İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MOTOR ELEKTRONİK KONTROL ÜNİTESİNİN GERÇEK ZAMANLI DONANIM İÇEREN SİMÜLASYONLARDA TESTİ

YÜKSEK LİSANS TEZİ

Mak. Müh. Alper ÇEBİ

Anabilim Dalı: Disiplinlerarası Program

Programı: Mekatronik Mühendisliği

Tez Danışmanı: Prof.Dr. Levent GÜVENÇ

OCAK 2005

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

MOTOR ELEKTRONİK KONTROL ÜNİTESİNİN GERÇEK ZAMANLI DONANIM İÇEREN SİMÜLASYONLARDA TESTİ

YÜKSEK LİSANS TEZİ
Mak. Müh. Alper ÇEBİ
(518021009)

Tezin Enstitüye Verildiği Tarih : 27 Aralık 2004
Tezin Savunulduğu Tarih : 25 Ocak 2005

Tez Danışmanı : **Prof. Dr. Levent GÜVENÇ**
Diğer Jüri Üyeleri **Prof. Dr. Metin ERGENEMAN**
Yrd. Doç. Dr. Bilin AKSUN GÜVENÇ

OCAK 2005

ÖNSÖZ

Motor kontrol ve yönetim sistemleri motor gücü, yakıt tüketimi ve emisyon seviyelerini, sensör girişlerini kullanarak denetler ve olası sensör hatalarını, diğer hataları da kontrol ederek gerekli diyagnostik sinyalleri yaratırlar. Bu sistemler bir elektronik kontrol ünitesinde (ECU) yazılım ve donanım olarak yer alırlar. Test odasında ya da yolda yapılan testlerde tüm sensör hatalarının test edilmesine olanak yoktur. Donanım içeren testlerde ECU bir motor simülatörüne bağlanarak test odasında ya da yol testlerinde mümkün olmayan bir çok değişik hata senaryosuna maruz bırakılır. Bu tezde motor elektronik kontrol ünitesinin donanım içeren testlerini yapmak için kullanılacak bir motor simülasyon ortamı geliştirilmiş ve çeşitli donanım ve yazılımlarla test edilmiştir.

Tecrübesi ve değerli bilgisi ile çalışmamın her aşamasında bana yardımcı ve destek olan saygıdeğer hocam Prof. Dr. Levent GÜVENÇ'e sonsuz teşekkürlerimi sunarım. ÜG 03.004 sayılı Motor ECU Test Sistemi Geliştirilmesi projesi ile çalışmama destek olan Ford/OTOSAN'a ve Ford/OTOSAN'ın Motor & Güç Aktarma Sistemleri Bölümünden sayın Ergün GÜRASLAN, Korhan KANAR ve Canan KAPLAN KARADENİZ'e de teşekkürü bir borç bilirim.

Ayrıca tez çalışmam ile ilgili sorularımı içtenlikle yanıtlayan sayın Yrd. Doç. Dr. Bilin AKSUN GÜVENÇ'e, tez savunmasındaki yararlı yorum ve önerileri için sayın Prof. Dr. Metin ERGENEMAN'a, gerçek zamanlı donanım içeren simülasyonların gerçekleştirilmesinde yardımını esirgemeyen Murat DEMİRCİ'ye, Matlab GUI ile ilgili çalışmalarda yardımcı olan Sertaç KARAMAN'a ve diğer OTGEM çalışanlarına teşekkür ederim. Tüm öğrenim hayatım boyunca bana destek veren aileme de sonsuz sevgi ve saygılarımı sunarım.

Ocak 2005 Alper ÇEBİ

İÇİNDEKİLER

KISALTMALAR ..... **iv**
TABLO LİSTESİ ..... **v**
ŞEKİL LİSTESİ ..... **vi**
ÖZET ..... **viii**
SUMMARY ..... **ix**

1. GİRİŞ…. **1**
1.1 Amaç 1
1.2 Günümüze Kadar Yapılmış Çalışmalar 1
1.3 Kapsam 11

2. SİNYALLERİNİN OLUŞTURULMASI VE GEREKSİNİMLER **12**
2.1 Giriş 12
2.2 Krank ve Kam Sinyalleri 13
2.3 Motor Devrinden Dolayı Olan Gereksinimler 17

3. REAL-TIME WINDOWS TARGET İLE YAPILAN GERÇEK ZAMANLI DONANIM İÇEREN SİMÜLASYONLAR **21**
3.1 Giriş 21
3.2 RTWIN (Real-Time Windows Target) 22
3.3 Real-Time Windows Target Simülasyon Ayarları 23
3.4 NI PCI-6025E Kartı ile Yapılan Testler 25

4. DSPACE SİSTEMLERİ VE YAPILAN ÇALIŞMALAR **34**
4.1 Giriş 34
4.2 DS 1103 ile Yapılan Çalışmalar 35

5. XPC TARGET DONANIMI **46**
5.1 Giriş 46
5.2 xPC Target 46
5.3 xPC TargetBox 53

6. MATLAB GUI VE HATALI SİNYALLERLE YAPILAN SİMÜLASYONLAR **58**
6.1 Giriş 58

7. SONUÇLAR VE ÖNERİLER **62**

KAYNAKLAR **65**

EKLER **68**

ÖZGEÇMİŞ **81**

KISALTMALAR

ECU	**:** Electronic Control Unit
OBD	**:** On-board Diagnostics
HILS	**:** Hardware-in-the-loop Simulations
CARB	**:** California Air Research Board
SAE	**:** Society of Automotive Engineers
DSP	**:** Digital Signal Processor
CAN	**:** Controller Area Network
ABS	**:** Anti-lock Braking System
TCS	**:** Traction Control System
SBW	**:** Steer-by-Wire
GUI	**:** Graphical User Interface
MVEM	**:** Mean Value Engine Model
RTWIN	**:** Real-Time Windows Target

TABLO LİSTESİ

SAYFA NO

Tablo 2.1 Silindir Sayısına Göre Açısal Boşluk Miktarı 14
Tablo 7.1 Kullanılan Kartların Özellikleri ve Kullanılan Yazılım Bilgisi 63
Tablo 7.1 Kullanılan Platformlarda Alınan Test Sonuçlarının Karşılaştırılması. 64

ŞEKİL LİSTESİ

SAYFA NO

Şekil 1.1 Dizel Enjeksiyon Sistemi Donanım İçeren Simülasyon Genel Şeması [5] 3
Şekil 1.2 CARTS-Gerçek Zamanlı Simülatörü ve ECU [6] 4
Şekil 1.3 Ortak Yakıt Hattı Yapısı ve ECU [6] 4
Şekil 1.4 LPG Motor Blok Diyagramı [7] 5
Şekil 1.5 Skoda Fabia 1.4 50 kW Motor Sensör ve Aktüatör Konfigürasyonu [8] 6
Şekil 1.6 Elektronik Kontrol Ünitesi Test Sistemi [8] 6
Şekil 1.7 Simülasyonlarda Kullanılan Donanım Konfigürasyonu [15] 7
Şekil 1.8 ABS'li ve ABS'siz Test Sonuçları [15] 8
Şekil 1.9 Pnömatik Fren Sistemi için Simülasyon Sistemi [17] 8
Şekil 1.10 ECU Tasarım Geliştirme Platformu Donanım Blok Diyagramı [18] 9
Şekil 1.11 Elektronik Direksiyon(steer-by-wire) Donanım İçeren Simülasyon Sistemi [23] 10
Şekil 2.1 Dizel Motor ECU Sensör Şeması 12
Şekil 2.2 Gerçek Motordan Alınan Kam ve Krank Sinyalleri 13
Şekil 2.3 Kam Mili ve Sensörü (Segment Speed Sensor) 14
Şekil 2.4 Krank Mili ve Sensörü (Increment Speed Sensor) 15
Şekil 2.5 Krank Açısı'na Bağlı Krank Sinyali 16
Şekil 2.6 Krank Açısı'na Bağlı Kam Sinyali 16
Şekil 2.7 RPM'e Bağlı Krank Sinyali (2500 rpm) 18
Şekil 2.8 RPM'e Bağlı Kam Sinyali (2500 rpm) 18
Şekil 3.1 Donanım İçeren Simülasyon Oluşum Şeması 21
Şekil 3.2 NI PCI-6025E Kartı ve Özellikleri [31] 23
Şekil 3.3 Real Time Workshop Bölümünde Tanımlı Hedefler (target) 24
Şekil 3.4 Simülasyon Parametrelerinin Ayarlanması-I 24
Şekil 3.5 Simülasyon Parametrelerinin Ayarlanması-II 25
Şekil 3.6 Modelin Genel Görünüşü 25
Şekil 3.7 Alt sistem 26
Şekil 3.8 Ölçüm Ekranı (2500 rpm) 26
Şekil 3.9 Sıcaklık Test Modeli 27
Şekil 3.10 Oluşturulan Sinüs Verisi 28
Şekil 3.11 Ölçüm Ekranından Sinüs Veri Görüntüsü 28
Şekil 3.12 Sinyal Oluşturucu ile Oluşturulan Rampa Verisi 29
Şekil 3.13 Ölçüm Ekranından Rampa Görüntüsü 29
Şekil 3.14 Gerçek Motor Test Verisi 30
Şekil 3.15 Ölçüm Ekranı Gerçek Veri Test Görüntüsü 30
Şekil 3.16 MVEM (Mean Value Engine Model) 31
Şekil 3.17 MVEM ile Elde Edilen Motor Devri Cevabı 31
Şekil 3.18 MVEM ile Oluşturulan Motor Devri Test Modeli 32
Şekil 3.19 Ölçüm Ekranından MVEM Motor Devri Sonucu 32
Şekil 3.20 S-Function ile Motor Devri Oluşturulan Model 33
Şekil 4.1 dSPACE Simülatörü, Kartları ve Bağlantı Panelleri [34, 35] 34
Şekil 4.2 dSPACE DS1103 PPC Kontrolcü Kartı [36] 35
Şekil 4.3 dSPACE LED'li Bağlantı Paneli [37] 35

