

**MOTOR DİZAYNI İÇİN OPTİMİZASYON
ALGORİTMASI HAZIRLANMASI**

Özkan ÖZLÜ

DANIŞMAN

Yrd. Doç. Dr. Murat CANER

ELEKTRİK EĞİTİMİ ANABİLİM DALI

Eylül, 2012

**AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**MOTOR DİZAYNI İÇİN OPTİMİZASYON ALGORİTMASI
HAZIRLANMASI**

Özkan ÖZLÜ

DANIŞMAN

Yrd. Doç. Dr. Murat CANER

ELEKTRİK EĞİTİMİ ANABİLİM DALI

Eylül, 2012

TEZ ONAY SAYFASI

Özkan ÖZLÜ tarafından hazırlanan “MOTOR DİZAYNI İÇİN OPTİMİZASYON ALGORİTMASI HAZIRLANMASI” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 21/09/2012 tarihinde aşağıdaki jüri tarafından oy birliği ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Elektrik Eğitimi Anabilim Dalı’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan	:Prof. Dr. Kubilay ASLANTAŞ	_____
	Afyon Kocatepe Üniversitesi Teknoloji Fakültesi	
Üye	:Yrd. Doç. Dr. Fatih HOCAOĞLU	_____
	Afyon Kocatepe Üniversitesi Mühendislik Fakültesi	
Üye	:Yrd. Doç. Dr. Murat CANER	_____
	Afyon Kocatepe Üniversitesi Teknoloji Fakültesi	

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
...../...../..... tarih ve
.....sayılı kararıyla onaylanmıştır.

.....
Enstitü Müdürü
Prof. Dr. Mevlüt DOĞAN

BİLİMSEL ETİK BİLDİRİM SAYFASI

Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

20/09/2012

Özkan ÖZLÜ

ÖZET

Yüksek Lisans Tezi

MOTOR DİZAYNI İÇİN OPTİMİZASYON ALGORİTMASI HAZIRLANMASI

Özkan ÖZLÜ

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik Eğitimi Anabilim Dalı

Danışman: Yrd. Doç. Dr. Murat CANER

Bu araştırmada, kuş ve balık sürülerinin eş zamanlı hareketlerinden esinlenerek geliştirilmiş, parçacık sürü optimizasyonu algoritmasının bir türevi olan, pasif katılımlı parçacık sürü optimizasyonu metodunu kullanarak daimi mıknatıslı senkron motor tasarımının sensörsüz kontrol açısından optimize edilmesi amaçlanmıştır.

Pasif katılımlı parçacık sürü optimizasyonu algoritması MATLAB ortamında geliştirilerek optimizasyon algoritmalarının testlerinde kullanılan, Rastrigin, Schwefel ve Michalewicz gibi, bir takım iyi-bilinen, doğrusal olmayan fonksiyonlar üzerinde sınanmış ve başarılı sonuçların alındığı gözlenmiştir. Yazılan optimizasyon algoritması, sonlu elemanlar tabanlı, motor tasarımı yapabilen ve performansını ölçebilen bir simülasyon yazılımı ile haberleşmeyi sağlayan, yine MATLAB ortamında önceden yazılmış aracı bir yazılıma adapte edilmiştir.

Motorun dış geometrisi sabit tutularak iç geometrisinde belirlenen sekiz değişken optimize edilmiştir. Tork ve sensörsüz kontrol kabiliyetini artırmak olmak üzere, iki farklı amaç için tasarım optimizasyonu yapılmıştır. Daimi mıknatıslı senkron motorların sensörsüz kontrol yeteneği optimize edilirken tork performansı düşmektedir. Yapılan üçüncü bir tasarım optimizasyonunda ise aynı dış geometriye sahip olan motorun

ürettiđi tork değeriinden daha az taviz vererek sensörsüz kontrol yeteneđi arttırılmaya çalıřılmıştır.

Yapılan her bir tasarım için performans sonuçları verilerek karşılaştırılmıştır. İlave olarak, her bir tasarım değışkeninin $\pm\%5-10$ değışimlerine göre performansa etkileri de incelenmiştir.

2012, xii + 69 sayfa

Anahtar Kelimeler: Motor tasarımı, Daimi mıknatıslı motorlar, Sensörsüz kontrol, Parçacık sürü optimizasyonu

ABSTRACT

M. Sc Thesis

PREPARATION OF AN OPTIMISATION ALGORITHM FOR MOTOR DESIGN

Özkan ÖZLÜ

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Electrics Education

Supervisor: Asst. Prof. Dr. Murat CANER

In this study, an optimization of a permanent magnet synchronous motor design has been aimed in terms of sensorless control using another variation of particle swarm optimization algorithm, which was developed inspired by bird or fish school moving synchronously, particle swarm optimization with passive congregation.

Particle swarm optimization with passive congregation algorithm first was developed on MATLAB environment and then examined on some well-known non-linear functions such as Rastrigin, Schwefel and Michalewicz which are widely used for examination of optimization algorithms. The algorithm written was then adapted to software, previously developed on MATLAB environment again, that can obtain communication with an external finite elements analysis based, capable of designing and measuring performance of a motor.

The given eight interior variables have been optimized by fixing the outer geometry of the motor. In order to enhance torque and sensorless control capability, design optimizations has been made for two different objectives. While optimizing the sensorless control capability of the permanent magnet synchronous motor, torque performance decreases. It has been tried to increase sensorless control capability by compromising torque value too little with a third design optimization.

The performance results for each design optimization have been given and compared. In addition to this, the effects of $\pm 5\text{-}10\%$ change for each design variable have been investigated.

2012, xii + 69 pages

Keywords: Motor design, Permanent magnet motors, Sensorless control, Particle swarm optimization

TEŞEKKÜR

Başta, bu araştırmanın konusu, çalışmaların yönlendirilmesi, sonuçların değerlendirilmesi ve yazımı aşamasında yapmış olduğu büyük katkı ve desteklerinden dolayı tez danışmanım Sayın Yrd. Doç. Dr. Murat CANER'e ve ardından, her konuda öneri, destek ve eleştirileriyle yardımlarını gördüğüm tüm arkadaşlarıma teşekkür ederim.

Her türlü desteğini hiçbir zaman benden esirgemeyen hayat arkadaşım Elif'e özel olarak şükranlarımı ifade etmek isterim.

Özkan ÖZLÜ

BOLU, 2012

İÇİNDEKİLER DİZİNİ

	Sayfa
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER DİZİNİ.....	vi
SİMGELER ve KISALTMALAR DİZİNİ	viii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ.....	xi
RESİMLER DİZİNİ	xii
1. GİRİŞ.....	1
2. LİTERATÜR BİLGİLERİ	5
2.1. Optimizasyon Algoritmaları	5
2.2. Elektrik Motoru Tasarımı İçin Optimizasyon Uygulamaları.....	10
3. MATERYAL VE METOT	15
3.1. Parçacık Sürü Optimizasyonu (PSO) Algoritması	15
3.1.1. Geliştirilen Algoritmanın Sınanması.....	22
3.1.1.1. Rastrigin Fonksiyonu.....	23
3.1.1.2. Michalewicz Fonksiyonu.....	26
3.1.1.3. Schwefel Fonksiyonu	27
3.3. Daimi Mıknatıslı Senkron Motorlar (DMSM)	29
3.3.1. Daimi Mıknatıs Malzeme Özellikleri.....	32

3.3.2. DMSM'nin Avantajları ve Dezavantajları	33
3.3.3. DMSM'nin Sensörsüz Kontrolü.....	33
3.4. Motor Tasarımı	35
3.5. Motor Tasarım Optimizasyonu.....	38
4. BULGULAR	45
4.1. Optimizasyon Algoritmasının Performansı	45
4.2. Motor Tasarım Optimizasyonuna Ait Sonuçlar.....	46
5. TARTIŞMA VE SONUÇ	52
6. KAYNAKLAR.....	60
6.1. İnternet Kaynakları	66
ÖZGEÇMİŞ.....	67
EKLER	68

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler

AlNiCo	Alüminyum-Nikel-Kobalt
CrFe	Krom-Demir
CrW	Krom-Tungsten
NdFeB	Neodmium-Demir-Boron
SmCo	Samaryum-Kobalt

Kısaltmalar

ASA	Artificial simulated annealing
DDMSM	Dahili daimi mıknatıslı senkron motor
DMSM	Daimi mıknatıslı senkron motor
DA	Doğru akım
DE	Diferansiyel evrim
FEA	Finite element analysis
GA	Genetik algoritma
gBest	Global best
KKO	Karınca kolonisi optimizasyonu
mH	milihenry
Mm	milimetre
ms	milisaniye
N	Newton
Nm	Newton-metre
pBest	Personal best
PID	Proportional – integral - derivative
PSO	Particle swarm optimization
PSOPC	Particle swarm optimization with passive congregation
TA	Tabu arama
W	Watt
YMDMSM	Yüzey montajlı daimi mıknatıslı senkron motor

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1	Deterministik optimizasyon modellemesi ve karar analizi.....2
Şekil 3.1	Doğrusal olmayan bir fonksiyon eğrisi.....16
Şekil 3.2	Fonksiyon üzerine rastgele yerleştirilmiş aday çözümler (parçacıklar).16
Şekil 3.3	Rastrigin fonksiyonu.....23
Şekil 3.4	Rastrigin fonksiyonu ile optimizasyon algoritmalarının sınanması.24
Şekil 3.5	Michalewicz fonksiyonu.....26
Şekil 3.6	Michalewicz fonksiyonu ile optimizasyon algoritmalarının sınanması.27
Şekil 3.7	Schwefel fonksiyonu.27
Şekil 3.8	Michalewicz fonksiyonu ile optimizasyon algoritmalarının sınanması.28
Şekil 3.9	Mıknatıs yerleşim şekillerine göre daimi mıknatıslı motorların rotor kısımları.30
Şekil 3.10	DMSM motor tasarım değişkenleri.36
Şekil 3.11	Optimize edilecek motora ait model görüntüsü.....37
Şekil 3.12	Motor tasarımı için geliştirilen PSOPC algoritmasının çalışması.41
Şekil 3.13	Ortalama tork ve tork dalgalanmasının ölçülmesi.43
Şekil 3.14	Tepeden tepeye endüktans değerlerinin ölçümleri.44
Şekil 4.1	Her üç optimizasyon için uygunluk değerinin değişimleri.....46
Şekil 4.2	Yapılan üç deney için bulunan optimum motor tasarım geometrileri.48
Şekil 4.3	Tasarım değişkenlerinin optimizasyon adımlarındaki değişimleri.49
Şekil 5.1	Birinci deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork ve tork dalgalanmasına etkileri.54

Şekil 5.2	İkinci deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork ve endüktans değerlerine etkileri.....	55
Şekil 5.3	Üçüncü deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork, tork dalgalanması ve endüktans değerlerine etkileri.	57

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1 Popülasyon sayısının arama başarısındaki etkisi.	25
Çizelge 3.2 Tasarımda kullanılan malzemeler	36
Çizelge 3.3 Tasarım değişkenleri ve sabitleri.....	37
Çizelge 3.4 Tasarım parametrelerinin alabilecekleri alt ve üst değerler.	38
Çizelge 3.5 Optimizasyon algoritmasında kullanılan sabitler.	40
Çizelge 4.1 Optimizasyon testlerindeki popülasyon, tekrar ve süre bilgileri.....	47
Çizelge 4.2 Tasarım parametrelerinin sınır değerleri ve deneyler sonunda aldıkları değerler.	47
Çizelge 4.3 Optimizasyon uygulamalarında elde edilen sonuçlar.	49

RESİMLER DİZİNİ

	Sayfa
Resim 3.1 PSO deneyleri için oluşturulmuş GUI görüntüsü.	22
Resim 3.2 YMDMSM'ye ait rotor görüntüsü (İnt. kyn. 4).....	31
Resim 3.3 DMSM yapımında kullanılan mıknatıs görüntüleri.....	32

1. GİRİŞ

İnsanlık daha önceleri çok uzun zaman alan işleri teknoloji sayesinde daha kısa sürede tamamlamayı başarmıştır. Bunlardan bir tanesi de hesaplaması çok uzun süren matematiksel işlemlerin, bilgisayar teknolojisi sayesinde, daha kısa sürede çözüme ulaştırılması olmuştur. Ancak günümüzde dahi bazı işlemler çok hızlı ve yetenekli bilgisayarlarla bile çözülememekte veya çözümü çok uzun zaman almaktadır. Dolayısıyla bu tür problemlerde çözüme farklı algoritma ve tekniklerle yaklaşmak gereği duyulmuştur.

Bilgi teknolojilerinin bilime katkıları kadar doğadaki olayların ve yaratıkların da teknolojinin gelişmesinde ve şekillenmesinde rolleri büyüktür. Doğada, sebebi yeni yeni anlaşılan yüzlerce oluşumun veya gelişimin, yapay ortamlarda taklit edilmesi yoluyla, bilim dünyasındaki problemlerin çözümünde kullanıldıkları görülmektedir. Uçaklar, kameralar, ultrasonik cihazlar veya insansı robotlar sadece ilk akla gelen örnekler olarak sayılabilir.

Teknolojinin gelişiminde, çevrede bulunan canlıların sadece fizyolojik ve biyolojik özelliklerinin taklit edilmesinin yanı sıra canlıların, sosyal ilişki ve davranış yönüyle de taklit edilmesine uzun yıllar önce başlanmıştır. Bu taklitler sayesinde gündelik hayatta karşılaşılan problemlerin çözümü için pek çok teknik ortaya çıkmıştır. Bunlardan biri de parçacık sürü optimizasyonu (PSO) algoritmasıdır. Bu algoritmanın geliştirilmesinde, doğadaki kuş ve balık sürülerinin ortak davranış ilişkilerinden ilham alınmıştır (Eberhart and Kennedy 1995).

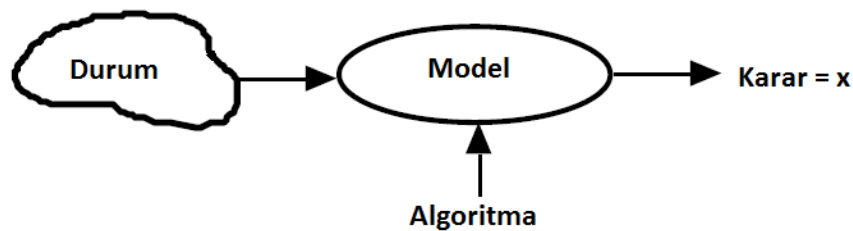
Bir sistemin kompozisyonunda pek çok değişken mevcuttur. Bu değişkenlerin birindeki değişim tüm sistemi, olumlu ya da olumsuz, etkileyebilir. Diğer bir deyişle, sadece bir boyutlu değişim her zaman faydalı olmayabilir. Bu yüzden sistemi optimize etmek için belirlenen amaçlara yönelik olarak değişkenlerin sadece birine değil, hepsine göre en uygun dengeyi bulmak gerekmektedir. Ancak hangi değişkenin, hangi değere ayarlanacağı asıl problemi oluşturmaktadır. Buradaki konu, sonsuz veya sonsuza yakın

sayıdaki ihtimaller içerisinde hangisinin en iyi olduğudur. Bu tür problemlerin çözümünde de hazırlanan optimizasyon algoritması rol alır.

Son yıllarda birtakım optimizasyon yöntemleri geliştirilmiştir. Bu yöntemler, optimum sonucu, işleme dayalı belli bir süreçle elde eden deterministik algoritmalar ile çözüm uzayında rastgele bir şekilde gezen stokastik algoritmalar olarak iki gruba ayrılmaktadır (Duan and Ionel 2011).

Bir dizi optimizasyon tekniği, farklı alanlardaki birçok araştırmacı tarafından öne sürülmüştür. Sayısız düzeltmeler ve geliştirmeler ile bu teknikler özgün uygulama türlerinde çalışmaktadırlar. Bütün bu prosedürler bazı ortak fikirler üzerine inşa edilmişlerdir. Bunun yanı sıra birkaç ek özgün özellikler ile karakterize edilmişlerdir. Optimizasyon algoritmalarında tekrarlı teknikler önemli bir rol oynarlar ve bu algoritmaların çoğunda doğrudan optimum (en iyi) çözüme ulaştıracak, genel bir prosedür yoktur (Hertz *et al.* 2000).

Deterministik optimizasyon algoritmaları, modele ait verilen parametreler ile en iyi çözümü belirler. Modelleme ve karar analizi süreci Şekil 1.1’de gösterilmiştir. Şekilde “durum” olarak belirtilen kısım, ele alınacak gerçek problemi temsil eder. Matematiksel bir model ortaya koymak için bazı varsayımlar, deterministik bilgisi de dâhil olmak üzere, probleme uygulanırlar. ‘X’ vektörü ile temsil edilen model, en iyi kararı belirleyecek olan algoritmanın olduğu bilgisayara girilir (Jensen and Bard 2003). Böylece optimum sonuç elde edilerek bir karara varılır.



Şekil 1.1 Deterministik optimizasyon modellemesi ve karar analizi.

Deterministik algoritmalar, daha az sayıda simülasyon ile optimum noktaya yakın sonuçlar verirler. Ancak lokal minimumlara takılıp optimum noktaya ulaşamama

ihtimali de vardır (Jensen and Bard 2003). Bunun yanında simülasyon sayısı, birbiriyle ilişkili olsun-olmasın, toplam tasarım değişken sayısına bağlıdır. Deterministik algoritmalar, başlangıç çözümü optimum çözüme yakın olduğu ve tasarım değişkenlerinin birbirleriyle ilişkilerinin iyi bilindiği durumlarda kullanmaya daha elverişlilerdir.

Stokastik optimizasyon algoritmaları, en iyi çözümün en kısa zamanda üretimini sağlarlar. Bilinen optimizasyon tekniklerinden en önemli farkı, türev almaya ihtiyaç duymadan çalışmalarıdır. Ayrıca, başlangıç çözümünün optimum çözüme çok uzak olduğu ve tasarım değişkenlerinin aralarındaki ilişkinin bilinmediği zamanlarda kullanılmaya uygundur (Günther *et al.* 2003). Bu sebeple, özellikle matematiksel modellerle ifade edilemeyen çok ihtimalli problemlerin optimize edildiği uygulamalarda tercih edilirler. Bu algoritmalar, doğrusal olmayan problemlerin çözümünde çok etkili sonuçlar verirler. Simülasyon sayısı tasarım değişkenlerinin sayısına değil, ilişkili değişkenlerin sayısına bağlıdır.

Deterministik optimizasyon algoritmalarında her giriş değeri için belirli bir çıkış vardır ve bu her zaman sabittir. Ancak stokastik algoritmalarda aynı giriş için her zaman aynı çıkış elde edilemez.

Gün geçtikçe; petrol rezervlerinin azalması ile insanlar alternatif enerji kaynağı olarak elektrik enerjisine yönelmektedirler. Dolayısıyla, kimyasal akümülatörlerle veya yakıt hücreleri ile çalışan sıfır emisyonlu taşıtların zorunluluk haline gelmesinden sonra gündelik hayatta kullanılan elektrik motorlarının kullanım alanlarının hızla yaygınlaşması sadece an meselesidir. Bu bağlamda çözüm olarak daimî mıknatıslı senkron motorlar (DMSM), düşük bakım maliyetine sahip olması ve yüksek verim sağlamasından dolayı geniş bir kullanım alanı bulmaktadır.

DMSM'lerin çalıştırılabilmesi için, hız ve konum bilgilerine ihtiyaç duyulmaktadır. Bu ihtiyacı karşılayan sensörler ise yüksek hızlarda kararlı davranmamaktadır. Ayrıca sensör ile kontrol gerektiğinde ek maliyetler açığa çıkmaktadır. Dolayısıyla, son yıllarda sensörsüz hız kontrolü önem kazanmıştır (Orman 2008).

Bu çalışma süresince yüksek tork ve sensörsüz kontrole uyum gibi avantajlarından dolayı ele alınan bir yüzey montajlı DMSM (YMDMSM) tasarımının optimize edilmesinde, PSO algoritmasının bir türevi olan Pasif Katılımlı Parçacık Sürü Optimizasyonu (PSOPC) algoritması kullanılmıştır. Pasif katılım etkisi ile, standart PSO algoritmasında göre, optimum noktaya daha kısa sürede ulaştığı ve daha verimli çalıştığı için bu algoritma tercih edilmiştir (He *et al.* 2004).

Motor, üç aşamada optimize edilmiştir. İlk olarak motorun tork verimliliği konusunda bir optimizasyonu gerçekleştirilmiştir. Daha sonra motorun sensörsüz kontrole uyumlu şekilde bir tasarım optimizasyonu gerçekleştirilmiştir. Ancak birinci ve ikinci durum tek başlarına bir anlam ifade etmedikleri için, üçüncü bir aşama ile tork değerinden taviz vermeden sensörsüz kontrole uyumlu bir motor tasarımı optimizasyonu denenmiştir. Çünkü motorun torku ve sensörsüz kontrol yeteneği arasında ters bir ilişki mevcuttur. Birindeki iyileşme diğere olumsuz yönde tesir etmektedir.

Yapılan deneyler ve izlenen metodoloji ile tork değerini belirli bir seviyede tutarken, sensörsüz kontrole uygun tasarım sonuçlarının elde edildiği ve daha önceden başka algoritmalarla yapılan benzer optimizasyon sonuçlarına göre makul ve kıyaslanabilir sonuçların elde edilmesi amaçlanmıştır.

2. LİTERATÜR BİLGİLERİ

2.1. Optimizasyon Algoritmaları

Verilen şartlar altında en iyi sonucun elde edilme işi olan optimizasyon için çok çeşitli algoritmalar mevcuttur. Optimizasyon alanındaki en önemli gelişmeler 18. yüzyılda Newton ve Lagrange tarafından yapılmıştır (İnt. Kyn. 1).

Genel olarak optimizasyon algoritmaları çok amaçlı ve çok modlu optimizasyon algoritmaları olarak ikiye ayrılır.

Çözülecek optimizasyon problemine birden fazla amaç yüklemek problemin karmaşıklığını artırır. Örneğin, yapısal bir tasarımı optimize etmek için, hem hafif hem de sağlam bir tasarım istenebilir. Bu durumda iki amaç arasında bir tezat oluşur. Böylece bir adet en hafif ve bir adet en sağlam tasarım ortaya çıkar. Sonsuz sayıdaki tasarımların bir kısmında ağırlıktan bir kısmında da sağlamlıktan ödün verilmiş olur. Hiçbir tasarımda bir amacı iyileştirirken diğerinden taviz vermemek mümkün değildir. Buradakine benzeyen problemlerin çözümünde çok amaçlı optimizasyon algoritmaları görev alır.

Çok modlu optimizasyon algoritmalarında birden fazla iyi çözüm bulunmaktadır. Bunların hepsi de global olarak iyi olabilirler. Çoklu çözüm elde etmek için kullanıldıklarında, tekrarlamalı yaklaşımlarından ötürü, klasik optimizasyon teknikleri tatminkâr derecede iş görememektedirler. Evrimsel algoritmalar çok-modlu optimizasyon görevlerinde çoklu çözümlerin elde edilmesinde daha yeteneklidirler.

Optimizasyon algoritmalarından biri olan doğrusal programlama, aralarında doğrusal ilişkilerin olduğu bir dizi gereksinim için verilen matematiksel bir modelde, (azami kâr veya asgarî maliyet gibi) en iyi sonucu elde etmek amacıyla kullanılan matematiksel bir yöntemdir. Doğrusal programlama ilk olarak 1939 yılında Leonid Kantorovich tarafından İkinci Dünya Savaşı'nda ordunun maliyetlerini azaltmak ve düşmanın kaybını arttırmak amacıyla geliştirilmiştir. Birkaç yıl gizli tutulan yöntem, 1947 yılında

George B. Dantzig'in yayınladığı bir metot ile bilim dünyasına katılmıştır. Doğrusal programlama yönteminin altında yatan teori, gözden geçirilecek muhtemel en iyi çözüm sayısını ciddi oranda azaltmaktır (İnt. Kyn. 2).

Yokuş Tırmanma tekniğinde ise tekrarlamalı yöntem kullanılır. Arama alanı tek bir noktadan başlayarak her bir adımda komşu noktalara bakılır. Eğer komşu nokta mevcut çözümden daha iyi sonuç veriyorsa en iyi çözüm olarak işaretlenir ve bir sonraki adımda o nokta kıyasa esas olarak alınır. Sonlanma şartına ulaşana kadar tekrarlama devam eder. Sonlanma şartı belirlenen limit adım sayısı olabileceği gibi algoritmanın artık daha iyi çözüm bulamaması da olabilir. Tekniğin dezavantajı, lokal minimum veya lokal maksimumlara takılmasıdır. Dolayısıyla aramaya farklı noktalardan başlamak gerekebilir. Başlama noktası rastgele, daha önceden belirlenerek veya önceki çözümler, ölçümler gibi diğer kısıtlara göre seçilebilir.

