T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

TEKERLEKLİ ARAÇLAR İÇİN DİJİTAL ŞASİ DİNAMOMETRESİNİN TASARIMI VE UYGULAMASI

Oğuzkağan AKÇAKAYA

YÜKSEK LİSANS TEZİ

MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI
KONYA, 2010

ÖZET

Yüksek Lisans Tezi

TEKERLEKLİ ARAÇLAR İÇİN DİJİTAL ŞASİ DİNAMOMETRESİNİN TASARIMI VE UYGULAMASI

Oğuzkağan AKÇAKAYA

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Makine Mühendisliği Makine Anabilim Dalı

Danışman: Prof. Dr. Şirzat KAHRAMANLI

2010, 55 sayfa

Jüri: Prof. Dr. Şirzat KAHRAMANLI
Prof. Dr. Ziya ŞAKA
Yrd. Doç. Dr. Behçet DAĞHAN

Bu çalışmada, motorlu taşıt testlerinde motor gücünün ölçülmesi için kullanılan şasi dinamometresinin hazırlanan bir bilgisayar yazılımı ve bu yazılımın üzerinde çalıştığı elektronik kart ile otomasyonu amaçlanmıştır. Bu sistemde dinamometre olarak kabul edilen sistemden gelen elektronik sinyaller ve bu sinyallerden elde edilen veriler bilgisayar ortamına aktarılarak ölçümler sağlanmıştır. Gelen sinyallerin verilere dönüştürülmesinde ve bilgisayara aktarılmasında tarafımızdan tasarlanan bir yazılım ve bu yazılımın üzerinde çalıştığı elektronik kart kullanılmıştır. Bu sistemle motorlu araç lastiklerinden aktarılan güç ve tork değerlerinin belirlenmesi ve grafiklerinin oluşturulması ile verilerin daha kolay değerlendirilmesi sağlanmıştır.

Anahtar Kelimeler: Şasi dinamometresi, Kodlayıcı, Tork, Güç

ABSTRACT

Master Thesis

DESIGN AND APPLICATION OF DIGITAL CHASSIS DYNAMOMETER FOR WHEELED VEHICLE

Oğuzkağan AKÇAKAYA

Selçuk University Graduate School of Natural and Applied Sciences
Department of Machine

Supervisor: Prof. Dr. Şirzat KAHRAMANLI

2010, 55 Pages
Jury: Prof. Dr. Şirzat KAHRAMANLI
Prof. Dr. Ziya ŞAKA
Yrd. Doç. Dr. Behçet DAĞHAN

In this study a computer software of the chasis dynamometer which is used for measuring engine power in auto tests and electronical card which this software functions on and its automation is targeted. In this system electronical signal which comes from the system called as dynamometer and the data which is get from these signals are transmitted into computer and measurements are provided. In the transformation of the coming signals into datas and transmission of the computer,a design which is designed by us and electronical card,which this software functions on,is used. By this system,determination of power and torque values transmitted by vehicle wheels and easier assesment of the datas by constituting of the graphics are provided.

Key Words: Chasis Dynamometer, Encoder, Torque, Power

ÖNSÖZ

Tezin hazırlanması sırasında bana destek olan danışmanım Prof.Dr. Şirzat KAHRAMANLI'ya, yardımlarını esirgemeyen Öğr.Gör. İbrahim KORANA'ya, Prof. Dr. Ziya ŞAKA'ya, mesai arkadaşım Öğr.Gör. Vahit TONGUR'a ve hep yanımda olan çok değerli eşim Selcan'a ve minik patatesime çok teşekkür ediyorum.

İÇİNDEKİLER

ÖZET i

ABSTRACT ii

ÖNSÖZ iii

ŞEKİL LİSTESİ vi

TABLO LİSTESİ viii

KISALTMALAR ix

1.GİRİŞ 1

1.1.İş, Güç, Beygir Gücü ve Tork Kavramları 1

1.2. Araçlarda Faydalı Güç ve Torkun Ölçülmesi 4

1.2.1. Motor dinamometresi 5

1.2.2. Şasi dinamometresi 6

2. KAYNAK ARAŞTIRMASI 9

3. MATERYAL VE METOD 11

3.1. Materyal 11

3.1.1. Dinamometre modeli 11

3.1.2. Doğru akım DC motoru 12

3.1.3. Kodlayıcı (Encoder) 13

3.1.4. Kontrol kartı (Mikrodenetleyici) 15

3.1.5. Bilgisayar 19

3.1.6. Paralel port kablosu 19

3.1.7. USB kablosu 20

3.1.8. Yazılım 21

3.2. Metod 22

3.2.1. Deney kartının tasarımı 22

3.2.1.1. Deney kartı yazılımının amacı 24

3.2.2. Sistemden istenen değerlerin hesaplanması 26

3.2.2.1. Encoder ile haberleşme süresinin belirlenmesi.....27
3.2.2.2. Sistem çıktıları için gerekli değerlerin hesaplanması.....28
3.2.3. Arayüz programı.....35
4. ARAŞTIRMA UYGULAMALARI VE SONUÇLARI.....38
5. SONUÇ VE ÖNERİLER.....46
KAYNAKLAR.....47
EK 1. Kontrol kartı ana programı ve buna ait başlık dosyası.....49
EK 2. Programda kullanılan kesme ve sayıcı fonksiyonlarının oluşturulduğu *EncoderTimer* dosyası ve buna ait başlık dosyası.....52
EK 3. Kontrol kartı ile ara yüz arasındaki USB seri port üzerinden haberleşmeyi sağlayan *encoderUART* dosyası ve başlığı.....54

ŞEKİL LİSTESİ

Şekil 1.1. Bir araca ait Güç-Tork grafiği 3
Şekil 1.2. Motor dinamometresi 6
Şekil 1.3. Şasi dinamometresi 7
Şekil 3.1. Kullanılan dinamometre modeli 12
Şekil 3.2. Kullanılan doğru akım fan motoru 13
Şekil 3.3. Opkon marka encoderin görünümü 14
Şekil 3.4. Bir mikrodenetleyicinin temel bileşenleri 15
Şekil 3.5. Atmega16 mikrodenetleyicinin pin diyagramı 18
Şekil 3.6. Program aktarma kablosu 20
Şekil 3.7. USB kablosu 21
Şekil 3.9. Deney kartının alttan görünümü 24
Şekil 3.10. Modelde kullanılan kayış-kasnak sistemi 29
Şekil 3.11. Açısal Hız-Zaman grafiği 30
Şekil 3.12. Kayış-kasnak sisteminde kuvvetler ve pozitif tork yönü 32
Şekil 3.13. Dinamometrede lastiğin silindirlere uyguladığı kuvvetler 33
Şekil 3.14. Motor gücünün zamana bağlı olarak değişimi 33
Şekil 3.15. Lineer grafiklerin birleşiminden oluşan Güç-Zaman grafiği 34
Şekil 3.16. Geliştirilen kullanıcı ara yüz yazılımı ana penceresi 35
Şekil 3.17. Ara yüze gelen değerleri okuma penceresi 36
Şekil 3.18. Kaydedilen verilerin görüntülendiği pencere 37
Şekil 4.1. 12 V - 3.5 A'de 10 defa yapılan testlerin sonucunda elde edilen Güç-Zaman grafikleri 39
Şekil 4.2. 15 V - 3.8 A'de 10 defa yapılan testlerin sonucunda elde edilen Güç-Zaman grafikleri 40

Şekil 4.4. Farklı güçlerde elde edilen birleştirilmiş Güç-Zaman grafikleri 42
Şekil 4.5. Tork-Devir grafiği 43
Şekil 4.6. Güç-Devir grafiği 43
Şekil 4.7. Devir-Zaman grafiği 44
Şekil 4.8. Hız-Zaman grafiği 44
Şekil 4.9. Tork-Zaman grafiği 45
Şekil 4.10. Hız-Devir grafiği 45

TABLO LİSTESİ

Çizelge 3.1. Kullanılan doğru akım fan motorunun teknik özellikleri…… ..13

Çizelge 3.2. Kullanılan encoderin teknik özellikleri……. 15

Çizelge 3.3. Atmega16 mikrodenetleyicinin pin açıklamaları…….. 19

Çizelge 3.4. Devre üzerindeki giriş-çıkış sayıları ve bağlı bulundukları port adları……...….... 22

KISALTMALAR

D_{sil}	Silindir çapı
D_{mot}	Motor kasnak çapı
n	Encoderdan okunan darbe sayısı
n_{sil}	Silindir devir sayısı
n_{mot}	Motor kasnak devir sayısı
t_1	İlk okuma zamanı
t_2	Son okuma zamanı
ω_1	İlk açısal hız
ω_2	Son açısal hız
$\propto_1$	İlk açısal ivme
$\propto_2$	Son açısal ivme
T	Tork (Moment)
P	Güç
I	Silindir atalet momenti
i	Kayış kasnak sistemindeki çevrim oranı
F	Sürtünme kuvveti
m	Silindirin kütlesi

1.GİRİŞ

Motorlu araçlar sektörü, Dünyada ve Türkiye'de yük ve insan taşıma amacına yönelik olarak imal edilmiş olan araçları kapsayan lokomotif sektörlerden biridir. Bugün, dünyada motorlu araçlar sektörünün ürettiği motorlu araç sayısı yaklaşık 58 milyon civarındadır. Bu araçların yaklaşık 41 milyon adedi otomobil, 17 milyon adedi kamyon, otobüs, çekici gibi ticarî araçlardır. Dünyada 1000 kişiye düşen araç sayısı 116'dır. Türkiye'de bu sayı 111, Avrupa Birliği ülkelerinde ise 534'tür (MEGAP, 2007). Bu araçlar gerek teknik gerekse fiziki özellikleri açısından farklılıklar göstermektedir. Özellikle motor gücü ve verimi bu araçların tercihlerinde önemli bir paya sahiptir.

Araçların laboratuar ortamında test edilmesi ile performans değerlendirmeleri yapılabilmektedir. Test sonuçlarının güvenilirliği çok fazla sayıdaki parametrenin ölçülmesi ve deneysel verilerin doğru bir şekilde kaydedilmesine bağlıdır. Bunun için birçok donanıma sahip motor test standlarına ihtiyaç duyulmaktadır. Araştırmalarla motorların laboratuar ortamında test edilmeleri ile hedeflenen amaca ulaşıp ulaşamadıkları kontrol edilmektedir. Bu testler yeni geliştirilen araçlar üzerinde yapılabileceği gibi üretim aşamasında üretimin denetlenmesi için de yapılabilir. Yenileştirilmiş motor ve taşıtlara da bu gibi testler uygulanarak sonuçların tatmin edici olup olmadığı araştırılabilir (Çelik, Bayır, Özdalyan, 2007).

1.1.İş, Güç, Beygir Gücü ve Tork Kavramları

İş; Tatbik edilen bir kuvvetin hareket meydana getirebilme kabiliyetine denir. Hareket itme veya çekme şeklinde olur.

İş = kuvvet × yol

formülü ile bulunur ve birimi Nm (Joule)'dir.

Güç; Birim zamanda yapılan iş olarak tanımlanır. Gücün formülü:

Güç = İş / Zaman = (kuvvet × yol) / Zaman

formülü ile bulunur ve birimi Nm/s (Watt)'dir (Petchers, 2003).

Beygir Gücü; Pratikte güç birimi olarak beygir gücü kullanılır. Motorlu araçların ve elektrik motorlarının güçlerinin belirlenmesi için "Beygir gücü" (HP) birimi kullanılmaktadır. 1 beygir: 75 kg'lık ağırlığı 1 sn'de 1 metre hareket ettirecek güç miktarı olarak tanımlanır. Bunun da yaklaşık bir atın gücüne denk geldiği düşünülerek "beygir gücü" denilmiştir.

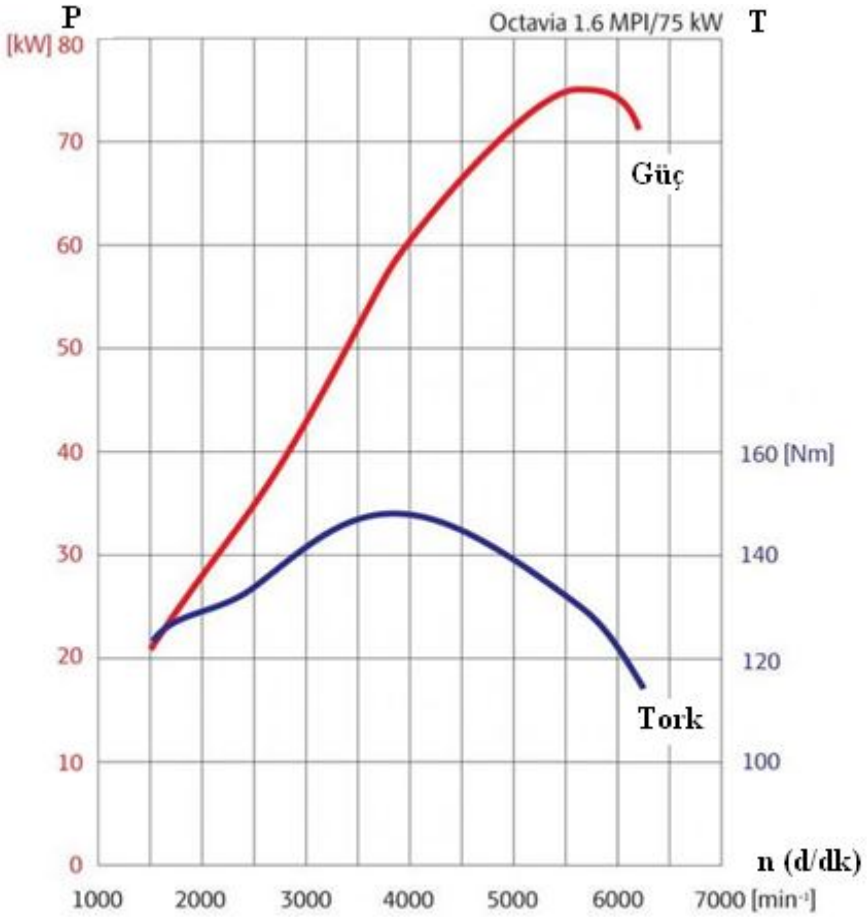
1 Beygir Gücü denilince saniyede 75 kgf.m lik iş yapıldığı kabul edilir. Yani;

1HP = 0,736 W' dır.

Motor gücü, araç kataloglarında HP, PS veya kW olarak verilir. HP ingilizce'de (horse power), PS Almanca'da (pferd starke) yani beygir gücü anlama gelen sözcüklerin baş harfleridir. HP ve PS beygir gücü olarak birbirine ihmal edilecek kadar yakındır. Bu yüzden her ikisi için de 75 kgf.m/s olarak söylenir. Bu ufak farkı ortadan kaldırmak için bazı motorların gücü, elektrikte güç birimi olarak kullanılan kW cinsinden verilir, kW olarak HP ve PS arasındaki fark;

1 PS = 0,736 kW; 1 HP = 0,746 kW' tır.

Beygir gücü, motorun güç çıkışının bir ölçüsüdür. Beygir gücü motora has değiştirilemez bir değerdir. Ama tork, aktarma organları, dişliler, vites kutusu, tekerlek çapı vs. gibi etkenlerle değiştirilebilir. Beygir gücü ise bunların tamamından bağımsız motorun karakteristik bir özelliğidir. Aslında tork da çıplak bir motorda belli bir değerle ifade edilebilir fakat bu güç yere iletilen itme kuvveti olarak tanımlandığından tek başına motora ait karakteristik bir veri olarak düşünülemez (URL1).



Şekil 1.1. Bir araca ait Güç-Tork grafiği

Tork; belirli bir motor devri için üretilen güç karşılığında ölçülen çevirme kuvveti olarak tanımlanır (Bolat, 2007). Kısaca tork, dairesel döndürme etkisi (Won, Langari, 2002) yani motordan tekerleğe iletilen dönme momentidir. Birimi Nm (Newtonmetre)'dir. Bu kuvvet motorun gücü ile doğrudan ilişkili değildir. Bu değer tekerlek çapı, vites kutusu ve motor tipi gibi etmenlere bağlı olarak değişir. Bir motorun tork eğrisi ne kadar düz bir çizgi şeklinde ilerliyorsa, motor o kadar verimlidir. Araçlar farklı devirlerde farklı bir tork verirler. Yani her bir devir için araç o devirde verdiği torka bağlı olarak farklı bir güce sahip olur. Genelde üreticiler en düşük devirde maksimum torka ulaşmayı ve bu torku maksimum gücün alındığı devire kadar fazla düşmeden tutmayı hedeflerler.

Bir motorun ürettiği tork motorun devrine göre değişir ve grafikte bir eğri ile gösterilir. Tam düz bir tork "eğrisi" olsaydı otomobil hangi viteste, hangi devirde olursa olsun, sabit ivme ile hızlanırdı. Ancak gerçek hayatta grafikte tork eğrileri düz değildir. Tavan yapmak üzere yükselirler ve çok yüksek devirlerde yine düşerler. Bir otomobil, motoru en yüksek torkunu verdiği devirde seyrettiği zaman en iyi şekilde hızlanır.