Şekil 4.4 Basınç, Sıcaklık ve Motor Devrinin Birlikte Oluşturulduğu Model 36
Şekil 4.5 Simülasyon Parametreleri Ayarları 37
Şekil 4.6 Real-Time Workshop Menüsü 37
Şekil 4.7 Real Time Workshop Bölümünde Tanımlı Hedefler (target) 38
Şekil 4.8 DAC Blok Açıklama Menüsü 39
Şekil 4.9 dSPACE Blok Kütüphanesi 39
Şekil 4.10 Master PPC Alt Kütüphanesi 40
Şekil 4.11 Control Desk Programı Genel Görünümü 41
Şekil 4.12 Control Desk Programında Oluşturulan Model 41
Şekil 4.13 Control Desk Programı Layout ve Gauge Görüntüsü 42
Şekil 4.14 Sensör Verileri Ölçüm Ekranı 42
Şekil 4.15 Motor Devri ve Yakıt Basıncı Ölçüm Ekranı 43
Şekil 4.16 MVEM ile Oluşturulan Veri 44
Şekil 4.17 Ölçüm Ekran Görüntüsü 44
Şekil 5.1 xPC Target Setup Menüsü 46
Şekil 5.2 Seri Bağlantı Gösterimi 47
Şekil 5.3 Network Bağlantı Gösterimi 47
Şekil 5.4 Real-Time Workshop Hedef Tanımlama Menüsü 48
Şekil 5.5 Model Genel Görünümü 48
Şekil 5.6 Motor Devri Modeli Alt Sistemi 49
Şekil 5.7 xPC Target 'www' Arayüzü 49
Şekil 5.8 xPC Target Remote Control Tool Penceresi 50
Şekil 5.9 xPC Target Spy Seçeneği ile Hedef Bilgisayardan Alınan Görüntü 51
Şekil 5.10 Monitör Görüntüsü 51
Şekil 5.11 Sıcaklık Sensörleri (xPC Target Sistemi için) 52
Şekil 5.12 Sıcaklık Sensörleri için Göstergelerin Olduğu Model (xPC Target) 52
Şekil 5.13 xPC TargetBox [38] 53
Şekil 5.14 Basınç Sensörlerinin Oluşturulduğu Model (xPC TargetBox) 54
Şekil 5.15 Basınç Göstergelerinin Olduğu Model (xPC TargetBox) 54
Şekil 5.16 Sinyal Özellikleri Menüsü 55
Şekil 5.17 Blok Özellikleri Menüsü 55
Şekil 5.18 Motor Devri ve Sıcaklık Kontrollü Model 56
Şekil 5.19 Motor Devri Alt Sistemi 56
Şekil 6.1 GUIDE Hızlı Başlangıç Penceresi 58
Şekil 6.2 Matlab GUI Model Oluşturma Penceresi 59
Şekil 6.3 Krank ve Kam Sinyali Bozma GUI Modeli 59
Şekil 6.4 15. Krank Dişinin Kırık Olduğu Motor Devri Sinyali 60
Şekil 6.5 4. Kam Dişinin Kırık Olduğu Motor Devri Sinyali 60
Şekil 6.6 15. Krank ve 4. Kam Dişinin Kırık Olduğu Motor Devri Sinyali 61
Şekil 6.7 Ölçüm Ekranı 61
Şekil A.1 NI PCI-6025E Kartı Blok Diyagramı [39] 69
Şekil A.2 NI PCI-6025E Kartı Pin Diyagramı [39] 69
Şekil C.1 DS 1103 Kontrolcü Kartı Blok Diyagramı [36] 72
Şekil D.1 DIAMOND MM-32 AT Kartının Blok Diyagramı [40] 73
Şekil D.2 DIAMOND MM-32 AT Kartının Pin Numaralandırılması [40] 73

MOTOR ELEKTRONİK KONTROL ÜNİTESİNİN GERÇEK ZAMANLI DONANIM İÇEREN SİMÜLASYONLARDA TESTİ

ÖZET

Bu çalışmada, gerekli sinyalleri oluşturabilmek için bir motor simülatörü geliştirilmiş ve motor elektronik kontrol ünitesinin çeşitli donanım ve yazılımlarla gerçek zamanlı donanım içeren simülasyon testleri yapılmıştır.

Motor kontrol ve yönetim sistemleri motor gücü, yakıt tüketimi ve emisyon seviyelerini, sensör girişlerini kullanarak denetler ve olası sensör hatalarını, diğer hataları da kontrol ederek gerekli diyagnostik sinyallerini yaratırlar. Bu sistemler bir elektronik kontrol ünitesinde (ECU) yazılım ve donanım olarak yer alırlar. Test odasında ya da yolda yapılan testlerde tüm sensör hatalarının test edilmesine olanak yoktur. Donanım içeren testlerde ECU bir motor simülatörüne bağlanarak test odasında yada yol testlerinde mümkün olmayan bir çok değişik hata senaryosuna maruz bırakılır. Motor ECU'nun donanım içeren simülasyonlarla test edilebilmesi için gerekli sensör sinyallerini oluşturan bir motor simülatörü geliştirilmesi gereklidir.

Birinci bölümde çalışmanın amacına değinilmiş ve günümüze kadar yapılan çalışmalar hakkında bilgi verilmiştir.

İkinci bölümde ise motor devri için krank ve kam sinyallerinin oluşturulması ve bununla ilgili olan gereksinimler anlatılmıştır.

Üçüncü bölümde Real-Time Windows Target ile National Instruments firmasının PCI 6025E kartı kullanılarak yapılan simülasyonlar verilmiştir.

dSPACE firmasının DS 1103 kartı ile yapılan çalışmalar ise dördüncü bölümde anlatılmaktadır.

Beşinci bölümde xPC Target donanımı ile yapılan uygulamalar, altıncı bölümde ise Matlab GUI (graphical user interface) ile yapılan çalışmalar ve simülasyonlar anlatılmıştır.

Yedinci ve son bölümde ise sonuçlar anlatılarak karşılaştırmalar yapılmış ve gelecek için öneriler verilmiştir.

TESTING AN ENGINE ELECTRONIC CONTROL UNIT IN REAL-TIME HARDWARE-IN-THE-LOOP SIMULATIONS

SUMMARY

In this study, a hardware-in-the-loop engine simulator that generates neccesary engine signals is developed and used in testing an engine electronic control unit (ECU) in different software and hardware platforms.

Engine control and management systems control engine power, fuel consumption and emission levels by using sensor inputs and create diagnostic signals by checking sensor and other failures. These systems exist as hardware and software in an electronic control unit. To create all combinations of failures is impossible to accomplish in the test room or in road tests. The engine ECU is subjected to various failure scenarios that are impossible to generate in the test room or in road tests by connecting an engine simulator for hardware-in-the-loop simulation. To test the engine ECU with hardware-in-the-loop simulations, developing an engine simulator that generates the neccesary signals is required.

In the first chapter, the goals of this thesis are mentioned and a literature review is given.

In the second chapter, generating crank and cam signals for engine speed and the associated requirements are explained.

The hardware-in-the-loop simulations with Real-Time Windows Target using the NI PCI-6025E board are presented in the third chapter.

The fourth chapter contains results with the DS 1103 board used in hardware-in-the-loop testing.

The use of xPC Target hardware is presented in the fifth chapter. The use of the Matlab GUI (graphical user interface) for automated erroneous signal generation is presented in the sixth chapter.

The seventh chapter contains a comparison of the results achieved, conclusions and recommendations.