Karıncaların yaşayışlarından esinlenerek 1996 yılında Marco Dorigo tarafından bilim dünyasına kazandırılan Karınca Kolonisi (KK) algoritması, temel olarak karıncaların yiyecek ile yuvaları arasındaki yolu bulmaya dayanmakta olan popülasyon tabanlı bir optimizasyon algoritmasıdır. Karıncalar geçtikleri yerde feromon adında bir salgı bırakarak dönüşte bunun kokusuyla yollarını bulabilmektedirler. KK algoritması,. Her bir karınca problem için aday bir çözümdür ve diğer karıncaların hareketini etkiler.

KK algoritması ayrık zamanlıdır. Her adımda her bir karınca bir birim hareket eder ve her hareketinde bir miktar feromon salgılar. Ancak çözüm arayışında kistas olarak sadece feromon ele alınırsa lokal minimumlara takılma riski doğar. Dolayısıyla global minimum değerini bulmak için bir takım ihtimal hesaplamalarının yapılması gerekmektedir. Feromon, doğası gereği, uçucudur. Dolayısıyla az kullanılan izin kaybolma ihtimali vardır.

Stokastik optimizasyon algoritmalarından birisi de Benzetilmiş Tavlama (BT) tekniğidir. Kirkpatrick ve arkadaşları 1983 yılında ortaya atılmıştır. Katıların ısıtılıp belirli bir sıcaklıktan sonra kontrollü bir şekilde soğutularak kristal hale dönmesine dayanan bir mantığa sahiptir.

Kısaca anlatmak gerekirse, ısı, atomların başlangıçtaki konumlarından uzaklaşmalarını ve kristal yapının bozulmasıyla atomların hareketlenmesine sebep olur. Kristalin ısıtılmış durumu ile soğutulmuş durumu arasındaki enerji seviyesinin farkı negatif ise yeni durum daha düşük enerji seviyesinde demektir ve sonraki adımlarda kıyaslama bu enerji seviyesine göre yapılır (Karaboğa 2011). Uygunluk değerlendirmesi durdurma ölçütü sağlanana kadar devam eder.

Başlangıç sıcaklığı, her sıcaklıkta üretilecek çözüm sayısı fonksiyonu, Sıcaklık azaltma fonksiyonu ve algoritmayı durdurma şartı, BT algoritması için gereken parametrelerdir. Durdurma şartı tekrar sayısı limitine ulaşılması, belirlenen minimum sıcaklık değerine ulaşılması veya hedeflenen çözüme ulaşılması olabilir.

Tabu Arama (TA) algoritması, Fred W. Glover tarafından 1986 yılında öne sürülmüştür. Değerlendirme fonksiyonu olarak her adımda en iyi komşu çözüme göre kıyaslama yapar. Daha önceden belirlenen bir tabu listesi ile çözüm arayışına belirli sınırlamalar getirilir. Daha önceden bulunmuş ama en iyi çözüm olamamış aday çözümlere tekrar ulaşılması engellenir ve arama uzayı her adımda daraltılır. Algoritmada kısa ve uzun dönem hafıza mevcuttur. Kısa dönem hafıza ile daha önceden karşılaşılan bilgiler tutulur, önceki çözümlere dönüş engellenir. Kısa dönemli hafıza her adımda güncellenir. Uzun dönemli hafıza ise aramayı iyileştirir. Elit çözümlerin bir listesini tutar. İleride bu elit çözümleri oluşturan niteliklere yeniden dönüş yaparak bu bölgelerin daha detaylı araştırılmasını sağlar.

Evrin teorisini temel edinmiş olan Genetik Algoritma (GA) tekniği ise 1970'li yıllarda John Holland tarafından geliştirilmiştir. Popülasyon tabanlı, stokastik bir optimizasyon algoritmasıdır. Popülasyon içindeki her birey bir aday çözümdür. Aday çözümler genellikle dizi şeklinde ele alınırlar.

GA tekniğinde, uygunluk değerlendirmesi probleme yönelik olarak çalışan en önemli kısımdır. Çözüm için kullanılan bireylere kromozom adı verilir. Kromozom içindeki anlamlı en küçük bilgi gen olarak adlandırılır. Tekrarlamalı olarak çalışan algoritmada

her adım jenerasyon olarak düşünülür. Çaprazlama ve mutasyon, genetik operatörlerdir. Çaprazlama ile seçilen iki birey belirli tekniklere göre iki veya daha fazla parçaya bölünerek takaslanır. Bu sayede yeni bireyler oluşturulur. Uygunluk değeri daha iyi olan kromozom yaşamaya devam eder. Mutasyonda ise herhangi bir sebebe gerek duyulmaksızın birey genlerinde değişikliğe gidilmesi söz konusudur. Bir diğer terim olan elitizm ile sonraki nesillere iyi olduğu kabul edilen bireyler herhangi bir değişikliğe tabi tutulmadan aktarılır.

Tipik bir GA şunları gerektirir:

1. Çözüm kümesinin genetik temsili,
2. Çözüm kümesini değerlendirmek için bir uygunluk fonksiyonu.

Çözümün standart gösterimi bir dizi bittin meydana gelir. Dizilerin diğer türleri ve yapıları aynı şekilde kullanılabilirler. Asıl önemli olan bu genetik gösterimlerin çaprazlama işlemlerinde uygunluk sağlamasıdır.

Uygunluk fonksiyonu genetik temsil üzerinden tanımlanır ve temsil edilen çözümün kalitesini ölçer. Uygunluk fonksiyonu daima probleme bağlıdır.

Başlangıçta birçok bireysel çözümler rastgele olarak üretilirler ve bir popülasyon formu oluştururlar. Popülasyon sayısı her ne kadar probleme göre değişiklik gösterse de genelde birkaç yüz veya bin muhtemel çözümü içerirler.

Her bir başarılı nesil boyunca, var olan popülasyondan bir kısmı yeni bir nesil üretmek için seçilir. Bu seçim uygunluk değerlendirmesine göre yapılır. Daha uygun çözüm sunan bireyler seçilmeye daha layıktır (İnt. Kyn. 3).

Her ne kadar GA yeniden üretim yöntemlerinde biyolojiden mülhem iki ebeveynden türetilen yeni bireyler üzerine kurulu olsa da, bazı araştırmalar, ikiden fazla ebeveyne sahip kromozomların daha kaliteli olarak üretildiğini öne sürmektedirler (Eiben *et al.* 1994, Ting 2005).

Çaprazlama ve mutasyonun iki ana genetik işlem olmasına rağmen, yeniden gruplama, veya göç gibi diğer işlemleri kullanmak da mümkündür (Akbari 2011).

Yukarıda anlatılan nesiller türetme işlemi herhangi bir sonlandırma şartı sağlandığında sona erer. Bazı sonlandırma şartları şunlardır:

- Minimum ölçütü tatmin edecek bir çözüm bulunduğunda,
- Sabitlenmiş nesil sayısına ulaşıldığında,
- Bundan sonraki tekrarlar daha yüksek uygunluk değeri bulunmadığından optimizasyon uygunluk değerinin yatay bir seyir izlemeye başladığı gözlemlendiğinde,
- Programcının iradesiyle,

Basit bir GA prosedürü şu şekilde gerçekleşir:

1. Bireylerin oluşturacağı başlangıç popülasyonunu belirle,
2. İlgili popülasyon içindeki her bireyin uygunluk değerlendirmesini yap,
3. Algoritma sonlandırılana kadar şu işlemleri tekrarla:
 - a. En uygun bireyleri yeniden üretim için ayır,
 - b. Seçilen bireyler üzerinden çaprazlama ve/veya mutasyon işlemleriyle yeni bireyler üret,
 - c. Yeni bireylerin bireysel uygunluk değerlendirmelerini yap,
 - d. Popülasyon içinde en az uygun olanlarla yeni bireyleri yer değiştir,

Popülasyon tabanlı stokastik optimizasyon tekniklerinden bir diğeri PSO'dur. İlk olarak 1995 yılında, J. Kennedy ve R. C. Eberhart tarafından bilim dünyasına sunulmuştur. Doğada sürü halinde dolaşan canlıların ortak davranışlarını taklit ederek geliştirilmiştir. Doğrusal olmayan problemleri çözmede başarısı kayda değerdir.

PSO algoritması GA ve diğer evrimsel algoritmalarla kıyaslandığında bir takım çekici özelliklere sahip olduğu görülür. Bir yandan, ayarlanacak çok az parametreye sahip olması yönüyle, parametrelerin optimum değerlerinin sağlanması için gereken hesaplama sayısı ve zaman yönünden tasarruf sağlarken, diğer yandan da çok hızlı bir şekilde optimum noktaya yaklaşabilmektedir. Çünkü bir adımda sadece iki hesaplama yapılmaktadır. Bu avantajlara rağmen, karmaşık optimizasyon problemlerinin çözümünde yerel en iyi noktalarda kalması da mümkündür (Khajehzadeh *et al.* 2010).

PSO algoritması, aramaya rastgele noktalardan başlar, GA’da olduğu gibi çözüm uzayında aday çözümler belirli noktalara rastgele veya belirli olarak konumlandırılırlar. Tekrarlama tekniğinin kullanıldığı algoritmada, ilgili adımda bütün parçacıklar uygunluk değerlendirmesine tabi tutulur ve bunlardan bir tanesi ilgili adım için optimum çözüm olarak belirlenir. Sonraki adımlarda, kıyaslama, bir önceki adımdaki optimum çözüme göre yapılır. Durdurma şartı sağlanana kadar işlem devam eder.

PSO algoritmasını uygulamak için gereken parametre sayısı diğer yaklaşımlara göre daha azdır. Her parçacığın bir konum ve hız bilgisi vardır. Bunun yanında her parçacık ulaştığı en iyi çözüm noktasını hafızasında tutar. Diğer bir deyişle, daha önce ulaştığı konumları sonraki adımlarında kıyaslamak üzere hafızaya kaydeder. Her adımda diğer parçacıklar en iyi olarak belirlenen parçacığa doğru hareket ederler. Bu sayede adımlar ilerledikçe bütün parçacıklar optimum çözüme doğru yaklaşmış olurlar. Parçacıkların hareketlerinin hızını belirleyen bir atalet katsayısı, optimum çözüme yaklaştıkça yavaşlamasını sağlar.

Yakın geçmişte, PSO algoritmasının performansının artırılması amaçlayan pek çok araştırma yapılmıştır (Mendes *et al.* 2004, He and Han 2006). Ancak bu algoritmanın geliştirilmesinde en iyi çalışmayı He ve arkadaşları (2004) başarmışlardır. Pasif katılım etkisi sayesinde standart PSO algoritmasının yakınlık ve doğruluk oranı ve verimli şekilde artırılmıştır (Khajehzadeh *et al.* 2010).

Bu tez çalışmasında yukarıda bahsi geçen algoritmalar incelenmiştir. PSO algoritmasının kullanım kolaylığı, geliştirmeye açık, diğer algoritmalara göre daha anlaşılır olması, çok modlu ve çok amaçlı kullanıma yatkın olması, kullanılması için bir tercih sebebi olmuştur.

2.2. Elektrik Motoru Tasarımı İçin Optimizasyon Uygulamaları

Leonardi vd. (1994) tarafından, Güney Avrupa Büyük Gözlem Teleskopu’na ait konum kontrol sisteminde kullanılan DMSM tasarımı, sonlu elemanlar analizi yöntemiyle

optimize edilmiştir. Çok düzgün bir tork elde etmek için tork dalgalanması çok düşük olması gereken bir motorun diş şekli, hava boşlukları, statorun uçları ve diş kafası kalınlığı optimize edilmiştir. Stator ünitesinin uzunluğu en önemli tasarım parametresi olarak tanımlanmıştır. Stator uçlarına ait tasarım optimizasyonu çok etkili olmamıştır. % 0,1 olması istenen tork dalgalanması % 0,05 olarak çıkmıştır ki bu değer oldukça düşüktür. Optimizasyon sürecinde, diş kafa kalınlığını 1,25 mm'den 2 mm'ye çıkartarak, stator dişleri üzerindeki kuvvet 8,5 N'den 7 N'ye düşürülmüştür.

Bianchi ve Bolognani (1998) tarafından kare ve sinüzoidal dalga ile beslenen DMSM'ler üzerinde ağırlık, tork, verimlilik ve sargı sıcaklığı gibi amaçları optimize etmek için genetik algoritma tabanlı optimizasyon teknikleri geliştirilmiş ve uygulanmıştır. Klasik yokuş tırmanma ve doğrudan arama metotları da ayrıca uygulanmıştır. Buna ek olarak analitik ve sonlu elemanlar analizi yaklaşımları motor performansını tahmin etmek için GA ile bağlanmıştır. Diğer geleneksel tekniklere nazaran GA genellikle daha çok tekrarlama gerektirmektedir. Ancak GA'nın en büyük avantajlarından birinin de yerel optimum değerlerden sıyrılıp global optimum değeri bulma konusunda başarılı olmasıdır. GA ile bulunan çözümlerin doğrudan arama yöntemleriyle elde edilen çözümlerden daha iyi olduğu saptanmıştır. GA'nın, optimizasyon prosedürü için iyi bir başlangıç noktası gerektirmemesiyle beraber en uygun değere doğru yakınsayabildikleri gözlenmiştir.

Łukaniszyn vd. (2004) tarafından, daha düşük vuruğu momenti için GA ile daimi mıknatıs şekli optimize edilmiştir. Servo motorlar, sabit disk motorları gibi uygulamalarda vuruğu momenti toleransı hiç yoktur. Dört dilimli bir mıknatıs şekli optimum çözüm olarak bulunmuş ve bunun uygulanması ile ilk durumdaki tasarıma göre vuruğu momentinde %97'lik azalma meydana gelmiştir.

Vaez-Zadeh ve Ghasemi (2005) tarafından, düşük mıknatıs hacmi ve yüksek tork dengesini elde etmek için daimi mıknatıs boyutları ve konumları için arama yapan bir optimizasyon algoritması dâhili daimi mıknatıslı bir motor üzerinde denenmiştir. Moment analizleri göstermiştir ki motor momenti, mıknatıs genişliğinden ziyade uzunluğuna bağlıdır. Bu yüzden tasarım optimizasyonu daha uzun ve daha ince daimi

mıknatıs kutbu sonucunu vermiştir. Böyle bir mıknatıs topolojisi daha az mıknatıs hacmi hedefine katkıda bulunmuştur. Tasarım optimizasyonu, mıknatıs kutuplarının rotor yüzeyinden çok uzakta olmaması gerektiğini göstermiştir. Sonuçta mıknatıs hacminde % 45 azalma ve momentte % 15 artış elde etmişlerdir.

An vd. (2007) tarafından yapılan çalışmada, yüksek verimli bir DMSM tasarımına ait mıknatıs optimizasyonu, kaotik optimizasyon ve PSO algoritmalarının birleştirilmesinden oluşan hibrid bir optimizasyon algoritması ile gerçekleştirilmiştir. Bu birleşimden manyetik alan hesaplamasının hızını arttırabilen ve şeklin, kutup-yay açısının, motor boyutunun küresel minimizasyonunu bulabilen bir teknik elde edilmiştir. Böylece optimum çözüme daha hızlı yaklaşma avantajı yakalanmıştır. Ek olarak, ana boyutları değiştirmeden, daimi mıknatısların ve motorun etkin malzemelerinin ağırlığının azaltılması başarılmıştır. Tasarlanan motorlar fabrika sınavasından geçirilmiş, enerji tasarruf kıyaslamasına tabi tutulmuştur. Sınama sonuçlarında, aynı şartlardaki asenkron motora göre güç verimliliğinde çok daha fazla artış gözlenmiştir. Test sonuçları, tasarım hedefinin yakalandığını ve tatmin edici nitelikte olduğunu göstermiştir.

Duan vd. (2008) DMSM tasarımı için bir analitik model geliştirmişlerdir. Bu modelde hava boşluğu çapı ve motorun eksensel uzunluğu serbest değişken olarak belirlenmiş diğer tüm değişkenler kullanıcıların performans gereksinimlerini karşılayacak şekilde ayarlanmıştır. Modeli FEA ile doğruluk kontrolünden geçirmişler ve analitik model üzerinde optimizasyon için PSO algoritmasını kullanmışlardır. Analitik model ile PSO algoritmasının birleşimiyle DMSM için hızlı ve verimli birçok-amaçlı tasarım yöntemi sağlanmıştır.

Brandon ve Scott (2009) tarafından gerçekleştirilen çalışmada, GA ve PSO gibi popülasyon tabanlı optimizasyon algoritmaları ile DMSM tasarımı yüksek verimlilik ve yüksek tork yoğunluğu yönleriyle optimize edilmiştir. Motora ait, dördü malzeme özelliği olmak üzere, on yedi fiziksel tasarım ögesi optimizasyon sürecinde yer almıştır. Optimizasyon algoritmalarının gerçekten global optimum noktaya ulaştığından veya

lokal optimumda kalmadığından emin olmak için aynı işlem 25 defa tekrarlanmıştır. Algoritma performansının probleme bağlı olduğu görülmüştür.

Duan vd. (2009) tarafından, DMSM tasarım optimizasyonunda GA ve PSO arasında doğruluk, sağlamlık ve algoritma katsayıları yönüyle bir kıyaslama gerçekleştirilmiştir. Zaman sınırlamasının olduğu durumlarda PSO algoritmasının GA'ya göre daha cazip olduğu yapılan kriter testlerinden sonra ortaya çıkmıştır. Hacim, ağırlık, verimlilik, mıknatıs ağırlığı ve akımdaki 1A değişime karşılık tork değerindeki değişim amaç fonksiyonunun bileşenlerini oluşturmuştur. On sekizer defa çalıştırılan algoritmaların sonunda PSO algoritmasının bulduğu en iyi ve en kötü uygunluk değerleri GA'nın bulduklarından daha iyi bir noktada olduğu gözlenmiştir. Standart sapması da daha düşük çıkmıştır. Doğru optimum değeri bulma sayısı PSO algoritmasında 7; GA'da 3 olarak kaydedilmiştir. Elde edilen verilere göre her iki algoritmanın da optimum noktayı bulma konusunda kabiliyetli oldukları ancak PSO algoritmasının başarısının biraz daha iyi olduğu söylenmiştir.

Hasanien (2011) tarafından, daimi mıknatıslı çapraz akılı doğrusal motor tasarımı optimizasyonu, PSO algoritması ile yapılmıştır. İlk tasarlanan daimi mıknatıslı motor için itme kuvvetini arttırmak, motor ağırlığını ve tutma kuvvetini azaltmak amacıyla optimizasyon yapılmıştır. Stator kutup uzunluğu, hava boşluğunun uzunluğu, sargı penceresinin genişliği ve stator kutup genişliği optimizasyon problemi için arama uzayını belirlemiştir. Kısaca, yapılan çalışmada hızlı tasarım ve doğru performans karşılaştırılması konularında başarılı olunmuş, bu yaklaşım uygulandığında optimizasyonun kararlılığını arttırmada daha verimli olduğu söylenmiştir.

Mutluer ve Bilgin (2011) tarafından yapılan çalışmada DMSM tasarım optimizasyonu için kullanılan üç optimizasyon uygulaması; GA, ASA ve diferansiyel evrim (DE) algoritmaları karşılaştırılmıştır. Maliyet, ağırlık, tork dalgalanması, verim ve rotordaki daimi mıknatıslar ile stator slotları arasındaki etkileşimden oluşan, vurutu momenti değerleri optimize edilmiştir. GA, ASA ve DE algoritmaları ile optimize edilen motor tasarımları FEA metoduyla analiz edilmiş ve sırasıyla %94,07, %94,57 ve %95 oranlarında verim değerleri elde edilmiştir. Bunun yanı sıra diğer iki optimizasyon

algoritmasından yaklaşık olarak yedi kat daha hızlı bir şekilde sonuç veren DE algoritması, diğerlerinden daha başarılı çıkmıştır.

Yukarıda bahsi geçen çalışmalara ve literatürdeki diğer pek çok süreli yayınlara bakıldığında DMSM'lerin rotorlarına ait konum tespiti için sensörlerden faydalanmaktan vazgeçildiği; atlanamayacak avantajlarından dolayı sensörsüz kontrole uygun tasarımların hedeflendiği bir eğilim gözlenmektedir. Bu çalışmada da buna bağlı olarak bir DMSM tasarımında, azami tork, azami sensörsüz kontrol yeteneği ve asgari tork dalgalanması optimize edilecektir.

Çalışmada optimize edilecek birden fazla amaç bulunmaktadır. Optimum tasarımın, arama uzayının hangi noktasında olduğu bilinmemektedir. Çünkü sistemin matematiksel modeli doğrusal bir fonksiyonla ifade edilememektedir. Optimum tasarımın arama uzayında birden fazla noktada olma ihtimali de vardır. Problemin birden fazla çözümü de olabileceği için, optimizasyon algoritmasının çok modlu olduğu söylenebilir.

PSO türevleri araştırıldığında, sensörsüz kontrole uyumlu, minimum tork dalgalanmasıyla maksimum tork yoğunluğunu sağlayabilecek bir DMSM tasarım optimizasyonu için, klasik PSO uygulamasına göre daha kararlı bir şekilde ve kısa sürede sonuca götürmesi göz önünde bulundurularak (He *et al.* 2004), pasif katılımlı PSO (PSOPC) algoritmasının kullanılmasına karar verilmiştir.

3. MATERYAL VE METOT

Bu bölümde, DMSM'lerin yapısı, çeşitleri, çalışma şekli ve sensörsüz kontrol yöntemleri bu bölümde ele alınmıştır. PSO algoritmasının çalışması hakkında detaylı bilgi verilmiştir. PSO ve PSOPC algoritmaları arasında, Rastrigin, Schwefel ve Michalewicz fonksiyonlarında minimum noktayı arama performanslarına göre bir kıyaslama yapılmıştır.

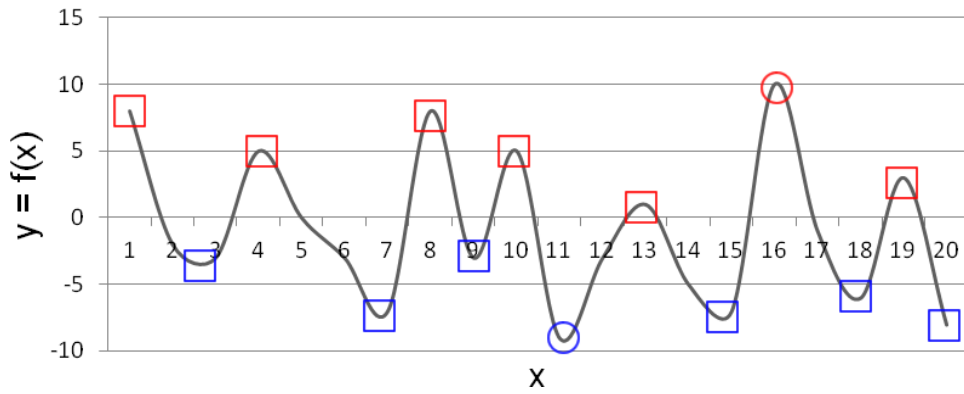
Optimizasyon sürecinde uygunluk değerlendirmesi Infolytica Magnet programı tarafından gerçekleştirilmiştir. Program verilen tasarım parametrelerinin değerleriyle motor tasarımını gerçekleştirmekte ve amaca göre, tork, tork dalgalanması ve sensörsüz kontrol yeteneğini hesaplamak için simülasyon işlemini koşturmaktadır. Elde edilen sonuç değerleri MATLAB programı tarafından geri okunarak uygunluk değerlendirmesine tabi tutulmuştur.

PSO algoritmasının motor tasarım optimizasyonuna uygun şekle getirildikten sonra motor tasarım optimizasyonuna geçilmiştir. Üç defa yapılan optimizasyon çalışmaları sonucunda optimum motor tasarımları elde edilmiştir. Birinci aşamada düşük tork dalgalanmasına ve yüksek torka sahip bir tasarımın optimizasyonu amaçlanmıştır. İkinci aşamada, sensörsüz kontrol yeteneği yüksek bir motor tasarımı, tork değerleri göz önüne alınmadan optimize edilmiştir. Hem tork ve tork dalgalanması açısından, hem de sensörsüz kontrol kabiliyeti açısından çok amaçlı bir optimizasyon çalışması ise üçüncü aşamada gerçekleştirilmiştir.

