error(e);
        JFrame frame = null;
        JOptionPane.showMessageDialog(frame,
            "Isc değeri 8.58 verilmiştir.",

```

EK – 10 (Devam) Menu.java kodu

```

"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setIsc(8.58);
}
try {
pvConstant.setN(Double.parseDouble(n.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"N değeri 54 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setN(54);
}
try {
pvConstant.setTempCoefficientVoc(Double.parseDouble(tcVoc.getText()));
} catch (NumberFormatException e) {
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
"Voc Sıcaklık Etkisi değeri -0.12 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setTempCoefficientVoc(-0.12);
}
try {
pvConstant.setTempCoefficientIsc(Double.parseDouble(tcIsc.getText()));
} catch (NumberFormatException e) {

```

EK – 10 (Devam) Menu.java kodu

```
logger.error(e);
JFrame frame = null;
JOptionPane.showMessageDialog(frame,
    "Isc Sıcaklık Etkisi değeri 0.00515 verilmiştir.",
    "Uyari",
    JOptionPane.INFORMATION_MESSAGE);
pvConstant.setTempCoefficientIsc(0.00515);
}
try {
    load.setR(Double.parseDouble(r.getText()));
} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "R değeri 6 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);
    load.setR(6);
}
try {
    pvVariable.setG(Double.parseDouble(g.getText()));
} catch (NumberFormatException e) {
    logger.error(e);
    JFrame frame = null;
    JOptionPane.showMessageDialog(frame,
        "G değeri 1000 verilmiştir.",
        "Uyari",
        JOptionPane.INFORMATION_MESSAGE);
    pvVariable.setG(1000);
```

EK – 10 (Devam) Menu.java kodu

```
}  
try {  
    pvVariable.setT(Double.parseDouble(t.getText()));  
} catch (NumberFormatException e) {  
    logger.error(e);  
    JFrame frame = null;  
    JOptionPane.showMessageDialog(frame,  
        "T değeri 25 verilmiştir.",  
        "Uyari",  
        JOptionPane.INFORMATION_MESSAGE);  
    pvVariable.setT(25);  
}  
try {  
    pvConstant.setRs(Double.parseDouble(rs.getText()));  
} catch (NumberFormatException e) {  
    logger.error(e);  
    JFrame frame = null;  
    JOptionPane.showMessageDialog(frame,  
        "Seri iç direnç değeri 0.165 verilmiştir.",  
        "Uyari",  
        JOptionPane.INFORMATION_MESSAGE);  
    pvConstant.setRs(0.165);  
}  
try {  
    pvConstant.setIdeal(Double.parseDouble(ideal.getText()));  
} catch (NumberFormatException e) {  
    logger.error(e);  
    JFrame frame = null;  
    JOptionPane.showMessageDialog(frame,
```

EK – 10 (Devam) Menu.java kodu

```
"Ideallik faktörü değeri 0.57 verilmiştir.",
"Uyari",
JOptionPane.INFORMATION_MESSAGE);
pvConstant.setIdeal(0.57);
}
converter.setD(0.5);
pvVariable = calc.tempVocIsc(pvConstant, pvVariable);
pvVariable = calc.workPoint(pvConstant, pvVariable, load, converter);
pvConstant.setVRs(pvVariable.getI() * pvConstant.getRs());
pvVariable.setV(pvVariable.getV() - pvConstant.getVRs());
pvVariable.setP(pvVariable.getV() * pvVariable.getI());
baslik.setText("PV modülüne ait değerler aşağıdadır:");
lvoc.setText("Açık Devre Gerilimi (Voc):"+pvVariable.getTempVoc()+" V");
lisc.setText("Kısa Devre Akımı (Isc):"+pvVariable.getTempIsc()+" A");
lg.setText("Işınım miktarı (G):"+pvVariable.getG()+" W/m"+(char)178);
lt.setText("Sıcaklık (T):"+pvVariable.getT()+" "+(char)176+"C");
lr.setText("Yük Direnci :"+load.getR());
lv.setText("Çıkış Gerilimi (V):"+Round(pvVariable.getV(),2)+" V");
li.setText("Çıkış Akımı (I):"+Round(pvVariable.getI(),2)+" A");
ln.setText("Güç değeri (P):"+Round(pvVariable.getP(),2)+" W");
//ContentPane'e eklenmesi
SpringLayout layout = new SpringLayout();
getContentPane().setLayout(layout);
getContentPane().setVisible(false);
getContentPane().removeAll();
getContentPane().add(baslik);
getContentPane().add(lvoc);
getContentPane().add(lisc);
getContentPane().add(lg);
```

EK – 10 (Devam) Menu.java kodu