1. GİRİŞ

1.1 Amaç

Motor kontrol ve yönetim sistemleri motor gücü, yakıt tüketimi ve emisyon seviyelerini, sensör girişlerini kullanarak denetler. Ayrıca olası sensör hatalarını ve diğer hataları da kontrol ederek gerekli diyagnostik sinyallerini yaratırlar. Bu sistemler bir elektronik kontrol ünitesinde (ECU) yazılım ve donanım olarak yer alırlar. Yeni bir motorun geliştirilmesi aşamasındaki zaman daraltıcı etkenlerden önemli biri de elektronik kontrol ünitesi'ndeki kontrol katsayı tablolarının kalibrasyonu ve tüm ECU yazılımının hatalara karşı sınanmasıdır. Test odasında ya da yolda yapılan testlerde tüm sensör hatalarının test edilmesine olanak yoktur. Dolayısıyla otomotiv sektöründe motor elektronik kontrol ünitesi'nin sınanmasını hızlandırma amaçlı olarak donanım içeren (hardware-in-the-loop) simülasyonlar kullanılır. Donanım içeren testlerde ECU bir motor simülatörüne bağlanarak test odasında ya da yol testlerinde mümkün olmayan bir çok değişik hata senaryosuna maruz bırakılır. ECU testleri, otomobil üreticilerinin kalite standartlarını arttırmak, maksimum verime ulaşabilmek ve olabilecek hataları en aza indirebilmek için yaptığı çeşitli testlerdendir.

Bu tezde bir motor elektronik kontrol ünitesinin donanım içeren testlerini yapmak için kullanılacak bir motor simülatörünün geliştirilmesi ve çeşitli donanım ve yazılımlarla test edilmesi anlatılmaktadır.

1.2 Günümüze Kadar Yapılmış Çalışmalar

Motor kontrol ünitesi olarak da adlandırılan "ECU" 1970'li yıllarda ortaya çıkmış ve 1980'lerden itibaren yol taşıtlarının standart bir parçası haline gelmiştir.

ECU, kontrol fonksiyonlarından sorumlu bir elektronik kontrol modülüdür. Motor ECU'sunun temel amacı, motor tarafından üretilip havaya atılan kirletici maddeleri azaltmak ve yakıt tasarrufu sağlamak için yakıt ve enjeksiyon sisteminin kapalı-çevrim kontrolünü sağlamaktır.

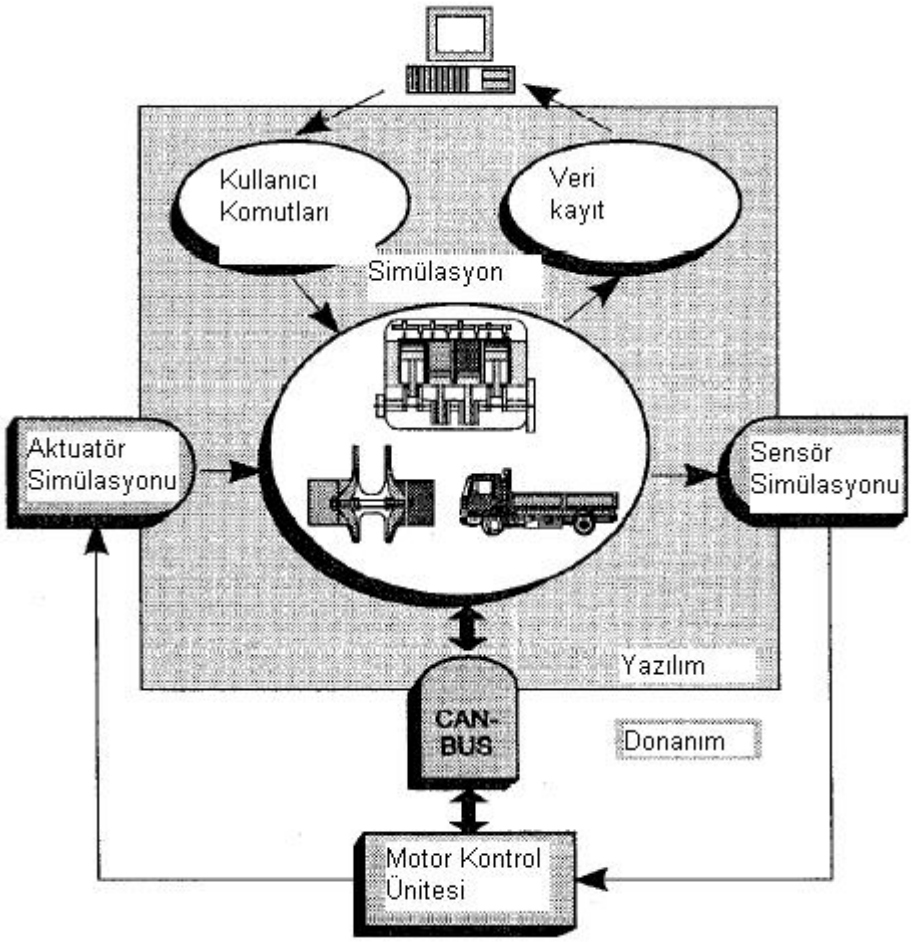
Elektronik motor kontrol sistemi, motor çalışma koşullarını sürekli ölçen sensörlerden ve bu sensörlerden aldığı girişleri veri tablolarını kullanarak gerekli aktüatör hareketleri olarak hesaplayıp kontrol çıktıları haline getiren elektronik kontrol ünitesinden (ECU) meydana gelir.

ECU motor kontrol sistemindeki işlevini tam olarak yerine getirebilmek için sensör bilgilerine ihtiyaç duymaktadır. Yani motor devri, krank ve kam durumu, hava miktarı, motor sıcaklığı, motor yükü, kelebek açısının değişimi gibi bilgileri (bu liste daha da uzatılabilir) ECU'nun alması gerekir. ECU aldığı sensör bilgileri ile enjeksiyon zamanlamasını ve gerekli yakıt miktarını belirler.

ECU içerisinde işletim sistemi ve gömülü uygulama olarak çalışan OBD (on-board diagnostics, bkz. [1]) denilen bir yazılım elemanı bulunmaktadır. Bu yazılım elemanı emisyon azaltımı ile ilgili konulan şartları sağlamak zorundadır. 1987 yılından itibaren California'da araçlar OBD I olarak bilinen yazılım özelliği ile satışa sunulmaktadır. 1996'dan itibaren OBD II kullanılmaya başlanmıştır. Araçların kullanıldıkları ülkelere göre CARB (California Air Research Board), SAE (Society of Automotive Engineers) gibi uzman kuruluşlar emisyon standartlarının belirlenmesinde yardımcı olmaktadır [2].

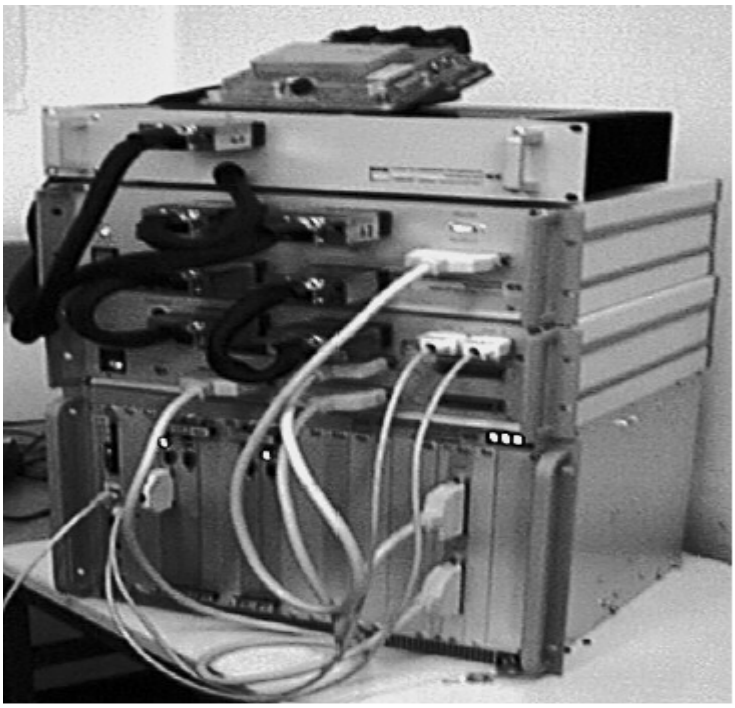
Otomotiv sektöründe motor elektronik kontrol ünitesi'nin sınanmasını hızlandırma amaçlı olarak donanım içeren (hardware-in-the-loop) simülasyonlar kullanılmaktadır. Donanım içeren simülasyonlar (HILS, hardware-in-the-loop-simulations), araca ait bir donanımın, birlikte çalıştığı diğer parçalarının bilgisayarda hazırlanmış sanal modelleriyle gerçek zamanlı olarak çalıştırılıp, test edildiği simülasyonlardır. Donanım içeren simülasyonlar ile ilgili bir çok çalışma yapılmış, ilk olarak uçuş simülasyonları ve araç parçalarının dinamik testlerini yapabilmek için kullanılmıştır. Kimura ve Maeda [3] motor kontrolü için gerçek zamanlı motor ve araç simülatörü geliştirmişlerdir. Isermann vd. [4] simülasyon yöntemleri hakkında bilgi verip gerçek zamanlı simülasyonlar arasında karşılaştırmalar yapmışlardır. Schaffnit vd. [5] gerçek zamanlı donanım içeren simülasyonlar hakkında bir çalışma yapmışlardır. Bahsedilen çalışmada (bkz. [5]) Mercedes-Benz B serisi 4 silindirli (125 kW) ve 8 silindirli (420 kW) kamyon motoru Matlab/Simulink kullanılarak modellendikten sonra üretilen sinyaller dSPACE sistemi (dijital sinyal işlemcisi-DSP, DEC alpha işlemcisi) kullanılarak motor kontrol ünitesine gönderilmiş ve testler yapılmıştır. Araç motor kontrol ünitesi entegre edilmiş elektronik sistemden CAN-ağı üzerinden aldığı sensör sinyal bilgilerini ve sürücü komutlarını kullanarak motor torkunu, fren torkunu, motor devrini ve referans değerlerini hesaplar. Bu referans değerleri bir motor Can-