3.1. Parçacık Sürü Optimizasyonu (PSO) Algoritması

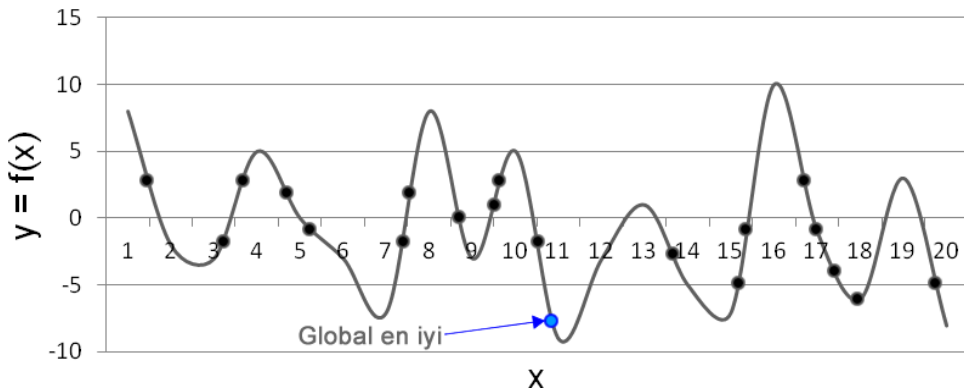
PSO algoritması, verilen bir problemin gerekli parametrelerini, belirli bir amaca göre maksimize veya minimize etmek için gereken çözümü, arama uzayında bulmakta kullanılan bir tekniktir. Bu teknik ilk defa James Kennedy ve Russel C. Eberhart (1995) tarafından ortaya atılmıştır. Bazı sürü halinde yaşayan hayvanların (kuşlar ve balıklar) sürü alışkanlık ve davranışlarının gözlemine dayalı sürü zekâsı fikri ve evrimsel hesaplama alanı fikri olmak üzere iki ayrı orijinden gelir.

Şekil 3.1’de, $y = f(x)$ şeklinde tanımlanmış, doğrusal olmayan bir fonksiyon eğrisi verilmiştir. Verilen bu örnek fonksiyonda birden fazla tepe değeri vardır. Optimizasyonda amaç, fonksiyon eğrisindeki herhangi bir tepe noktasını bulmaktan ziyade, en yüksek veya en alçak değer noktasına yaklaşılmaktır. Maksimizasyon uygulamalarında en iyi değer, girilen x değerine karşılık elde edilen en yüksek y değeridir. Minimizasyon uygulamalarında ise hedef, girilen x değerine karşılık elde edilen en küçük y değerine ulaşmaktır.



Şekil 3.1 Doğrusal olmayan bir fonksiyon eğrisi.

Şekil 3.2’de, fonksiyon eğrisi üzerinde rastgele atanmış parçacıkların, yerleşimleri gösterilmektedir. Yatay eksen giriş değişkenine ait değeri, dikey eksen ise sonuç değerini göstermektedir. Mavi renkli noktanın, bir minimizasyon uygulaması için global en iyi nokta olduğu söylenebilir. Çünkü tüm parçacıklar içinde mavi renkli parçacığın y eksenindeki değeri en küçüktür.



Şekil 3.2 Fonksiyon üzerine rastgele yerleştirilmiş aday çözümler (parçacıklar).

Anlatımı daha kolay kılmak için optimizasyon uygulamalarında minimizasyon yöntemi ele alınacaktır ve bir aramadaki minimum nokta optimum nokta olarak düşünülecektir.

Başlangıçta PSO algoritması arama uzayı içinden aday çözümleri rastgele seçer. Amaç fonksiyonunun ne olduğunu ve hangi aday çözümün küresel veya yerel en iyi değerlere yakın ya da uzak olduğunu bilmez. Sadece amaç fonksiyonunu aday çözümleri değerlendirmek için kullanır ve edinilen uygunluk değerine göre işlem yapar.

Her parçacığın bir konumu ve bir hızı mevcuttur. Uygunluk değerlendirmesinde konumu gBest değerine uzak olan parçacığın hızı yakın olanlara göre daha fazladır. Zira uzaktaki parçacığın daha kısa sürede en uygun değere ulaşması gerekmektedir.

Her bir parçacık kendi konumunu (x), uygunluk değerini ve hızını (v) muhafaza etmektedir. Buna ek olarak optimizasyon süreci içinde şimdiye kadarki elde ettiği en iyi değeri (pBest) ve tüm parçacıklar arasında en iyi değeri (gBest) hafızasında tutar. Genellikle arama adımları ilerledikçe pBest ve gBest değerleri de değişir. Bir noktadan sonra artık değişmediği gözleniyorsa optimum noktalardan bir tanesine ulaşılmış demektir. Ancak bu fonksiyondaki yerel veya global minimum noktaya ulaşıldığı anlamına gelmez.

Her adım sonrasında, diğer parçacıkların hız vektörleri, gBest değerine sahip parçacığın bulunduğu noktaya doğru hareket edecek şekilde güncellenir. Parçacıklar birbirlerine yaklaştıkça hareket alanları da daralır. Bu sayede, her adımda, arama uzayı bir bakıma küçülmüş olur. Bunun avantajı, optimum noktanın çok kısa sürede bulunmasıdır. Ancak arama uzayı içinde global minimuma, hem konum hem de değer olarak, çok yakın lokal minimumlar olabilir. Diğer taraftan, gBest değerine sahip parçacık, böyle bir lokal minimum noktayı bulmuş olabilir. Geçerli gBest noktasına doğru hareket eden parçacıklar, sonraki adımlarda, arama kapsamı daralacağı için, arama uzayının dışında kalmış olan global minimum noktasını hiç bulamayabilirler. Bu da bir dezavantaj olarak görülebilir. Bunun önüne geçmek için literatürde çeşitli yöntemler geliştirilmiştir.

PSO algoritması durdurma şartı sağlanana kadar tekrar eden üç ana adımdan oluşur (Van Den Bergh 2001):

1. Her bir parçacığın uygunluk değerlendirilmesi yapılır.
2. pBest ve gBest uygunluk değerleri güncellenir.
3. Her bir parçacığın hız ve konumları güncellenir.

Uygunluk değerlendirilmesi, amaç fonksiyonuna aday çözümün uygulanması ile yürütülür. Önceki uygunluk değeri ile şimdiki gBest ve pBest değerleri kıyas edilerek değerlendirme yapılır. Yeni değerler önceki adıma göre daha iyi ise güncellenir, değilse herhangi bir işlem yapılmaz.

PSO algoritmasındaki mantığa göre, popülasyon içindeki bir parçacığın hızı, gBest değerine sahip parçacıkla aralarındaki mesafe ile doğru orantılıdır. Böylece parçacık, optimum noktaya çok daha çabuk yaklaşabilmektedir.

Optimizasyon uygulamasında, minimum noktaya en yakın olan parçacık gBest olarak belirlenir. Bir sonraki adımda her bir parçacığın yeni konumunun belirlenmesinde kullanılan hız vektörleri hesaplanırken gBest olarak belirlenmiş parçacığın konumu dikkate alınır. Bu şekilde her adımda hem gBest, minimum noktaya, diğer parçacıklar da gBest değerine sahip parçacığa doğru yaklaşırlar.

Bir sistemi optimize etmek için gereken parametre değerlerini aramak, kuşların yerini bilmediği yiyecekleri aramalarına benzer. Hangi parametrenin hangi değere ayarlanacağı belirsizdir. Kuşlardan birisi yiyecek bulduğu zaman, sürüdeki diğer kuşlara da haber verir. Diğer kuşlar da yiyeceğe en yakın olan kuşa doğru uçmaya başlar. Buradaki her bir kuş olası bir çözüm olarak düşünülebilir. Her bir aday çözümün pozisyonu uygunluk değerlendirmesine alınır. Yiyeceğe en yakın kuş gBest olarak işaretlenir.

PSO algoritması, doğrusal olmayan problemleri optimize etmek için kullanılır. Başlangıç durumunda optimum noktanın çok uzağında bulunduğu ya da optimum nokta

ile ilgili hiçbir fikrin olmadığı anlarda çok kullanışlıdır. Çok parametreye sahip (çok boyutlu) optimizasyon problemlerinde başarılı sonuçlar vermiştir.

PSO algoritması, GA gibi evrimsel hesaplama yöntemlerine benzerlik gösterir. Algoritma öncelikle parçacıkları arama uzayında rastgele noktalara yerleştirir. Her adımda (jenerasyonda) konumlar güncellenerek optimum noktaya doğru bir hareket başlar. PSO algoritmasında parçacık olarak adlandırılan aday çözümler o andaki gBest parçacığı sonraki adımlarda takip ederler. Ayarlanacak parametre sayısı az olduğu için PSO algoritmasının uygulaması çok esnek ve kolaydır.

PSO algoritması, birkaç aday çözümün arama uzayında eş zamanlı olarak denenmesiyle çalışan, ayrık zamanlı bir algoritmadır. Algoritmanın her adımında, her bir aday çözüm optimize edilecek amaç fonksiyonu tarafından değerlendirmeye alınır. Her bir aday çözüm, amaç fonksiyonunun minimum değerini bulmak için çözüm uzayında dolaşan bir parçacık olarak düşünülebilir.

PSO algoritmasında tanımlanması gereken iki ana unsur vardır: Bir tanesi arama uzayının kaç boyutlu olacağı, diğeri ise uygunluk ölçütünün ne olduğudur. Arama uzayının boyut sayısı optimize edilecek parametre sayısı kadardır. Örneğin 4 parametrelili bir optimizasyon dört boyutlu bir arama uzayında gerçekleştirilmelidir. (3.1)'deki fonksiyonu sıfıra minimize etmek için, en uygun x ve y değişkenlerini aramak gerekmektedir. Fonksiyonda iki adet değişken olduğu için arama uzayı iki boyutludur. (3.2)'de ise x, y ve z olmak üzere üç adet değişken söz konusu olduğu için arama uzayı üç boyutlu olacaktır. Fonksiyonların alması istenen değer uygunluk ölçütüdür.

$$f_a = x^2 - \frac{5y}{4} + 6 \quad (3.1)$$

$$f_b = 3x - 2y - z \quad (3.2)$$

Başlangıçta bütün parçacıkların hızları belirli bir değere ayarlanabilir. Temelde PSO algoritmasının hız vektörü (3.3)'e ve yeni konum (3.4)'e göre güncellenir.

$$v_i^{(k+1)} = v_i^k + r_{1i} (p_i - x_i^k) + r_{2i} (G_i - x_i^k) \quad (3.3)$$

$$x_i^{(k+1)} = x_i^k + v_i^k \quad (3.4)$$

Burada,

- i : parçacık numarası,
- k : iterasyon numarası
- v : parçacığın hızı
- x : parçacığın konumu
- r_1 : 0 ile 1 arasında rastgele ondalıklı bir sayı
- r_2 : 0 ile 1 arasında rastgele ondalıklı bir sayı
- p : parçacığın o ana kadar elde ettiği en iyi uygunluk değerini döndüren parçacık (pBest)
- G : Tüm popülasyon içinde o ana kadar elde edilen en iyi uygunluk değerini döndüren parçacık (gBest) anlamına gelir.

Bir kuş yiyeceğe doğru sürekli sabit hızla uçmaz. Yiyeceğe doğru yaklaştıkça hızını azaltır ve yiyeceğin bulunduğu noktada durur. Bu davranışı algoritmaya yansıtmak için hız vektörüne atalet katsayısı ω getirilmiştir. ω 'nın her adımda azalmasıyla, her adımdan sonra hız vektörleri daha küçük olacaktır. Bu sayede arama algoritmasının kararlılığı artmış ve arama daha az sayıda tekrar ile başarı sağlamıştır. Khajehzadeh (2010), ω değerinin değişimi için (3.5)'teki formülü kullanmıştır. Böylece atalet katsayısı ω , her adımda doğrusal olarak azalacaktır.

$$\omega_k = \omega_{max} - (\omega_{max} - \omega_{min}) \times \frac{k}{T} \quad (3.5)$$

ω_{max} ve ω_{min} , atalet değerinin en yüksek ve değerinin en düşük değerleridir. T , yapılacak tekrar sayısını; k , iterasyon numarasını temsil eder.

Hız vektörünün güncellenmesi işleminde kişisel tecrübe ile sosyal etkinin ağırlıkları kontrol edilebilir. (3.6). Bu, β_1 ve β_2 katsayıları ile mümkündür. β_1 , kişisel tecrübenin,

β_2 ise popülasyonun tecrübesinin hız vektörüne etkisini belirler. Güncellenecek hız vektörü hesaplanırken bu katsayılar ile hangi bilginin daha ağırlıklı olacağı belirlenir. Bu değerler optimizasyonun amacına göre değiştirilebilir.

$$v_i^{(k+1)} = r_{0i} \omega_i v_i^k + \beta_{1i} r_{1i} (p_i - x_i^k) + \beta_{2i} r_{2i} (G_i - x_i^k) \quad (3.6)$$

Bu tez çalışmasında kullanılan optimizasyon algoritması, PSO türevlerinden biri olan PSOPC algoritmasıdır. Pasif katılım etkisi, popülasyon içerisinde rastgele bir parçacığın konum bilgisinin hız vektörüne eklenmesidir. Bu işlem, parçacık, optimum noktaya yaklaşırken yaklaştığına dair bilgiyi bir başka parçacıktan teyit etmesi gibi düşünülebilir. Bu sayede arama daha küresel bir hal alır ve optimum çözüme doğru yaklaşıırken r_0 , r_1 ve r_2 rastgele sayılarından ötürü aksi yönde bir hareket ihtimali düşürülmüş olur.

PSOPC algoritmasına ait hız vektörünün formülü (3.7)'de verilmiştir. β_3 , pasif katılım etkisinin ağırlığını belirleyen katsayıdır. r_3 , 0 ile 1 arasında rastgele bir sayıdır. z , rastgele seçilmiş bir parçacık, ∂ ise, popülasyon içerisinden bir parçacığı seçebilmek için gereken, 1 ile popülasyon sayısı arasında bir tamsayıdır.

$$v_i^{(k+1)} = r_{0i} \omega_i v_i^k + \beta_{1i} r_{1i} (p_i - x_i^k) + \beta_{2i} r_{2i} (G_i - x_i^k) + \beta_{3i} r_{3i} (z_{\partial i} - x_i^k) \quad (3.7)$$

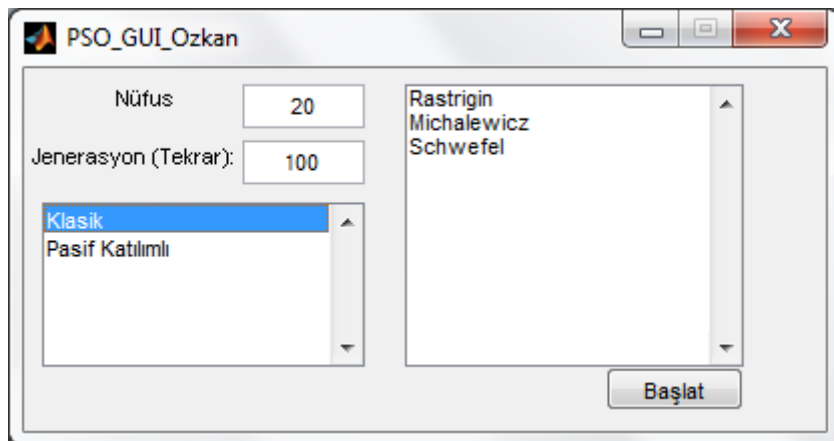
Arama uzayı ne kadar geniş olursa olsun, PSO algoritmaları için çok fazla kalabalık, parçacıklar arasında kararsızlıklar oluşmasına, böylece farklı iki nokta arasında parçacıkların hareket etmesine ve optimizasyonun çok uzun veya sonsuza kadar sürmesine yol açmaktadır. Çok fazla kalabalık bir popülasyonda, sürü zekası, yerini fikir ayrılıklarına bırakmaktadır. Nüfusun çok تنها olması ise sürü içerisinde yanlış kararlar alınmasına, bir diğer deyişle, aramanın, lokal minimumlarda son bulmasına sebep olmaktadır. Bu yüzden yine arama uzayındaki aramalar çok uzun sürmekte veya hiç bitmemektedir. Etkili bir optimizasyon için sürü popülasyonu 15 ile 25 arasında tutulmalıdır (Clerc 2006). Optimizasyon algoritmasının geçerliliğinin test edildiği çalışmalarda, 10, 20, 30, 40 ve 50 parçacıktan oluşan sürüler, Rastrigin fonksiyonu

üzerinde arama yapmaları sağlanmıştır. Aramalara ilişkin sonuçlara ait grafikler EK 1’de verilmiştir.

3.1.1. Geliştirilen Algoritmanın Sınanması

MATLAB ortamında hazırlanan optimizasyon algoritması, gerçekten işe yarayıp yaramadığını anlamak için, öncelikle, yerel minimumları fazla olan, optimizasyon algoritmalarını sınamak için yapılmış, bazı özel amaç fonksiyonları üzerinde denenmiştir.

Kullanımı kolaylaştırmak için bir grafiksel kullanıcı arayüzü (GUI) tasarlanmıştır (Resim 3.3). Arayüzde, popülasyon sayısını, tekrar sayısını, PSO algoritması türünü ve optimize edilecek amaç fonksiyonunu seçmek için çeşitli denetimler bulunmaktadır. Seçimler yapıldıktan sonra “Başlat” düğmesine basıldığında optimizasyon algoritması koşmaya başlar ve her adımda her bir parçacığın bulunduğu konum MATLAB pencerelerinden “Command Window” (komut penceresi) içinde listelenir. Bunun yanında optimizasyonu görselleştirmek için bir de üç boyutlu fonksiyon çıktısını gösteren grafik penceresi mevcuttur. Her adımda, küçük üçgenler ile temsil edilen parçacıklar, fonksiyon eğrisinin etrafında gösterilerek optimizasyon adımlarını ayrıık zamanlı olarak göstermektedir.



Resim 3.1 PSO deneyleri için oluşturulmuş GUI görüntüsü.

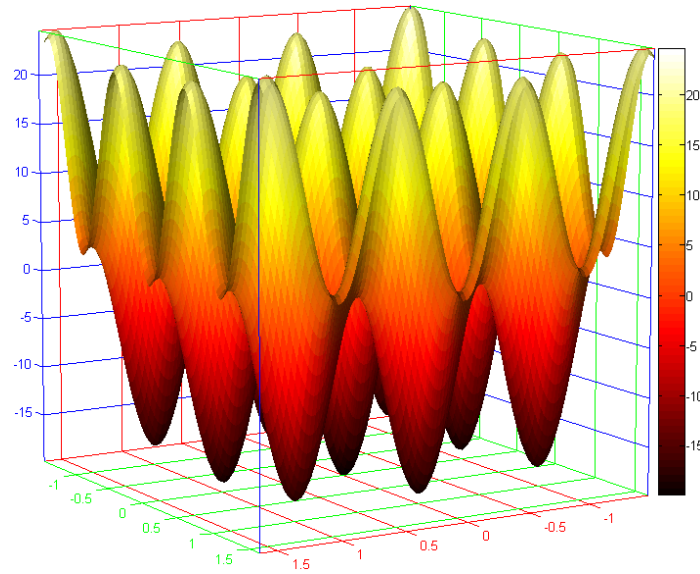
Yazılan optimizasyon algoritması programı, çok fazla yerel minimum noktasına sahip, diğer bir ifade ile parçacıkların aramasını oldukça zorlaştıran fonksiyonlar olan

Rastrigin, Michalewicz, Schwefel fonksiyonları üzerinde denenmiştir. Her fonksiyon için 20 parçacıktan oluşan bir sürü ile 100 defa tekrar eden bir arama gerçekleştirmiştir.

Gerçek hayattaki modellerde, tasarım optimizasyonu sonuçlarını gözlemek bu kadar kolay değildir. Fakat kullanılan amaç fonksiyonları, optimizasyon algoritmalarının geliştirilmesi ve gerçek anlamda kullanışlı olup olmadığının anlaşılması hususunda aydınlatıcı veriler sunmaktadırlar.

3.1.1.1. Rastrigin Fonksiyonu

Fonksiyonun eğrisi Şekil 3.3'te görülmektedir. Fonksiyonun belirlenen aralıktaki global minimum değeri -19,9002 olarak hesaplanmıştır. Şekilden de anlaşılağı üzere fonksiyonda çok fazla sayıda yerel minimum noktası mevcuttur. Arama yapılırken parçacıkların çoğunun yerel minimumlarda kalması galip ihtimaldir.



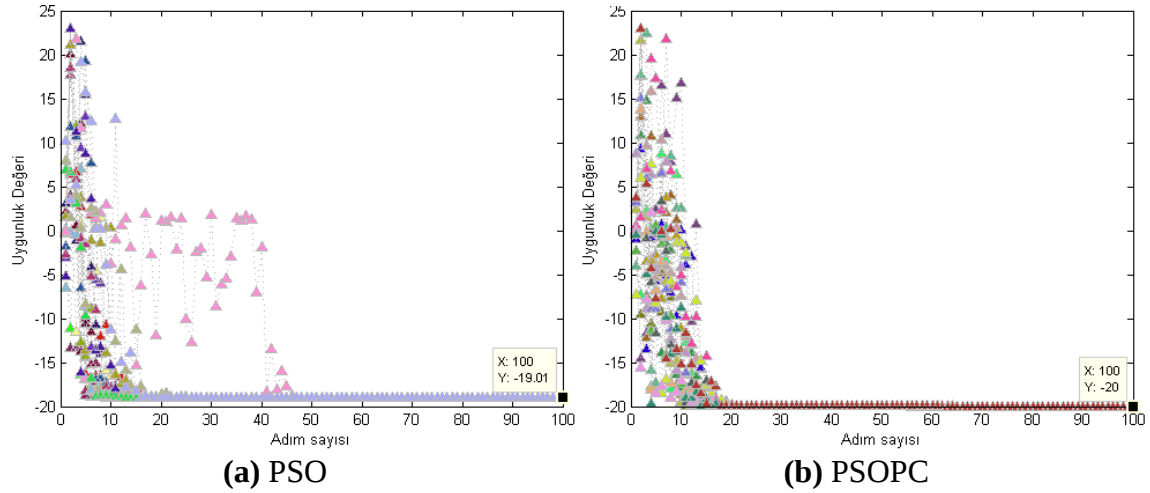
Şekil 3.3 Rastrigin fonksiyonu.

Rastrigin fonksiyonu, matematiksel optimizasyon uygulamalarında, sınama problemi olarak kullanılan, çok modlu, doğrusal olmayan bir fonksiyondur. Rastrigin tarafından iki boyutlu bir fonksiyon olarak ortaya atılan fonksiyon, 1991 yılında, Mühlenbein ve arkadaşları tarafından genelleştirilmiştir (3.8).

$$f(x) = 10n + \sum_{i=1}^n x_i^2 - 10 \cos(2\pi x_i) \quad (3.8)$$

$$-5,12 \leq x_i \leq 5,12$$

Şekil 3.4'te, PSO ve PSOPC algoritmalarının arama sonuçları, adım adım gösterilmiştir. Parçacıkların hareketleri çizgilerle ve her adımdaki parçacık konumu küçük üçgenlerle verilmiştir. Şekilde, aynı renk olan üçgenler aynı parçacığın farklı adımlardaki pozisyonunu belirtmektedir. Her iki algoritma da fonksiyonun minimum değerini başarılı bir şekilde bulmuşlardır. Yatay eksen adım numaralarını; dikey eksen ise fonksiyon üzerinde parçacıkların bulunduğu noktalara ait değerleri göstermektedir.



Şekil 3.4 Rastrigin fonksiyonu ile optimizasyon algoritmalarının sınanması.

PSO algoritması ile yapılan aramada, parçacıklar, ilk 15 adımdan sonra belli bir çizgiyi tutturmayı başarsa da, bazı parçacıklar, optimum noktaya yaklaşmakta biraz gecikmişlerdir. Pasif katılım etkisi (Şekil 3.4(b)), bu iki grafiğe bakıldığında daha açık bir şekilde görülmektedir. Parçacıklar birbirlerine daha yakın olarak hareket etmektedirler. Diğer bir deyişle, parçacıkların komşularıyla arasında daha sıkı bir ilişki vardır. PSO algoritmasında ise parçacıkların bazıları sürüden ayrılabilir.

Grafiklerden de anlaşılacağı üzere PSO algoritmasında PSOPC algoritmasına göre kararlılık ve isabet daha sonraki adımlarda gelmektedir.

Rastrigin fonksiyonu üzerinde PSO ve PSOPC algoritmaları ile yirmi er defa yapılan aramalarla ilgili bazı bilgiler  izelge 3.1’de verilmi tir. Pop lasyon sayısı, EK-1’de verilen  alı ma sonu larına dayanarak, 20 olarak ayarlanmı tır. Her adımda pop lasyon miktarı kadar uygunluk de erlendirmesi yapılır. Uygunluk de erlendirme sayısı, optimum de erin bulundu u adım numarası ile pop lasyonun  arpımı olarak ele alınmı tır.