```

getContentPane().add(It);
getContentPane().add(lr);
getContentPane().add(lv);
getContentPane().add(li);
getContentPane().add(ln);
//Yerlerinin belirlenmesi
layout.putConstraint(SpringLayout.WEST,          baslik,          10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,          baslik,          10,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          lvoc,            10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,          lvoc,            40,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          lisc,            10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,          lisc,            70,
SpringLayout.NORTH,getContentPane()
);
layout.putConstraint(SpringLayout.WEST,          lg,              10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,          lg,              100,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          lt,              10,
SpringLayout.WEST,getContentPane());
layout.putConstraint(SpringLayout.NORTH,          lt,              130,
SpringLayout.NORTH,getContentPane());
layout.putConstraint(SpringLayout.WEST,          lr,              10,
SpringLayout.WEST,getContentPane());

```

EK – 10 (Devam) Menu.java kodu

layout.putConstraint(SpringLayout.NORTH,	lr,	160,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	lv,	10,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	lv,	190,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	li,	10,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,		li, 220,
SpringLayout.NORTH, getContentPane());		
layout.putConstraint(SpringLayout.WEST,	ln,	10,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	ln,	250,
SpringLayout.NORTH, getContentPane());		
getContentPane().setVisible(true);		
}		
protected void about() {		
//ContentPane'e eklenmesi		
SpringLayout layout = new SpringLayout();		
getContentPane().setLayout(layout);		
getContentPane().setVisible(false);		
getContentPane().removeAll();		
getContentPane().add(labout);		
//Yerlerinin belirlenmesi		
layout.putConstraint(SpringLayout.WEST,	labout,	10,
SpringLayout.WEST, getContentPane());		
layout.putConstraint(SpringLayout.NORTH,	labout,	10,
SpringLayout.NORTH, getContentPane());		
getContentPane().setVisible(true);		

EK – 10 (Devam) Menu.java kodu

```

}
protected void thank() {
    //ContentPane'e eklenmesi
    SpringLayout layout = new SpringLayout();
    getContentPane().setLayout(layout);
    getContentPane().setVisible(false);
    getContentPane().removeAll();
    getContentPane().add(lthank);
    //Yerlerinin belirlenmesi
    layout.putConstraint(SpringLayout.WEST,          lthank,          10,
        SpringLayout.WEST, getContentPane());
    layout.putConstraint(SpringLayout.NORTH,         lthank,          10,
        SpringLayout.NORTH, getContentPane());
    getContentPane().setVisible(true);
}
private double Round(double pmp, int i) {
    double p = (double)Math.pow(10,i);
    pmp = pmp * p;
    double tmp = Math.round(pmp);
    return tmp/p;
}
}

```


EK – 11 Run.java kodu

Run.java

```
package view;
import javax.swing.JFrame;
import org.apache.log4j.Logger;
public class Run extends Menu{
    static Logger logger = Logger.getLogger(Run.class);
    private static final long serialVersionUID = -7847419050711300612L;
    public Run(){
        setTitle("PV Uygulaması");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 500);
        setDefaultCloseOperationDecorated(false);
        setJMenuBar(menuBar);
        calcWorkPoint();
        setVisible(true);
    }
    public static void main(String[] args) {
        new Run();
    }
}
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : YÜMSEK, Özer
Uyruğu : T.C.
Doğum tarihi ve yeri : 02.05.1979 / Almanya
Medeni hali : Evli
Telefon : 0 (533) 3421285
e-mail : ozeryumsek@yahoo.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Dumlupınar Üniv. / Elk-Elk Müh.	2001
Lise	Kırıkkale Fen Lisesi	1996

İş Deneyimi

Yıl	Yer	Görev
2005	Gümrük Müsteşarlığı	Programcı
2006	Maliye Bakanlığı	Çözümleyici

Yabancı Dil

İngilizce

Hobiler

Müzik, Bilgisayar, Futbol, Satranç.