Çalışmamızda sonuç veri olarak lastiklerden aktarılan güç ve tork değerleri belirlenmiş ve grafikleri oluşturulmuştur. Gerçek sürüş koşullarında aracı etkileyen ana kuvvetler; itici güç (tork), frenleme kuvveti, aerodinamik yapı ve yerçekimi (yamaçlarda) kuvvetidir.

1.2. Araçlarda Faydalı Güç ve Torkun Ölçülmesi

Araç motorlarında iki çeşit güç vardır. Bunlardan birincisi, silindir içerisine giren karışımın yanması ile meydana gelen ısı enerjisinin mekanik enerjiye dönüşmesi sırasında elde edilen, indike güçtür. İyi bir yakıt ekonomisi ve emisyon için, motor yanma veriminin, motorun bütün çalışma şartlarında maksimum seviyede tutulması gerekmektedir. Her bir işletme parametresinin yanma üzerindeki etkisi farklı olmaktadır. Dolayısıyla mümkün olan çok sayıda parametreden, yakıtın en verimli biçimde enerjiye dönüşümünü sağlayacak şekilde yararlanılmalıdır (Çelik, Balcı, 2002). Diğeri ise mil, fren veya faydalı güç olarak adlandırılan ve motorun volanından veya kasnağından ölçülen güç olup motorun gerçek gücüdür. Yani motor içerisindeki sürtünme ve diğer kayıplardan sonra kalan gerçek güçtür.

Çalışmamızda hesaplanmış olan güçler araç lastiklerinden elde edilmiş olan gerçek güçtür. Bu güç değerlerinin belirlenebilmesi için araç üzerinde bazı performans testlerinin yapılması gereklidir.

Faydalı güç dinamometre ile ölçülür (Batmaz, 1998). Dinamometre güç ölçü cihazlarının genel adıdır. Genel olarak ağırlık ve cisme etkiyen kuvveti ölçen aletler olarak tanımlanmaktadırlar. Döner bir makinenin çıkış kuvvetini ölçmede kullanılan aygıttır.Test amacı için imal edilmiş stantlar yapısal olarak birbirlerine benzemekle beraber teknolojik imkânlarına ve ölçme sistemlerindeki farklılıklara göre değişiklikler

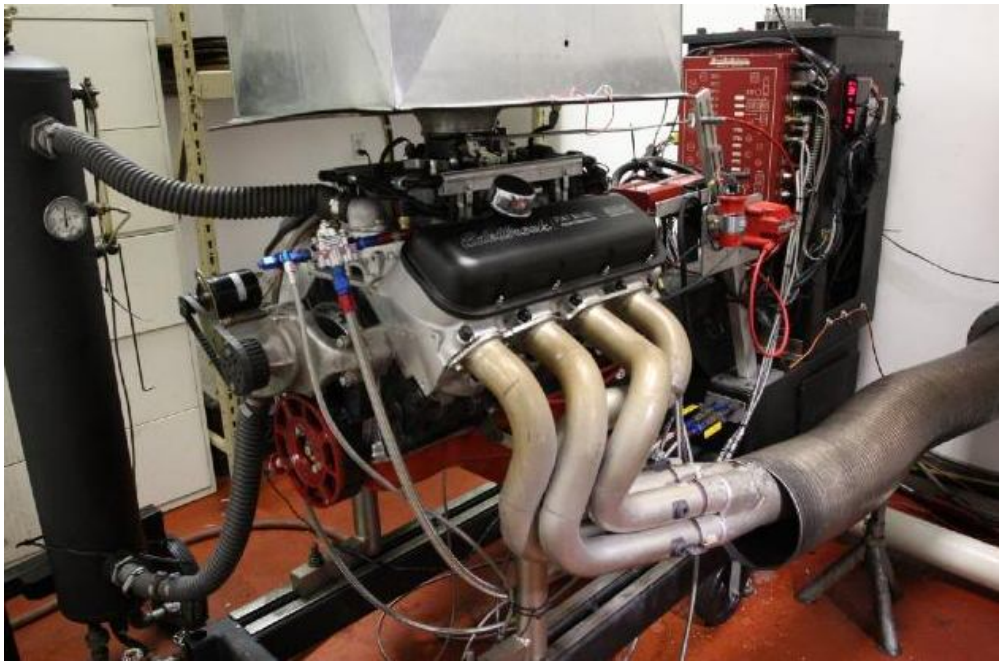
gösterirler. Dinamometre en çok, bir elektrik motorunun ya da bir otomobil motorunun beygir gücünü ölçmede kullanılır. Döner parçaya uygulanan burma kuvveti (moment) ile açısal hızın çarpımı, kuvveti verir. Kuvvet ölçen dinamometrelerden en yaygın kullanılanı, esnek bir metal halkadan oluşur. Bu halkayı sıkıştıracak biçimde bir kuvvet yüklendiğinde halka burulur ve burulma miktarına göre kuvvet ölçülür. Cismin uyguladığı kuvvet ne kadar büyükse yay o kadar gerilir. Dinamometrede ölçülen değerler, Newton birimiyle (N) gösterilir. Bu cihazlar motorun ürettiği gücü kendi bünyelerine alarak harcarlar.

Dinamometreler ölçtükleri vasıtaya göre birçok kategoriye ayrılırlar. Motor dinamometreleri, şasi dinamometreleri, endüstriyel dizel motor dinamometreleri, karting dinamometreleri, marine dinamometreler, motosiklet dinamometreleri, ve snowmobile dinamometreleri bunlardan bazılarıdır.

Ancak dinamometreler yapısal olarak motor dinamometresi ve şasi dinamometresi olarak ikiye ayrılabilir.

1.2.1. Motor dinamometresi

Genellikle fabrikalarda kullanılır. Araç üzerinden sökülen motorun volan çıkışı özel aparatlar yardımı ile dinamometreye bağlanır. Elektrik bağlantıları yapılarak motor çalıştırılır ve volan çıkışındaki gücün ölçümü yapılır. Ölçülen güç sadece motordaki güçtür ve sistemin pratikte en büyük dezavantajı motorun ve elektrik bağlantılarının araç üzerinden sökülmesi gereğidir. Bir motor dinamometresi örneği Şekil 1.2'de görülmektedir. Bu dinamometrenin en önemli özelliği, yüksek ölçme hassasiyeti ile birlikte, hem bir elektromotor olarak hareket üretimi ve motorlara güç verilmesinde, hem de jeneratör olarak hareket üreten makinelerin güçlerinin yutulmasında kullanılabilmesidir (Çelik, Bayır, Özdalyan, 2007).



Şekil 1.2. Motor dinamometresi

1.2.2. Şasi dinamometresi

Her marka araç üzerine çalışan servislerin ve ölçüm merkezlerinin kullandığı sistemdir. Aracın çekiş, tekerlekleri yer seviyesinde bulunan iki adet silindir arasına alınır. Araç sabitlenir ve tam gaz pozisyonunda, araç sabit devirde tutulmak suretiyle ölçüm yapılır. Ölçümler motor çıkışı veya tekerlek çıkışı olarak farklı farklı yapılabilir.

Şasi dinamometreleri araçların yol simülasyon testleri için çok faydalı araçlardır. Ancak bu sistemlerin maliyetleri oldukça yüksektir. Şasi dinamometrelerinin tasarımında birçok varyasyon vardır, ama hepsi ortak bazı temel özelliklere sahiptir: ve sistem temelde aynıdır. En kolay ölçme yöntemi zeminde döndürülen tekerleklerden iletilen güç ile yapılandır. Bu sistemler çok fazla kullanılmaktadır ve taşınabilir özelliktedir (French, Stark, 2000).

Hafif tip araç testleri çoğunlukla bir şasi dinamometre üzerinde önceden tanımlanmış bir test çevrimine göre yapılır. Bu test aynı zamanda gerçek sürüş koşullarının da bir simülasyonudur (Pelkmans, Debal, 2006).

Çalışmamızda kullandığımız şasi dinamometresi, araç motorlarında yapılan tüm ayarların atölye içinde ölçülmesi için gerçek yol şartlarını sağlayan cihazlardır. Bu cihazların hidrolik ve elektronik tipte olanlar vardır. Şasi dinamometresinde ölçülen gücün özelliği, tekerlek gücü olmasıdır (Anthony, Alexander, 2007; Ribbens, Mansour, 2003). Yani motordan tekerlere kadar olan tüm sürtünmeler bu gücün dışındadır. Burada ölçülen güç, araca hareket veren tekerleklerden alınan tamamı faydalı güçtür. ölçüm, motor araç üzerinde iken yapılır. Araca hareket veren tekerler cihazın silindirleri (tamburları) üzerine bindirilir. Motor çalıştırılır ve vitese takılır. Tekerlekler tamburlar üzerinde döner. Cihazın fren tertibatı yardımıyla tamburların döndürülmesi istenildiği kadar zorlaştırılır. Bu şekilde motorun, istenilen devirde değişik yüklerle yüklenmesi sağlanır.



Şekil 1.3. Şasi dinamometresi

Dinamometreler metallerin esneklik özelliğinden yararlanılarak yapılmıştır. İç içe geçmiş iki borudan oluşur. İçteki boruda yay asılıdır. İçteki borunun üzeri eşit olarak bölmelendirilmiştir. Cisim içteki borunun ucundaki çengele takılır. Yer cismi ne kadar kendine doğru çekebilirse o cismin ağırlığı o kadardır.

Bu çalışmada, motorlu taşıt testlerinde motor gücünün ölçülmesi için kullanılan şasi dinamometresinin hazırlanan bir bilgisayar yazılımı ve bu yazılımın üzerinde çalıştığı elektronik kart ile otomasyonu amaçlanmıştır. Bu sistemde test motorundan gelen elektronik sinyaller ve bu sinyallerden elde edilen veriler bilgisayar ortamına aktarılarak ölçümler sağlanmıştır. Gelen sinyallerin verilere dönüştürülmesinde ve bilgisayara aktarılmasında tarafımızdan tasarlanan bir yazılım ve bu yazılımın üzerinde çalıştığı elektronik kart kullanılmıştır.

Hazırlanan düzenekte bir deney motoru, bir şasi dinamometresi modeli, bilgisayar, veri alış veriş kartı ve motordan gelen gücü hesaplayan bir yazılım ünitesi mevcuttur. Dinamometre kontrolü için bilgisayar kontrollü dijital sinyal işlemci, geliştirdiğimiz bir yazılım kullanılmıştır. Bu sistemle motorlu araç lastiklerinden aktarılan güç değerlerinin belirlenmesi ve grafiklerinin oluşturulması ile motorlara ait performans değerlendirmesi kolaylıkla yapılabilmektedir. Oluşturulan test düzeneği piyasada mevcut ünitelere göre kıyaslanamayacak kadar daha ucuza mal edilebilmektedir ve geliştirilmeye de oldukça müsaittir.

2. KAYNAK ARAŞTIRMASI

Anthony, Alexander (2007); Bu kitap motor test uygulamaları ve teoriyi bir arada sunarak geniş ve dağınık bilgileri biraya getirmiştir. Motor testlerinin yeni motor gelişiminin temel bir parçası olduğundan bahsedilmektedir.

Batmaz (1998) yaptığı çalışmada; motorlu taşıt testlerinde kullanılan şasi dinamometresi, yakıtmetre ve egzoz gaz analiz cihazlarının bilgisayar ile otomasyonu amaçlanmıştır.

Bolat (2007) tarafından yapılan yüksek lisans tez çalışmasında, traktörlerin daha ucuz maliyetle işletilebilmesi ve bunun karşılığı olarak motorda iş başarısının etkilenme derecelerinin bulunması amacıyla karakteristik değerlerinin değişimleri üzerinde durulmuştur.

Çelik, Balcı (2002), sabit bir yük ve sabit hız şartlarında sıkıştırma oranının motor karakteristiklerine etkisini çeşitli testlerle incelemişlerdir.

Çelik ve ark. (2007), çalışmalarında içten yanmalı bir motoru bilgisayar kontrolünde test edebilen ve deneysel verileri eşzamanlı olarak bilgisayar ortamına aktarabilen bir test standının tasarımı ve imalatı yapılmıştır.

French, Stark (2000), bu makalede şasi dinamometreleri ile ilgili temel bilgiler verilmekte özellikleri anlatılmaktadır.

MEGEP (2007) Milli Eğitim Bakanlığınca hazırlanan bir kitapçıktır ve otomotiv sanayi hakkında bilgi verildikten sonra bu alandaki iş kolları tanıtılmaktadır.

Pelkmans, Debal (2006), yaptıkları çalışmada belirli bir mesafe-hız profilinde bir aracın yakıt tüketimini tahmin etmek için geliştirilmiş bir yazılım paketi kapsamında yürütülen çalışmalar açıklanmaktadır. Modelde, özel hafif ticari araçlar için motor dinamometreleri ile yoğunluk ölçümleri, şasi dinamometreleri ile de zemin ispatı ve gerçek trafik verileri girişi sağlanmıştır.

Petchers (2003), mortların çalışma prensipleri hakkında genel bilgiler veren bir elkitabıdır.

Ribbens, Mansour (2003); kitapta otomobil elektrik operasyonları ile ilgili en son teknolojik gelişmeleri kapsamakta ve elektronik sistemler ve bileşenlerini tanıtmaktadır. Şasi dinamometreleri hakkında bilgi içermektedir.

Tongur (2008); bu tez çalışmasında Atmega128 mikrodenetleyici tabanlı bir PLC tasarımı gerçekleştirilmiştir. Tasarlanan PLC modeli için kontrol devrelerinin simülasyonunu yapabilen, hex kodlar üretebilen bir arayüz yazılımı geliştirilmiştir. Hex kodlarının çalışmasını sağlayan bir yorumlayıcı geliştirilerek mikrodenetleyicinin sabit belleğine yerleştirilmiştir.

Won, Langari (2002), yaptıkları çalışmada elektrikli araçlar için enerji yönetimi kapsamında bulanık mantık yöntemi ile bir bulanık tork dağılımı kontrol önerilmiştir.

3. MATERYAL VE METOD

Bu tez çalışmasında bir şasi dinamometresinin simülasyonu için bir model üretip, dinamometrenin gerçeklenmesi amaçlanmıştır. Dinamometreden gelen darbelerin değerlendirilmesi için Atmel firmasına ait atmega serisi bir mikrodenetleyicili kart kullanılmıştır. Mikrodenetleyicide çalışacak yazılım, yine aynı firma tarafından geliştirilen Avr Codevision C derleyici programı kullanılmıştır.

C++ programlama dili kullanılarak mikro denetleyici programlanmış ve bir arayüz programı hazırlanmıştır. Encoderdan gelen darbeler mikrodenetleyici üzerindeki 1 baytlık değişken üzerinden bilgisayara taşınması USB portu aracılığı ile gerçekleşmektedir. Bunun için FTDI firmasına ait ft232BM entegresi kullanılmıştır.

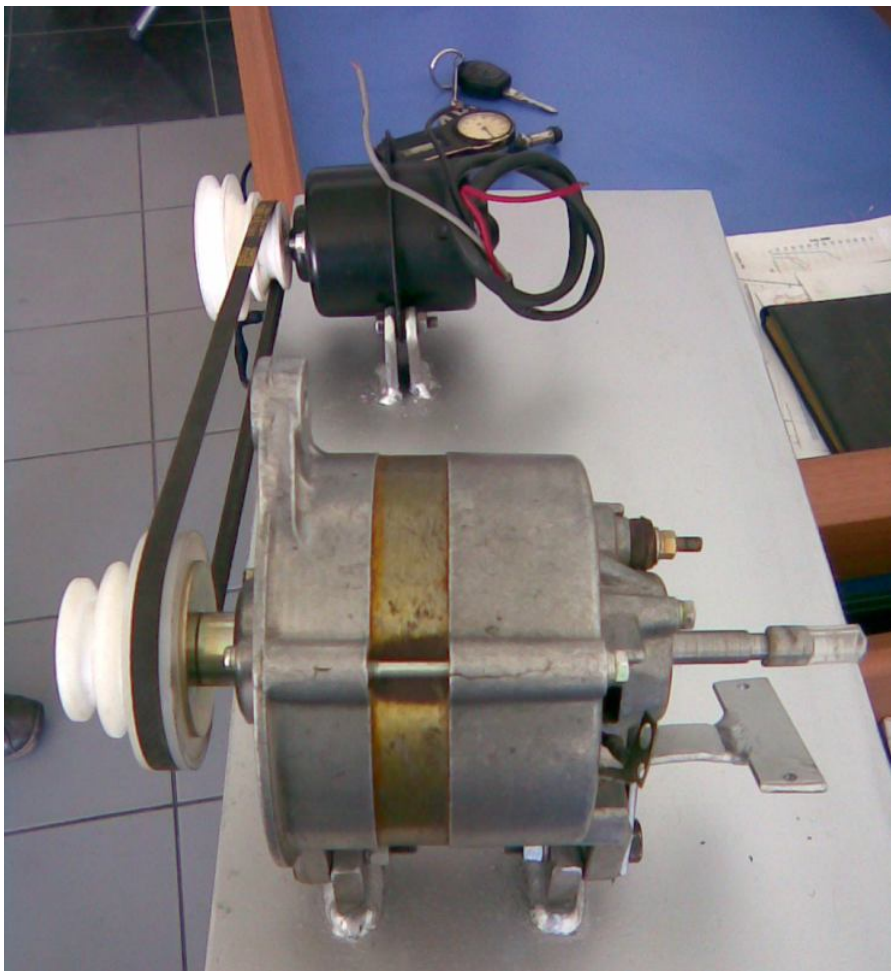
Yorumlayıcının mikrodenetleyici üzerine aktarılması paralel port aracılığı ile gerçekleştirilmiştir.