ağı ile pompa hattı enjeksiyon memesi kontrol ünitesine manyetik enjeksiyon valflerin ayarlanması için gönderilir. Bu şekilde gerçek motor olmadan istenilen simülasyonlar yapılmış ve hata etkileri incelenmiştir. Şekil 1.1'de donanım içeren simülasyonun genel şeması gösterilmektedir.



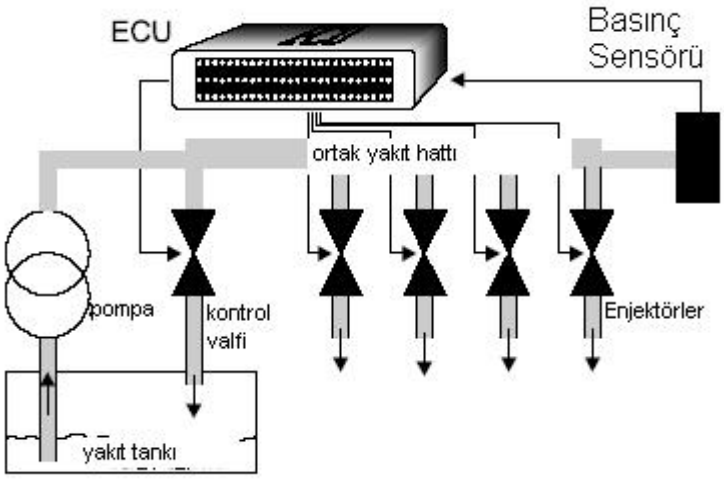
Şekil 1.1 Dizel Enjeksiyon Sistemi Donanım İçeren Simülasyon Genel Şeması [5]

Gerçek zamanlı donanım içeren simülasyonlara bir örnek de Woermann vd.'nin [6] çalışmasıdır. Bu çalışmada gerçek zamanlı donanım içeren simülasyonlarda kullanmak için pompa, ortak yakıt hattı (common rail), kontrol valfi ve enjektörlerin yakıt kütle akışı ve basıncının dinamik karakteristikleri gözönüne alınarak oluşturulan model bir dizel motor modeli ile birleştirilmiştir. Oluşturulan model bir dizel ECU ve gerçek zamanlı simülatör CARTS (bkz. [6] ve Şekil 1.2) ile bağlanarak simülasyonlar yapılmış, alınan sonuçlar gerçek testlerdeki sonuçlarla karşılaştırılmıştır. Sonuçlarda motor sabit bir devirde çalıştırılırken gaz pedalı pozisyonu değiştirilerek motor torku değişimi ve farklı motor devirlerindeki yakıt basıncı değişimi izlenmiş ve karşılaştırılmıştır.



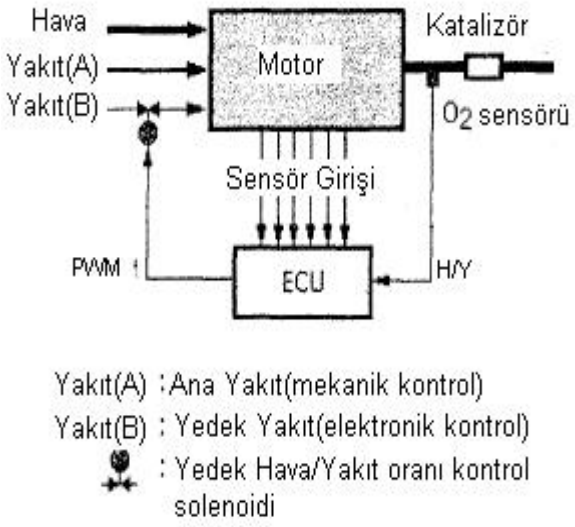
Şekil 1.2 CARTS-Gerçek Zamanlı Simülatörü ve ECU [6]

Şekil 1.3'te ECU'nun motor devri ve yüküne bağlı olarak basınç kontrol valfini ayarladığı basit bir model gösterilmektedir. Ortak yakıt hatlı sistem için ana enjeksiyondan önce başlangıç koşullarının sağlanması için bir ön enjeksiyon gereklidir. ECU basınç sensörü, motor devri ve yükü bilgilerini alıp basınç kontrol valfine komut vererek solenoid valflerinin kusursuz çalışmasını sağlar. Ön enjeksiyon ile yanma sırasında daha yüksek bir sıcaklık elde edilmektedir. Aryıca ön enjeksiyon olmadığında yanma sırasında daha fazla gürültü emisyonu oluşmaktadır. Woermann vd. [6] bu çalışmalarında gürültü, yakıt tüketimi ve hava kirletici atıkları azaltabilmiştir.



Şekil 1.3 Ortak Yakıt Hattı Yapısı ve ECU [6]

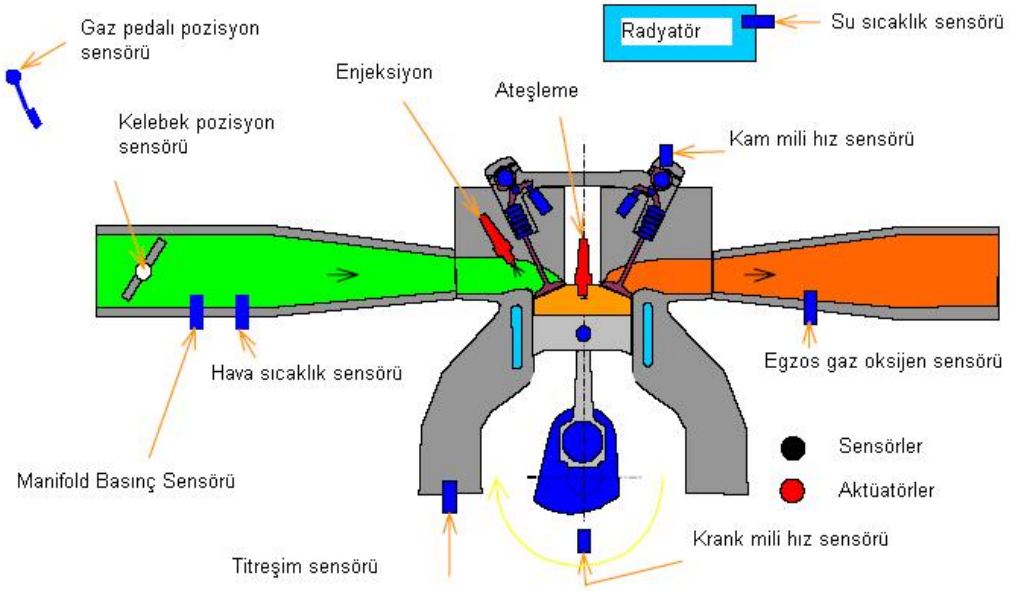
ECU'nun donanım içeren simülasyonlarda kullanıldığı başka bir çalışma da Sunwoo vd. [7] tarafından yapılmıştır. Bu çalışmada bir LPG'li motorun hava-yakıt karışımı kontrolünü sağlamak amacı ile Motorola MCU MC68HC05 işlemcisi ile ECU tasarımı yapılmış ve prototip ECU ile performans arttırılıp, emisyon azaltılmıştır. Bu çalışmada oksijen sensöründen (BOSCH model LSM11) gelen sinyale göre ECU'nun enjeksiyon zamanlaması kontrol edilmektedir. LPG motorunun hava-yakıt kontrolünün blok diyagramı Şekil 1.4'te gösterilmektedir.