Bulunan optimum noktaların ve optimum noktanın bulunma zamanlarına ait standart sapma verileri incelendi inde PSOPC algoritmasının PSO algoritmasına g re optimum noktaya ula mada daha hızlı ve kararlı oldu u a ık a g r lmektedir. PSO algoritmasının buldu u optimum de erlerin standart sapması 0,528 iken, PSOPC algoritmasında 0,2’dir. Ayrıca fonksiyonun minimum de erinin bulundu u adım sayılarında da standart sapmalar yakla ık olarak yarı yarıyadır. Do al olarak de erlendirme sayılarındaki standart sapmalar da aynı orandadır.

 izelge 3.1 Pop lasyon sayısının arama ba arısındaki etkisi.

Deney Nu	PSO			PSOPC		
	Opt. De�. Bulundu�u Adım	De�. Say. (Adım�Pop.)	Bulunan De�er	Opt. De�. Bulundu�u Adım	De�. Say. (Adım�Pop.)	Bulunan De�er
1	52	1040	-19,0050	26	520	-19,9001
2	7	140	-19,9025	15	300	-19,9010
3	52	1040	-18,0101	33	660	-19,9003
4	53	1060	-19,9067	30	600	-19,0050
5	7	140	-19,8984	21	420	-19,9016
6	25	500	-19,9029	50	1000	-19,8998
7	24	480	-19,8997	37	740	-19,9002
8	18	360	-19,9003	16	320	-19,8975
9	11	220	-19,8996	28	560	-19,9001
10	14	280	-19,8998	20	400	-19,9000
11	8	160	-19,9091	15	300	-19,9002
12	9	180	-19,8986	33	660	-19,8994
13	17	340	-19,9018	19	380	-19,9014
14	48	960	-19,0050	29	580	-19,9001
15	17	340	-19,9054	7	140	-19,9006
16	51	1020	-19,0050	17	340	-19,9001
17	36	720	-19,9024	25	500	-19,8993
18	13	260	-19,8999	21	420	-19,8994
19	63	1260	-19,8994	29	580	-19,9001
20	53	1060	-19,0050	24	480	-19,9016
Ort.	28,900	578,000	-19,628	24,750	495,000	-19,855
Std. Sp.	19,633	392,664	0,528	9,53	190,608	0,200

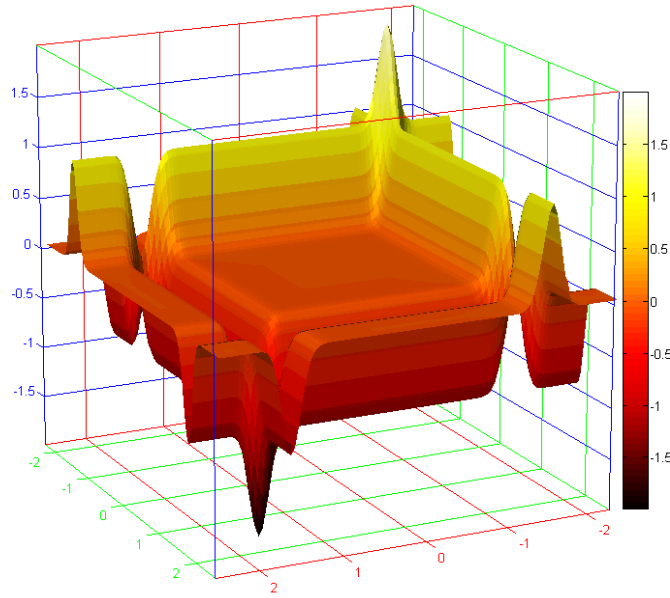
Her iki algoritma için yapılan yirmi er denemeden sonra PSOPC algoritması, PSO algoritmasına g re  st nl   n  g stermi tir. Ancak kesin olarak bir ifade kullanılamaz.  ok d   k bir ihtimal de olsa, hız vekt r  g ncellenirken kullanılan rastgele sayılar  st  ste  ok uygun oranda gelebilir. Bu durumda algoritma belki ilk birkaç adımda optimum noktayı bulacaktır. Bu durumun tam tersi de ge erlidir.

3.1.1.2. Michalewicz Fonksiyonu

Michalewicz fonksiyonu  ok modlu bir fonksiyondur. $n!$ adet lokal minimuma sahiptir. m de i keni, fonksiyondaki vadilerin dikli ini belirler. m de erinin artması aramayı zorla tırır (3.9). PSO ve PSOPC aramalarında m de eri 10 olarak alınmı tır. x de eri ise $[-2,4, 2,4]$ arasında sınırlandırılmı tır. Fonksiyon grafi i  ekil 3.5'te verilmi tir. Fonksiyonda, belirtilen aralık i in minimum nokta -1,998 olarak hesaplanmı tır.

$$f(x) = -\sum_{i=1}^n \sin(x_i) \left(\sin\left(\frac{ix_i^2}{\pi}\right) \right)^{2m} \quad (3.9)$$

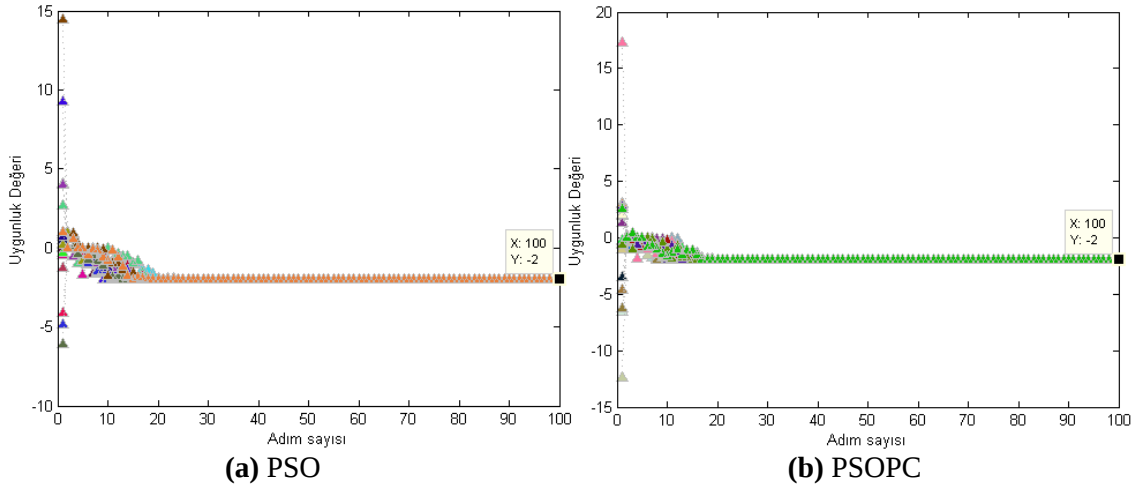
$i = 1:n \quad m = 10 \quad 0 \leq x_i \leq \pi$



 ekil 3.5 Michalewicz fonksiyonu.

Par acıkların minimum noktaya ula ma adımları, her iki optimizasyon algoritması i in,  ekil 3.6'da g sterilmi tir. Aramanın, PSOPC algoritması kullanılarak tekrar

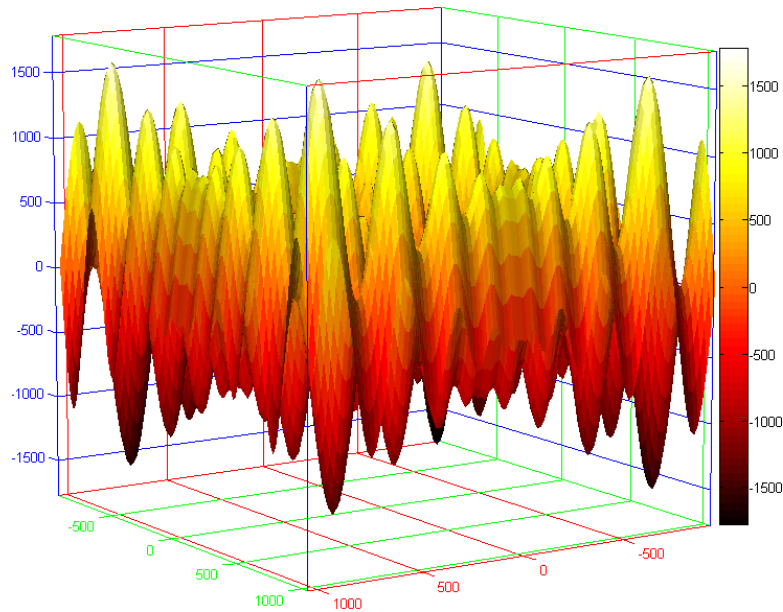
yapıldığında, PSO algoritmasına göre daha kısa sürede bulunduğu gözlemlenmiştir. Hareketlere göre PSOPC algoritmasının PSO algoritmasına göre %20-30 daha başarılı olduğu söylenebilir. Fakat hesaplamaların içine rastgele katsayıların girmesi durumu öngörülemez hale getirmektedir.



Şekil 3.6 Michalewicz fonksiyonu ile optimizasyon algoritmalarının sınanması.

3.1.1.3. Schwefel Fonksiyonu

Schwefel fonksiyonuna ait eğri, Şekil 3.7’de görülmektedir.



Şekil 3.7 Schwefel fonksiyonu.

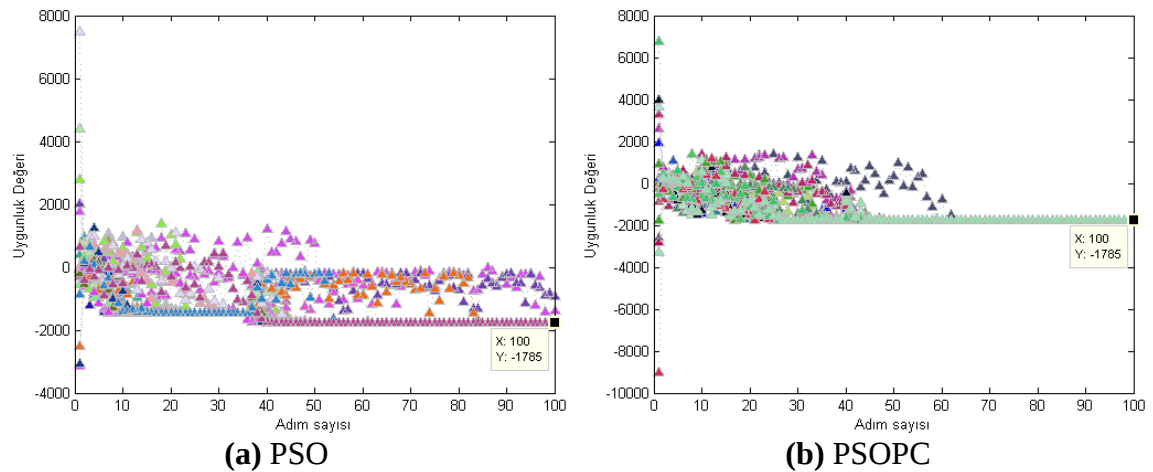
Schwefel fonksiyonu, arama için oldukça zordur. Fonksiyonun global minimum noktası, kendisine en yakın değerde olan en iyi yerel minimumdan uzaktadır. Bu yüzden algoritma, arama uzayında genellikle yanlış tarafa yönelmektedir.

Fonksiyonun formülü (3.10)'da verilmiştir. Genellikle, x değeri $[-500, 500]$ arasında sınırlandırılmıştır. Verilen aralıkta fonksiyonun global minimum noktası $-1790,8792$ olarak hesaplanmıştır.

$$f(x) = \sum_{i=1}^n -x_i \sin \sqrt{|x_i|} \quad (3.10)$$

$$-500 \leq x_i \leq 500$$

PSOPC algoritması (Şekil 3.8(b)), bu fonksiyon için de başarısını kanıtlamıştır. Her ne kadar iki algoritma da fonksiyonun global minimum noktasını bulamayıp, ona yakın olan bir yerel minimum noktasına (-1785) ulaşmışlarsa bile, parçacıkların arasındaki uyumlu hareket davranışı, kolay bir şekilde farkedilmektedir. PSOPC algoritmasının kararlılığının ve isabet oranının, PSO algoritmasına göre çok daha yüksek olduğu görülmektedir. Aynı zamanda parçacıkların birbirini etkileme oranının da daha fazla olduğu grafiklerden gözlemlenebilmektedir.



Şekil 3.8 Michalewicz fonksiyonu ile optimizasyon algoritmalarının sınanması.

Maalesef, ne PSO, ne de PSOPC algoritmalarının başarı oranı bu fonksiyon için çok yüksek değildir. Ancak PSOPC algoritmasının arama yaparken parçacıkların hareketleri

gözlemlendiğinde, pasif katılım etkisinin olumlu yönünün açığa çıktığı da rahatlıkla anlaşılabılır.

3.3. Daimi Mıknatıslı Senkron Motorlar (DMSM)

Günümüz dünyasında enerji kaynakları ihtiyacı karşılamakta zorlanmakta, dolayısıyla enerji tüketen tüm endüstriyel ürünlerde enerji tasarrufu ve verimliliği öncelik kazanmıştır.

DMSM'lerin tasarımı, uygun sonuçların neredeyse sonsuz sayıdaki kombinasyonlarla elde edilebildiği ve tasarım için önemli olduğu kabul edilen faktörlere bağlı olan bir çalışma alanıdır. Yapısındaki karmaşıklık, optimum tasarımı zor bir işe dönüştürmektedir. Çok fazla tasarım parametresi, karmaşık denklemler ve sistemin doğrusal olmayışı, DMSM modelini karmaşık hale getirmektedir (Duan and Habetler 2008). GA ve PSO gibi algoritmalar, bu ve buna kompleks sistem optimizasyon problemlerini çözebilmektedirler ve DMSM tasarım optimizasyonu için uygulanmışlardır (Wrobel and Mellor 2006).

DMSM, elektrik motorları arasında en yüksek güç yoğunluğuna sahip, yüksek tork ve esnek kontrol imkânı gerektiren uygulamalarda kullanışlıdır. Boyutlarının küçük olması gibi özellikleriyle diğer motorlara ciddi üstünlükler sağlamaktadır. (Krishnan 1987). Tahrik alanında, akım yerine içindeki daimi mıknatıs ile tahrik edilen senkron bir motordur. Veriminin ve kalkış momentinin yüksek, bunun yanı sıra çalışırken çıkardığı gürültünün az olması gibi bir takım karakteristik özelliklerine istinaden yaygın bir kullanım alanına sahiptir.

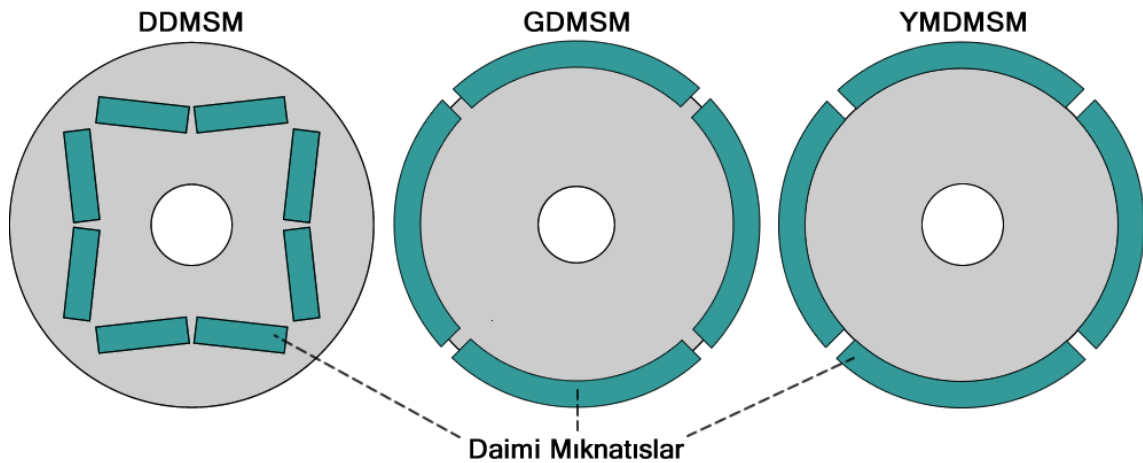
Motor mimarisinde mıknatısların kullanılması; rotor sargılarının elimine edilmesine ve radyal aralığın verimli kullanılmasına imkân sağlar. Bakır kayıpları da bu sayede azaltılmış olur. Mıknatıs malzemeleri olarak kullanılan neodmium-demir-boron (NdFeB) çok yüksek güç yoğunluğunu devam ettirirken, motor boyutlarında önemli bir azalmaya izin verir (Vas 1998). Rotor ağırlığı düşük olduğu için DMSM'nin uygulanan gerilim değişimine tepkisi çok hızlıdır. Ev kullanıcıları için de kararlı hareketler

sergilemesi istenen, elektrikli güç kontrol ihtiyacını karşılayan tüketim malzemelerinin de, kullanıcıların ihtiyaçlarını karşılayabilmesi için asgari tork dalgalanması gereklidir. (Thomas 1996).

DMSM'ler daimi mıknatıslardan kaynaklanan bazı kısıtlamalara tabi tutulurlar. Bu kısıtlamalardan bazıları, motor sargılarındaki aşırı akımlardan veya mıknatısın aşırı ısınmasından dolayı demanyetizasyon riskini içermesidir. Buna ek olarak, daimi mıknatısların üretebildikleri akı yoğunluğu biraz sınırlıdır. Ancak neodmium-demir-boron NdFeB ve SmCo gibi yeni manyetik malzemelerin geliştirilmesiyle, DMSM tasarımındaki bu tür kısıtlamalar gittikçe azalmaktadır (Fitzgerald *et al.* 2003).

DMSM'lerde sargılar, stator slotlarının içlerine dağıtılmıştır. Stator akımları tarafından üretilen akı hemen hemen sinüzoidal formdadır. Bu yüzden endüksiyon motorlarına uygulanan kontrol yöntemleri DMSM'lere de aynen uygulanabilir (Özçira vd., 2007). Yüksek verim nedeniyle; atalet oranındaki yüksek momentine ve kontrol problemlerine rağmen, DMSM'lere ilgi artışı olmuştur.

Genel olarak, mıknatıs yerleşimlerine göre DMSM'ler üçe ayrılırlar. Mıknatısın rotor içine yerleştirildiği motora dâhili DMSM (DDMSM), rotora gömülü olarak monte edilmiş haldeki motora gömülü DMSM (GDMSM) ve rotorun yüzeyine monte edilmiş haldeki motora yüzey montajlı DMSM (YMDMSM) adı verilir. Her üç montaj türü için örnek şekiller Şekil 3.9'da verilmiştir.



Şekil 3.9 Mıknatıs yerleşim şekillerine göre daimi mıknatıslı motorların rotor kısımları.

Resim 3.2’de, bu tezde üzerinde çalışılan yüzey montajlı DMSM’ye (YMDMSM) ait bir fotoğraf görülmektedir. Stator sargısı, eskisinde olduğu gibi, genellikle üç fazlıdır.



Resim 3.2 YMDMSM’ye ait rotor görüntüsü (İnt. kyn. 4).

Kalıcı mıknatısın alan şiddetindeki azalma, demanyetizasyon riskini ortaya çıkarır. Bu durumu düzeltmek için mıknatıs, rotor içerisine doğru büyütülerek yerleştirilir. Elde edilen bu yeni yapı içsel DDMSM olarak adlandırılır. DDMSM, yüksek hız uygulamalarında, YMDMSM’nin yapısına göre daha pürüzlü bir yapı oluşturur ve sensörsüz kontrolde faydalı bir endüktans salınımı meydana getirir.

GDMSM’lerde mıknatıslar motorun rotoruna gömülüdür. Bir kısmı veya tamamı rotora girmiş şekilde olabilir. İhtiyaca göre gömülme miktarı değiştirilebilir.

YMDMSM, DMSM’ler içinde en yaygın olarak kullanılan türdür. Özellikle yüksek hızlarda, mıknatıs ile rotor arasındaki bağlantıda yüksek bir gerilim vardır. Böyle durumlarda bağlantı, karbon fiber gibi yüksek gerilmeleri izale edebilen malzemeden yapılmış özel bir bant ile sağlanır. Yüzey montajlı tasarım diğer yerleşimlere göre daha düşük maliyete sahiptir. Bu tip motorların rotor direnci ve hava aralığı etkisi fazladır. Bu çalışmada YMDMSM motor tasarımının optimizasyonu üzerinde durulmuştur.

Daimi mıknatısların rotor bloğuna kuzey – güney kutup çifti oluşturacak şekilde yerleştirilmeleri ile DMSM’nin rotorundaki kutup çiftleri elde edilir. Stator kutupları, sargılardan geçen akımlarla sağlanır. Rotor kutupları sabittir. Fakat stator kutupları

uygulanan gerilimin frekansı ile ilişkilidir. Frekans ile değişen kutuplara bağlı olarak bir dönel alan oluşur. Motorun dönmesi için gerekli olan moment, rotor kutuplarının oluşturduğu alan ile stator dönel alanı tarafından üretilir (Kazan 2009).

3.3.1. Daimi Mıknatıs Malzeme Özellikleri

Geçtiğimiz yüzyıl içerisinde yeni daimi mıknatıs malzemeleri geliştirilmiştir. Önceleri kobalt-tungsten (CoW) ve krom-demir (CrFe) alaşımlarına dayanan malzemeler, motor imalatındaki çalışmalarda kullanılmışlardır. Alüminyum-nikel-kobalt (AlNiCo) alaşımlara ek olarak 1970 yılında geliştirilen SmCo mıknatıslar (Resim 3.3(a)) ve daha sonra 1982 yılında üretilen NdFeB alaşımlı mıknatıslar (Resim 3.3(b)) geliştirilmişlerdir.



Resim 3.3 DMSM yapımında kullanılan mıknatıs görüntüleri.

NdFeB mıknatıslar, yaklaşık 1 T gibi yüksek bir akı yoğunluğuna sahip oldukları için endüstriyel motor tasarımlarında daha cazip hale geldiler. Oda sıcaklığında NdFeB mıknatısın artık akı yoğunluğu (residual flux density) 1,1 – 1,25 T aralığında olduğu için geniş bir hava aralığı boyunca 0,8 – 0,9 T değerinde bir akı yoğunluğu üretmede yeterlidir.

Tipik olarak, daimi mıknatıslarda sıcaklık sınırlaması 120 °C'nin altındadır. Son yıllarda özel alaşımların kullanımıyla bu sınır bazı mıknatıslar için yaklaşık 180 °C'ye yükseltilebilir hale gelmiştir.

3.3.2. DMSM'nin Avantajları ve Dezavantajları

DMSM'nin rotorunda bilezik ve fırçaların yer almaması, bakım maliyetlerini azaltırken motor verimliliğini de artırır. Bir diğer avantaj da rotor sargısında oluşan bakır kayıplarının olmamasıdır. Dolayısıyla rotor sargısına sahip senkron motora göre aynı güçteki bir motorun ürettiğinden daha fazla tork üretilebilir. Bakır ve demir kayıpları daha çok statorda yoğunlaşır. Bu da soğutmanın stator etrafında yapılması avantajını getirir.

DMSM, 2 ile 50 arasında veya daha fazla mıknatıs kutuplu olarak imal edilirler. Çok fazla kutup sayısı genellikle aynı akımda daha fazla moment üretir. Böylece, üretiminde $\text{Sm}_2\text{Co}_{17}$ ve NdFeB gibi yüksek mıknatıslı malzemelerin kullanıldığı küçük boyutlardaki motorlardan yüksek güçler elde edilmesini sağlar.

Uygulamada, daimi mıknatıslar pek çok kullanışlı avantajlar sunarlar. Makine içinde manyetik alan üretmek için harici bir uyarım akımına gerek duyulmaması, bu avantajların başında gelir. Daimi mıknatıslar için gereken boşluk, rotor sargısının kaplayacağı yerden daha azdır. Böylece DMSM'ler daha küçük, bazı durumlarda da daha ucuz olmaktadır.

Stator alanı ile senkronize edilmesi gerektiği için rotor pozisyonunun bilinmesi gerekir. Hassas pozisyon bilgisi bir pozisyon sensörü ile sağlanabilir. Ancak bu da maliyeti ve yapının karmaşıklığını arttıracaktır.

Mıknatısların üretiminde kullanılan malzemenin değerli olmasından dolayı, üretim maliyeti daha fazla ve üretim süreci daha karmaşıktır. (Kurronen 2003).

3.3.3. DMSM'nin Sensörsüz Kontrolü

Son yıllarda çok farklı motor topolojileri için yüksek ve düşük hızlarda sensörsüz kontrol tekniklerinin araştırılmasına rağmen bu araştırmalar, robotik ve ulaşım endüstrilerinde de elektrikli motorların kullanılmaya ve yaygınlaşmaya başlamasıyla, iyi

bir ivme kazanmıştır. DMSM'lerin kontrolü için rotor pozisyonunun kesin bilgisine ihtiyaç vardır. Genellikle, rotor pozisyonu sensörler, dekoder ve enkoderler ile algılanmaktadır. Bu cihazların her biri kendi tasarım uygulama kısıtlamaları içinde etkili olmakla birlikte, bir konum algılayıcının sisteme dâhil edilmesi ek bir maliyet oluşturmakta, adanmış elektriksel bağlantılar gerektirmektedir. Buna ek olarak, bu bilgiyi sağlayan sensörlerin hem maliyeti arttırması, hem de kararsız davranışlar sergilemesi açısından dezavantajlı olması kontrolün sensörsüz olarak gerçekleştirilmesine doğru bir yönelime sebep olmuştur. Bilhassa motor karmaşık bir sistem içine gömülü ise veya basınçlı ya da sert ortamlarda çalıştırılıyorsa, pozisyon sensörünün motora entegre edilmesi birtakım problemlere sebep olur.