3.1. Materyal

3.1.1. Dinamometre modeli

Çalışmanın en önemli elemanlarından birisi dinamometredir. Motorlar ürettiği efektif gücü veya döndürme momentini işe veya elektrik enerjisine dönüştürerek yutan makinelere dinamometre adı verilmektedir. Dinamometreler ile motorların değişik devir sayılarındaki dönme momentleri ölçülebilmektedir. Herhangi bir doğru akım elektrik jeneratörü, motoru yüklemek için, bir dinamometre olarak kullanılabilir. Doğru akım dinamometreleri elektrik motoru olarak da kullanılabildiği için değişik hızlarda test motorunun döndürülmesi ile sürtünme güçleri de belirlenebilmektedir (Çelik, Bayır, Özdalyan, 2007). Bu çalışmada dinamometre modeli olarak Şekil 3.1'deki yapı kullanılmıştır.

Çalışmada kullandığımız dinamometre modelinde;
Silindirin çapı (D_{sil})=0,082 m.
Silindirin kütlesi =1.835 kg'dır.



Şekil 3.1. Kullanılan dinamometre modeli

3.1.2. Doğru akım DC motoru

Tekerleklerden gelen gücün temsili için 12 voltluk bir fan motoru tercih edilmiştir (Şekil 3.2). Çalışmamızda temsili tekerlek kuvvetini vermesi için seçilen doğru akım fan motorunun teknik özellikleri Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Kullanılan doğru akım fan motorunun teknik özellikleri

Marka/Model	OPS / OAT03-300000
Voltaj	12 V DC motor
Max. Hız	3300 d/dk
Ağırlık	0,30 kg



Şekil 3.2. Kullanılan doğru akım fan motoru

3.1.3. Kodlayıcı (Encoder)

Encoderler yani sinyal üreticiler, bağlı olduğu şaftın hareketine karşılık, sayısal (dijital) bir elektrik sinyali üreten elektromekanik bir cihazdır. Kısaca dakikada maximum devir sayısını ölçerler. Encoderler çalışma şekillerine göre dönel olarak

çalışan Şaft Encoderlar ve doğrusal olarak çalışan Lineer Encoderlar olmak üzere ikiye ayrılırlar. Testere dişli bir kol yada ölçü tekerleği gibi uygun bir mekanik dönüştürme cihazı ile birlikte kullanıldığı zaman şaft encoderlar, açısal yer değiştirme, lineer ve dairesel hareket, dönüş hızı, ivme,gibi büyüklükleri ölçmek için kullanılabilir (URL2). Çalışmamızda encoder tekerleğin dönüş hızını bulmak için kullanılmıştır. Encoderlar özelliklerine göre farklı sayılarda darbeler üretirler. Bu darbeler encoderin bir tam tur attığında üreteceği darbe sayısıdır. Darbe sayısı ne kadar yüksek olursa encoder o kadar hassas olur. Çünkü darbe sayısı arttıkça nokta aralıkları daha da azalır.

Encoder olarak Opkon markasının 021209-03 serisi ve PRI50 TP360 Z V3 2m5 numaralı modeli kullanılmıştır. Encoder Şekil 3.3'de görülmektedir. Encoderin teknik özellikleri Çizelge 3.2'de verilmiştir.



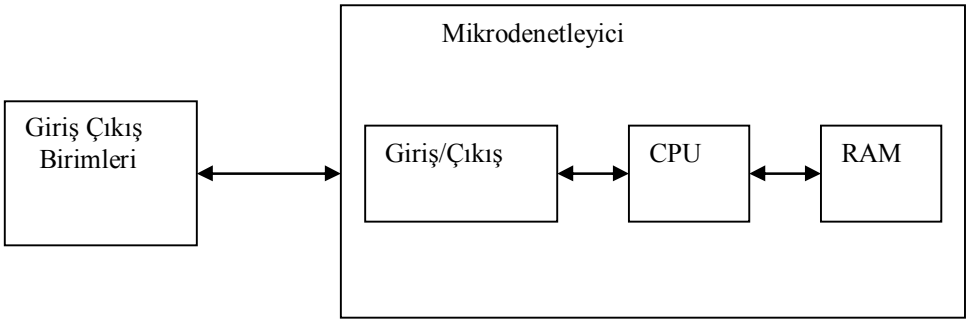
Şekil 3.3. Opkon marka encoderin görünümü

Çizelge 3.2. Kullanılan encoderin teknik özellikleri

Çözünürlük	360 darbe/tur
Besleme Voltajı	5 V DC
Çalışma Sıcaklığı	-20° C / + 80° C
Mil Çapı	8 mm
Gövde Çapı	50 mm
Max. Tur Hızı	5000 d/dk
Çalışma Sıcaklığı	-20° C / + 80° C

3.1.4. Kontrol kartı (Mikrodenetleyici)

Sayısal devreler günümüzde kullanılan elektronik sistemlerinde sıkça karşımıza çıkmaktadır. Bu devreler sayesinde birçok aritmetik ve mantıksal işlemi yapmamız mümkündür. Endüstriyel ortamlarda da kullanılan bu devrelerin amacı giriş sinyallerine göre çıkışlar üretmektir. Bu işlemi yapabilmek için de merkezi işlem birimi ve işlem yapabilmek için hafıza bölgelerine ihtiyaç duyulur. Mikrodenetleyiciler bu noktada öne çıkarlar. Veri işlenebilmesi için işlemci biriminin giriş ünitelerinden aldığı verileri geçici hafıza bölgelerinde işleyerek çıkış ünitelerine aktarması gerekmektedir.



Şekil 3.4. Bir mikrodenetleyicinin temel bileşenleri

Mikrodenetleyiciler merkezi işlem birimi, giriş/çıkış birimleri, geçici hafıza denilen RAM bellekleri ve kalıcı program hafızalarını tek bir entegre içerisinde toplamıştır. Bir mikrodenetleyici genel olarak aşağıdaki gibi ifade edilir (Tongur, 2008).

Çalışmada kullanılan Atmega16 Mikrodenetleyicinin temel özellikleri aşağıda verilmiştir;

1. 8 bit veri genişliği
2. Gelişmiş RISC mimarisi içerisinde
    a. 133 komut
    b. 32 x 8 genel amaçlı saklayıcı ve çevresel kontrol saklayıcısı
    c. 16 MIPS e kadar 16 MHz de çalışma imkanı
    d. Dahili iki çevrim saklayıcı
3. Kalıcı program ve veri hafızası
    a. 128 Kbyte programlanabilir FLASH
    b. 4 Kbyte EEPROM
    c. 4 Kbyte dahili SRAM
    ç. Belleklerin yazılıp silinebilme ömrü: 10.000 FLASH Bellek, 100.000 EEPROM
    d. 64 Kbyte a kadar dâhili seçimlik hafıza alanı
    e. Yazılım güvenliği için program kilitleme özelliği
    f. Sistem programı için SPI arayüzü
4. Çevresel özellikleri
    a. İki tane 8 bitlik zamanlayıcı, sayıcı ve karşılaştırma modu
    b. İki tane genişletilmiş 16 bitlik zamanlayıcı, sayıcı ve karşılaştırma modu
    c. Gerçek zaman sayıcısı
    ç. İki tane 8 bitlik PWM kanalı
    d. 2 bitten 16 bite kadar program çözünürlüğünde 6 PWM kanalı
    e. Karşılaştırmalı çıkış modülatörü
    f. Byte uyumlu iki girişli seri arayüz

g. Çift programlanabilir seri USART

ğ. Master/Slave SPI seri arayüzü

h. Dahili osilatör ile programlanabilir watchdog zamanlayıcısı

ı. Dahili analog karşılaştırıcı

5. Farklı Özellikleri

a. Güç ve program kontrolüyle resetleme

b. Dâhili kalibrasyonlu RC osilatörü

c. Harici ve dâhili kesme kaynakları

ç. 6 adet uyku modu

ç.1. Rölanti modu

ç.2. ADC gürültü indirgeme modu

ç.3. Güç koruma modu

ç.4. Güç kesilme modu

ç.5. Bekleme (Standby) modu

ç.6. Genişletilmiş bekleme modu

d. Seçilebilir saat frekansı yazılımı

6. Girişler / Çıkışlar

a. 53 tane programlanabilir giriş çıkış portu

7. Çalışma gerilimleri

a. ATmega 16L için 2,7 – 5,5 V arası

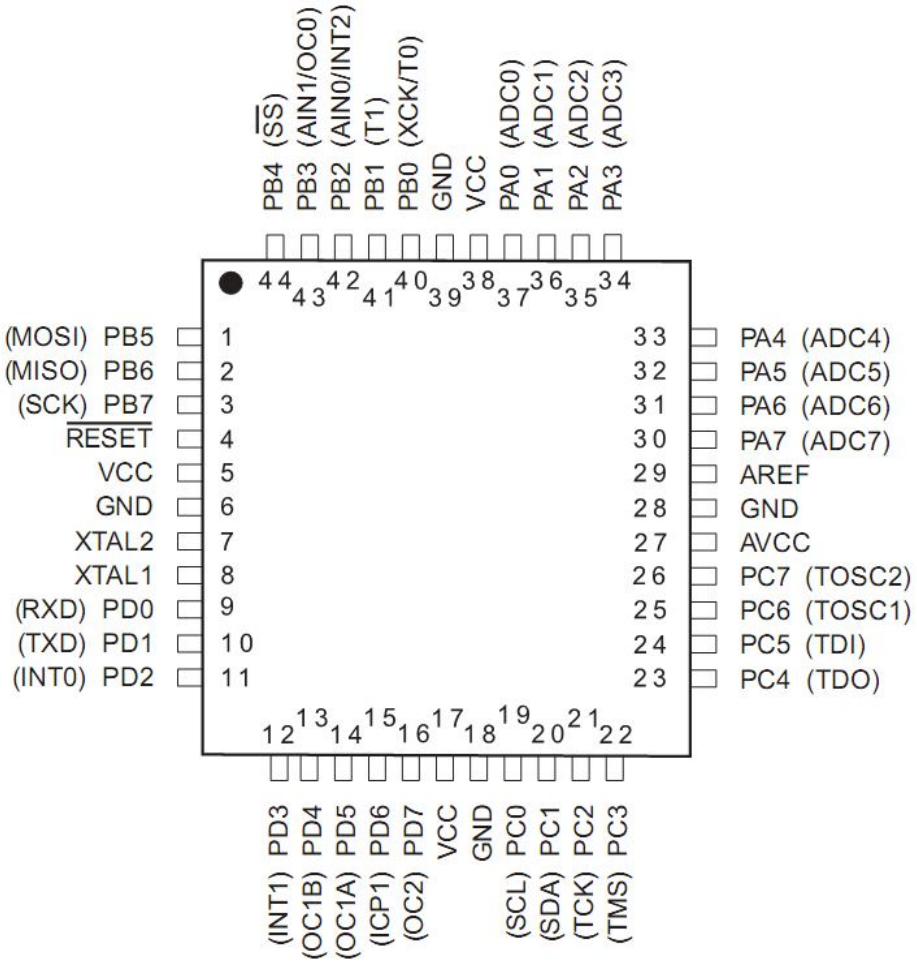
b. ATmega 16 için 4,5 – 5,5 V arası

8. Hız

a. ATmega 16L için 0 – 8 Mhz arası

b. ATmega 16 için 0 – 16 Mhz arası

Atmega16 Mikrodenetleyicinin pin diyagramı Şekil 3.5'da gösterildiği gibidir.



Şekil 3.5. Atmega16 mikrodenetleyicinin pin diyagramı

Atmega16 mikrodenetleyicinin pin açıklamaları kısaca Çizelge 3.3'de verilmiştir. (URL3).

Çalışılan kartın tasarımı konusuna tezin Metod başlığı altındaki Deney kartının tasarımı başlığı altında değinilmiştir. Deney kartı hakkında detaylı bilgi bu bölümde verilmiştir.

Çizelge 3.3. Atmega16 mikrodenetleyicinin pin açıklamaları

Pin Adı	**Açıklaması**
VCC	Sayısal besleme gerilimi
GND	Toprak
PortA (PA7..PA0)	8 bitlik giriş/çıkış portu
PortB (PB7..PB0)	8 bitlik giriş/çıkış portu
PortC (PC7..PC0)	8 bitlik giriş/çıkış portu
PortD (PD7..PD0)	8 bitlik giriş/çıkış portu
RESET	Reset girişi
XTAL1	Tersleyen osilatör yükseltecine ve dahili saat işletim devresine giriş
XTAL2	Tersleyen osilatör yükseltecinden çıkış
AVCC	PortF ve A/S dönüştürücü için besleme pini
AREF	AREF A/S dönüştürücü için anolog frekans pini
PEN	SPI seri programlama modu için programlamayı etkinleştirme pini

3.1.5. Bilgisayar

Çalışmada HP Nc6120 Centrino Sonoma dizüstü bilgisayar kullanılmıştır. Bilgisayarın işlemci hızı 1,73 GHz, belleği 1 GB, harddiski 60GB'tır ve 128MB paylaşımlı ekran kartı kullanmaktadır.

3.1.6. Paralel port kablosu

Şekil 3.6'daki kablo, oluşturulan yorumlayıcıyı PC tarafından kart üzerindeki mikrodenetleyiciye aktarmamızı sağlamaktadır. Kablo PC tarafında paralel porta takılırken deney kartına ise Şekil 3.8'deki "D" ile isimlendirilen pinlere takılmaktadır. Bu sayede Atmega üzerindeki FLASH veya EEPROM belleklere veri yazılabilmekte veya mevcut veriler okunabilmektedir.

Bu projede Şekil 3.6'daki kablo, yorumlayıcının FLASH belleğe yazılması için kullanılmaktadır. Yorumlayıcı kart üzerine yerleştikten sonra yorumlayıcının işlemesi için gerekli olan veri aktarımı USB portu aracılığı ile gerçekleştiğinden dolayı yorumlayıcının eğer varsa bir sonraki güncellenmesine kadar bu kabloya ihtiyaç duyulmaktadır.



Şekil 3.6. Program aktarma kablosu

3.1.7. USB kablosu

Şekil 3.7'deki USB kablosu, PC ile kart arasındaki veri aktarım görevini üstlenmektedir. Entegre üzerindeki yorumlayıcıya işlenecek veri hazırlanılan programcı arayüz ortamında oluşturulduktan sonra USB portu kanalı ile PC tarafından entegredeki EEPROM belleğe yazılmaktadır. Her yeni programın kart üzerine aktarılması esnasında bu kabloya da ihtiyaç duyulmaktadır.

USB üzerinden bu veri aktarımının gerçekleşmesi için ft232BM sürücüsünün PC tarafına yüklü olması gerekmektedir.



Şekil 3.7. USB kablosu

3.1.8. Yazılım

Deney kartı üzerindeki yorumlayıcı için hazırlanan yazılım mikrodenetleyiciler için geliştirilmiş olan C++ programlama dili ile yapılmıştır. Ara yüz için hazırlanan yazılımlar ise genel C++ programlama dili ile yapılmıştır. C++, 1980'lerin başlarında Bjarne Stroustrup tarafından geliştirilen C'yi kapsayan ve çok paradigmalı, yaygın olarak kullanılan, genel amaçlı bir programlama dilidir. İlk olarak *C With Classes* (Sınıflarla C) olarak adlandırılmış, 1983 yılında ismi C++ olarak değiştirilmiştir. C++ dili 1979'da ABD'de Bell Laboratuvarları'nda geliştirilmiştir.

Genel olarak her C programı aynı zamanda bir C++ programıdır, ancak her C++ programı bir C programı değildir. Bu durumun bazı istisnaları mevcuttur. C++'ı C'den ayıran ve C++'ın nesne yönelimli bir dil olmasına da olanak sağlayan en temel özellik, sınıflardır. Sınıflar sayesinde yeni veri türleri yaratılabilir veya varolan türlerden yenileri türetilebilir. Çalışmada oluşturulan yazılımlar Metod başlığı altında incelenmiştir.

3.2. Metod

3.2.1. Deney kartının tasarımı

Tasarladığımız elektronik devrenin ana kısmını mikrodenetleyici oluşturmaktadır. Kullanılan entegre Atmel firmasına ait ATmega serisi mikrodenetleyicidir. Kullanılan deney kartında giriş-çıkış sayılarının çok olması çalışmamızda, denemeler ve kullanım açısından birçok faydayı beraberinde getirmiştir.

Deney kartında bulunan giriş çıkış sayıları ve bunların entegre üzerine bağlı bulundukları portlar aşağıdaki çizelgede verilmiştir.