Şekil 1.4 LPG Motor Blok Diyagramı [7]

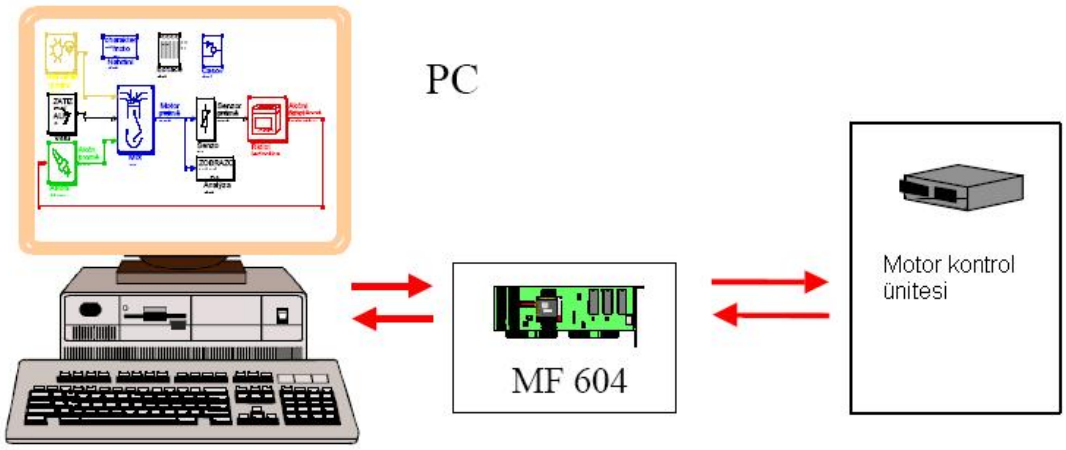
ECU motor devri, kelebek pozisyonu, su sıcaklığı, oksijen sensörü ve motor boşta şalterinden aldığı sinyallere göre yedek hava/yakıt oranı kontrol solenoidine PWM (20 kHz) sinyali göndermektedir. Bu şekilde yakıt akışı kontrolü ve yedek enjeksiyon sağlanmaktadır.

Donanım içeren simülasyonlara başka bir örnek de Jurak ve Tuma [8] tarafından yapılan çalışmadır. Bilgisayarda Matlab/Simulink kullanılarak oluşturulan benzinli (SI) motor modeli Weeks ve Moskwa [9] tarafından yapılan çalışmadan alınmıştır. Motor modeli verileri Skoda Fabia 1.4 50 kW motorundan alınmıştır. Bu motorun sensör ve aktüatör konfigürasyonu Şekil 1.5'te gösterilmektedir. Simülasyonlarda Matlab/Simulink, Real-Time Windows Target ve Real-Time Workshop ile birlikte SIEMENS SIMOS 3PB elektronik kontrol ünitesi kullanılmıştır. Simulink motor modeli ile motor kontrol ünitesinin iletişimi çok fonksiyonlu bir kart (MF 604 tipi) kullanılarak sağlanmıştır. Şekil 1.6'da elektronik kontrol ünitesi testinde kullanılan sistem gösterilmektedir.



Şekil 1.5 Skoda Fabia 1.4 50 kW Motor Sensör ve Aktüatör Konfigürasyonu [8]

Oluşturulan modelle sensörlerin ve gerçek bir benzinli motorun davranışı simüle edilerek kontrol ünitesi prototiplenmiştir. Normal çalışma koşullarında 10 kHz'lik örnekleme hızıyla 4000 rpm motor devrine ulaşılmıştır.

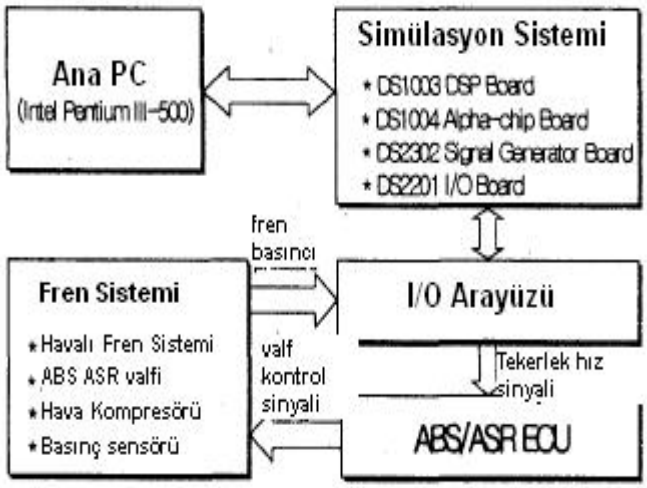


Şekil 1.6 Elektronik Kontrol Ünitesi Test Sistemi [8]

Günümüzde bir standart haline gelen anti blokaj fren sistemleri (ABS) aracın kararlılığını kısmen sağlamakta, kaymayı engelleyerek bazı kritik durumların oluşmasını engellemektedir. İlk ABS sistemi Bosch [10] firması tarafından 1978 yılında otomotiv pazarının kullanımına sunulduktan sonra Mercedes Benz S serisi ve BMW 700 serisi araçlarda kullanılmıştır. 1980'li yılların başında Bosch [10] çekiş kontrol sistemleri (TCS/ASR) geliştirmeye başlamış ve 1986 yılından itibaren bu teknolojiyi ticari araçlarda kullanmaya başlamıştır. TCS aracın ivmelenme sırasında

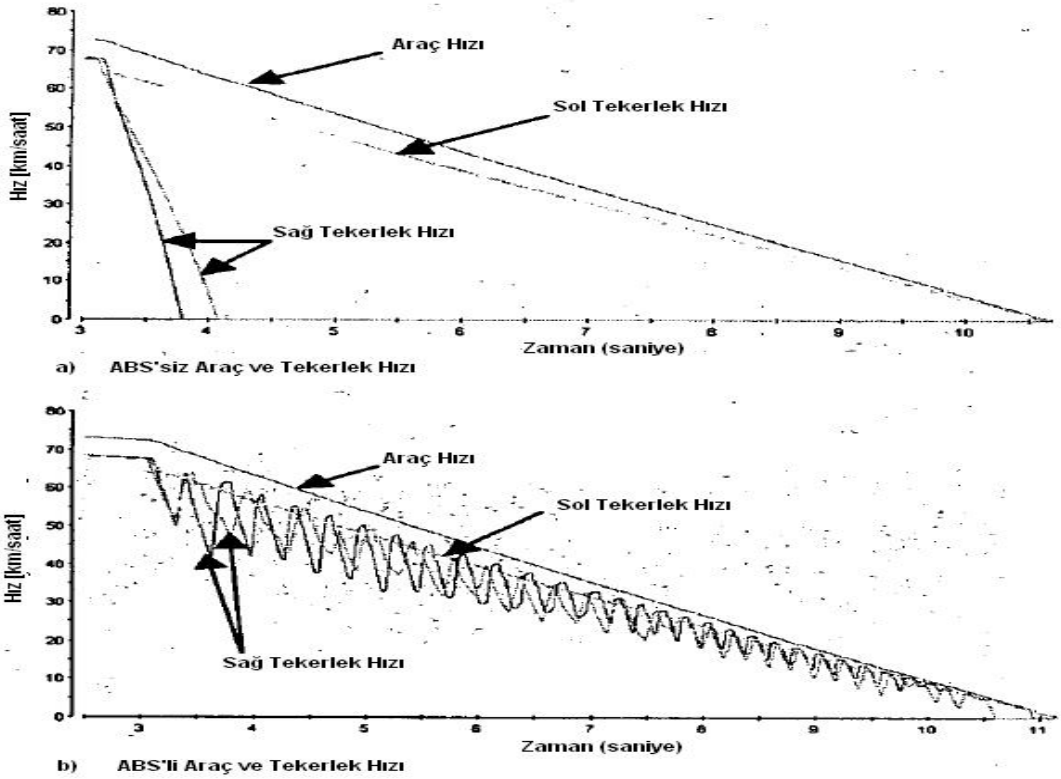
tekerlek hızlarını gözleyerek tekerleklerin arasındaki hız farkını belirler, eğer tekerleklerden biri diğerlerine göre hızlı dönüyorsa ABS sisteminin aktüatörlerini kullanarak o tekerleğe kısa süreli fren kuvveti uygulayarak patinaj çekmeyi engeller.

Günümüzde araçlarda kullanılan bu sistemler donanım içeren simülasyonlarda ECU kullanılarak test edilmektedir. Yapılan birçok çalışmada donanım içeren simülasyonlarda yolcu araçları için ABS ECU'su test edilmiştir [11-14]. Cho vd. [15] ise ticari araçlar için ABS ECU'nun gerçek zamanlı donanım içeren simülasyonlarını yapmışlardır. Gerçek zamanlı simülasyonlar için dijital sinyal işlemcisi ile alpha-chip işlemci kartı kullanılmıştır. Şekil 1.7'de simülasyonda kullanılan donanım konfigürasyonu gösterilmektedir.