Hareket sensörünün olmayışı, servo motorlar ve otomotiv gibi özellikle maliyet konusunda hassas olan uygulamalarda çok ciddi avantajlar sağlamıştır. Ayrıca, bir mekanik parçanın eksik olması, bozulma olasılığını da ortadan kaldırır. Sensörsüz kontrol stratejileri motora uygulanan gerilim ve akımlardan gerekli bilgiyi çıkartmak için çeşitli sinyal işleme tekniklerini içermektedir (Budden *et al.*, 2006).

DMSM için geleneksel sensörsüz kontrol şemaları iki tipte sınıflandırılabilir. Bunlardan biri, zıt EMK geriliminin tahmini, diğeri ise endüktans salınımlarından elde edilen endüktans farkı ile tahminidir.

Zıt EMK geriliminin tahmini tipinde, gerilim modeli, durum gözlemleyicisi veya Kalman filtreleri kullanılmaktadır. Gerilim modeli yönteminde, ölçülen gerilim ve akımlar, DMSM'nin gerilim denklemlere göre rotor akısının bağlantı yerini tahmin etmek için kullanılır. Sıcaklık ve doygunluk sebebiyle parametrelerdeki değişimler rotor pozisyonunun tahmini performansını ciddi düzeyde etkiler. Durum gözlemleyicisi ve Kalman filtresi yöntemlerinde, parametre değişimleri gerilim modeli yönteminde olduğu kadar kritik değildir. Fakat bahsi geçen bu üç yöntemde ortak problem, rotor pozisyonunun tahmin edilme performansının, rotor hızıyla orantılı olan zıt EMK geriliminin genliğine bağlı olmasıdır. Zıt EMK geriliminin çok küçük olduğu düşük hız bölgelerinde, performans önemli ölçüde düşmektedir. Bu sorunu çözmek için dalgalanan yüksek frekanslı akım sinyali enjeksiyonu kullanılır. Bu yöntemdeki temel

fikir, durma noktasındaki bir motora yüksek frekanslı bir test akımının enjekte edilmesiyle zıt EMK geriliminde oluşacak küçük dalgalanmayı tespit etmeye dayanmaktadır. Ancak bu yöntem sadece motor, yüke esnek kavramayla bağlıysa geçerlidir.

Endüktans salınım farkından elde edilen pozisyon bilgisi yönteminde, motora harici bir gerilim ve akım enjekte edilir. Enjekte edilen sinyale karşılık gelen sinyaller rotor pozisyonu tespit etmek için kullanılır. Bazı algoritmalarda, motora enjekte edilen gerilim bir örnekleme periyodundadır. Bu yöntemlerle kısa sürede gerilim sinyallerini kullanan motorun endüktansı tespit edildiğinden, parametrelerdeki değişim ve ölçümlerdeki gürültü önemsiz kalmaktadır. Ayrıca, enjekte edilen sinyal sebebiyle, periyodik bir sinyal olarak Kalman filtresi gibi bir düzeltme şeması gereklidir. Diğer bir algoritma dönen yüksek frekanslı gerilim sinyallerini enjekte eder ve bir izleme algoritması kullanır. Dönen yüksek frekanslı sinyalleri kullandığından dolayı bu yöntemin dinamik karakteristiği, enjekte edilen sinyal frekansıyla ilişkili olduğu için, kısıtlanmaktadır. Dalgalanan yüksek frekanslı gerilim sinyalinin enjeksiyonu metodu, sıfır hızlı yüklü şartlar dâhil, düşük hızlı bölgelerde iyi performans göstermektedir. (Jang *et al.* 2003).

3.4. Motor Tasarımı

Tasarımı yapılacak motorun dış geometrisi sabit tutularak tork dalgalanması en düşük; tork ve sensörsüz kontrol kabiliyetinin en yüksek olmasını sağlayacak çok amaçlı bir optimizasyon hedeflenmiştir.

YMDMSM tasarımı için, FEA tabanlı bir yazılım olan Infolytica MagNet isimli programın sunduğu ortam kullanılmıştır. MagNet, kişisel bir bilgisayar üzerinde elektro manyetik modelleme yapabilmek için geliştirilmiş bir yazılımdır. Sunduğu sanal laboratuvar sayesinde, çeşitli manyetik malzemelerden imal edilmiş modeller tasarlanabilir, çizim ve grafik şeklinde sonuçlar gözlemlenebilir, akı çizgileri ve kuvvet gibi nicelikler hesaplanabilir. İki ve üç boyutlu modeller tasarlamaya ve bu modellerin simülasyonunu yaparak gerçeğe yakın çözümlemeler yapmaya imkân sağlamaktadır.

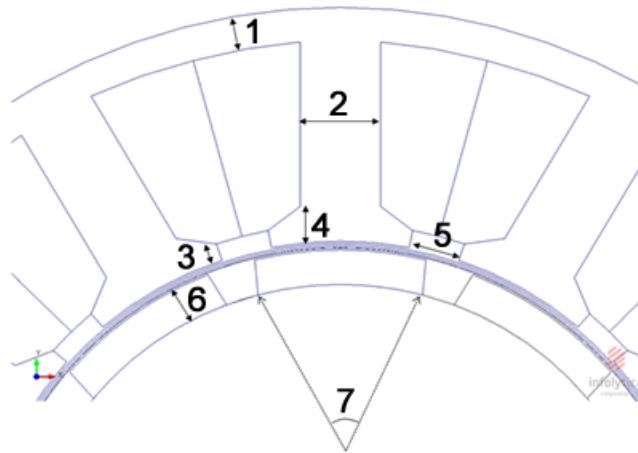
YMDMSM tasarımı yapılırken dikkat edilecek hususlardan bir tanesi de motorun kullanım amacıdır. Zira üretim aşamasında, motorda kullanılacak mıknatıs tipinin ve yerleşim şeklinin seçimi motor karakteristiğini etkilemektedir. Farklı seçimler ile motor boyutları aynı fakat güçleri farklı olabilir.

Tasarım için kullanılan malzemeler Çizelge 3.2’de gösterilmiştir. Motorun dış çapı bu çalışma için sabit tutulmuştur. Çizelge 3.3’te verilen geometrik parametreler optimize edilmiş ve verilen bir ısı transfer katsayısı için bakır kayıpları sabitlenmiştir. Demir kayıpları ise ihmal edilmiştir.

Çizelge 3.2 Tasarımda kullanılan malzemeler

Malzeme	Bileşen
M330-35A silikon çelik	Stator ve rotor demiri
NdFeB: 40/15	Mıknatıslar
Bakır: $5.77e7$ S/m	Bobinler

Motor tasarımı sekiz adet geometrik değişkenin optimize edileceği şekilde hazırlanmıştır. İlk değişken statorun iç çapının dış çapına oranlanmasıyla bulunan bölünme oranıdır. Kalan yedi değişkenin motor üzerinde ait olduğu bölgeleri gösteren çizim, Şekil 3.10’da verilmiştir. Çizelge 3.3’te değişkenlerin isimleri ve Şekil 3.10’da görülen numaraları ile diğer sabit tutulan parametreler verilmiştir.

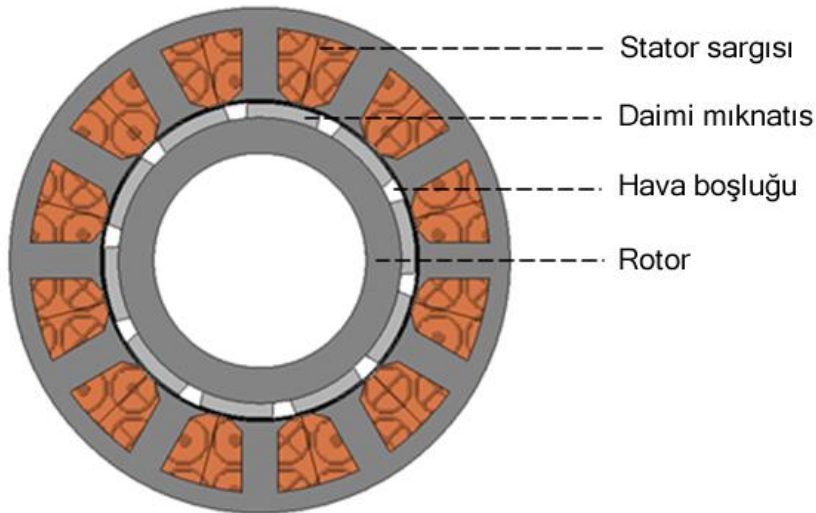


Şekil 3.10 DMSM motor tasarım değişkenleri.

Çizelge 3.3 Tasarım değişkenleri ve sabitleri.

Değişken Numarası	Tasarım parametreleri	Değer
-	Bölünme Oranı (Split ratio)	Değişken
1	Stator sırt demir kalınlığı (Stator back iron thickness)	Değişken
2	Diş genişliği (Tooth width)	Değişken
3	Slot açılma yüksekliği (Slot opening height)	Değişken
4	Diş köprü yüksekliği (Tooth bridge height)	Değişken
5	Slot açılma genişliği (Slot opening width)	Değişken
6	Mıknatıs kalınlığı (Magnet thickness)	Değişken
7	Mıknatıs yay açısı (Magnet span)	Değişken
-	Dış çap (Outer diameter)	67.5mm
-	Hava boşluğu (Air gap)	1mm
-	Bakır kaybı (Copper loss)	165W
-	Kutup sayısı (Pole number)	10
-	Slot sayısı (Slot number)	12
-	Motor uzunluğu (Motor length)	100mm

Daimi mıknatıslar rotor yüzeyine yerleştirilmişlerdir. Fakat kutuplar arası boşluk mesafesi değişkenlik gösterebilir. Tasarımı yapılan motorun 10 kutbu ve çift katmanlı sargıya sahip 12 yuvası bulunmaktadır. Her faz dört adet bobin ve yoğunlaştırılmış yapıda birer sargıya sahiptir. Motor geometrisi Şekil 3.11’de gösterilmiştir.



Şekil 3.11 Optimize edilecek motora ait model görüntüsü.

Motor tasarım parametrelerine ait alt ve üst sınır değerleri Çizelge 3.4’te verilmiştir. Çizelgedeki çözünürlük sütunu, ilgili parametredeki en küçük değişim miktarını belirler.

Çizelge 3.4 Tasarım parametrelerinin alabilecekleri alt ve üst değerler.

Tasarım Parametresi	Alt değer	Üst değer	Çözünürlük
Bölünme oranı	0,5	0,7	0,01
Diş kalınlığı (mm)	6	10	0,01
Mıknatıs yay açısı (°)	130°	150°	0,01
Mıknatıs kalınlığı (mm)	3	4	0,01
Stator sırt demiri kalınlığı (mm)	2,5	7	0,01
Diş köprü yüksekliği (mm)	2	6	0,01
Yuva açılma yüksekliği (mm)	0,1	0,8	0,01
Yuva açılma genişliği (mm)	2	10	0,01

3.5. Motor Tasarım Optimizasyonu

YMDMSM tasarımını PSOPC algoritması ile optimize etmek için, MATLAB ortamında uygunluk değerlendirmesini yapmak üzere, optimizasyon algoritması ile MagNet programı arasında haberleşmeyi sağlayan, önceden yazılmış ikinci bir program kullanılmıştır.

Zaman adımlı bir simülasyon, ortalama torku, tork dalgalanmasını ve sensörsüz kontrol yeteneğini hesaplamak üzere, her bir aday geometri için çalıştırılmıştır. Simülasyonda bakır kayıpları sabitlenmiştir. Daha sonra elde edilen sonuçlar PSOPC algoritmasında, uygunluk değerlendirmesine tabi tutulmuştur.

Simülasyon sonunda ölçülen tork, tork dalgalanması ve sensörsüz kontrol parametrelerinin değerleri tekrar MATLAB ortamına gönderilir. Her adımda, bütün parçacıklar için işlemler tamamlandığında, en iyi (gBest) parçacık belirlenir ve diğer parçacıkların hız vektörleri bu parçacığa yönelecek şekilde, algoritmada belirtilen formüle göre yeniden hesaplanır. Azami tekrar sayısına ulaşıldığında optimizasyon süreci son bulur.

Optimize edilecek parametre sayısı sekiz olduğu için arama uzayı da sekiz boyutlu olmalıdır. Diğer bir ifade ile her bir parçacık sekiz elemanlı bir diziden oluşmalıdır. Buna ek olarak, PSO algoritmasının doğası gereği, her bir parçacığın kendisinin ulaştığı en iyi değeri ve popülasyon içinde ulaşılan en iyi değeri hafızasında tutması gerektiği için, dizinin eleman sayısı artmaktadır.

PSO algoritmasının en temel işlemleri, sırasıyla, başlangıç, amaç fonksiyonunun belirlenmesi ve bilgi güncellemeleridir. PSO algoritmasında her bir parçacık için konum ve hız vektörleri mevcuttur. Konum bilgisi, bir parçacık için motor tasarım parametrelerinin o anda aldığı değerler olarak düşünülebilir. Bu tez çalışmasında motor tasarımı için önceki başlık altında bahsedilen sekiz parametre, sekiz boyutlu bir uzaydaki koordinat bilgisidir.

PSO algoritmasında sürü parçacıklardan oluşur ve her bir parçacık, ilk başta sınırları önceden belirlenen çözüm uzayının içinde rastgele bir noktaya atanır.

Her bir tasarım parametresinin optimum noktasını bulmak için sayısız kombinasyon vardır. Bu yüzden tasarım değerleri için en uygun yöntem optimum noktaya en yakın olan değeri almaktır.

Amaç fonksiyonu, optimizasyon işleminin amacına göre belirlenir. Optimizasyonun uygunluk değerinin yönüne göre minimize veya maksimize edilir. Uygunluk fonksiyonu olarak da adlandırılır.

Popülasyon içinde her bir parçacığın uygunluğu değerlendirilir. Bu değerlerle PSO algoritması, önceki adıma göre, yeni ve daha gelişmiş bir değer üretir. Bu değer parçacığın yeni konumu olarak da düşünülebilir. Uygunluk değerlendirmesinde her bir parçacığın konumu an içindeki en iyi parçacığın (gBest) konumu ile kıyas edilir.. Eğer bir parçacık gBest değerinden daha iyi bir noktaya geldiğinde artık kendisi gBest olarak güncellenir ve hafızaya alınır. Bu şekilde, diğer parçacıklar, kısa sürede en iyi parçacığın yanına yaklaştırılmaya çalışılır. Sonunda, her adımda parçacıklar birbirlerine

yaklaşırlar ve ortaya bir sonuç çıkar. Bu sonuç optimum tasarım için gerekli olan tasarım parametrelerini içerir.

Motor tasarımı için kullanılan PSOPC algoritmasını çalıştırmak için başlangıçta belirlenen nicelikler Çizelge 3.5’te verilmiştir.

Çizelge 3.5 Optimizasyon algoritmasında kullanılan sabitler.

Nicelik	Değer
Atalet (ω)	0,9-0,4 (Her adımda oğrusal olarak azalır).
Arama Uzayının Boyutu (Tasarım Parametreleri)	8
Ağırlık katsayıları ($\beta_1, \beta_2, \beta_3$)	$\beta_1 = 0,8; \beta_2 = 0,8; \beta_3 = 0,6$
Rastgele katsayı aralığı (r_0, r_1, r_2, r_3)	0-1

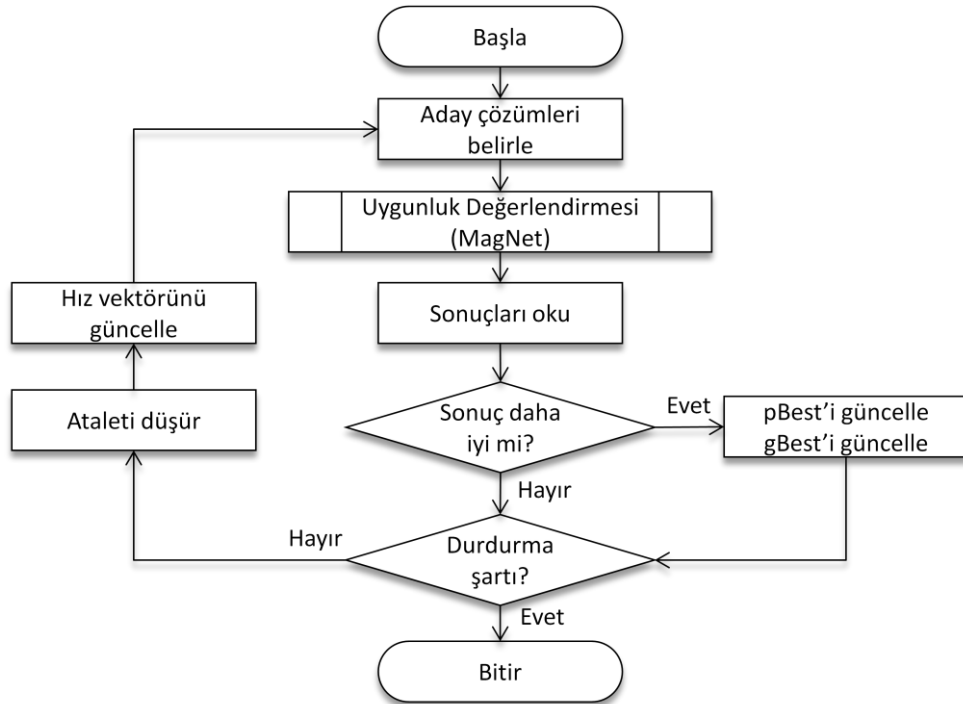
Parçacıklar arasındaki mesafe ile parçacıkların hızları arasında doğru bir orantı vardır. gBest olarak belirlenen parçacığın bulunduğu noktaya uzak olan parçacığın hızı, yakın olan parçacığın hızından daha fazla olmalıdır ki, parçacık, kısa sürede gBest’e yaklaşabilsin. gBest’in pozisyonu her adımda değişebilir. gBest bulunduğu noktadan daha iyi bir noktaya ulaşabilir veya başka bir parçacık artık yeni gBest’tir.

Bir diğer ana etmen de sistemdeki popülasyon sayısıdır. Eğer popülasyon küçük bir sayıda ise çözüme ulaşmak çok fazla zaman alabilir ve hatta bazen çözüme ulaşılamayabilir. Diğer taraftan, tasarım parametrelerinin alt ve üst sınırları arasındaki değerlerin çözünürlüğü arttırılmadıkça, çok büyük sürü popülasyonuna ihtiyaç kalmamaktadır.

Popülasyonda çok fazla parçacık olması, arama uzayında çok daha fazla ihtimali denemek ve çözümün daha hızlı yapılması şeklinde düşünülebilir. Fakat algoritmanın kaç defa tekrar edileceği daha önemlidir. Çünkü çoğunlukla gerçek problemlerde değerlendirmeler çok fazla zaman almaktadır. Bir adımda yapılan uygunluk değerlendirmesi parçacık sayısı kadardır. Sonuç olarak, eğer değerlendirme sayısı azaltılmak isteniyorsa bunu tekrar sayısını azaltarak değil, parçacık sayısını azaltarak yapmak gerekir. Popülasyon miktarının az olması, karşılaştırma işlemleri için elverişli değildir (Clerc 2006).

Rastgele katsayılar olan r_0 , r_1 , r_2 ve r_3 , önceden kestirilemeyen, algoritmanın tamamen sürpriz yapmasını sağlayan katsayılardır ve hız vektörünün güncellenmesinde çok farklı sonuçların çıkmasına sebep olurlar. Lokal minimum değerlere takılmayı önlemek adına oldukça kullanışlıdır. Diğer bir ifade ile sürüyü tek bir yönde değil, mümkün olan tüm yönlerde aramaya yöneltirler.

Şekil 3.12’de verilen akış şeması, PSOPC algoritmasına göre hazırlanmış bir optimizasyon uygulamasının işleyişini göstermektedir. Akış şemasına göre PSOPC kodları MATLAB ortamında çalıştırılır ve MATLAB, elde edilen aday çözümleri uygunluk değerlendirmesi için etkileşim içinde olduğu MagNet’e gönderir. MagNet, gelen tasarım parametrelerine göre motor geometrisini oluşturur ve motorun simülasyonunu yapar. MATLAB, MagNet’in elde ettiği sonuçları tekrar ana bellekten okur ve uygunluk değeri en iyi olan parçacığı tespit eder. Elde edilen uygunluk değeri, o ana kadar elde edilen en iyi çözümden daha iyi ise uygunluk değerini, parçacıkların hız vektörlerini, pBest ve gBest değerlerini günceller. Değilse, sadece her bir parçacığın konumunu ve hız vektörünü günceller.



Şekil 3.12 Motor tasarımı için geliştirilen PSOPC algoritmasının çalışması.

Yukarıda anlatılan işlemler, her bir aday çözümü içeren parçacık için tekrarlanır. 16 parçacıktan oluşan 20 tekrarlı bir optimizasyon için program $16 \times 20 = 320$ defa tasarım ve simülasyon yapar. Bu da oldukça uzun zaman almaktadır. Optimizasyon uygulamaları, Windows 7 Professional işletim sistemi yüklü, Intel Core2Duo (2,93 GHz, 3 MB önbellek, 2 çekirdekli) ve 2 GB ana belleğe sahip bir bilgisayar sisteminde bir günden fazla sürede tamamlanmaktadır.

Optimizasyon işlemleri üç ayrı amaç için üç ayrı uygunluk fonksiyonu kullanılarak uygulanmıştır. İlk uygunluk fonksiyonu f_1 , en az tork dalgalanmasıyla en fazla tork değerini elde etmek için (3.11), ikinci uygunluk fonksiyonu f_2 , sensörsüz kontrol kapasitesinin arttırılması için kullanılmıştır. (3.12). İlk iki fonksiyonun birleşimi olarak belirlenen son uygunluk fonksiyonu f_3 , her iki optimizasyon amacını gerçekleştirmek üzere kullanılmıştır (3.13).

$$f_1 = k_T \left((-1) \frac{T_{ort}}{30} \right) + k_R \left(\frac{T_R}{1} \right) \quad (3.11)$$

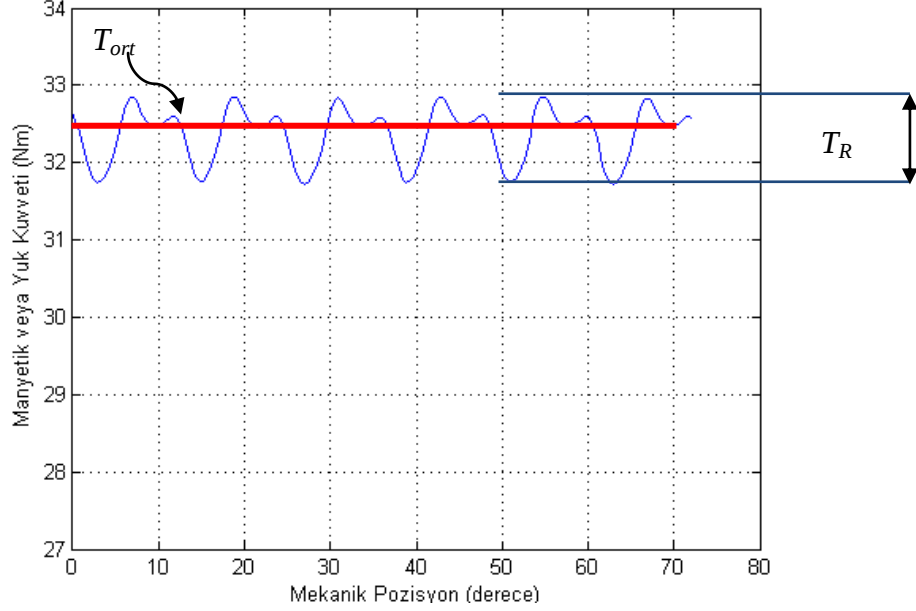
$$f_2 = k_S \left[-1 \quad Lq_{ort} - Ld_{ort} \right] + k_D \left(\frac{Ld_{pp}}{0,2} \right) + k_Q \left(\frac{Lq_{pp}}{0,2} \right) \quad (3.12)$$

$$f_3 = f_1 + f_2 \quad (3.13)$$

k_T ve k_R , sırasıyla, ortalama tork (T_{ort}) ve tork dalgalanma değerinin (T_R) önem katsayılarıdır (3.11). k_S , q ve d endüktans farkının ve k_D ile k_Q tepeden tepeye diferansiyel endüktans farklarının (L_{dpp} ve L_{qpp}) önem katsayılarıdır. Optimizasyon uygulamalarında $k_T = 0,95$, $k_R = 0,05$, $k_S = 0,6$ ve $k_Q = k_D = 0,2$ değerlerine ayarlanmıştır. Optimizasyon sürecinde amaçların önemine göre katsayılar değiştirilebilir.