Çizelge 3.4. Devre üzerindeki giriş-çıkış sayıları ve bağlı bulundukları port adları

Giriş/Çıkış	**Bağlı Bulundukları Port Adları**
8 adet giriş/çıkış	A portu
6 adet giriş/çıkış	D portu
5 adet çıkış	B portu

Deney kartı üzerinde A ile belirtilen kısım; sayısal giriş ve çıkışları ifade etmektedir. 8 adet sayısal giriş/çıkış mevcuttur ve bu giriş/çıkışlar mikrodenetleyicinin A portuna bağlıdır. Resme göre en üstteki çıkış topraklama için şase bağlantısıdır.

B ile gösterilen bölge sayısal giriş ve çıkışları ifade etmektedir. 6 adet sayısal giriş/çıkış mevcuttur ve bu giriş/çıkışlar mikrodenetleyicinin D portuna bağlıdır. A portu ile beraber düşünüldüğünde giriş/çıkış sayısı toplam 14 adettir.

C ile belirtilen kısım; karta beslemenin (12 V DC) verildiği yerdir.

D ile belirtilen kısım; programlama pinleridir. PC'nin paralel portundan haberleşmesini sağlayarak mikrodenetleyici üzerine (FLASH veya EEPROM) program veya veri aktarımı sağlar.

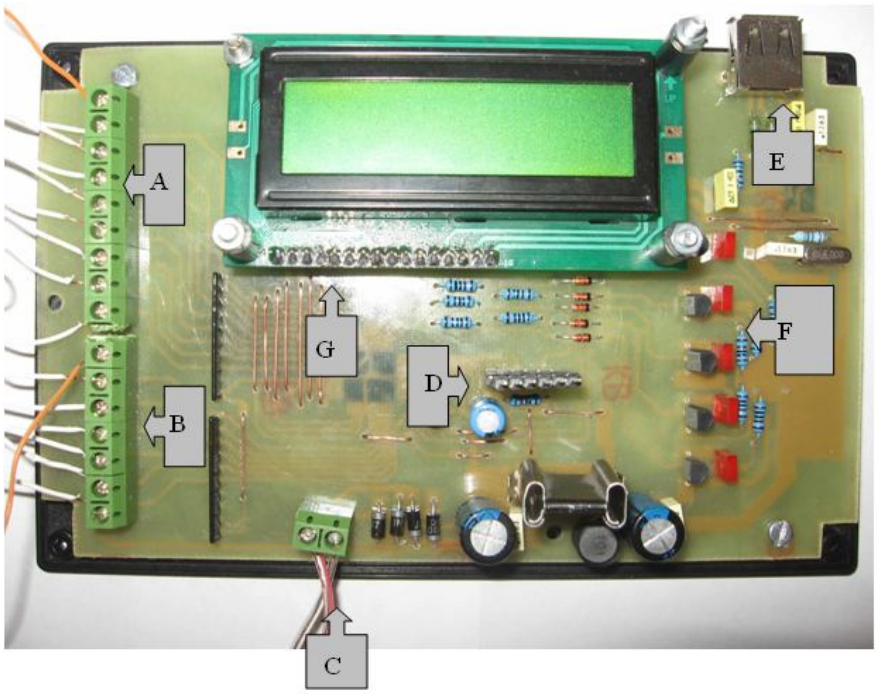
E ile belirtilen kısım; USB portudur. Uygulama esnasında mikrodenetleyici üzerine daha önce yüklenmiş olan yorumlayıcının işleneceği verilerin EEPROM'a

aktarılmasında kullanılması amaçlanmıştır. USB portundan PC tarafındaki terminal ile haberleşme de sağlanabilmektedir.

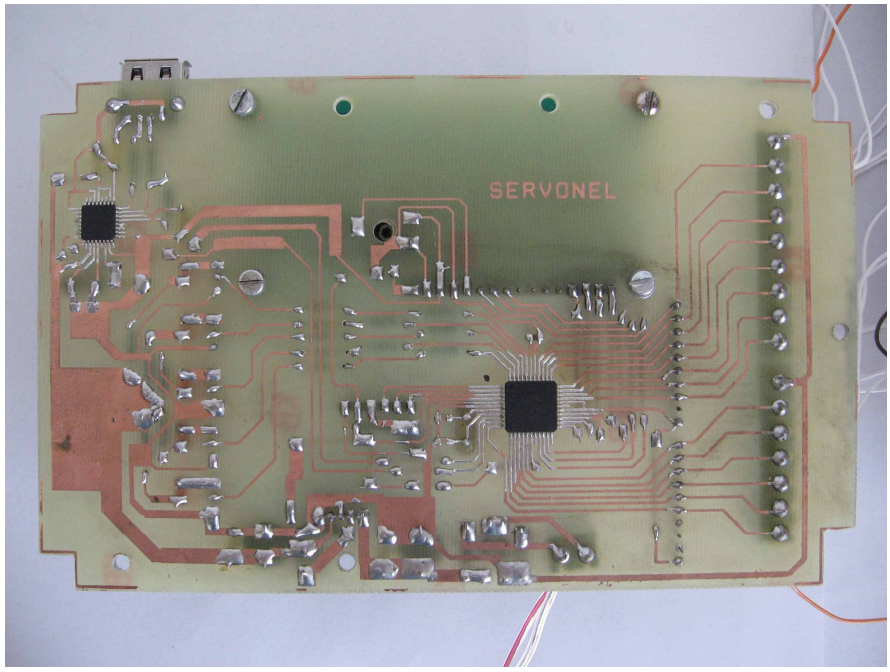
F ile belirtilen kısım; sayısal çıkışları ifade etmektedir. Toplam 5 adet çıkış mevcuttur. Çıkışlar ledlere bağlıdır. Mikrodenetleyici tarafında ise B portunun 0, 1, 2, 3 ve 4. pinlerine bağlıdır.

G ile belirtilen kısım; LCD ekrandır. 2x16'lık bu LCD ekran mikrodenetleyicinin C portuna bağlıdır. Programın çalışması esnasında kullanıcıyı bilgilendirme maksatlı kullanılmıştır.

Devre üzerinde A ve D portları hem giriş hem de çıkış olarak kullanılabilmektedir. C veya B portunda olduğu gibi LCD ve ledlerle portun giriş ya da çıkış olacağı sabitlenmemiştir. Bu konuda programcıya biraz daha hareket alanı kazandırılmıştır.



Şekil 3.8. Deney kartının üstten görünümü



Şekil 3.9. Deney kartının alttan görünümü

Yukarıdaki şekilden de görüldüğü üzere kart üzerine Atmega mikrodenetleyicisi ve USB portu için kullanılan FTDI firmasına ait ft232BM USB RS232 dönüşümünü sağlayan entegre yerleştirilmiştir. Şekilde diğerine göre biraz daha büyük olan entegre Atmega mikrodenetleyicisidir. Devre üzerindeki veri yolları ve entegrelerin bacaklarının ilgili yerlere bağlantıları görülmektedir. LCD ekranı için de deney kartının alt tarafından bir delik açılarak potansiyometre ile karakter parlaklığının ayarlanması sağlanmıştır.

3.2.1.1. Deney kartı yazılımının amacı

Deney kartı için hazırlanan yazılımın amacı, ölçüm işleminin başlamasından itibaren, ölçme işleminin sonuna kadar, encoder'dan gelen darbeleri sayıp, belirli aralıklarla USB-seri portu üzerinden bilgisayarda çalışan arayüze aktarmaktır.

Bunun için karttaki yazılım, arayüzden "başla" komutu gelene kadar "Hazır" (stand-by) durumunda bekler. Bu komutu almadan önce hiçbir komuta cevap vermemektedir. "Başla" komutunu aldığında, mikro denetleyici üzerindeki tüm kesmeler

aktif hale geçer. Dolayısı ile bizim kullandığımız TIMER1 kesme fonksiyonu ve COUNTER0 sayaç fonksiyonu da aktif hale geçmiş olur.

Counter0 sayaç fonksiyonu encoderden gelen darbeleri saymakla görevlidir. Darbe sayıları 1 byte uzunluğundaki işaretsiz bir değişkende tutulur. Değişken uzunluğu, ara yüzle haberleşme zamanını ve haberleşmeden doğacak kayıpları en aza indirme maksadıyla 1 byte uzunluğunda tutulmuştur.

Bu durumun kötü yönü olarak da, değişkende tutulacak sayının büyüklüğünü kısıtlanması sorunu ortaya çıkar. Yani bir byte uzunluğundaki bir değişkende tutulabilecek en büyük sayının 255 olduğunu düşünürsek, bu değişkende tuttuğumuz darbe sayısının bu rakama ulaşmadan ara yüze gönderilmesi ve sayacın tekrar sıfırlanmasını zorunlu kılmaktadır.

Modelde kullandığımız encoderin bir devirde 360 darbe gönderdiğini de göz önünde tutarsak, sorunun çözümü için okuma süresinin optimum bir değerde tutulması gerekliliği ortaya çıkmaktadır. Yapılan deneylerden bu sürenin 10 ms olduğu görülmüştür. Bu sürenin üzerindeki zaman aralıklarında değişken değerinde taşmalar oluşmuş, daha düşük sürelerde ise devir sayısının aşırı büyük hesaplandığı görülmüştür.

Karttaki program “Dur” komutunu alana kadar 10 ms aralıklarla ara yüze saydığı darbe sayılarını göndermeye devam eder. “Dur” komutunu aldığında, tüm kesme fonksiyonları durur, böylelikle sayma işlemi durdurulur. Bu komutun hemen ardından “Reset” komutunu alır ve karttaki program başlangıç durumuna dönerek yeni bir test için “Hazır” konumunda bekler.

Bunların dışında kullandığımız PLC kartında bulunan bazı ledler yakılıp söndürülerek ve “display”de bazı mesajlar gösterilerek çalışan programın hangi pozisyonda olduğu konusunda mesajlar vermektedir, ancak bu işlemlerin test için her handi bir olumlu yada olumsuz katkısı bulunmamaktadır.

C++ dilinde hazırlan kontrol kartında çalışan programın ana dosyası ve buna ait başlık dosyası *Encoder.h, Encoder.c* dosyaları EK 1’de verilmiştir.

Programda kullanılan kesme ve sayıcı fonksiyonlarının oluşturulduğu *EncoderTimer* dosyası EK 2'de verilmiştir.

Kontrol kartı ile ara yüz arasındaki USB seri port üzerinden haberleşmeyi sağlayan *encoderUART* dosyası ve başlığı EK 3'de verilmiştir.

3.2.2. Sistemden istenen değerlerin hesaplanması

Dinamometre modelimizde kullanılan silindirin çapı 0.082 m, kütlesi ise 1.835 kg'dır. Elektrik motoru, silindiri kayış-kasnak mekanizması ile tahrik etmektedir. Testlerin yapıldığı andaki çevrim oranı 1,67'dir. Burada silindirin bağlı olduğu kasnak çapı elektrik motorunun bağlı olduğu kasnağın çapından büyüktür.

Encoder silindir miline bağlı olmasına rağmen atalet momenti çok küçük olması nedeniyle ihmal edilmiştir. Benzer nedenlerle kayış-kasnak sisteminin de kütleleri çok küçük olduğundan atalet momentleri ihmal edilmiştir. Modelimiz Şekil 3.1'de görülmektedir.

Modelimizde kullanılan silindirin atalet momenti (3.1) bağıntısından hesaplanır.

$$I = \frac{1}{2} m r^2 \qquad (3.1)$$

Modelimizin sabit değerlerini (3.1) bağıntısını kullanırsak sistemin atalet momenti;

$$I = \frac{1}{2} x 1.835 x \left(\frac{0,082}{2}\right)^2 = 0,0015\ kgm^2$$

olarak bulunur.

3.2.2.1. Encoder ile haberleşme süresinin belirlenmesi

Kontrol kartı arayüze 10 ms aralıklarla USB-Seri port üzerinden 1 byte uzunluğunda veri göndermektedir. Bu da 8 bit data, bir bit de stop biti olmak üzere 9 bitlik veriler gönderecek demektir. Haberleşme kalanı için seçtiğimiz haberleşme hızı 9600 bps'dir. Yani saniyede 9600 bit iletilebilecektir. Bu duruda bir bit için gereken süre;

$$\frac{1}{9600} = 1{,}0416.10^{-4}\ s$$

olacaktır.

Çalışmamızda 1 byte 9 bitten oluştuğundan, bu bitlerin iletimi için;

$$1{,}0416.10^{-4}x9 = 0{,}0009375\ s$$

Kadarlık bir süre gerekmektedir. Bir bu kadarlık süreyi'de bir sonraki byte için bekleme süresi olarak kullanırsak;

$$0{,}0009375x2 = 0{,}001875\ s$$

süre gerekmektedir. Bu süre de 10 ms'den küçük olduğundan 9600 bps'lik iletişim hızı yeterli olmaktadır.

Veri kaybına sebep olmamak için daha yüksek hızlarda iletişimden kaçınılmıştır.

3.2.2.2. Sistem çıktıları için gerekli değerlerin hesaplanması

Şasi dinamometreleri güç ve tork ölçümlerinde, silindirin birim zamandaki hızından faydalanırlar. Bu devir sayısını da silindirin dönüşüyle encoderin ürettiği palselerin kontrol kartı aracılığıyla sayılmasıyla bulmak mümkündür.

Modelimizde kullandığımız encoder 1 devirde 360 darbe üretmektedir. buna göre, ölçümün başlangıç anına t_0 dersek bu durumda tüm değerler gelen darbe sayısı 0 olduğundan sıfırdır. Bir sonraki okuma anı

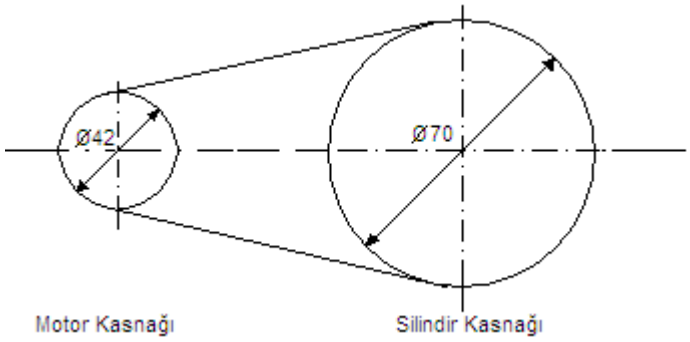
$$t_1 = \frac{10}{1000} = 0{,}01\ s$$

sonra gerçekleşecektir. Bu durumda silindirin devir sayısı d/dk olarak (3.2) bağıntısıyla belirlenebilir.

$$n_{sil} = \frac{n}{0{,}01} x \frac{60}{360} = \frac{100n}{6} \tag{3.2}$$

olarak bulunur. Modelimizde silindir, motor tarafından kayış-kasnak sistemi aracılığıyla tahrik edilmektedir.

Şekil 3.10'daki kayış kasnak sistemindeki çevrim oranı (3.4) eşitliğiyle bulunabilir.



Şekil 3.10. Modelde kullanılan kayış-kasnak sistemi

$$n_{sil} \times d_{sil} = n_{mot} \times d_{mot} \tag{3.3}$$

eşitliğinden,

$$\frac{n_{mot}}{n_{sil}} = \frac{d_{sil}}{d_{mot}} = i \tag{3.4}$$

Burada modelimizdeki kasnak çapları yerine konursa çevrim oranı

$$i = \frac{0{,}07}{0{,}042} = 1{,}67$$

gibi olur.

Bu çevrim oranı yardımıyla motorun devir sayısı;

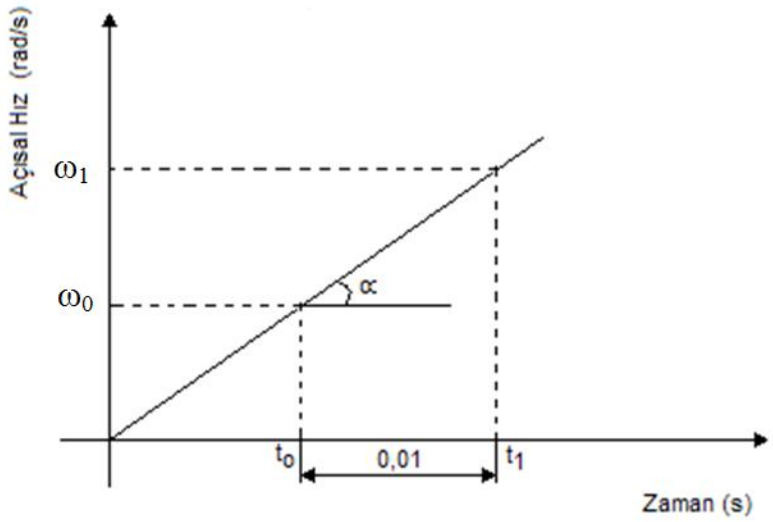
$$n_{mot} = i \times n_{sil} = 1.67 \times n_{sil} \tag{3.5}$$

olarak bulunur.