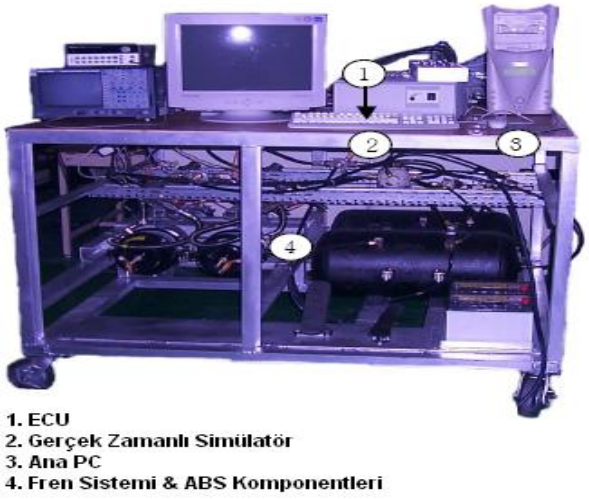
Şekil 1.7 Simülasyonlarda Kullanılan Donanım Konfigürasyonu [15]

Araç dinamik modeli araç gövde ve tekerlek dinamiğinden meydana gelmektedir. 14 serbestlik derecesine sahip bir araç modeli ve Dugoff vd. [16] tekerlek modeli kullanılmıştır. dSPACE Control Desk yazılımı kullanılarak araç hızı, tekerlek hızı, tekerleğin ivmelenmesi (her iki yönde), tekerlek fren basıncı ve tekerlek kayması görsel olarak modellenerek izlenmiştir. ABS performansı 70 km/saat başlangıç araç hızı ile farklı tekerlek yol tutunma katsayılarında (μ=0.3 buzlu yol sağ tekerleklere, μ=0.8 normal yol sol tekerleklere) fren uygulanarak test edilmiş ve ABS performansı arttırılmaya çalışılmıştır. Şekil 1.8'de bu çalışmadan alınan ABS'li ve ABS'siz test sonuçları gösterilmiştir.



Şekil 1.8 ABS'li ve ABS'siz Test Sonuçları [15]

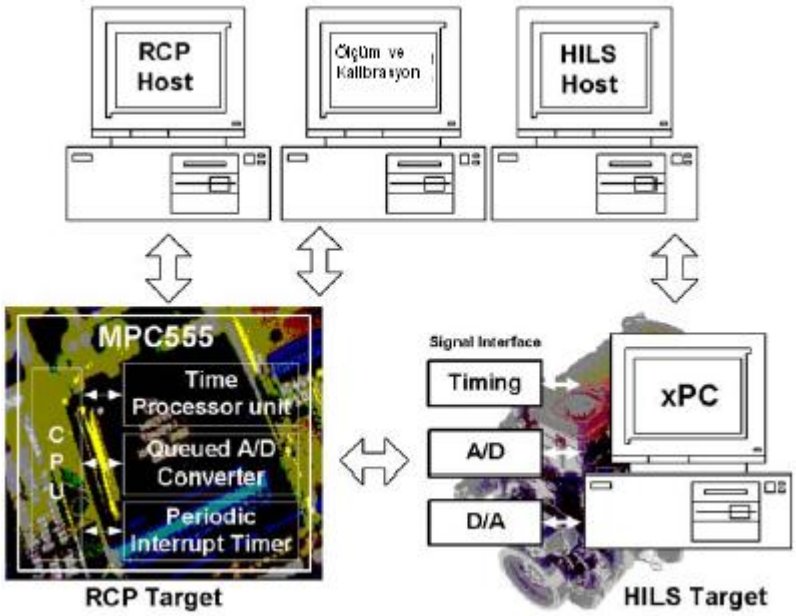
Lee vd. [17] pnömatik fren sistemi için ECU ile ABS kontrol algoritması performans testleri yapmıştır. Bu çalışmada ECU hem gerçek araçta hem de donanım içeren simülasyonlarda kullanılmış ve aynı sonuçların alındığı gözlenmiştir. Şekil 1.9'da gerçek zamanlı simülasyonda kullanılan sistem gösterilmektedir.



Şekil 1.9 Pnömatik Fren Sistemi için Simülasyon Sistemi [17]

ECU için yapılan bazı çalışmalarda öncelikle prototipleme üzerinde durulup daha sonra oluşturulan prototip, donanım içeren simülasyonlarda kullanılmış ve kontrol

algoritmaları üzerinde iyileştirme çalışmaları yapılmıştır. Lee vd. [18] yaptıkları çalışmada hızlı kontrolcü prototiplendirme (RCP, rapid control prototyping) ve donanım içeren simülasyonlar üzerinde durmuşlardır. Simülasyonlar Matlab/Simulink/Real-Time Workshop, xPC Target ve standart I/O kartları ile yapılmıştır. Çalışmada hava/yakıt kontrol sistemi test edilmiş ve ECU tasarım platformu geliştirilmesi amaçlanmıştır. Şekil 1.10'da bu çalışmada kullanılan ECU tasarım geliştirme platformu donanım blok diyagramı gösterilmektedir.

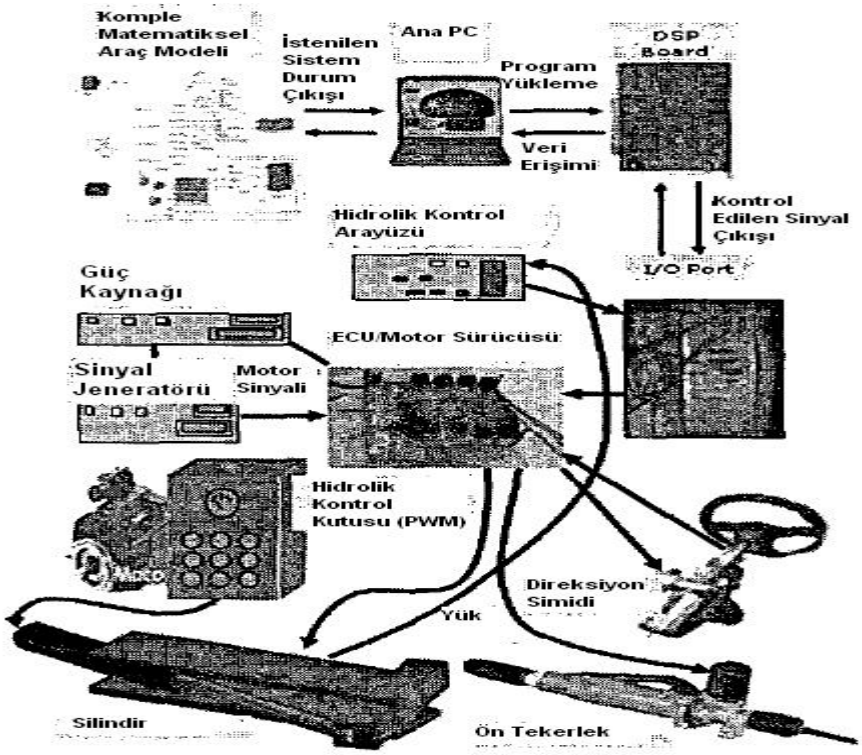


Şekil 1.10 ECU Tasarım Geliştirme Platformu Donanım Blok Diyagramı [18]

ECU prototipleme ile ilgili bir başka çalışma da Weber ve Isermann [19] tarafından yapılmıştır. Çalışmada bir kamyon dizel motoru Matlab/Simulink ile modellenmiş ve gerekli sinyaller için dSPACE I/O kartları kullanılmıştır. Motor kontrol fonksiyonları geliştirilmesi için oluşturulan gerçek zamanlı sistem DaimlerChrysler AG ile birlikte kurulmuş ve bir lineer olmayan aşırı doldurma kontrol algoritması tasarımı için gerekli ECU fonksiyonlarının geliştirilmesi amaçlanmıştır. Yine benzer bir çalışma Hafner vd. [20] tarafından yapılmış, Matlab/Simulink, dijital sinyal islemcisi, ControlDesk yazılımları kullanılarak farklı kontrol algoritmaları test edilmiştir.

Son yıllarda elektronik direksiyon (steer-by-wire) üzerine çalışmalar yoğunlaşmıştır. Elektronik direksiyon sisteminin güvenlik ve konfora getirdiği yenilikler Ackermann tarafından ifade edilmiş ve kritik durumlar için elektronik direksiyon kontrolcüsüne sahip araç modeli analiz edilmiştir [21, 22].

Elektronik direksiyon sisteminde iki adet aktüatör bulunmaktadır. Bu iki aktüatörün görevi, sırasıyla tekerleği döndürmek ve direksiyon simidine istenen yol etkilerini geri beslemektir. İki aktüatörden birincisi direksiyon sistemine girilmesi gereken kontrol sinyalini iletirken diğeri ise sürücünün sürüş yeteneğini koruyabilmesi ve tekerleklere gelen kuvvetler hakkında değerlendirmeler yapabilmesi için direksiyon sistemindeki kuvvetleri sanal olarak oluşturarak direksiyon simidine uygular. Elektronik direksiyon sistemi de donanım içeren simülasyonlar vasıtasıyla test edilebilir. Jang vd. [23] Şekil 1.11'de gösterilen donanım içeren simülasyon düzeneği ile elektrik direksiyon sistemini test etmişlerdir.