Minimizasyon uygulaması olduğundan dolayı, uygunluk fonksiyonlarında yüksek çıkması istenen ortalama torkun ve sensörsüz kontrolde gerekli olan self endüktans farkının önüne (-1) katsayısı getirilmiştir. Uygunluk fonksiyonunun değerini 1'e eşitlemek için bazı kısımları belirli sayılara bölünmüştür. Örneğin hedeflenen ortalama tork değeri 30 Nm olarak hedeflendiği için T_{ort} değeri 30'a bölünmüştür.

Optimizasyon sonucunda tasarlanan motora ait ortalama tork ve tork dalgalanmasına ilişkin hesaplama Şekil 3.13'te gösterilmiştir.



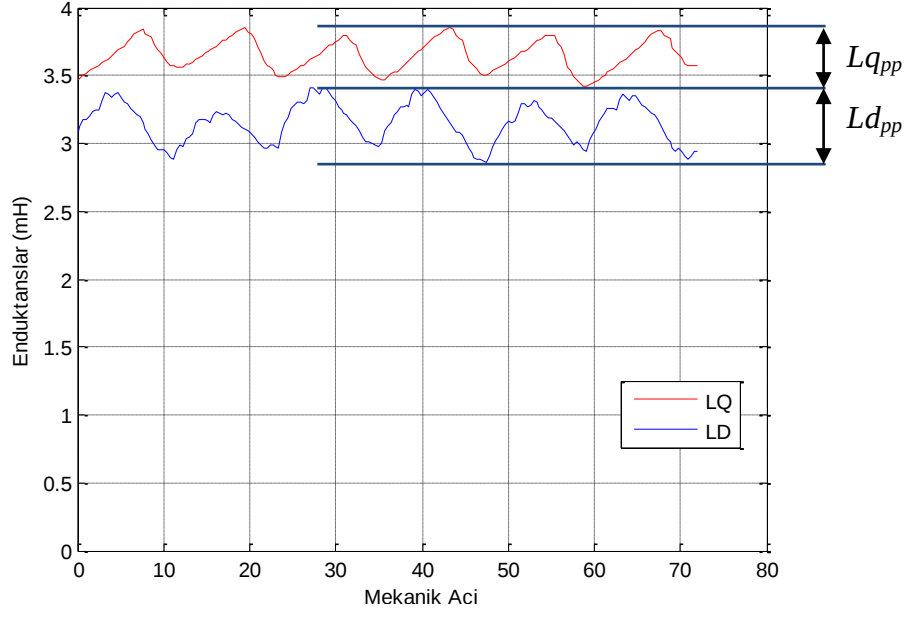
Şekil 3.13 Ortalama tork ve tork dalgalanmasının ölçülmesi.

T_{ort} , torkun ortalama değeri kırmızı renkli çizgi ile yaklaşık olarak belirtilmiştir. T_R , tork eğrisindeki dalgalanmadır.

L_{dpp} ve L_{qpp} , sırasıyla, d ve q eksenlerine karşı, diferansiyel self endüktans değişimlerinin dalgalanmalarıdır. Bu dalgalanmalar, tepeden tepeye fark değerleri alınarak hesaplanmıştır. L_{dpp} ile L_{qpp} değerleri arasındaki vektörel fark L_{dqfark} olarak isimlendirilmiştir (3.14).

$$L_{dqfark} = L_{dpp} - L_{qpp} \quad (3.14)$$

L_D ve L_Q diferansiyel endüktansları arasındaki dikey mesafe ne kadar fazla ise sensörsüz kontrol o kadar başarılı olur. Yapılan optimizasyonlar sonucu elde edilen tasarımların endüktans değişimleri ve L_{dpp} ve L_{qpp} değerlerinin hesaplanması Şekil 3.14'te verilmiştir. Bu değerler bir adet nominal ve iki adet değiştirilmiş akım şartlarında yapılan üç simülasyon sonucunun mukayesesi yapılarak hesaplanabilir.



Şekil 3.14 Tepeden tepeye endüktans değerlerinin ölçümleri.

Değerleri hesaplayabilmek için, evvelâ akı değişimleri, i_q ve i_d eksen akımlarına ilave edilen küçük birer test sinyali (ΔI_q ve ΔI_d) kullanmak şartıyla hesaplanır. Ardından akı değişimleri (3.14) ve (3.15) kullanılarak hesaplanır.

$$L'_d = \frac{1}{\Delta i_d} \phi D_{\Delta i_d} - \phi D \quad (3.15)$$

$$L'_q = \frac{1}{\Delta i_q} \phi Q_{\Delta i_q} - \phi Q \quad (3.16)$$

4. BULGULAR

4.1. Optimizasyon Algoritmasının Performansı

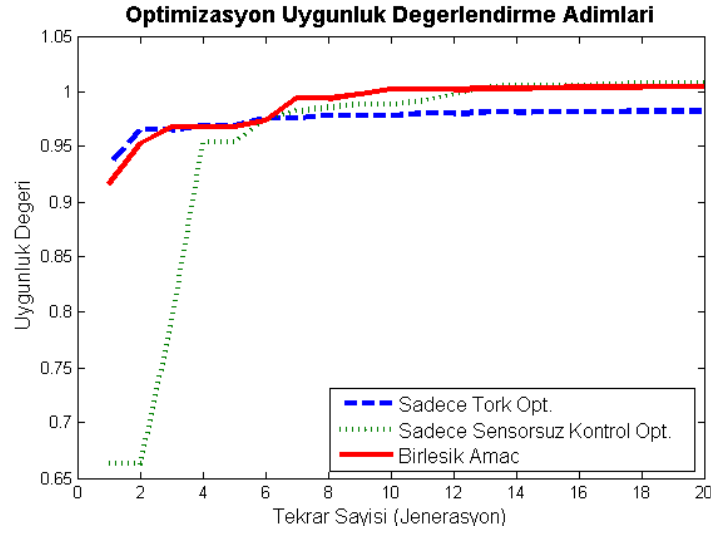
Test fonksiyonları üzerinde yapılan deneylerde PSOPC algoritması, PSO algoritmasına göre daha kısa sürede daha kaliteli sonuçlar vermiştir. Motor tasarımı yapılırken de, gerek kararlı davranışlar sergilemesi, gerekse kısa sürede optimizasyonu gerçekleştirmesinden dolayı, PSOPC algoritması tercih edilmiştir.

PSOPC algoritmasında optimum değerlere ulaşmak için uygunluk değerlendirmeleri yapılmıştır. Çizelge 3.3'te verilmiş olan alt ve üst sınır değerlerinde değişimin 0,01 puanlık çözünürlükte olmaları sağlanmıştır. Her bir parametre için yüz değerden birisi her parçacığa atanmıştır. Dolayısıyla motor optimizasyonda optimize edilen sekiz parametre için muhtemel çözüm sayısı 100^8 veya diğer bir ifade ile bir milyar adettir. Bu kadar çok ihtimalin içinden en iyi olanını isabetli olarak bulmak için popülasyon miktarını ve tekrarlama sayısını arttırmak iyi bir yaklaşım olarak düşünülebilir. Ancak bunun bazı kusurları da gözden kaçırılmamalıdır.

PSO algoritması, rastgele olarak arama uzayında en iyi çözümü bulmak ister. Popülasyon sayısının az olması sosyal etkiyi zayıflatır ve arama çok daha uzun ve hatta sonsuza kadar sürebilir. Bunun yanı sıra en iyi değer in lokal minimumlarda takılmasına yol açabilir. Buna karşılık popülasyonun çok kalabalık olmasının da sakıncaları vardır. Kalabalık bir popülasyon hesaplama işlemlerini arttırır ve optimizasyon işlemini uzatır (Chen and Zhao 2009).

Şekil 4.1'de, grafikler her üç amaç için kullanılan optimizasyon çalışmalarının elde ettiği uygunluk değerlendirmesindeki gelişimi göstermektedirler. Tekrar sayısının bir yere kadar etkili olduğu grafiklerden de anlaşılmaktadır. Sadece tork optimizasyonundaki uygunluk değişimi mavi renkli kesikli çizgi, sadece sensörsüz kontrol optimizasyonuna ait uygunluk değişimi yeşil renkli, noktalı çizgi ile ve son olarak, her iki amacın birleşik olarak optimize edildiği aşamanın uygunluk değeri değişimi kırmızı renkli çizgi ile gösterilmektedir. Her üç adımda da ilk on adımdan

sonra ciddi deęişimlerin olmadığını görmek mümkündür. Bunun yanı sıra, sadece sensörsüz kontrol yeteneęi için yapılan optimizasyonda PSO algoritmasının karakteristięinde olan hızlı yaklaşma özellięi de gözlenebilmektedir. Optimum değere optimizasyonun henüz ilk adımlarında yaklaştıęı görülmektedir (Uygunluk Deęeri = 1).



Şekil 4.1 Her üç optimizasyon için uygunluk değerin deęişimleri

4.2. Motor Tasarım Optimizasyonuna Ait Sonular

Sürünün popölasyon sayısı ve tekrar (jenerasyon) sayısı döngü ile kontrol edilmektedir. Testlerde popölasyon sayısı 20 ve tekrar sayısı 100 olarak ayarlanmıştı. Ancak çok uzun süren motor tasarımında hem popölasyon sayısı hem de tekrar sayısı düşürölmüşür. Çizelge 4.1’te gerçekleştirilen optimizasyon testlerinde popölasyon, tekrar ve süre bilgilerine ilişkin veriler mevcuttur.

Deney 1, sadece ortalama tork ve tork dalgalanması optimizasyonu için yapılmıştır. Deney 2, sadece sensörsüz kontrol optimizasyonu için ve Deney 3 hem tork hem de sensörsüz kontrol optimizasyonu için gerçekleştirilmiştir. Deney 3 dięer deneylerden daha uzun sürmüştür. Sadece tork optimizasyonunda sistem sadece tork ve tork dalgalanmasını optimize etmeyi amaçladığı için sensörsüz kontrol parametrelerini optimize etmemiş, sadece kaydetmiştir. Endüktans hesaplamaları için üç defa hesaplama yapmasına gerek olmadığından, altı saat gibi bir sürede, tork optimizasyonu tamamlanmıştır. Ancak ikinci ve üçüncü deneylerde sensörsüz kontrol parametrelerinin

hesaplanması için üç defa tekrarlı şekilde ölçüm yaptığı için optimizasyon süresi üç kat kadar artmıştır. Üçüncü deney, hem torku hem de sensörsüz kontrol yeteneğini optimize ettiği için işlem süresi daha da hızlanmıştır.

Çizelge 4.1 Optimizasyon testlerindeki popülasyon, tekrar ve süre bilgileri.

Tasarım Parametresi	Deney 1	Deney 2	Deney 3
Popülasyon	16	16	16
Jenerasyon (Tekrar sayısı)	20	20	20
Deney süresi	6 sa 3 dk	24 sa 34 dk	41 sa 22 dk

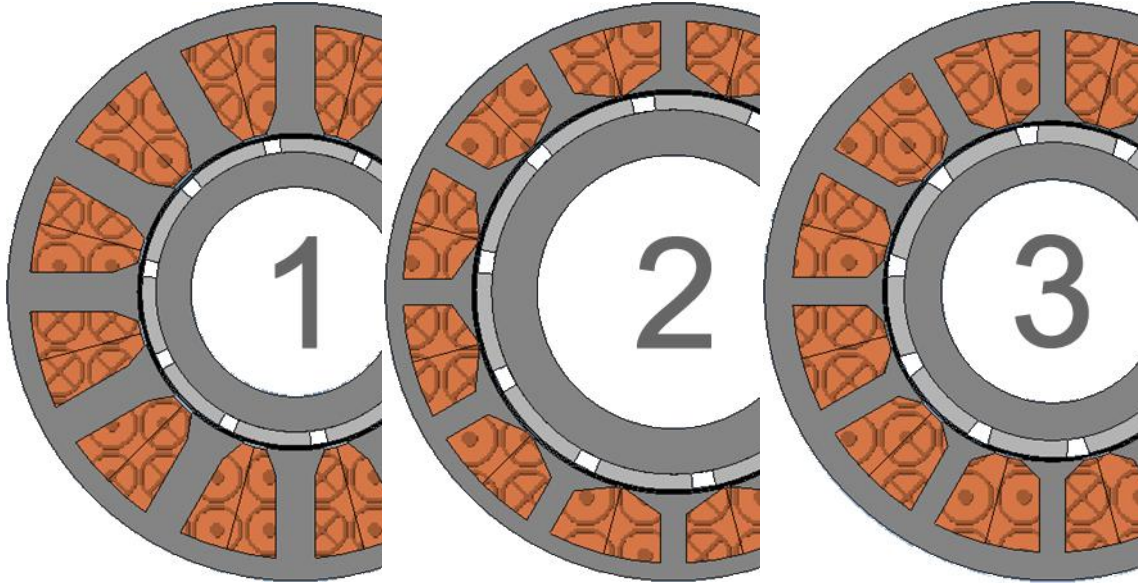
MagNet ile motora ait endüktans salınım hesaplamasını gerçekleştirilebilecek bir ortam sağlanmıştır. Sensörsüz kontrol ve tork değerlerine ilişkin tasarım değişkenleri PSOPC algoritması kullanılarak optimize edilmiştir. Azami tork üretim kapasitesinden çok az bir fedakârlık yaparak sensörsüz kontrol açısından verimli sonuçlar elde edilmiştir. Deneyler sonucunda başta belirtilen tasarım parametrelerine ait alt ve üst sınır değerleri ve deney sonunda elde edilen sonuçlar Çizelge 4.2’de verilmiştir. Tasarım parametreleri verilen sınırların dışına çıkmamıştır.

Çizelge 4.2 Tasarım parametrelerinin sınır değerleri ve deneyler sonunda aldıkları değerler.

Parametre	Alt sınır	Üst sınır	Deney 1	Deney 2	Deney 3
Bölünme oranı	0,5	0,7	0,55	0,70	0,59
Diş kalınlığı (mm)	6	10	9,43	6,16	6,34
Mıknatıs yay açısı (°)	130	150	148,15	145,93	149,95
Mıknatıs kalınlığı (mm)	3	4	3,31	3,65	4
Stator sırt demiri kalınlığı (mm)	2,5	7	5,35	4,17	6,66
Diş köprü yüksekliği (mm)	2	6	5,57	4,63	3,81
Yuva açılma yüksekliği (mm)	0,1	0,8	0,46	0,10	0,32
Yuva açılma genişliği (mm)	2	10	5,86	2,02	6,40

3 deney için elde edilen tasarım parametrelerine göre oluşturulan motor tasarımlarına ait geometrilerinin görsel olarak gösterimi Şekil 4.2’de gösterilmiştir. Amaçlar göz önüne alındığında, azami tork hedeflenerek optimize edilmiş olan birinci tasarımda, stator sargılarının bulunduğu slotlar ince ve yüksek bir formdadır. Diş genişliği oldukça yüksek çıkmıştır. Diş kalınlığının artması, daha fazla manyetik akı elde edilmesi ve

dolayısıyla motorun tork değerinin artması demektir. Daimi mıknatıslar, diğer iki tasarıma göre, hem daha ince hem de daha kısa boyutlardadır. Sadece sensörsüz kontrole uyumluluk için optimize edilen ikinci tasarımda ise bu slotlar daha geniş ve kısa bir şekil almıştır. Slot yüksekliklerinin orta noktasının daimi mıknatıslara yakın olması ile daha dengeli bir akı yoğunluğu sağlanmaktadır. Mıknatısların sargıya etkisi arttığından endüktans salınımlarının d ve q bileşenlerindeki vektörel fark daha fazla çıkmaktadır. İki amacın aynı anda optimize edilmesinden elde edilen üçüncü tasarım her iki tasarıma da benzerlikler göstermektedir. Slotlar, optimizasyon algoritması tarafından, birinci tasarımdaki kadar yüksek ve ikinci tasarımdaki kadar geniş olarak ayarlanmıştır.



Şekil 4.2 Yapılan üç deney için bulunan optimum motor tasarım geometrileri.

Yapılan deneylerin sonucunda elde edilen tork, tork dalgalanması ve endüktans değerleri Çizelge 4.3'te gösterilmektedir. Tork dalgalanması, en fazla 1 Nm dolaylarında olacak kadar düşük tutularak yaklaşık 30 Nm tork ve en az 1 mH endüktans farkı elde edilmiştir.

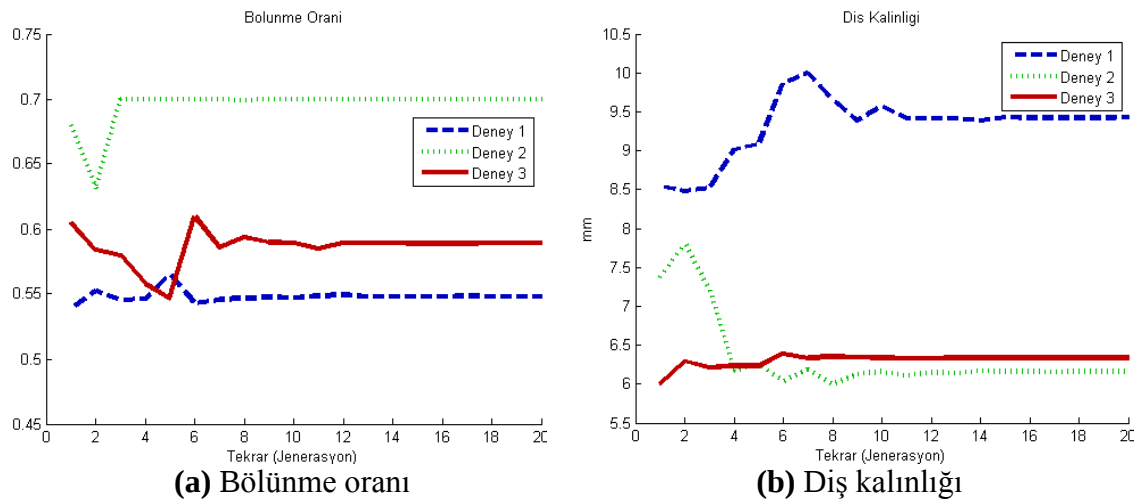
Birinci deneyde, tork değeri 32 Nm değerini aşmış ve tork dalgalanması 1 Nm altında kalmıştır. Ancak sensörsüz kontrol parametrelerinin kabul edilebilir düzeyde olmadıkları da görülmektedir. L_{dqfark} değerinin küçülmesi sensörsüz kontrolü zorlaştırmaktadır. Yine, L_{dpp} ve L_{qpp} değerlerinin sensörsüz kontrol uygulamalarında

istenilen seviyede olmadıkları görülmektedir. İkinci deney, birinci deneyle tam bir tezat oluşturmaktadır. Bu deneyde sensörsüz kontrol parametreleri makul düzeylerde seyretmiştir. Ancak tork değeri hedeflenen değerin altında kalmıştır. Buna ek olarak tork dalgalanması da yine hedeflenen değerin neredeyse iki katına çıkmıştır. İlk iki optimizasyon amaçlarının birleştirilmesiyle gerçekleştirilen üçüncü optimizasyon deneyinde, hem sensörsüz kontrol parametreleri oldukça iyi çıkmış, hem de tork ve tork dalgalanması değerleri de kabul edilebilir düzeydedirler.

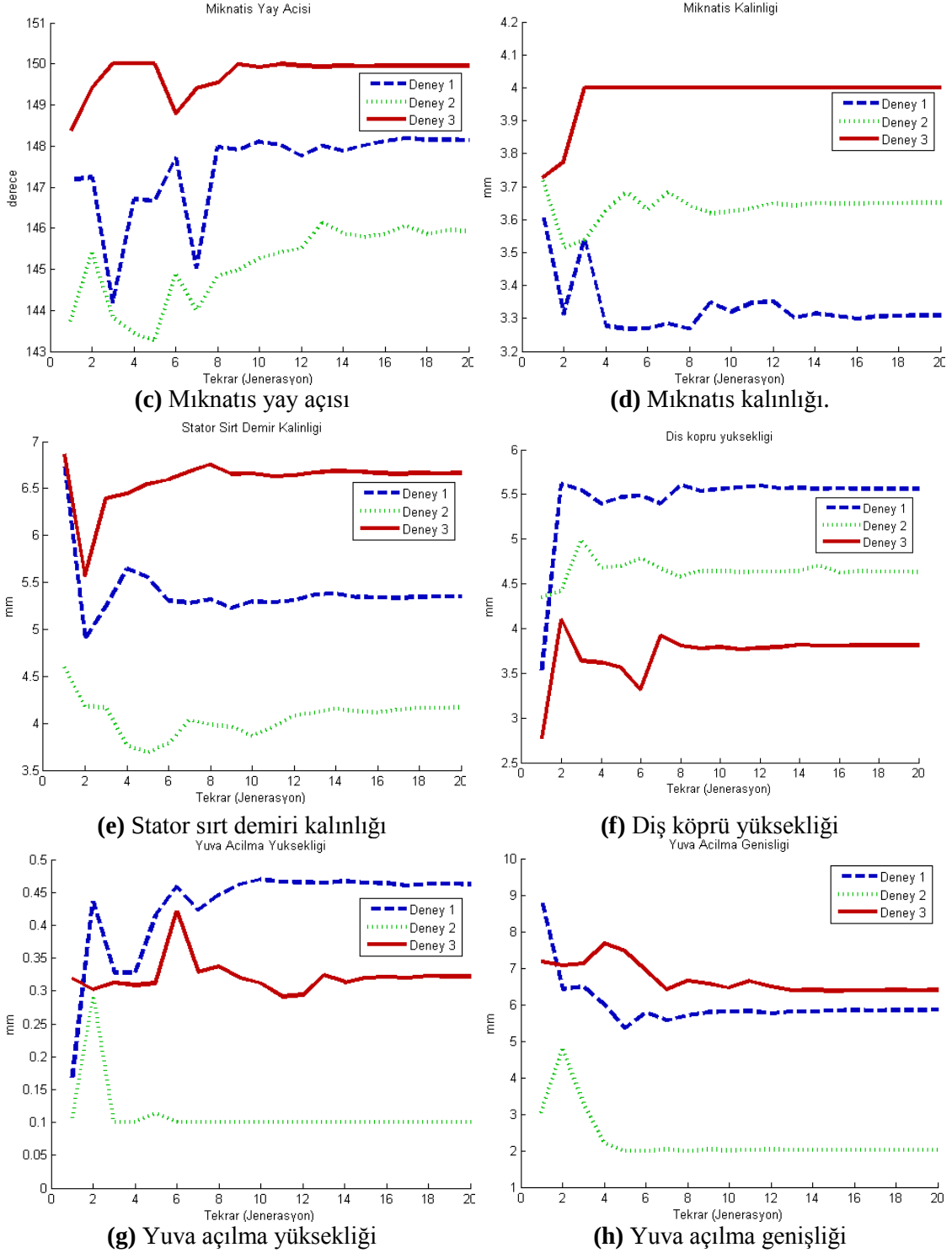
Çizelge 4.3 Optimizasyon uygulamalarında elde edilen sonuçlar.

Nicelik	Deney 1	Deney 2	Deney 3
Tork (Nm)	32,579	28,227	29,306
Tork dalgalanması (Nm)	0,991	1,969	0,889
Ld _{pp} (mH)	0,473	0,343	0,301
Lq _{pp} (mH)	0,371	0,230	0,070
Ldq _{fark} (mH)	0,494	1,097	1,020

Yapılan optimizasyon uygulamalarında elde edilen her bir tasarım değişkeninin, optimizasyon süresince değişimlerini gösteren grafikler Şekil 4.3'te gösterilmektedir. Mavi renkli kesikli çizgi sadece tork optimizasyonu, yeşil renkli noktalı çizgi sadece sensörsüz kontrol optimizasyonu ve kırmızı renkli çizgi her iki amaç için yapılan tasarım optimizasyonu için ilgili tasarım değişkeninin değişimini göstermektedir.



Şekil 4.3 Tasarım değişkenlerinin optimizasyon adımlarındaki değişimleri.



Şekil 4.3 (Devam) Tasarım değişkenlerinin optimizasyon adımlarındaki değişimleri.

Gerçek hiçbir değişimi kaybetmemek adına, modelleme yazılımından anlamlı değerler elde etmek için, simülasyon süresi ve çözünürlüğü için doğru değerleri seçmek önemlidir. Çözünürlüğü yükseltmek daha doğru sonuçlar verse de çok fazla hesaplama süresi gerektirmektedir.