Devir sayıları belirlendikten sonra artık motorun açısal hızı bulunabilir. (3.6) ve (3.7) eşitlikleri silindirin ve motorun açısal hızlarını göstermektedir.

$$\omega_{sil} = \frac{\pi \times n_{sil}}{30} \tag{3.6}$$

$$\omega_{mot} = \frac{\pi \times n_{mot}}{30} \tag{3.7}$$



Şekil 3.11. Açısal Hız-Zaman grafiği

Şekil 3.11'den de fark edileceği gibi, motora ait açısal hızlar belirlendikten sonra motorun ve silindirin açısal ivmeleri de rahatlıkla belirlenebilir. (3.8) ve (3.9) eşitlikleri silindire ve motora ait ivmelerin belirlenmesinde kullanılmaktadır.

Kontrol kartı darbeleri sabit zaman aralıklarıyla arayüze gönderdiğinden iki okuma arasındaki süre sabit olarak yerine konmuştur.

$$\alpha_{sil} = \frac{\omega_{sil1} - \omega_{sil0}}{t_1 - t_0} \tag{3.8}$$

$$\alpha_{mot} = \frac{\omega_{mot1} - \omega_{mot0}}{t_1 - t_0} \tag{3.9}$$

Sabit okuma süresi yerine konursa, açısal hızlar;

$$\alpha_{sil} = 100 \times (\omega_{sil1} - \omega_{sil0})$$

ve

$$\alpha_{mot} = 100 \times (\omega_{mot1} - \omega_{mot0})$$

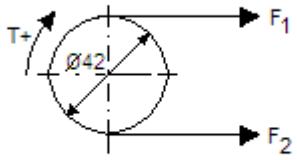
olarak bulunur.

Silindirin anlık ivmesi belirlendiğine göre torku da bulabilmek mümkündür. Bu tork kütle atalet momenti ve silindirin kazandığı ivmeyle orantılı olduğundan (3.10) eşitliğiyle belirlenebilir.

$$T_{sil} = I_{sil} \times \alpha_{sil} \tag{3.10}$$

Burada I_{sil} silindirin kütle atalet momenti olup, (3.1) eşitliğinde verilmiştir.

Silindirin kazandığı bu tork aynı zamanda kayış-kasnak sistemide, kayışın uçlarındaki kuvvetler farkının oluşturduğu torka da eşit olacaktır (Şekil 3.12).



Şekil 3.12. Kayış-kasnak sisteminde kuvvetler ve pozitif tork yönü

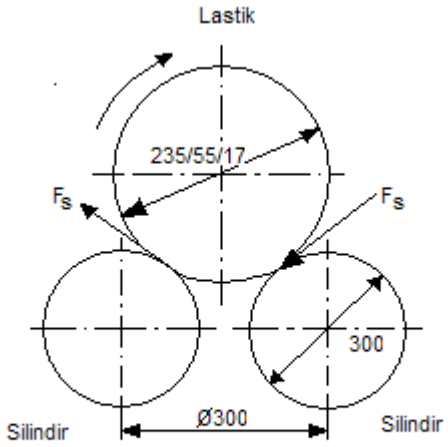
Şekil 3.13'den de görüleceği gibi Silindirin kazandığı tork, kayışlardaki kuvvetlerin farklarının oluşturduğu torka eşit olacaktır. Bu nedenle (3.11) eşitliği yazılabilir.

$$T_{sil} = I_{sil} \times \alpha_{sil} = (F_1 - F_2) \times \frac{D_{sil}}{2} \tag{3.11}$$

Kuvvetlerin farkına F dersek, bu kuvvet (3.12)'deki gibi bulunabilir.

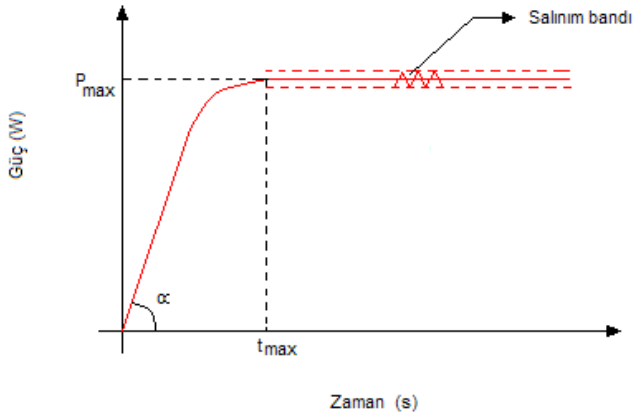
$$F = \frac{2 \times T_{sil}}{D_{sil}} \tag{3.12}$$

Bu kuvvet kayış-kasnak sistemi nedeniyle motor kasnağına da etkidiğinden motorun uyguladığı momentin tespitinde de kullanılabilir. Dinamometrede bu kuvvet Şekil 3.13'deki gibi silindir lastik arasındaki sürtünme kuvveti olarak da anılabilir. Bu durumda motora ait tork (3.13)'deki eşitlikle bulunabilir.



Şekil 3.13. Dinamometrede lastiğin silindirlere uyguladığı kuvvetler

$$T_{mot} = \frac{F \times D_{mot}}{2} \tag{3.13}$$



Şekil 3.14. Motor gücünün zamana bağlı olarak değişimi

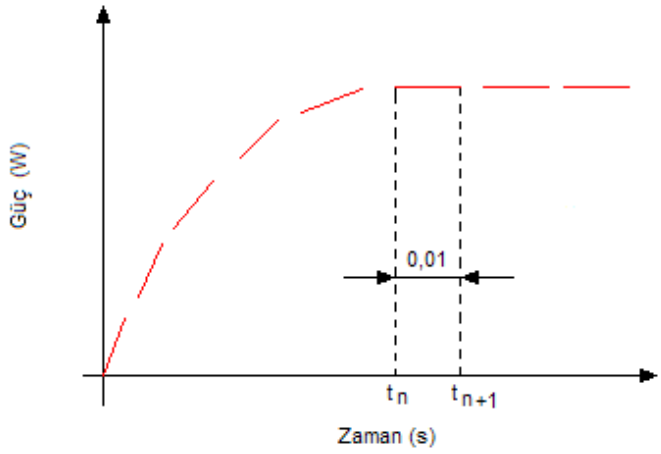
Şekil 3.14'de görüldüğü gibi motor maximum devir sayısına ulaşana kadar motor karakteristiğine bağlı olarak yaklaşık lineer olarak güç artışı gösterecek, maximum güce ulaşdığında da belirli bir band genişliğinde salınım gösterecektir. Çünkü encoderden

okunan darbe sayısında küçük değişimler olacaktır. (3.2) eşitliğine bağlı olarak da bu palse sayısından devir sayısı elde edilirken sabit katsayılar nedeniyle devir sayısında değişimler meydana gelecektir. Bu durumda söz konusu salınımı oluşturacaktır.

Bu noktada silindir ve motor güçleri eşit olacağından silindir gücünün hesaplanmasına gerek duyulmamıştır. Motor gücü (3.14) eşitliğinde olduğu gibi hesaplanabilir.

$$P_{mot} = T_{mot} \times \frac{\omega_{mot1} + \omega_{mot0}}{2} \tag{3.14}$$

Şekil 3.16'daki grafik 10 ms aralıklarla elde edilmiş Güç-Zaman grafiğidir. Bu nedenle motor gücü hesaplanırken kontrol kartı okuma aralığının ortalaması olarak alınmıştır. Şekil 3.15'de bu lineer grafikler görülmektedir.

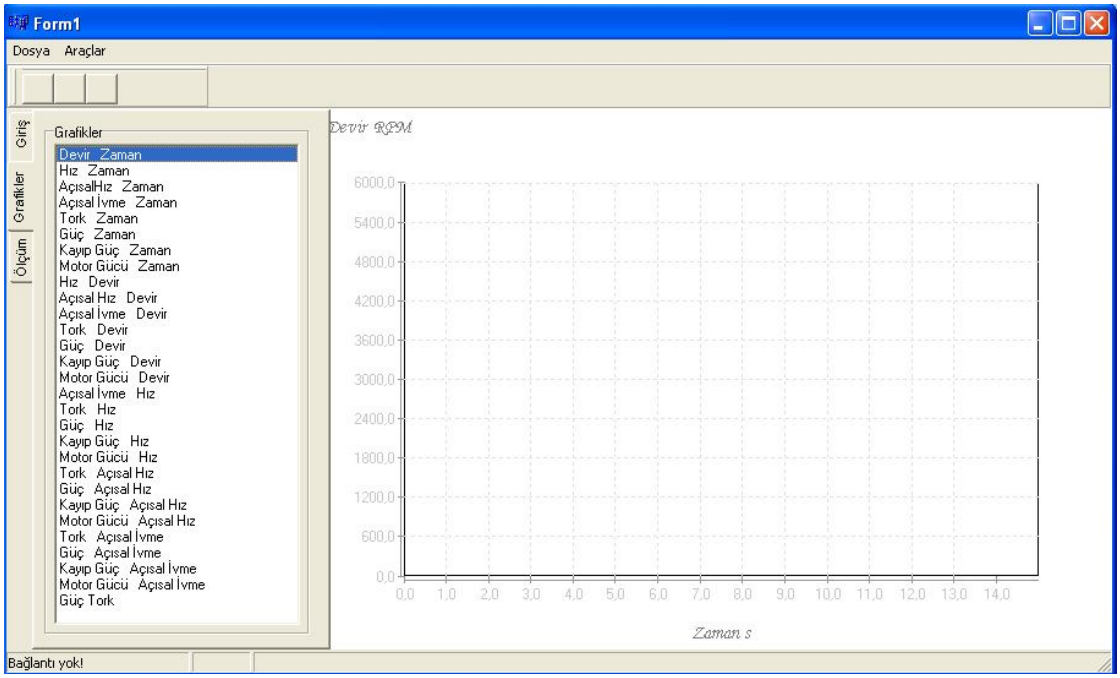


Şekil 3.15. Lineer grafiklerin birleşiminden oluşan Güç-Zaman grafiği

3.2.3. Arayüz programı

Mikrodenetleyici içerisine yazılan program vasıtası ile deney düzeneğinden gelen değerler yorumlanmaktadır. Ancak bu veriler kullanıcı açısından hiçbir sonuç içermez. Bu verilerin kullanıcılar tarafından anlaşılır hale gelip yorumlanabilmeleri ve sistemi denetlemek için sayısal verilere dönüştürülmesi ihtiyacı vardır. Bunun için bir ara yüz programı geliştirilerek istenilen şartlar sağlanmıştır.

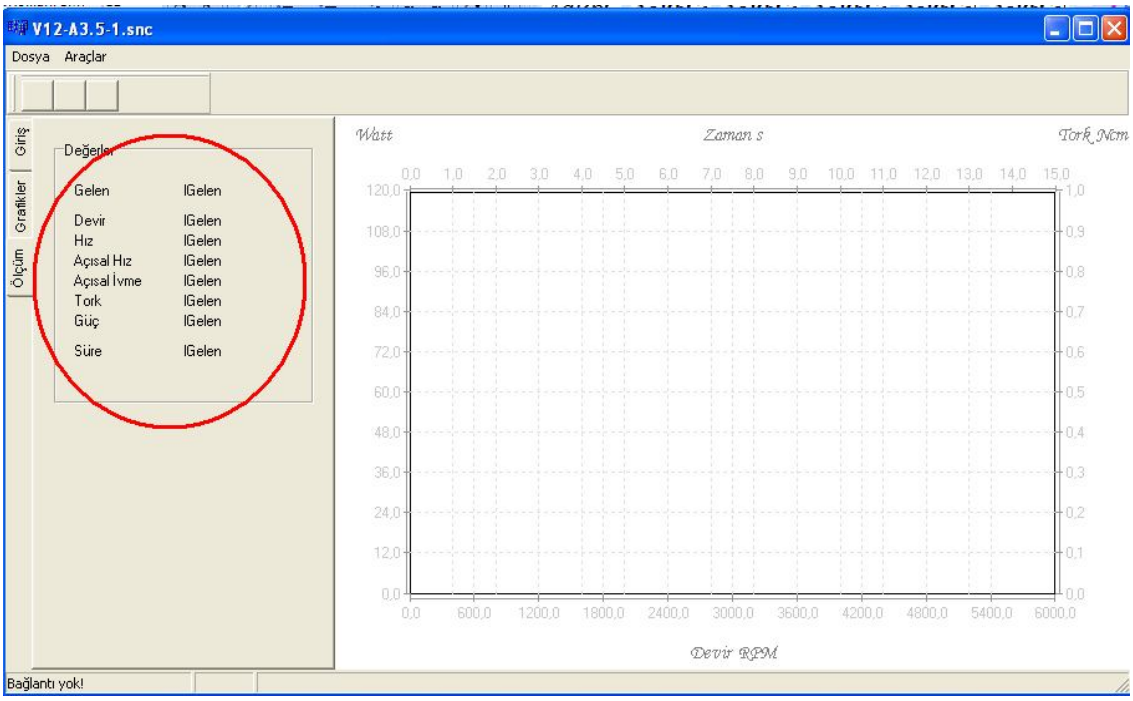
Hazırlanan ara yüz programı ile hesaplanan devir, hız, açısal hız, açısal ivme, tork, güç ve daha birçok değere ait grafikler kullanıcıya sunulmaktadır. Değerlerin grafikler halinde sunulması kullanıcının değerleri hızlı bir şekilde yorumlayabilmeleri için önemlidir.



Şekil 3.16. Geliştirilen kullanıcı ara yüz yazılımı ana penceresi

Kullanıcı ara yüz yazılımı, C++'da geliştirilmiştir. Bu yazılım vasıtası ile deney motoruna ait bilgiler deney kartı kullanılarak sistemden bilgisayara aktarılmaktadır. Bu

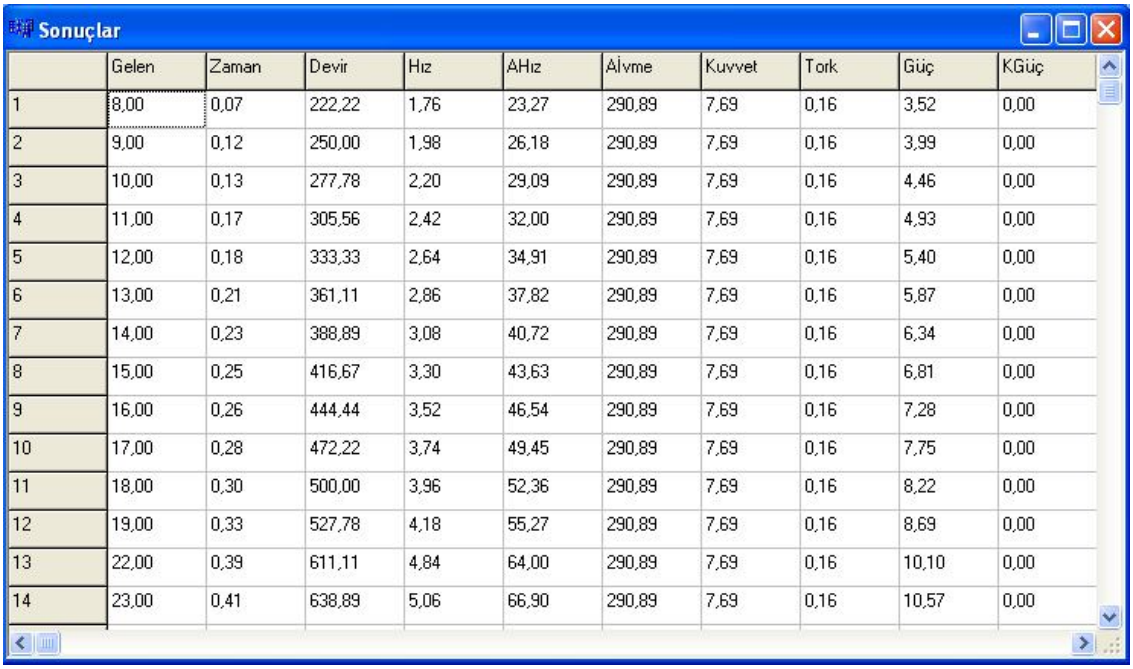
yazılımın ana penceresi Şekil 3.16'da verilmiştir. Ara yüz programından elde edilebilecek grafikler Şekil 3.16'da görülen ekran penceresinin sol kısmında "Grafikler" başlığı altında gösterilmiştir. Menü penceresinin sol kısmından "Giriş" butonu ile araca ait lastik tipi değerleri ve çevre şartları değerleri girilebilir.



Şekil 3.17. Ara yüze gelen değerleri okuma penceresi

Araçlar menüsünde bulunan Başla komutu ile önce tasarlanan deney kartına başla komutu gönderilir. Kart komutu aldığı andan itibaren 10 ms sayma işlemi gerçekleştirir. Bu süre sonunda ilk okunan toplam darbe gönderilir. Bu arada arayüz programı kartın ilk okumaları tamamlayabilmesi için 10 ms bekletilir. Gelen darbe sayısı encoderden alındıktan sonra sistem bunu dakika/devir sayısına çevirir ve bu değer silindirin devir sayısı değeri olur. Silindir ve elektrik motoru arasındaki çevirim oranından faydalanılarak motorun devir sayısı bulunur.