Şekil 1.11 Elektronik Direksiyon (steer-by-wire) Donanım İçeren Simülasyon Sistemi [23]

Günümüzde donanım içeren simülasyonlarda kullanılabilecek ticari olarak satılan simülatör bilgi işlem platformları mevcuttur. Çeşitli simülatörler, gerçek zamanlı donanım içeren simülasyonlarda kullanılan farklı platformlar ve uygulamaları ile ilgili olarak Hanselmann [24], Boot vd. [25], Caraceni vd. [26], Lefarth vd. [27] çalışmaları örnek gösterilebilir. Ayrıca ECU tasarım ve özellikleri ile bazı ek uygulama alanları Bortolazzi vd. [28] tarafından açıklanmıştır. Müller-Glaser vd. [29], Giusto ve Demmeler [30] bu bilgilerin yanında ilgili teknolojiler, simülasyon test platformları ve çeşitli modelleme dilleri hakkında bilgi verip otomotiv ECU uygulamalarından örnekler vermişlerdir.

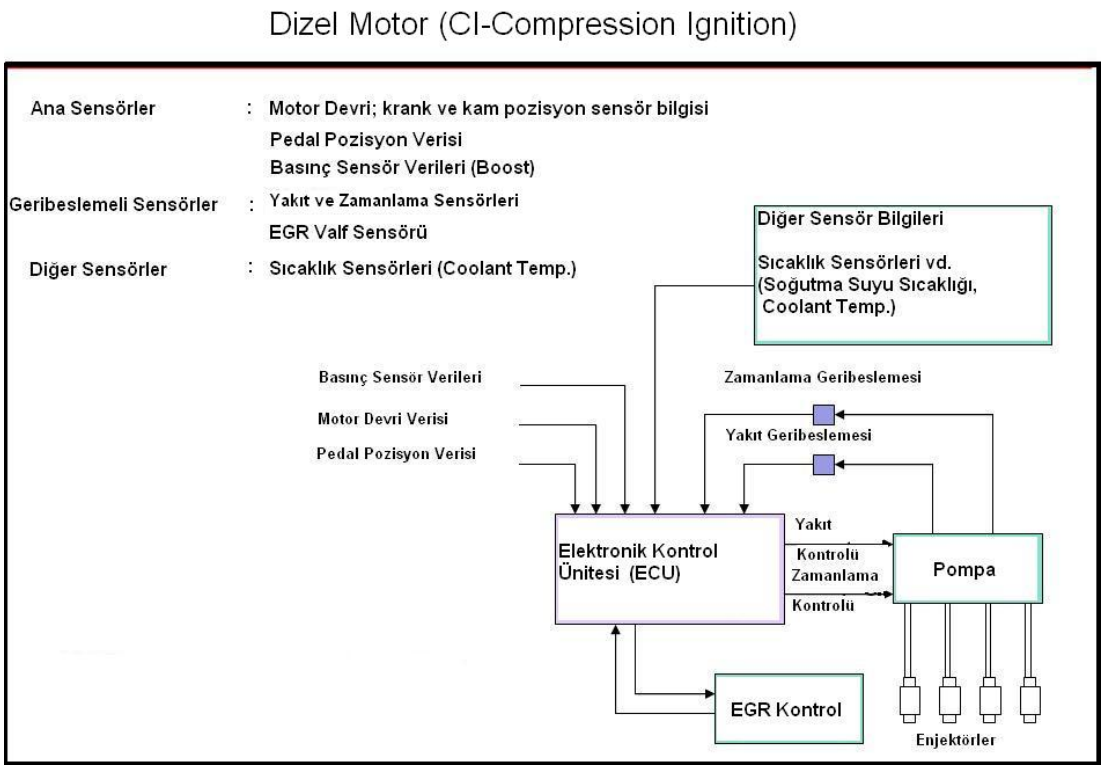
1.3 Kapsam

Bu tezde bir motor elektronik kontrol ünitesinin donanım içeren testlerini yapmak için kullanılacak bir motor simülatörü geliştirilmiş, çeşitli donanım ve yazılımlarla test edilmiştir. Birinci bölümde motor ECU'su tanıtılmış ve günümüze kadar yapılan çalışmalar üzerinde durulmuştur. İkinci bölümde motor devri için krank ve kam sinyallerinin oluşturulması ve bununla ilgili gereksinimler anlatılmıştır. Üçüncü bölümde Real-Time Windows Target ile National Instruments firmasının PCI 6025E kartı kullanılarak yapılan simülasyonlar verilmiştir. dSPACE firmasının DS 1103 kartı ile yapılan çalışmalar ise dördüncü bölümde anlatılmaktadır. Beşinci bölümde xPC Target donanımı ile yapılan uygulamalar, altıncı bölümde ise Matlab GUI (graphical user interface) ile yapılan çalışmalar ve simülasyonlar anlatılmıştır. Yedinci bölümde sonuçlar ve öneriler verilmektedir.

2. SİNYALLERİNİN OLUŞTURULMASI VE GEREKSİNİMLER

2.1 Giriş

ECU motor kontrol sistemindeki işlevini tam olarak yerine getirebilmek için sensör bilgilerine ihtiyaç duymaktadır. Yani motor devri, krank ve kam durumu, hava miktarı, motor sıcaklığı, motor yükü, kelebek açısının değişimi gibi bilgileri ECU'nun alması gerekir. ECU aldığı sensör bilgileri ile enjeksiyon zamanlamasını ve gerekli yakıt miktarını belirler. Bir dizel motorun elektronik kontrol ünitesine gelen sinyaller Şekil 2.1'de basit olarak gösterilmiştir. Bu tezde ECU'nun alması gereken sensör sinyal bilgilerinden öncelikli olarak krank ve kam sinyalleri ile yağ basıncı (oil pressure), aşırı doldurma basıncı (boost pressure), yakıt basıncı (rail pressure), soğutma suyu sıcaklığı (coolant temperature), aşırı doldurma sistemi sıcaklığı (boost temperature) sinyalleri oluşturulmuş ve test edilmiştir.

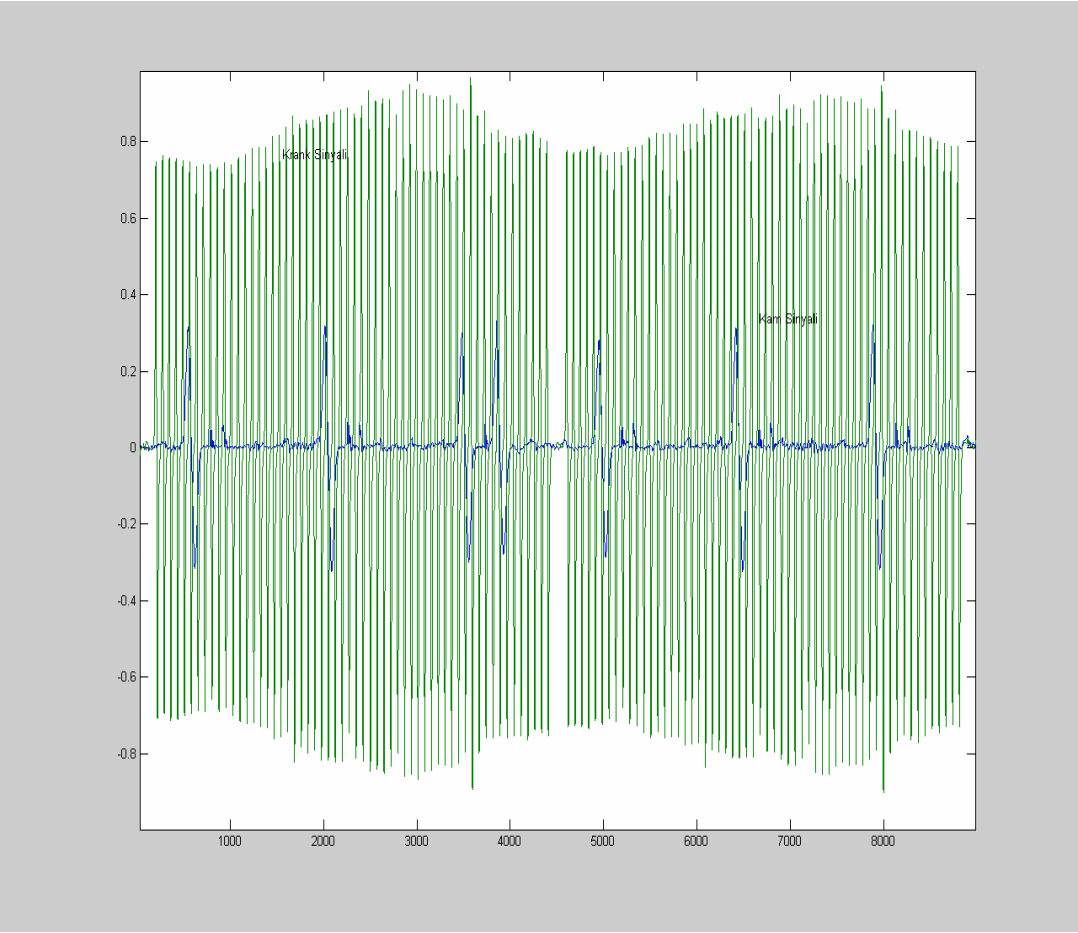


Şekil 2.1 Dizel Motor ECU Sensör Şeması

Bu bölümde motor devri için krank ve kam sinyallerinin oluşturulması ve bu yüzden olan gereksinimler anlatılacaktır.