Motorun nominal tur değeri 3000 d/dk olduğu için bir saniyede 50 d/d hız ile döner. Buradan, bir tam tur için geçen süre 0,02 s olarak hesaplanır. Motor yapısı beş çift kutuptan oluştuğu için sadece bir kutup çiftinin dönüşündeki hesaplamalardan elde edilen sonuçlar bütün kutup çiftleri için aynı olacaktır. Bu yüzden, bir tam tur, yani 360° yerine, 72° 'lik dönüşün hesaplanması yeterli olacaktır. Dolayısıyla bu çalışmada, simülasyon süresi bir tam tur için geçen sürenin beşte biri olan 4 ms olarak seçilmiştir. Örnekleme adım aralığı ise L_{dpp} , L_{qpp} ve T_{ort} eğrilerindeki dalgalanmaların yeteri kadar detaylı çıkması için 20 μ s olarak belirlenmiştir.

5. TARTIŞMA VE SONUÇ

Belirlenen ölçütlere göre motor tasarım optimizasyonuna uygun bir PSOPC algoritması MATLAB ortamında hazırlanmıştır. Hazırlanan algoritmanın gerçekten güvenilir sonuçlar verdiğini anlamak için Rastrigin, Michalewicz ve Schwefel fonksiyonları üzerinde denenmiştir. PSO algoritması ile PSOPC algoritması aynı fonksiyonlar üzerinde denenmişlerdir. Rastrigin fonksiyonu üzerinde yirmişer defa çalıştırılmak suretiyle elde edilen sonuçların ortalamalarına göre PSOPC algoritmasındaki pasif katılım etkisinin algoritmayı daha başarılı kıldığı görülmüştür. Her iki algoritmada fonksiyonun minimum noktasına ulaşma zamanında bariz farklar gözlenmemiştir. Ancak, optimum noktaya doğru yaklaşma açısından bakıldığında, PSOPC algoritmasının daha başarılı olduğu gözlemlenmiştir.

Test edilerek kullanılmasına karar verilen optimizasyon algoritması, MagNet yazılımı ile haberleşmeyi sağlayan, daha önceden yazılmış bir programa adapte edilmiştir. Bu program sayesinde, MATLAB ortamında, önerilen optimizasyon algoritması tarafından oluşturulan aday çözümler MagNet'e gönderilmiştir. MagNet programında elde edilen sonuçların, tekrar, MATLAB tarafından okunması bu program sayesinde sağlanmıştır.

Sadece tork verimliliği için optimize edilen bir motor geometrisinin sensörsüz kontrol performansı açısından zayıf kaldığı görülmüştür. Bunu göz önüne alarak yapılan yeni tasarımlar kayda değer gelişmeler göstermişlerdir. Çok-amaçlı fonksiyonların bir engeli, bütün parametrelerin azami düzeyde optimize edilememesidir.

Tork değerini arttırmak çok kolay gibi gözükebilir. Ancak sistemde amaçlar birbiriyle ilişkili oldukları için diğer amaçlardan da en az şekilde taviz vererek bir orta yolun bulunması esastır. Yine buna benzer olarak, sensörsüz kontrol kabiliyetinin artması, tork değerinin düşmesine sebep olmaktadır. Bu yüzden, tasarım değişkenlerinin değerlerinin daha düşük veya daha yüksek olmaları durumunda amaç fonksiyonuna, dolayısıyla, tork, tork dalgalanması ve sensörsüz kontrole uyumluluk etkisinin araştırılması için elde edilen sonuçlar tekrar analiz edilmiştir. Analizde, uygulanan her üç optimizasyonda, optimum değerleri elde edilen tasarım değişkenlerinin %5 ve %10 artması ve

azalmasının gözlenen parametrelere etkileri incelenmiştir. Bu, bir bakıma, farklı amaçlara göre yapılan optimizasyonların sağlaması anlamına da gelmektedir.

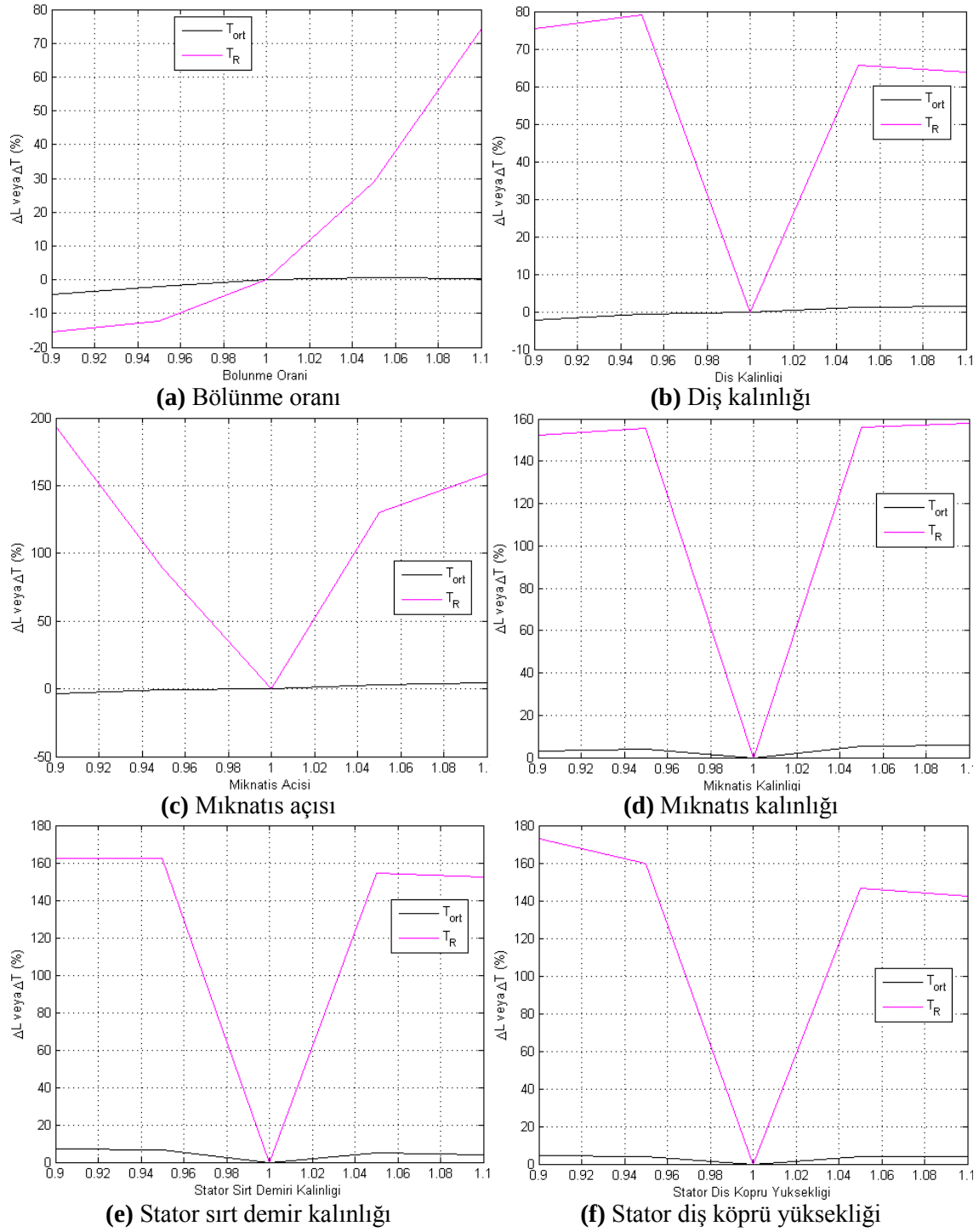
Sadece tork optimizasyonunun yapıldığı deneyde, bulunan optimum değerlerdeki değişimlerin tork dalgalanmasına etkilerini gösteren grafikler Şekil 5.1’de verilmiştir. Dikey eksen (ΔL veya ΔT), optimizasyon sonucu elde edilen değerine göre, tork ve endüktans salınımı farkını yüzde olarak göstermektedir.

Siyah renkli çizgi ile ortalama tork değeri, pembe renkli çizgi ile tork dalgalanması gösterilmiştir. Grafiklerin yatay ekseninin tam ortası, algoritma tarafından bulunan optimum değeri temsil eder. Sol tarafta %5 ve %10’luk azalmanın ve sağ tarafta da %5 ve %10’luk artışın, tork ve tork dalgalanmasına olan etkileri gözlenebilir. Grafik çizgileri, uygunluk değerlendirmesindeki önem katsayılarıyla ilişkili olduklarından dolayı farklı amaçlarda farklı şekilde görünebilirler.

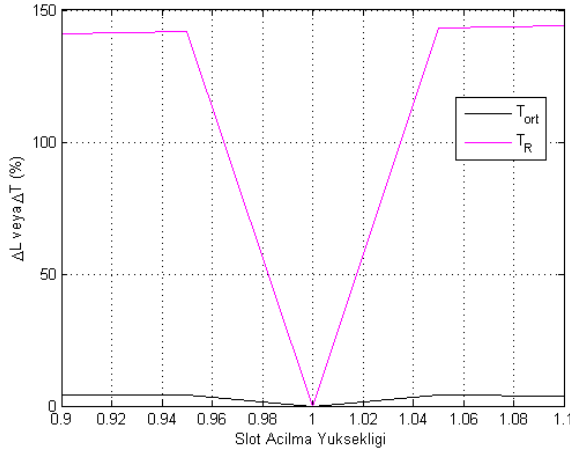
Bölünme oranındaki %10’luk azalma, tork dalgalanmasını %16 civarında azaltmıştır ama bunun yanında ortalama tork değerini de %4 civarında azaltmıştır. Optimizasyon sonucunda elde edilen optimum değerle bulunan tork dalgalanması değeri 0,991 Nm olarak hesaplanmıştı (Çizelge 4.3). Buna göre, bölünme oranındaki %10’luk azalma ile yeni tork dalgalanması değeri yaklaşık olarak 0,85 Nm olur. Optimizasyon sonucunda 32,579 Nm olarak hesaplanan ortalama tork değeri de yaklaşık olarak 31,276 Nm olur. Ortalama tork değerinde 1,3 Nm, tork dalgalanmasında ise 0,14 Nm azalma görülmüştür. Sonuçta tork daha önemli olduğu için dalgalanmadaki azalma göz ardı edilebilir.

Bölünme oranı haricinde diğer tüm değişkenlerin optimum değerinin hem azalması hem de artması tork dalgalanmasında ciddi artış getirmektedir. Ortalama tork değerine ise, ne artışın ne de azalmanın, kayda değer bir etkisi yoktur. Tüm değişkenler için, %10 artışa karşılık ortalama tork değeri sadece %0,1 ile %4 arasında bir artış gösterecektir ki bu da genel anlamda en fazla 34 Nm tork demektir. Fakat tork dalgalanması %65 ile %165 arasında bir oranda artacaktır. Bu artış, tork dalgalanmasını 1,6 Nm ile 2,6 Nm

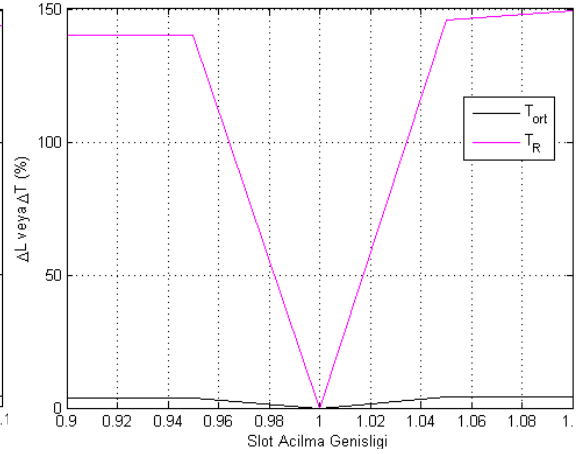
dolaylarına çıkarır. Çıkan değer ise amaç fonksiyonunda belirlenen maksimum değer olan 1 Nm'nin hayli üstündedir.



Şekil 5.1 Birinci deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork ve tork dalgalanmasına etkileri.



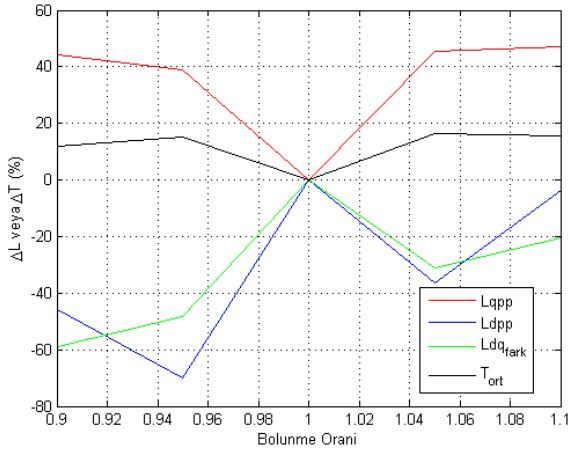
(g) Slot açılma yüksekliği



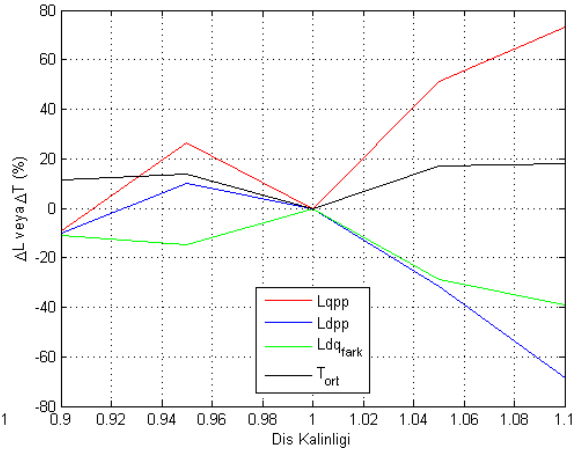
(h) Slot açılma genişliği

Şekil 5.1 (Devam) Birinci deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork ve tork dalgalanmasına etkileri.

Sensörsüz kontrol optimizasyonu için yapılan ikinci deneyde elde edilen sonuçların değişimleri gösteren grafikler Şekil 5.3'te verilmiştir. Kırmızı renkli çizgi L_{qpp} , siyah renkli çizgi ortalama torku, mavi renkli çizgi L_{dpp} ve yeşil renkli çizgi L_{dort} ile L_{qort} arasındaki fark olan L_{dqfark} değerini temsil etmektedir. Grafiklerin tamamından anlaşılacağı gibi, çok az tork artışı için önemli ölçüde sensörsüz kontrol kabiliyetinden taviz vermek gerekmektedir. Yine tasarım değişkenlerinin hiçbirisindeki artış ya da azalma, sensörsüz kontrol için gerekli olan L_{dqfark} değerine olumlu bir etki sağlamamaktadır.

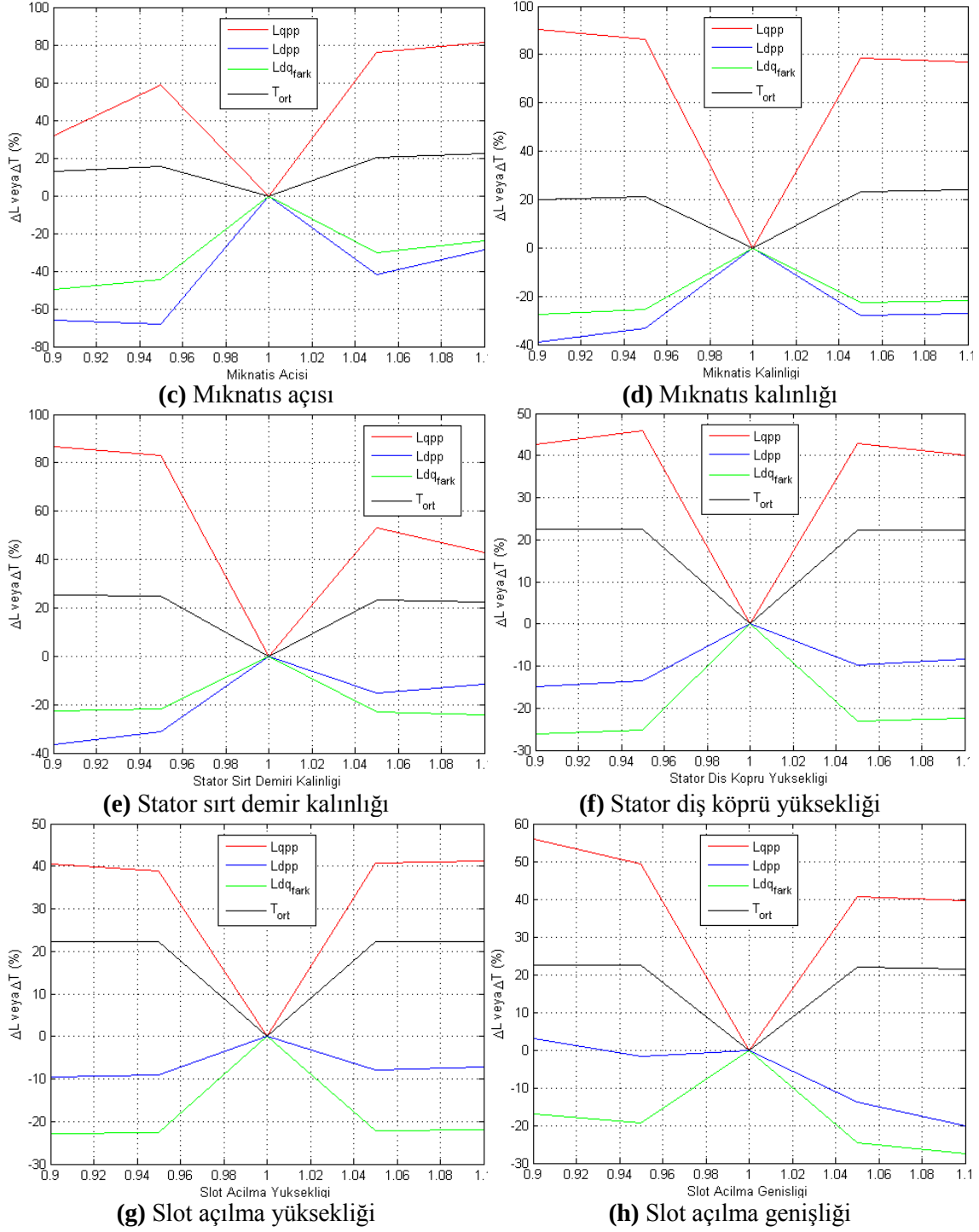


(a) Bölünme oranı



(b) Dis kalınlığı

Şekil 5.2 İkinci deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork ve endüktans değerlerine etkileri.

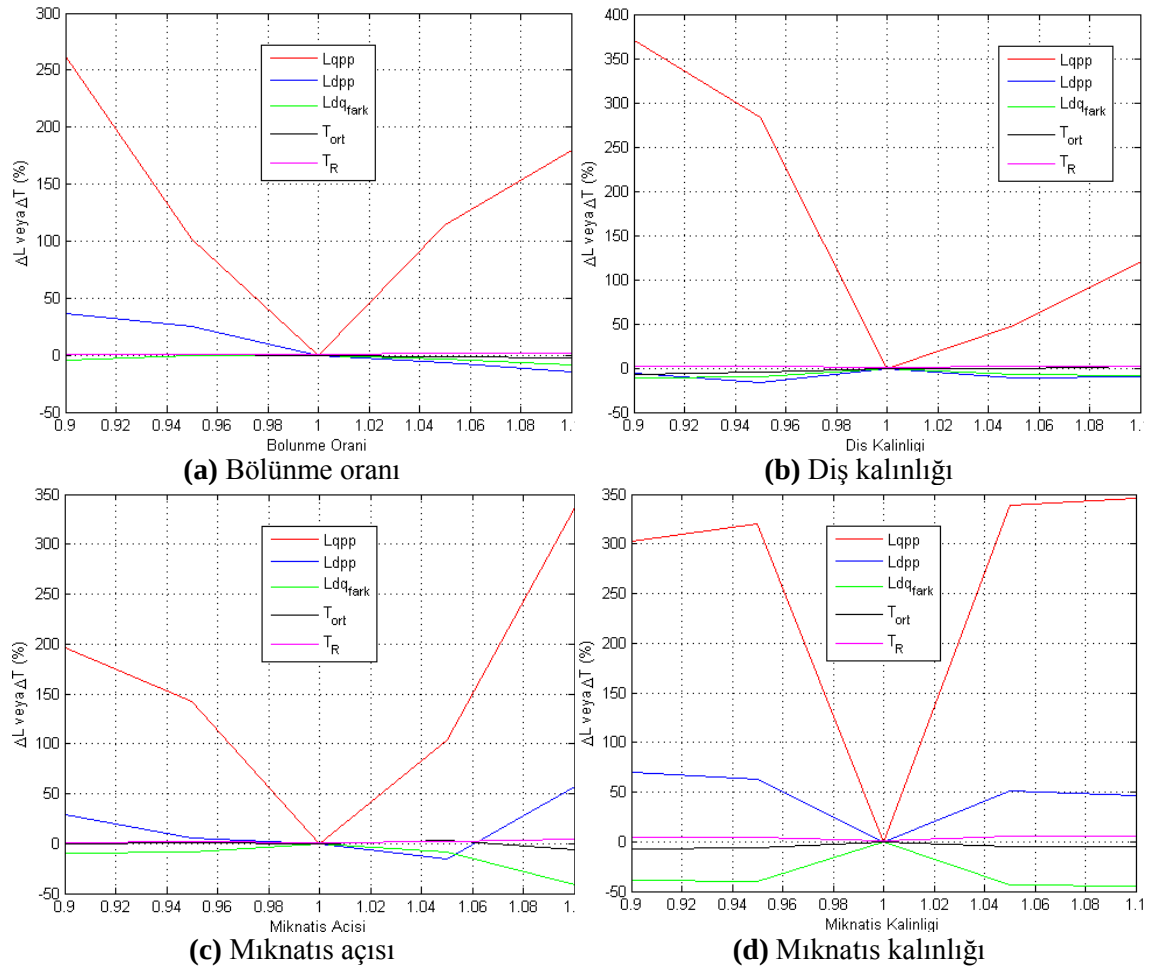


Şekil 5.2 (Devam) İkinci deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork ve endüktans değerlerine etkileri.

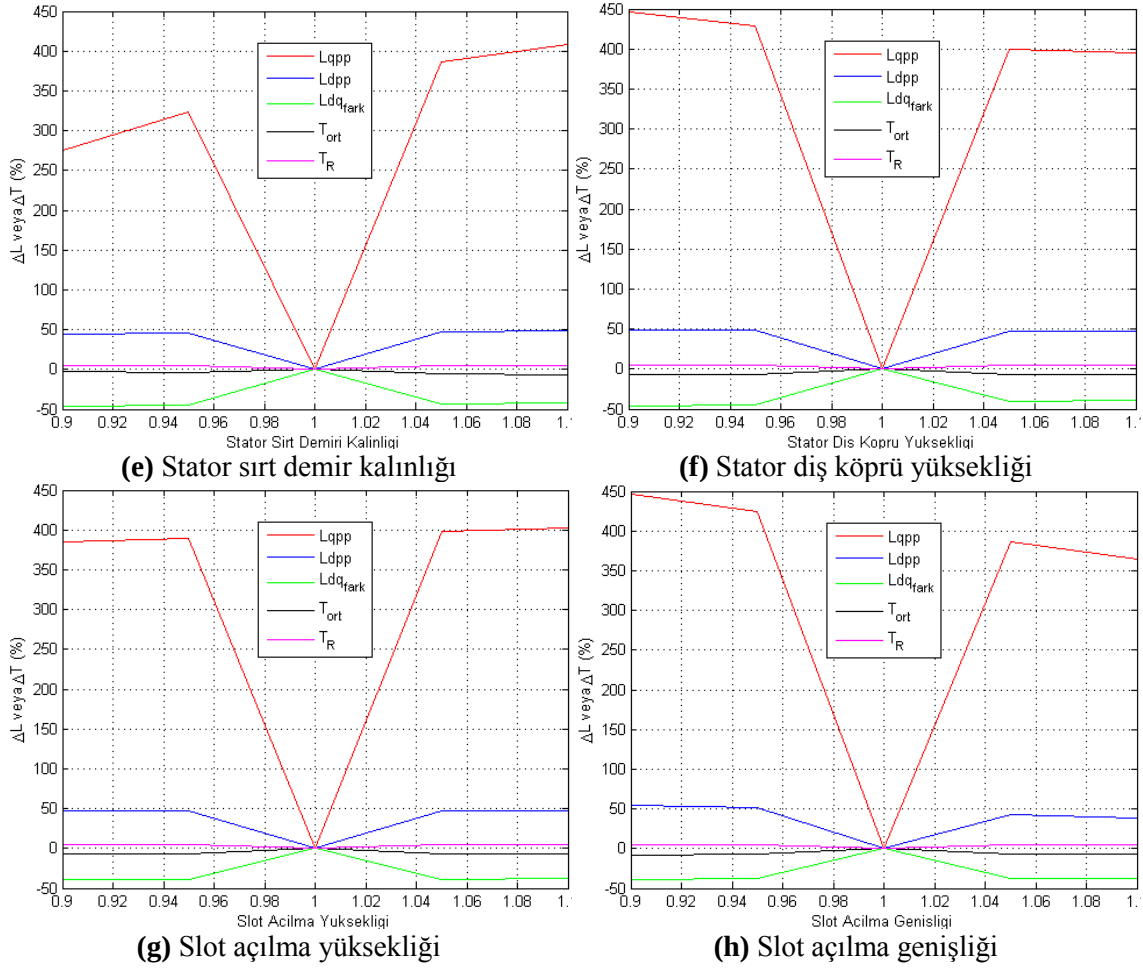
$L_{dq fark}$ değerini temsil eden yeşil renkli çizginin V harfi şeklini alması durumunda yeni tasarım değerleri tartışılabilirdi ancak çizgi ters V şeklindedir. L_{dpp} değerinin mümkün olduğunca yüksek ve L_{qpp} değerinin de mümkün olduğunca düşük çıkması, sensörsüz

kontrolü kolaylaştırmaktadır. Bulunan optimum değerlerdeki değişimlerin gerçekleşmesi durumunda ise L_{dpp} değeri düşmekte ve L_{qpp} değeri yükselmektedir.