Ölçüm işleminde deney motorundan gelen değerlerde anlık olarak gözlenebilmektedir (Şekil 3.17). Yapılan ölçümlerin kaydedilmesi istenirse "Dosya" menüsünde bulunan "Kaydet" düğmesine basılmalıdır. Excel dosyasına kaydedilen bu ölçüm sonuçları görülmek istenirse, "Araçlar" menüsündeki "Sonuçlar" düğmesine basılarak veriler kontrol edilebilir (Şekil 3.18).



	Gelen	Zaman	Devir	Hız	AHız	Alvme	Kuvvet	Tork	Güç	KGüç
1	8,00	0,07	222,22	1,76	23,27	290,89	7,69	0,16	3,52	0,00
2	9,00	0,12	250,00	1,98	26,18	290,89	7,69	0,16	3,99	0,00
3	10,00	0,13	277,78	2,20	29,09	290,89	7,69	0,16	4,46	0,00
4	11,00	0,17	305,56	2,42	32,00	290,89	7,69	0,16	4,93	0,00
5	12,00	0,18	333,33	2,64	34,91	290,89	7,69	0,16	5,40	0,00
6	13,00	0,21	361,11	2,86	37,82	290,89	7,69	0,16	5,87	0,00
7	14,00	0,23	388,89	3,08	40,72	290,89	7,69	0,16	6,34	0,00
8	15,00	0,25	416,67	3,30	43,63	290,89	7,69	0,16	6,81	0,00
9	16,00	0,26	444,44	3,52	46,54	290,89	7,69	0,16	7,28	0,00
10	17,00	0,28	472,22	3,74	49,45	290,89	7,69	0,16	7,75	0,00
11	18,00	0,30	500,00	3,96	52,36	290,89	7,69	0,16	8,22	0,00
12	19,00	0,33	527,78	4,18	55,27	290,89	7,69	0,16	8,69	0,00
13	22,00	0,39	611,11	4,84	64,00	290,89	7,69	0,16	10,10	0,00
14	23,00	0,41	638,89	5,06	66,90	290,89	7,69	0,16	10,57	0,00

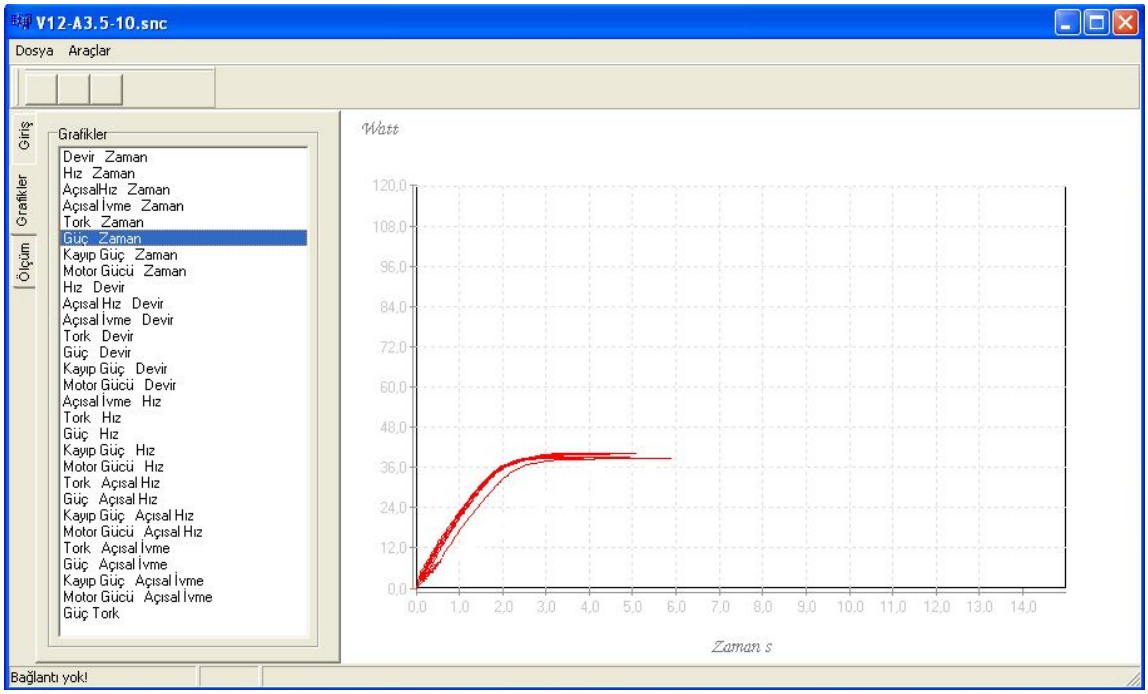
Şekil 3.18. Kaydedilen verilerin görüntülendiği pencere

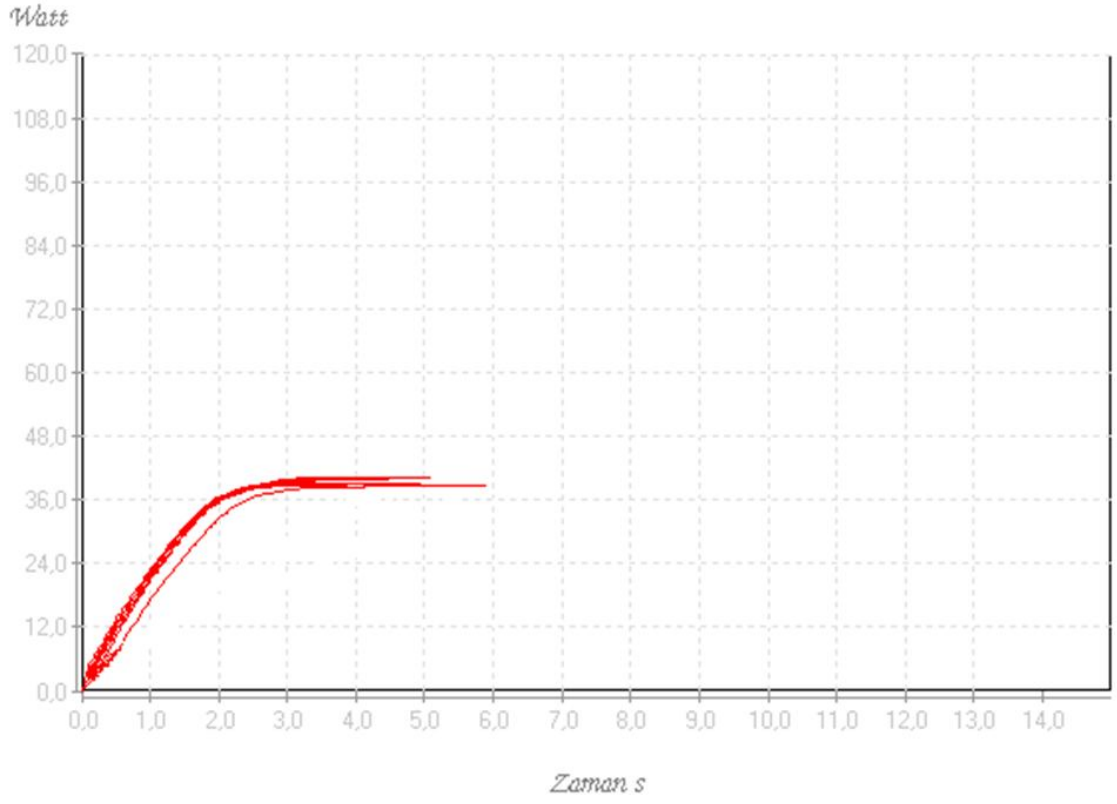
4. ARAŞTIRMA UYGULAMALARI VE SONUÇLARI

Bu bölümde oluşturulan model üzerinde yapılan deneylere yer verilmektedir. İlk olarak deney motoru 12 V, 3,5 A'de çalıştırılmış ve on tane farklı test yapılmıştır. 12 V, 3,5 A'de teorik olarak motorun çektiği güç 42 W'tır. Test sonuçlarına göre ise yapılan on testin ortalaması 39 W olarak bulunmuştur. Buna göre yaklaşık %7'lik bir kayıp söz konusudur. Bu kayıp değerin motor verimi ve sistemin toplam kayıpları olduğu düşünülmektedir. Elde edilen on farklı grafiğin birleştirilmiş hali Şekil 4.1'de görülmektedir.

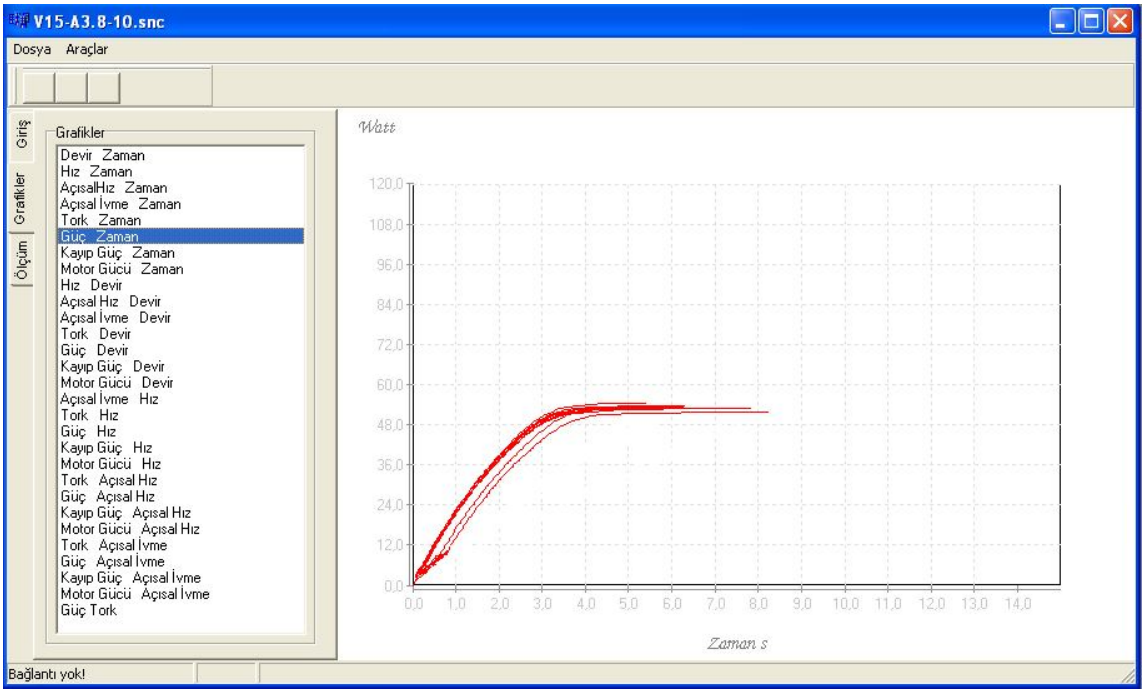
Benzer şekilde deney motoru 15 V, 3,8 A'de çalıştırılmış ve yine on tane farklı test yapılmıştır. 15 V, 3,8 A'de teorik olarak motorun çektiği güç 57 W'tır. Test sonuçlarına göre ise yapılan on testin ortalaması 52,8 W olarak bulunmuştur. Buna göre yine yaklaşık %7'lik bir kayıp söz konusudur. Bu kayıp değerin motor verimi ve sistemin toplam kayıpları olduğu düşünülmektedir. Elde edilen on farklı grafiğin birleştirilmiş hali Şekil 4.2'de görülmektedir.

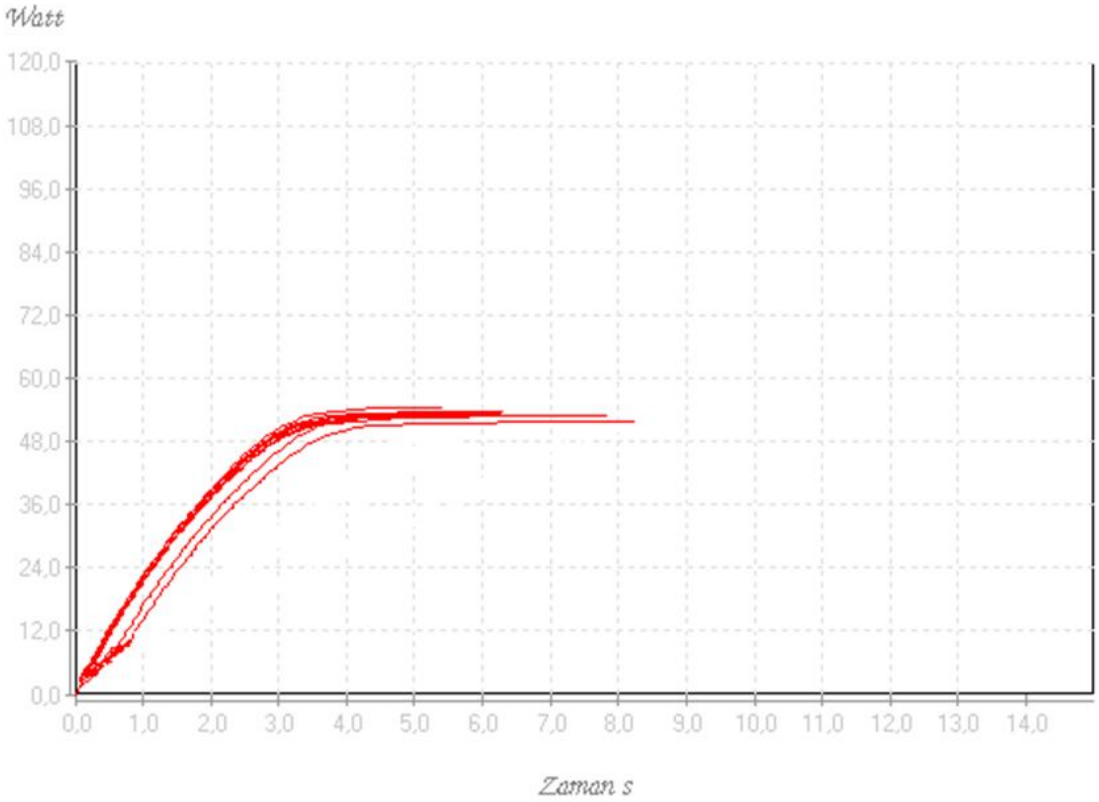
Üçüncü olarak deney motoru 20 V, 4 A'de çalıştırılmış ve on tane farklı test yapılmıştır. 20 V, 4 A'de teorik olarak motorun çektiği güç 80 W'tır. Test sonuçlarına göre ise yapılan on testin ortalaması yaklaşık 77 W olarak bulunmuştur. Buna göre yaklaşık % 4'lük bir kayıp söz konusudur. Bu kayıp değerin motor verimi ve sistemin toplam kayıpları olduğu düşünülmektedir. Elde edilen on farklı grafiğin birleştirilmiş hali Şekil 4.3'de görülmektedir.



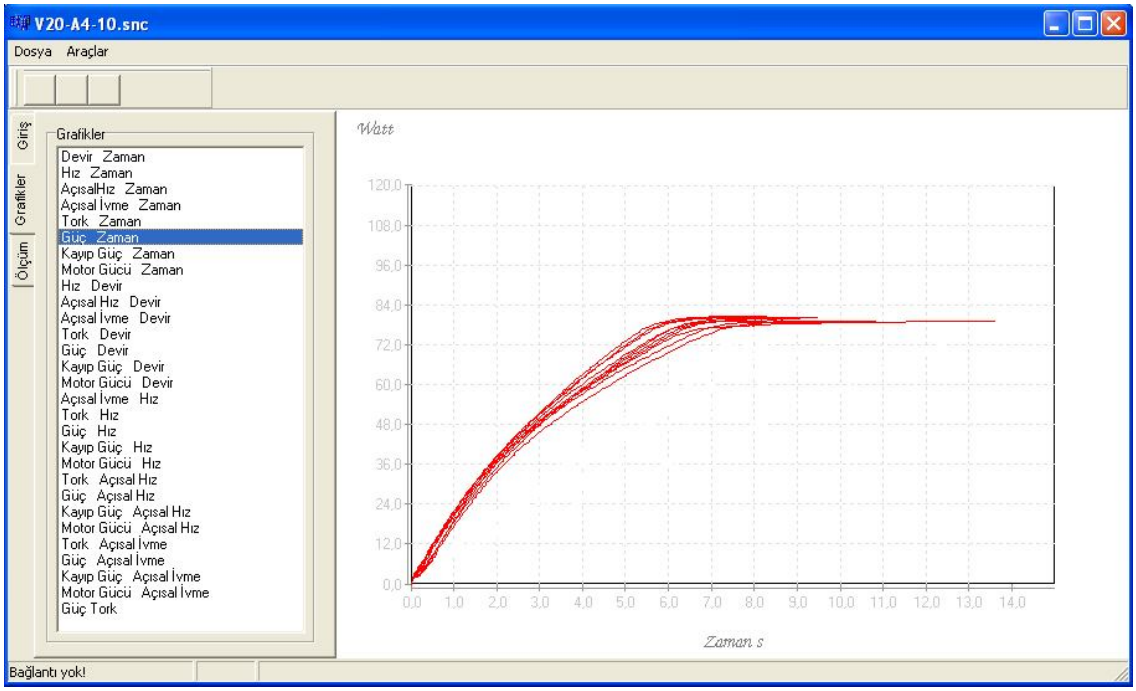


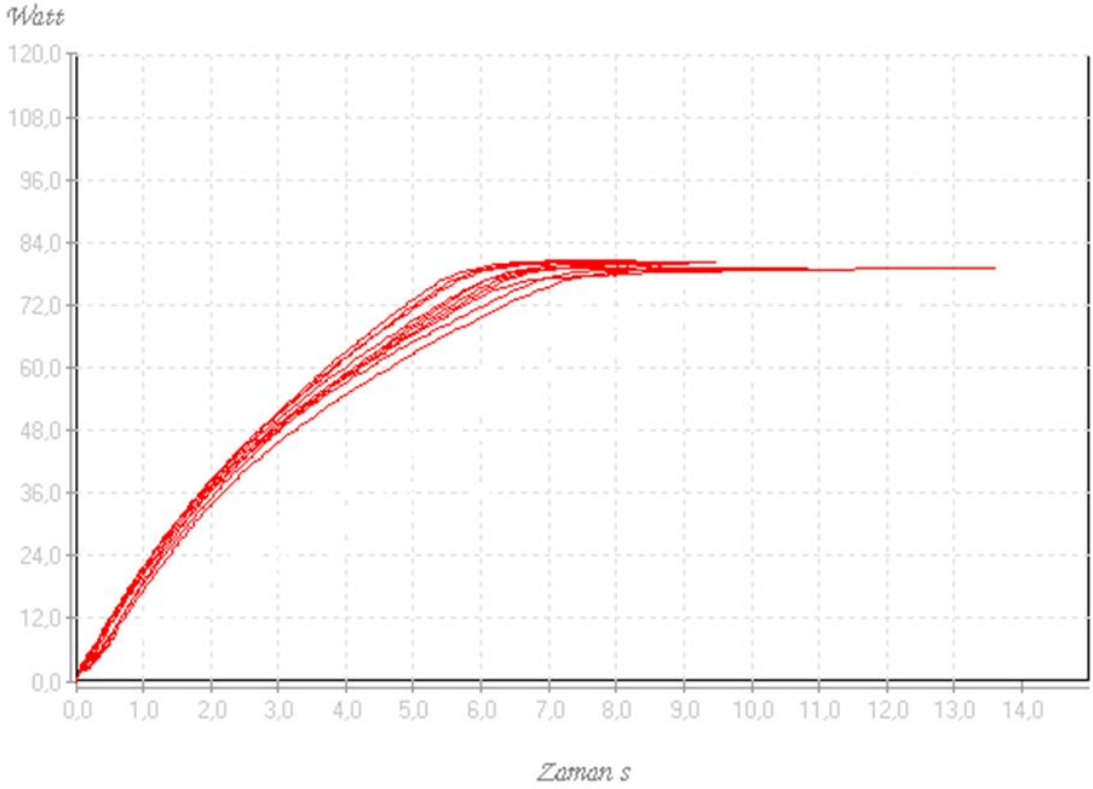
Şekil 4.1. 12 V - 3.5 A'de 10 defa yapılan testlerin sonucunda elde edilen Güç-Zaman grafikleri





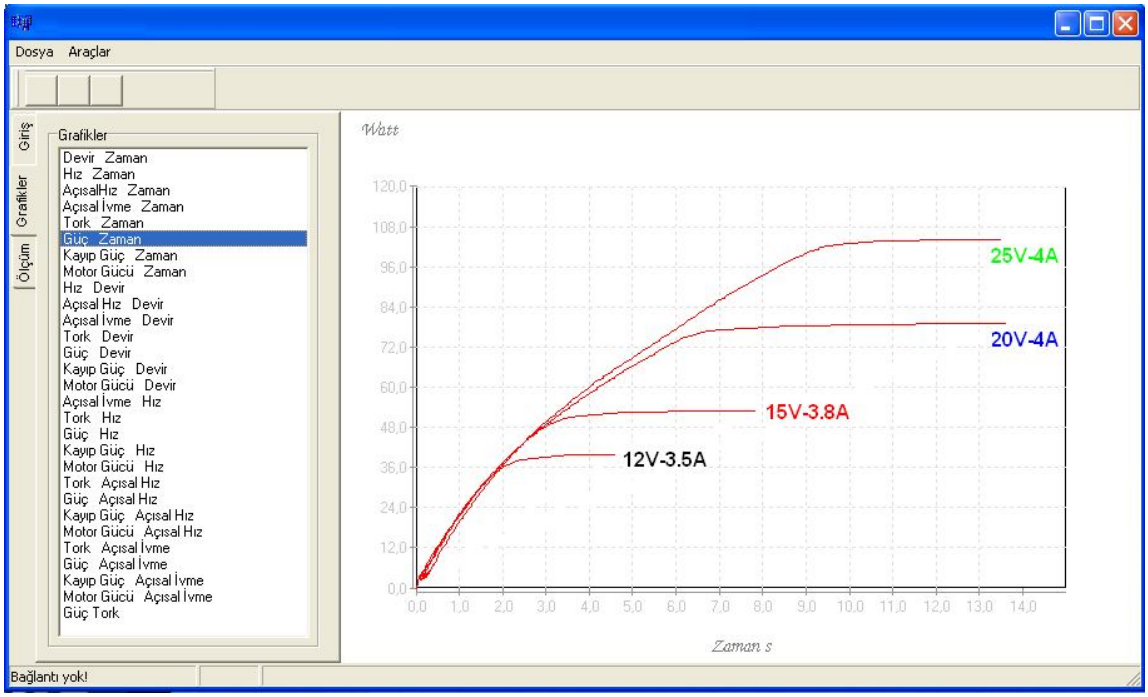
Şekil 4.2. 15 V - 3.8 A'de 10 defa yapılan testlerin sonucunda elde edilen Güç-Zaman grafikleri





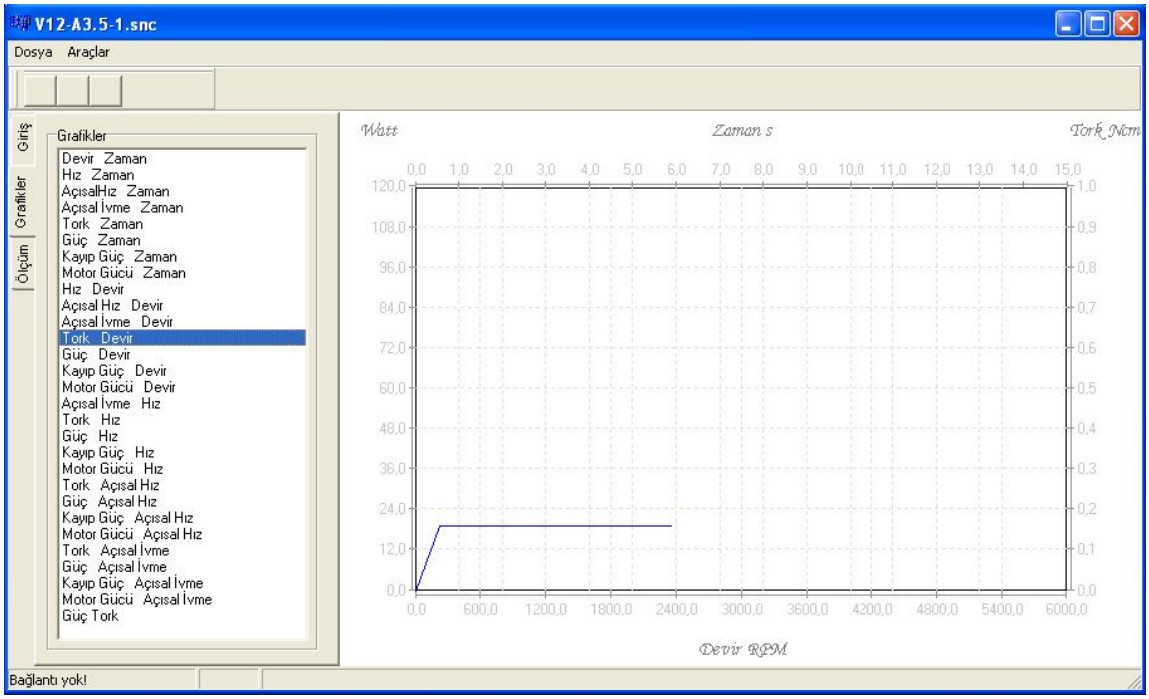
Şekil 4.3. 20 V - 4A'de 10 defa yapılan testlerin sonucunda elde edilen Güç-Zaman grafikleri

Farklı gerilimlerde (12 V - 3.5 A, 15 V - 3.8 A, 20 V - 4 A, 25 V - 4 A) yapılan testlere ait güç-zaman grafiklerinin birleştirilmiş hali Şekil 4.4'de verilmiştir. Şekle göre motora uygulanan güç arttırıldıkça motorun maximum güce ulaşma süresi artmakta ancak motor karakteristiği bütün testlerde aynı kalmaktadır. Bu durum Şekil 4.4'de de açıkça görülmektedir.

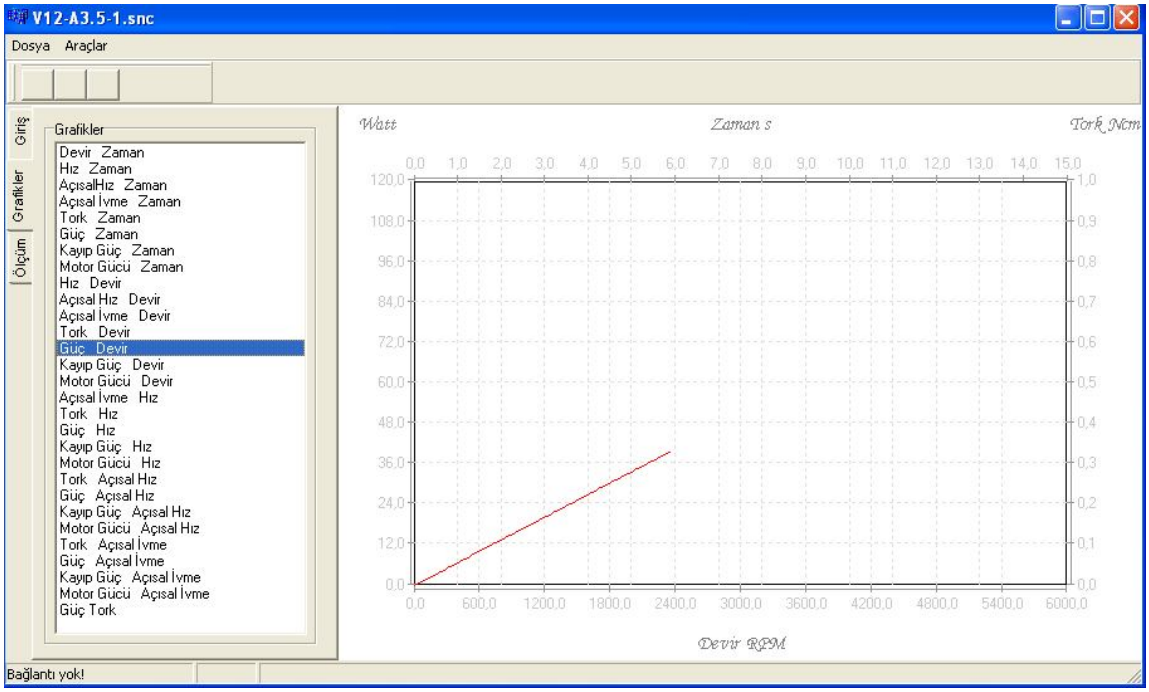


Şekil 4.4. Farklı güçlerde elde edilen birleştirilmiş Güç-Zaman grafikleri

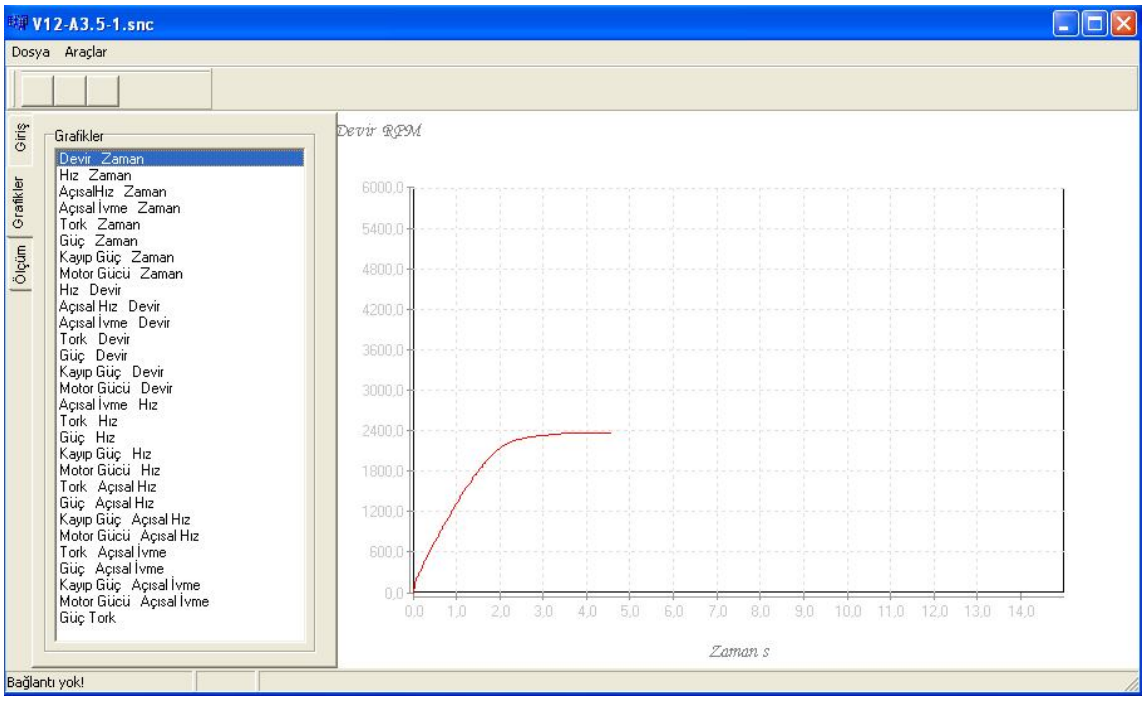
Yapılan testlerden elde edilen sonuçlardan çıkartılabilecek grafiklerden bazıları aşağıda verilmektedir.