Üçüncü deney olan, hem tork, hem de sensörsüz kontrol optimizasyonu deneyi için değişimleri gösteren grafikler Şekil 5.3'te verilmiştir. Sekiz tasarım parametresinin hepsindeki artış veya azalma, L_{qpp} değerini arttırmaktadır ki bu sensörsüz kontrole uygun bir motor tasarımından uzaklaşmak anlamına gelmektedir. Bunun yanı sıra ortalama tork ve tork dalgalanması değerlerinde olumlu değişimler de görülmemektedir. Deneyde, tork ve tork dalgalanması amaç fonksiyonuna dahil edilmemiştir. Dolayısıyla eldeki sistem, belirlenen sınır değerler dâhilinde, sensörsüz kontrole uyum açısından mantıklı bir tasarım olarak sayılabilir.



Şekil 5.3 Üçüncü deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork, tork dalgalanması ve endüktans değerlerine etkileri.



Şekil 5.3 (Devam) Üçüncü deneyde bulunan optimum tasarım değerlerindeki değişimlerin ortalama tork, tork dalgalanması ve endüktans değerlerine etkileri.

Yukarıdaki sonuçların tamamı hem tork, hem de sensörsüz kontrol kabiliyetinin nasıl artırılabilirliği hususunda fikir vermektedir. Bölünme oranı ve stator sırt demirinin kalınlığı, endüktans salınımı açısından son derece hassastırlar. Yukarıdaki sonuçlara göre, sensörsüz kontrol yeteneğinin, başka amaçlardan ödün vermeden geliştirilmesi mümkün değildir. Torkun artması istendiğinde aynı oranda sensörsüz kontrol kabiliyetinin azalması gerekmektedir. Böylece bu sonuçlar, PSOPC algoritmasının, kolayca geliştirilemeyen genel tasarım optimizasyonu bakımından oldukça iyi bir noktaya ulaştığını ispatlamıştır.

Tüm bunların yanında, yukarıdaki sonuçlar, hem tork ve tork dalgalanması, hem de sensörsüz kontrol kabiliyetinin optimize edilmesinde, optimizasyon sürecinin geçerliliğini de göstermektedir.

Motor tasarımında sabit bir dış geometri için hedeflenen değerlere PSOPC algoritması ile kolayca ulaşılmıştır. Uygulanan bu algoritmanın, hızlı bir şekilde, motor sürücü kontrol tekniklerinin daha başarılı uygulanabilmesi için uygun sonuçlar bulduğu gözlenmiştir.

YMDMSM için sensörsüz kontrol özellikleri, motorun tasarım aşamasında göz önünde bulundurulmalıdır. Bu çalışmada, PSOPC kullanarak bir YMDMSM tasarımı, sensörsüz kontrol ve tork yoğunluğu için optimize edilmiştir. Motor geometrisinden kaynaklanan endüktans salınımını izleyerek frekans enjeksiyonu yöntemiyle sensörsüz kontrol yapabilen sürücülere daha uyumlu bir motor tasarımı elde edilmiştir.

6. KAYNAKLAR

An, Y., Sun, C., Meng, Z., Che, D., Kong, Q., Cao, J. (2007). Optimization Design of High Efficiency Permanent Magnet Spinning Motor with Hybrid Algorithm of PSO and Chaos. International Conference on Electrical Machines and Systems, Seoul, Korea, October 8-11, 1778-1780.

Arellano, J., Gerada, C., Asher, G., Sumner, M. (2010). Inductance Characteristics of PMSMs and Their Impact on Saliency-based Sensorless Control. 14th International Power Electronics and Motion Control Conference, Ohrid, Republic of Macedonia, September, 6-8, S1-1-S1-9.

Berg, F. (2001). An Analysis of Particle Swarm Optimizers. Ph. D. Thesis, University of Pretoria, Pretoria.

Bianchi, N., Bolognani, S. (1998). Design Optimisation of Electric Motors by Genetic Algorithms. *Electric Power Applications*, **145**: 475-483.

Brandon, N. C., Scott, D. S. (2009). Population-Based Design of Surface-Mounted Permanent-Magnet Synchronous Machines. *IEEE Transactions on Energy Conversion*, **24(2)**: 338-346.

Brownlee, J. (2011). *Clever Algorithms: Nature Inspired Programming Recipes*. Lulu Enterprises, First Edition, ISBN: 978-1-4467-8506-5, Melbourne, Australia.

Budden, A. S., Wrobel, R., Holliday, D., Mellor, P. H., Dinu, A., Sangha, P., Holme, M. (2006). Impact of Rotor Design on Sensorless Position Estimation, IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference, Paris, France, November, 6-10, 787-792, DOI: 10.1109/IECON.2006.347358.

Chen, D. B., Zhao C. X. (2009). Particle Swarm Optimization with Adaptive Population Size and Its Application. *Applied Soft Computing*, **9**: 39-48.

Chuan-Kang, T. (2005). On the Mean Convergence Time of Multi-Parent Genetic Algorithms Without Selection. *Advances in Artificial Life*, 403-412, ISBN: 978-3-540-28848-0.

Clerc, M. (2006). Particle Swarm Optimization. ISTE Ltd., 2nd Edition, London, UK, ISBN-13: 978-1-905209-04-0.

Dorigo, M., Birattari, M. Stützle, T. (2006). Ant Colony Optimization. *IEEE Computational Intelligence Magazine*, **1 (4)**: 28-39, ISSN: 1556-603X, 28-32.

Dorigo, M. (1992). Ottimizzazione, Apprendimento Automatico, ed Algoritmi Basati su Metafora Naturale (Optimization, Learning and Natural Algorithms). Ph.D. Thesis, Politecnico di Milano, Italy.

Duan, Y., Harley, R. G., Habetler, T. G. (2008). Multi-objective Design Optimization of Surface Mount Permanent Magnet Machine with Particle Swarm Intelligence. IEEE Swarm Intelligence Symposium, St. Louis, MO, USA, September 21-23, 1-5.

Duan, Y., Harley, R. G., Habetler, T. G. (2009). Comparison of Particle Swarm Optimization and Genetic Algorithm in the Design of Permanent Magnet Motors. IEEE 6th International Power Electronics and Motion Control Conference, Wuhan, China, May 17-20, 822-825.

Duan, Y., Ionel, D. M. (2011). A Review of Recent Developments in Electrical Machine Design Optimization Methods with a Permanent Magnet Synchronous Motor Benchmark Study. IEEE Energy Conversion Congress and Exposition, Marlborough, USA, September 17-22, 3694-3701, ISBN: 978-1-4577-0541-0/11.

Eberhart, R. C., Shi, Y. (1998). Comparison between Genetic Algorithms and Particle Swarm Optimization. Evolutionary Programming VII, San Diego, CA, USA, March 25-27.

Eiben, A. E., Raué, P. E., Ruttkay, Z. (1994). Genetic Algorithms with Multi-parent Recombination. PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature, London, UK, ISBN: 3-540-58484-6, 78-87.

Emel, G. G., Taşkın, Ç. (2002). Genetik algoritmalar ve uygulama alanları. *Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi*, **13 (1)**: 129-152.

Fitzgerald, A. E., Kingsley, C., Umans, S. D. (2003). Electric Machinery. McGraw-Hill Higher Education, 6th Edition, New York, USA, ISBN-13: 978-0-07-366009-7.

Gudisz, V. G., Venayagamoorthy, G. K. (2003). Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks, IEEE Swarm Intelligence Symposium, Indianapolis, Indiana, USA, April 24-26, 110-117.

Hasanien, H. M. (2011). Particle Swarm Optimization of Transverse Flux Linear Motor for Weight Reduction and Improvement of Thrust Force. *IEEE Transactions on Industrial Electronics*, **58**: 4048-4056.

Haykin, S. O. (1999). Neural networks: A Comprehensive Foundation. Prentice Hall, Hoboken, McMaster University, Ontario, Canada, ISBN-13: 978-0-13-273350-2.

He, S., Wu, Q. H., Wen, J. Y., Saunders, J. R., Paton, R. C. (2004). A Particle Swarm Optimizer with Passive Congregation. *Biosystems*, **78(1-3)**: 135-147.

He, Q., Han, C. (2006). An Improved Particle Swarm Optimization Algorithm with Disturbance Term. *Computational Intelligence and Bioinformatics*. **4115**: 100-108.

Hertz, A., Taillard, E., Werra, D. D. (1995). A Tutorial on Tabu Search. Technical Report, EPFL, Departement de Mathematique, Ecublens.

Jang, J. H., Sul, S. K., Ha, J. I., Ide, K., Sawamura, M. (2003). Sensorless Drive of Surface-Mounted Permanent-Magnet Motor by High-Frequency Signal Injection Based on Magnetic Saliency. *IEEE Transactions on Industry Applications*, **39(4)**: 1031-1039.

Jenigri, S. (1996). Comparative Study of Efficiency of Genetic Algorithms and Particle Swarm Optimization Technical to Solve Permutation Problems. Computer Society of India, Internal Report, Mysore, India.

Jensen, P. A., Bard, J. F. (2003). Operations Research Models and Methods. John Wiley & Sons Inc., Hoboken, New Jersey, NY, USA, ISBN: 978-0-471-38004-7.

Karaboğa, D. (2011). Yapay Zekâ Optimizasyon Algoritmaları. Nobel Yayın Dağıtım, Genişletilmiş 2. Baskı, Ankara, Türkiye.

Kazan, F. A. (2009). Kalıcı Mıknatıslı Senkron Motorun Kontrolü. Yüksek Lisans Tezi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Konya.

Kennedy, J., Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, **4**: 1942-1948.

Khajehzadeh, M., Taha, M. R., El-Shafie, A., Eslami, M. (2010). Economic Design of Retaining Wall Using Particle Swarm Optimization with Passive Congregation. *Australian Journal of Basic and Applied Sciences*, **4(11)**: 550-557, ISSN: 1991-8178.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., (1983). Optimization by Simulated Annealing. *Science*, **220**: 671–680. DOI: 10.1126/science.220.4598.671. JSTOR 1690046 PMID 17813860.

Krishnan, R. (1987). Selection Criteria for Servo Motor Drives, *IEEE Transactions on Industrial Applications*, **23**: 270–275.

Kurronen, P. (2003). Torque Vibration Model of Axial-Flux Surface-Mounted Permanent Magnet Synchronous Machine. *Diss. Lappeenranta University of Technology*, ISBN: 951-764-773-5.

Leonardi, F., Venturini, M., Vismara, A. (1994). Design and Optimization of Very High Torque, Low Ripple, Low Cogging PM Motors for Direct Driving Optical Telescope. Industry Applications Society Annual Meeting, Denver, Colorado, USA, October 2-6, 272-278, ISSN: 0-7803-1993-1/94.

Łukaniszyn, M., JagieŁa, M., Wróbel, R. (2004). Optimization of Permanent Magnet Shape for Minimum Cogging Torque Using a Genetic Algorithm. *IEEE Transactions on Magnetics*, **40(2)**: 1228-1231.

Mendes, R., Kennedy, J., Neves, J. (2004). The Fully Informed Particle Swarm: Simpler, Maybe Better. *IEEE Transactions on Evolutionary Computation*, **8(3)**: 204-210.

Mühlenbein, H., Schomisch, D., Born, J. (1991). The Parallel Genetic Algorithm as Function Optimizer. *Parallel Computing*, **17**: 619-632.

Orman, K. (2008). Daimi Mıknatıslı Senkron Motorlar İçin Akım Gözetleyici ve Hız Kestirimi. Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Erzurum.

Özçira, S., Bekiroğlu, N., Ayçiçek, E. (2007). Simulation of Direct Torque Controlled Permanent Magnet Synchronous Motor Drive. Proceedings of ELECO '07 Conference – Bursa, Turkey, December 5 – 9, 2007, pp. 272-276.

Rzesnitzek, T., Müllerschön, H., Günther, F. C., Wozniak, M. (2003). Two-Stage Stochastic and Deterministic Optimization. Nichtlinear Optimierung und Stochastische Analysen mit LS-OPT, Stuttgart, March 27.

Settles, Mr., Rodebaugh, B. (2003). Comparison of Genetic Algorithm and Particle Swarm Optimizer When Evolving a Recurrent Neural Network. Genetic and Evolutionary Computation Conference, Chicago, USA, July 9-11, 148-149.

Şimşek, F., Mergen, F. (2009). Asenkron Motorun Çok Hedefli Tasarım Optimizasyonu. Elektrik-Elektronik-Bilgisayar ve Biyomedikal Mühendisliği 13. Ulusal Kongresi ve Fuarı Bildirileri, Ankara, Aralık 23-26.

Thomas, M. J., Wen, L. S. (1996). Pulsating Torque Minimization Techniques for Permanent Magnet AC Motor Drives - A Review. *IEEE Transactions on Industrial Electronics*, **43**: 321-330.

Vaez-Zadeh, S., Ghasemi A. R. (2005). Design Optimization of Permanent Magnet Synchronous Motors for High Torque Capability and Low Magnet Volume. *Electric Power Systems Research*, **74 (2)**: 307-313.

Vas, P. (1998). Sensorless Vector and Direct Torque Control. Oxford University Press, New York, USA.

Wang, Y. Peng, X. Wei, B. (2008). A New Particle Swarm Optimization Based Auto-Tuning of PID Controller. IEEE Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, July 12-15, 1818-1823, ISBN: 978-1-4244-2096-4/08.

Wrobel R., Mellor, P.H. (2006). Particle Swarm Optimization for the Design of Brushless Permanent Magnet Machines, IEEE Conference on Industrial Applications, Glasgow, UK, October 8-12, **4**:1891 – 1897.

6.1. Internet Kaynakları

1. http://www.nptel.iitm.ac.in/courses/Webcourse-contents/IISc-BANG/OPTIMIZATION%20METHODS/pdf/Module_1/M1L1slides.pdf, 07.04.2012
2. http://en.wikipedia.org/wiki/Linear_programming, 07.04.2012
3. http://en.wikipedia.org/wiki/Genetic_algorithm, 04.04.2012
4. <http://www.geophysik.uni-muenchen.de/~igel/downloads/inviiannealing.pdf>, 05.04.2012
5. http://www.ewh.ieee.org/soc/es/Nov1997/09/B_8POL2.JPG, 10.04.2012
6. http://img.directindustry.com/images_di/photo-g/neodymium-iron-boron-magnet-ndfeb-270482.jpg, 14.05.2012
7. http://img.directindustry.com/images_di/photo-g/samarium-cobalt-ring-magnet-smco-583939.jpg, 14.05.2012

ÖZGEÇMİŞ

Adı Soyadı : Özkan ÖZLÜ
Doğum Yeri ve Tarihi : Afyonkarahisar, 14.07.1982
Yabancı Dili : İngilizce
İletişim (Telefon/e-posta) : (374) 254 1000 – 1822 / ozkanozlu@hotmail.com

Eğitim Durumu (Kurum ve Yıl)

Lisans : Marmara Üniversitesi
Teknik Eğitim Fakültesi,
Bilgisayar ve Kontrol Öğretmenliği (İngilizce),
İstanbul, 2005

Lise : Gazi Anadolu Meslek Lisesi,
Afyonkarahisar, 2000

Çalıştığı Kurumlar ve Yıl

1. Abant İzzet Baysal Üniversitesi, Bilgi İşlem Daire Başkanlığı, Bolu, 2012-Halen
2. Mimar İzzet Baysal Teknik ve Endüstri Meslek Lisesi, Bolu, 2010-2012
3. Afyonkarahisar Teknik ve Endüstri Meslek Lisesi, Afyonkarahisar, 2005-2010

EKLER

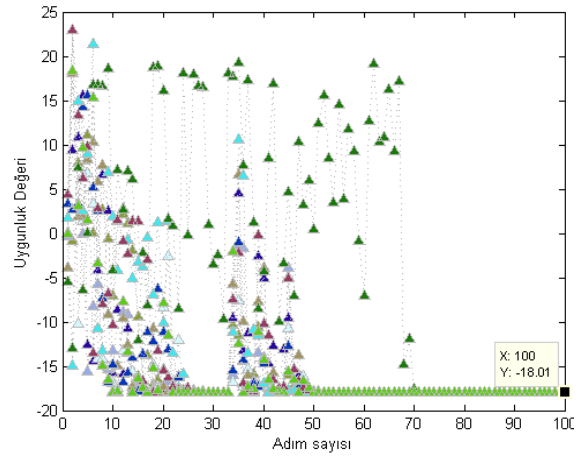
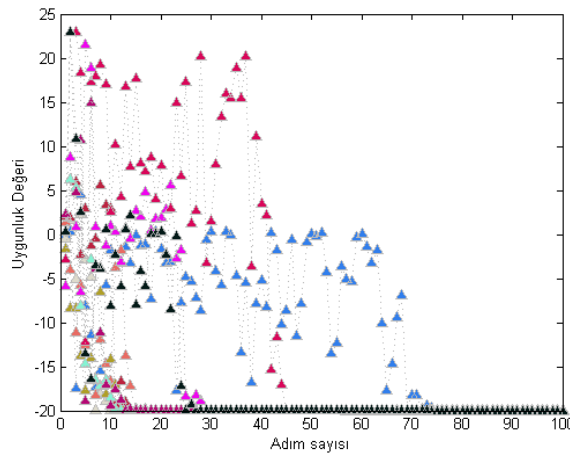
Ek. 1 PSO ve PSOPC Algoritmalarının Rastrigin Fonksiyonu Üzerinde Sınanması

Fonksiyonun uygunluk değeri dikey eksende gösterilmiştir. Adım sayıları yatay eksen üzerindedir. Aynı renkteki her bir üçgen tek bir parçacığın farklı adımlardaki konumunu göstermektedir. Fonksiyonun minimum noktası -19,9002'dir. Sol taraftaki grafikler PSO algoritmasındaki parçacıkların hareketlerini göstermektedir. Sağ taraftakiler ise PSOPC algoritmasına ait hareketleri temsil etmektedir.

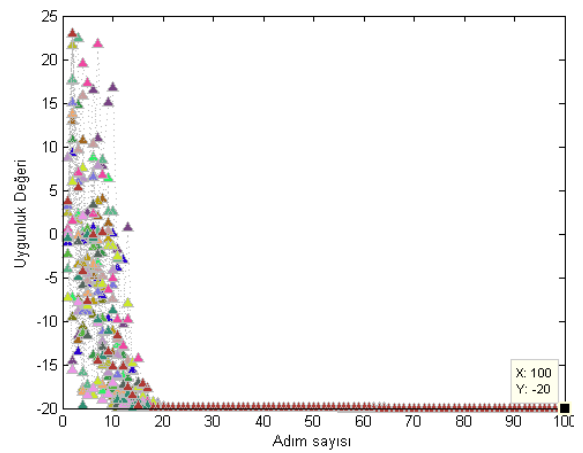
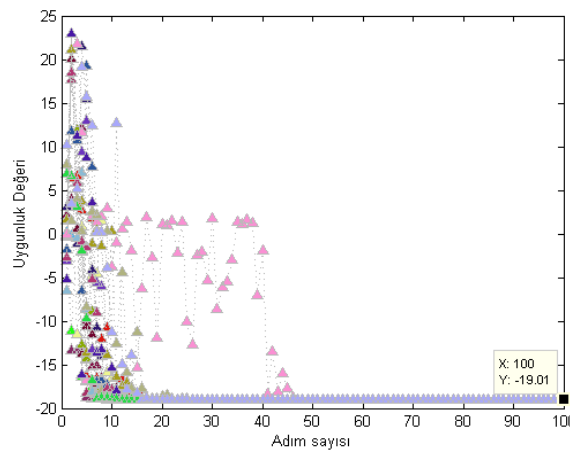
PSO algoritmasının arama sonuçları

PSOPC algoritmasının arama sonuçları

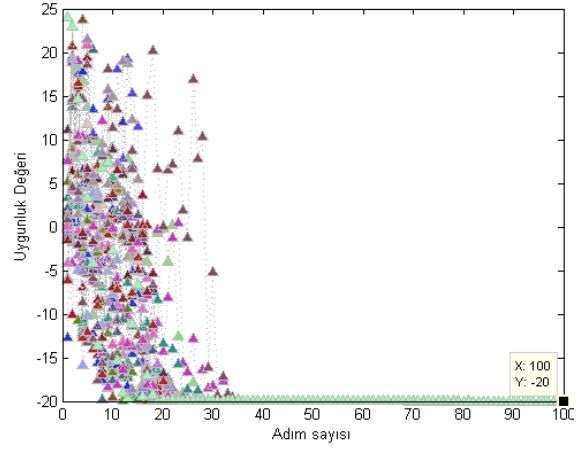
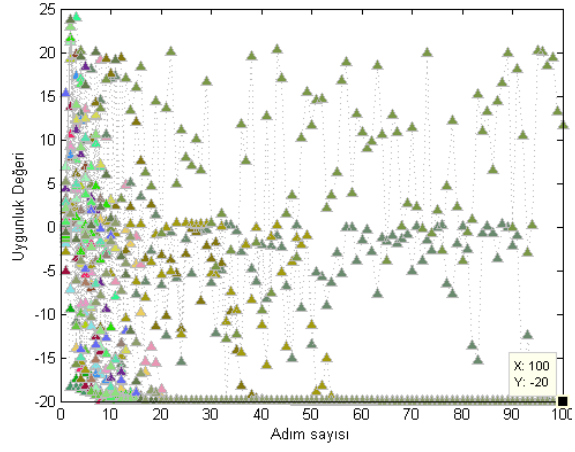
Popülasyon: 10



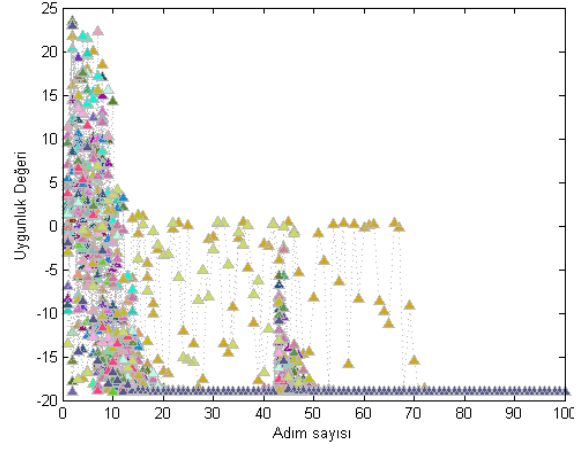
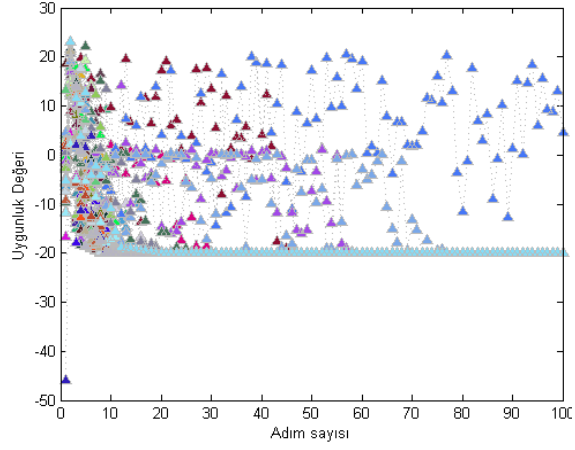
Popülasyon: 20



Popülasyon: 30



Popülasyon: 40



Popülasyon: 50