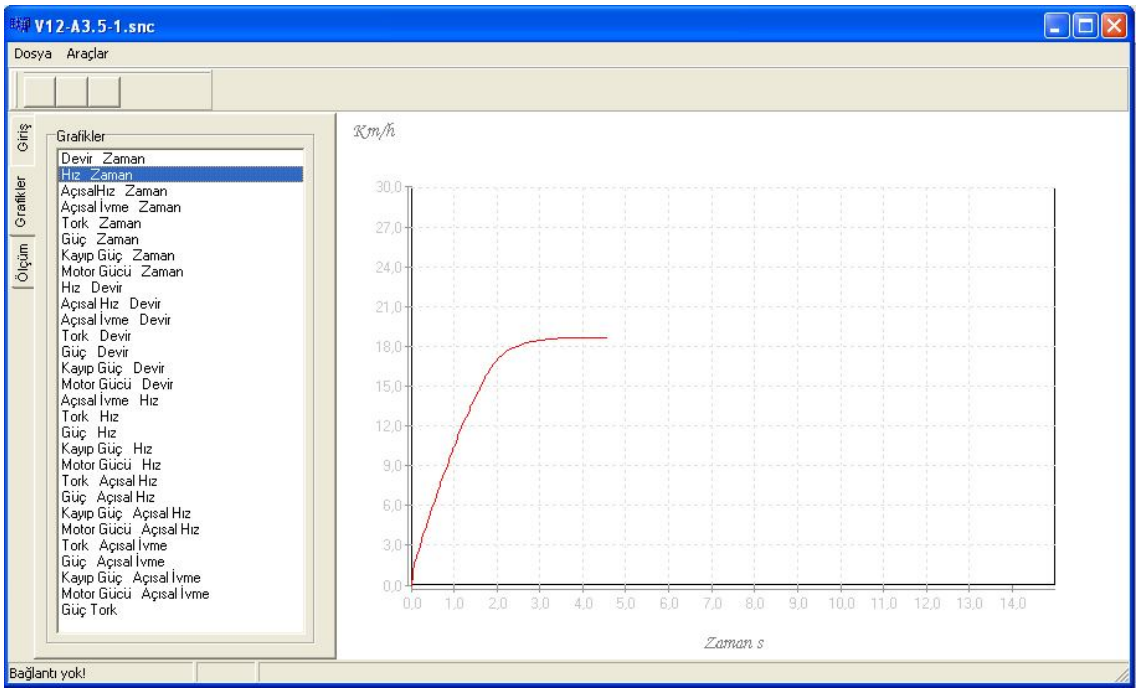
Şekil 4.5. Tork-Devir grafiği



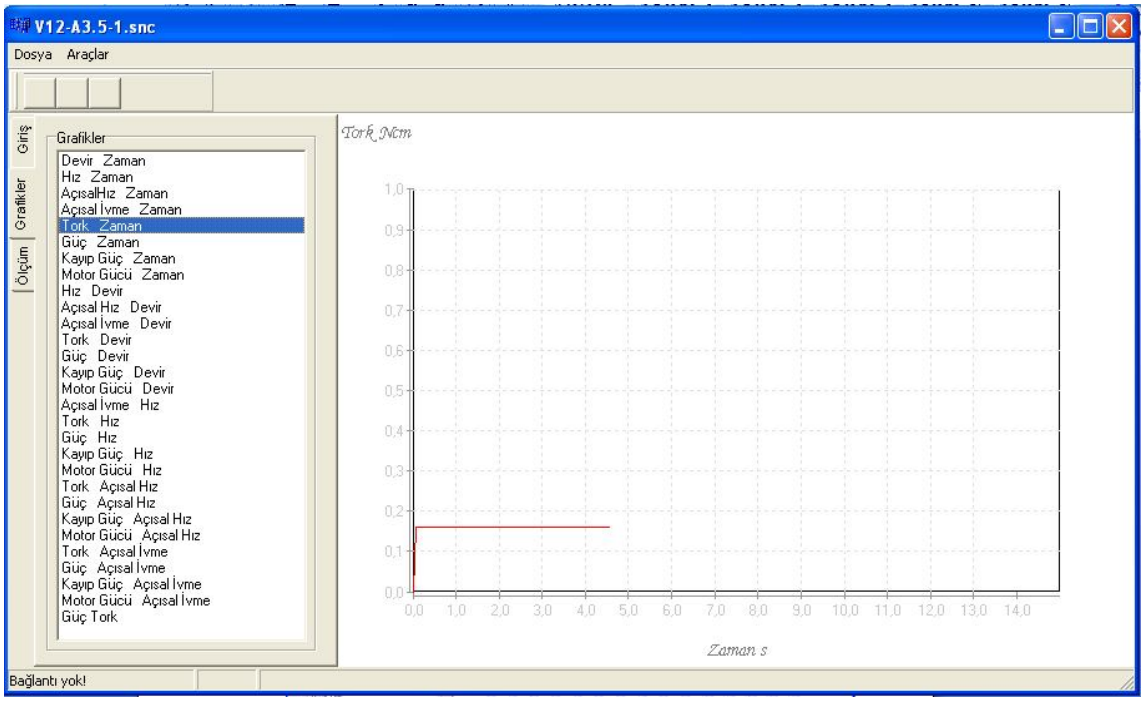
Şekil 4.6. Güç-Devir grafiği



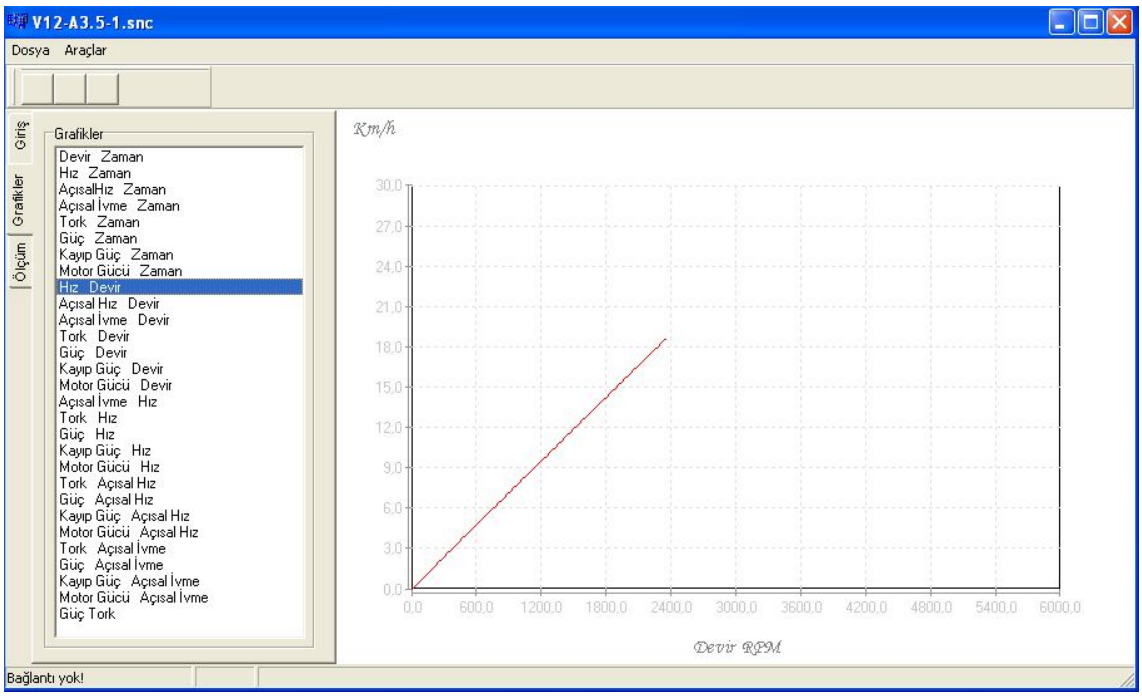
Şekil 4.7. Devir-Zaman grafiği



Şekil 4.8. Hız-Zaman grafiği



Şekil 4.9. Tork-Zaman grafiği



Şekil 4.10. Hız-Devir grafiği

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, motorlu taşıt testlerinde araç performansının hesaplanabilmesi için kullanılan şasi dinamometresinin hazırlanan bir bilgisayar yazılımı ve bu yazılımın üzerinde çalıştığı elektronik kart ile otomasyonu amaçlanmıştır. Bu sistemde test motorundan elde edilen elektronik sinyaller ve bu sinyallerden elde edilen veriler bilgisayar ortamına aktarılarak hesaplamalar sağlanmıştır. Gelen sinyallerin verilere dönüştürülmesinde ve bilgisayara aktarılmasında tarafımızdan tasarlanan bir yazılım ve bu yazılımın üzerinde çalıştığı elektronik kart kullanılmıştır. Bu sistemle motorlu araç lastiklerinden aktarılan güç değerlerinin belirlenmesi ve araca ait performans değerlerinin hesaplaması yapılmış ve bu değerler ara yüz programı ile kolay anlaşılır grafikler haline getirilerek hem anlaşılması hem de yorumlanmaları kolaylaştırılmıştır.

Oluşturulan test düzeneği piyasada mevcut ünitelere göre kıyaslanamayacak kadar daha ucuza mal edilebilmektedir ve geliştirilmeye de oldukça müsaittir. Yapılan çalışmanın, şasi dinamometrelerinin ve yazılımlarının yerli üretim çalışmalarına katkı sağlayacağına inancımız tamdır.

KAYNAKLAR

Anthony, A., Alexander, M. (2007). Engine testing: theory and practice, Butterworth-Heinemann, 442 sayfa.

Batmaz, İ. (1998). Bilgisayar ve veri toplama sisteminin taşıt performans ölçümlerinde kullanılması, Gazi Üniv. Müh. Mim. Fak. Der., Cilt 13, No 2, 171-179.

Bolat, A. (2007). Orta Segment Bir Tarım Traktöründe Biyodizelin Motor Performansı Üzerine Etkileri Ve Biyodizelin Türkiye İçin Önemi, Yüksek Lisans Tezi, Trakya Üniversitesi.

Çelik, M. B., Balcı, M. (2002). Sabit yük ve hız şartlarında sıkıştırma oranının motor karakteristiklerine etkisi, Teknoloji / Z.K.Ü. Karabük Teknik Eğitim Fakültesi Dergisi, 5(3-4):39-45.

Çelik, M. B., Bayır, R., Özdalyan, B. (2007). Bilgisayar Destekli Motor Test Standının Tasarımı ve İmalatı, Teknoloji, Cilt 10, Sayı 2, Sayfa:131-141.

French, M., Stark, A. (2000). Chassis Dynamometers, Experimental Techniques, sayfa:45-46.

MEGEP (2007). Motorlu Araçlar Teknolojisi Alanı, MEB, Ankara.

Pelkmans, L., Debal, D. (2006). Comparison of on-road emissions with emissions measured on chassis dynamometer test cycles, Transportation Research Part D: Transport and Environment, Volume 11, Issue 4, sayfa 233-241.

Petchers, N. (2003). Combined Heating, Cooling, and Power Handbook, Fairmont Pres, Inc.

Ribbens, W. B.,Mansour, N. P. (2003). Understanding automotive electronics, Newnes, 470 sayfa.

Tongur, V. (2008). Atmega128 Tabanlı PLC Tasarımı, Y. Lisans tezi, Selçuk Üniversitesi.

Won, J. S., Langari, R. (2002). Fuzzy torque distribution control for a parallel hybrid vehicle, Expert Systems, Volume 19 Issue 1, Sayfa 4-10.

URL1: http://www.bilgiustam.com/ Erişim:07.07.2010.

URL2: http://www.otomasyon.gen.tr/ Erişim:11.07.2010.

URL3: http://www.atmel.com Erişim:18.05.2010.

EK 1. Kontrol kartı ana programı ve buna ait başlık dosyası

```
//Encoder.h : header file for the Encoder project
//

#ifndef _ENCODER_H_
#define _ENCODER_H_

#include <io.h>
#include <wdt.h>
#include <sleep.h>
#include <Interrupt.h>
#include <inttypes.h>
#include <stdbool.h>

//-----------------------------------------------------------------------------
//Encoder

#define LOW(i)                  (i & 0xff)
#define HIGH(i)                         ((i >> 8) & 0xff)
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
unsigned char Sayi;
#endif // _ENCODER_H_
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
//-----------------------------------------------------------------------------
//Encoder.c : source file for the Encoder project
//
#include "Encoder.h"
#include "EncoderUART.h"
#include "EncoderTimer.h"
#include <lcd.h>
#include <wdt.h>
#include <delay.h>
////////////////////////////////////////////////////////////////////////////
//Encoder
static void io_init(void)
{
  //{{WIZARD_MAP(General)
  //}}WIZARD_MAP(General)

  //{{WIZARD_MAP(I/O Ports)
  // PortA
  PORTA = 0x00;
```

```
  DDRA = 0x00;
  // PortB
  PORTB = 0x01;
  DDRB = 0x00;
  // PortC
  PORTC = 0x00;
  DDRC = 0x00;
  // PortD
  PORTD = 0x00;
  DDRD = 0x00;
  //}}WIZARD_MAP(I/O Ports)

  //{{{WIZARD_MAP(Watchdog)
  // Watchdog Enabled, Prescaler: OSC/16k
  wdt_enable(WDTO_15MS);
  //}}WIZARD_MAP(Watchdog)

  //{{{WIZARD_MAP(Analog Comparator)
  // Analog Comparator Disabled
  ACSR = (1<<ACD);
  //}}WIZARD_MAP(Analog Comparator)
}

static void reset_checking(void)
{
  if (MCUCSR & (1 << PORF))
  {       // Power-on Reset

  }
  else if (MCUCSR & (1 << EXTRF))
  {       // External Reset

  }
  else if (MCUCSR & (1 << WDRF))
  {       // Watchdog Reset

  }
  MCUCSR &= 0xE0;
}
int main(void)
{
  // Reset source checking
  reset_checking();
  //{{{WIZARD_MAP(Initialization)
  io_init();
  timers_init();
  uart_init();
```

```
lcd_init(16, 2, &PORTC);     // LCD Using PORTC
//}}WIZARD_MAP(Initialization)
// TODO: Add extra initialization here
        //{{WIZARD_MAP(Global interrupt)
sei();
//}}WIZARD_MAP(Global interrupt)
TIMSK=0x00;Sayi=0;TCNT0=0;
lcd_clear();lcd_gotoxy(0,0);
lcd_puts("Hazir");
PORTB=2;
while(1)
{
        // TODO: Add your code here
        wdt_reset();
        while(!(UCSRA & (_BV(RXC))))
        {
                wdt_reset();
        }
        if(UDR=='b' && TIMSK==0x00)
        {
                lcd_gotoxy(0,1);
                lcd_puts("Basladi");
                PORTB|=4;TCNT0=0;
                Sayi=0;TIMSK=0x05;
                wdt_reset();
        }
        else if(UDR=='d' && TIMSK==0x05)
        {
                lcd_gotoxy(0,1);
                lcd_puts("Durdu");
                PORTB-=4;
                TIMSK=0x00;
                wdt_reset();
        }
        else if(UDR=='r' && TIMSK==0x00)
        {
                _delay_ms(200);
        }
        else if(UDR=='t' && TIMSK==0x00)
        {
                Sayi=UDR;
                ByteGonder(Sayi);
        }
        wdt_reset();
        reset_checking();
  }
}
```

EK 2. Programda kullanılan kesme ve sayıcı fonksiyonlarının oluşturulduğu *EncoderTimer* dosyası ve buna ait başlık dosyası

```
//EncoderTimer.h : header file for the Encoder Timers
//
#ifndef _ENCODERTIMER_H_
#define _ENCODERTIMER_H_
/////////////////////////////////////////////////////////////////////////////
//EncoderTimer
void timers_init(void);
#endif // _ENCODERTIMER_H_
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
//---------------------------------------------------------------------------
//EncoderTimer.c : source file for the Encoder Timers
//

#include "Encoder.h"
#include "EncoderTimer.h"

//---------------------------------------------------------------------------
//EncoderTimer

void timers_init(void)
{
  //{{WIZARD_MAP(Timers)
  // Timer/Counter0 Clock source: System Clock
  // Timer/Counter0 Clock value: Stopped
  // Timer/Counter0 Mode: Normal
  // Timer/Counter0 Output: Disconnected
  OCR0 = 0x00;
  TCNT0 = 0x00;
  TCCR0 = 0x06;

  // Timer/Counter1 Clock source: System Clock
  // Timer/Counter1 Clock value: 7.813kHz
  // Timer/Counter1 Mode: Normal
  // Timer/Counter1 Output: A: Disconnected, B: Disconnected
  OCR1A = 0x0024;
```

```
  OCR1B = 0x0024;
  TCNT1 = 0xffdc;
  TCCR1A = 0x00;
  TCCR1B = (1<<CS12) | (1<<CS10);

  // Timer/Counter2 Clock source: System Clock
  // Timer/Counter2 Clock value: Stopped
  // Timer/Counter2 Mode: Normal
  // Timer/Counter2 Output: Disconnected
  ASSR = 0x00;
  OCR2 = 0x00;
  TCNT2 = 0x00;
  TCCR2 = 0x00;

  TIMSK = (1<<TOIE1) | (1<<TOIE0);
  //}}WIZARD_MAP(Timers)
}
//-----------------------------------------------------------------------------
ISR(SIG_OVERFLOW0)
{
  // TODO: Add your code here
  }
//-----------------------------------------------------------------------------
ISR(SIG_OVERFLOW1)
{
  // Reinitialize Timer1's value
  TCNT1 =0xffdc;
  // TODO: Add your code here
  Sayi=(unsigned char)TCNT0;
  wdt_reset();
  cli();
            ByteGonder(Sayi);
  sei();
  TCNT0=0;Sayi=0;
}
void ByteGonder(unsigned char a)
{
  wdt_reset();
  while(!(UCSRA & (_BV(UDRE))));
  UDR=a;
}
unsigned char ByteOku()
{
  wdt_reset();
  while(!(UCSRA & (_BV(RXC))));
  return(UDR);
}
```

EK 3. Kontrol kartı ile ara yüz arasındaki USB seri port üzerinden haberleşmeyi sağlayan *encoderUART* dosyası ve başlığı

```
//EncoderUART.h : header file for the Encoder UART

//

#ifndef _ENCODERUART_H_

#define _ENCODERUART_H_

//EncoderUART

void uart_init(void);

#endif // _ENCODERUART_H_

//-------------------------------------------------------------------------
//-------------------------------------------------------------------------
//-------------------------------------------------------------------------
//EncoderUART.c : source file for the Encoder UART
//

#include "Encoder.h"
#include "EncoderUART.h"

////////////////////////////////////////////////////////////////////////////
//EncoderUART

#ifndef LOW
#define LOW(i)                  (i & 0xff)
#endif // LOW
#ifndef HIGH
#define HIGH(i)                         ((i >> 8) & 0xff)
#endif // HIGH

#define COM_BAUD_RATE           9600Ul
#define COM_UBRR                (F_CPU/(COM_BAUD_RATE*16)-1)

void uart_init(void)
```

```
{
  //{{WIZARD_MAP(UART)
  // Baud Rate: 9600
  // Character Size: 8-bit
  // Mode: Asynchronous
  // Parity: Disabled
  // Stop Bit: 1-bit
  UBRRL = LOW(COM_UBRR);
  UBRRH = HIGH(COM_UBRR);
  UCSRA = 0x00;
  UCSRC = (1<<URSEL) | (1<<UCSZ1) | (1<<UCSZ0);
  UCSRB = (1<<RXCIE) | (1<<TXCIE) | (1<<RXEN) | (1<<TXEN);
  //}}WIZARD_MAP(UART)
}

ISR(SIG_UART_RECV)
{
  // TODO: Add your code here

}

ISR(SIG_UART_TRANS)
{
  // TODO: Add your code here

}
```