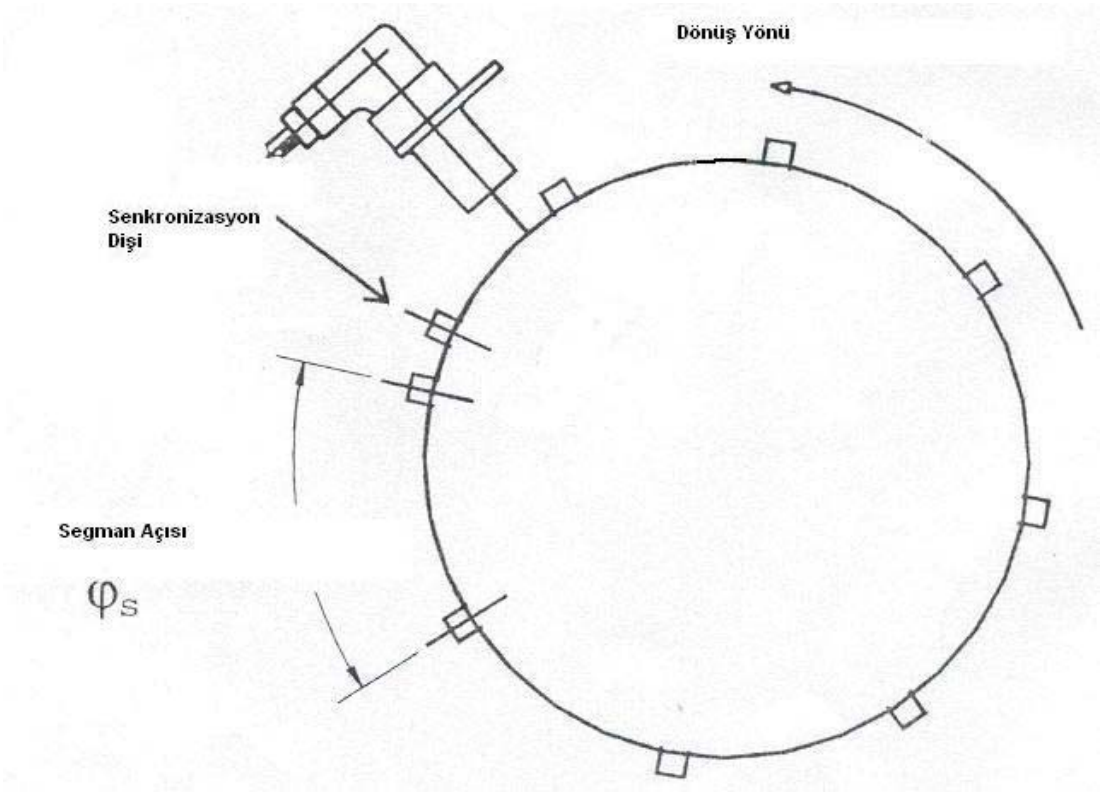
2.2 Krank ve Kam Sinyalleri

Bu tez kapsamında oluşturulması gereken en zor sinyaller krank ve kam sinyalleridir. Motor devrini oluşturan krank ve kam sinyalleri birbiriyle senkronize olarak çalışmaktadır. Birbiriyle senkronize olarak çalışması bu sinyalleri oluşturmaktaki zorluklardan biridir. Bu tezde krank ve kam sinyalleri oluşturulurken bir ağır ticari araç motorundan alınan veriler ve sensör özellikleri dikkate alınmıştır. Şekil 2.2'de bahsi geçen motordan alınan krank ve kam sinyalleri gösterilmektedir.



Şekil 2.2 Gerçek Motordan Alınan Kam ve Krank Sinyalleri

Şekil 2.2'de gösterildiği gibi gerçek motordan alınan krank ve kam sinyalleri birbiriyle senkronize olarak şekildeki gibi devam etmektedir. Krank milinin 2 turuna karşılık (720°) kam mili 1 tur (360°) atmaktadır. Şekil 2.3'de kam mili ve sensörü gösterilmektedir.



Şekil 2.3 Kam Mili ve Sensörü (Segment Speed Sensor)

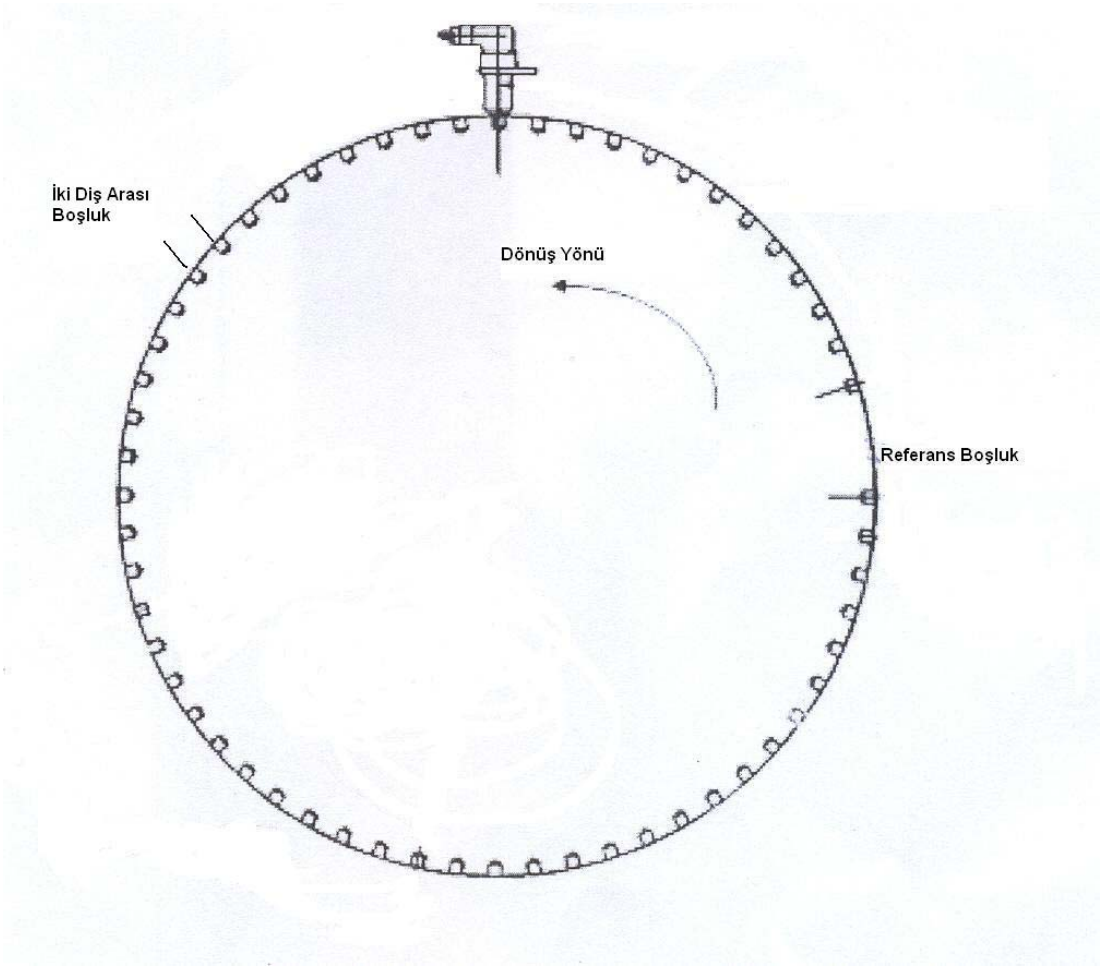
Kam diş sayısı motorun silindir sayısına bağlı olarak değişmektedir. Aşağıda kam diş sayısı ve açısının silindir sayısına bağlı olarak değişimini gösteren formül ve tablo gösterilmektedir. (z: silindir sayısı)

Kam Diş Sayısı = z + 1 (2.1)

Diş Açısı (Segment Angle) = $\varphi_s = \frac{360}{z} \pm 0.1°$ (2.2)

Tablo 2.1 Silindir Sayısına Göre Açısal Boşluk Miktarı

Silindir Sayısı (z)	4, 5, 6	8
Açısal Boşluk Miktarı (°)	15°	12°

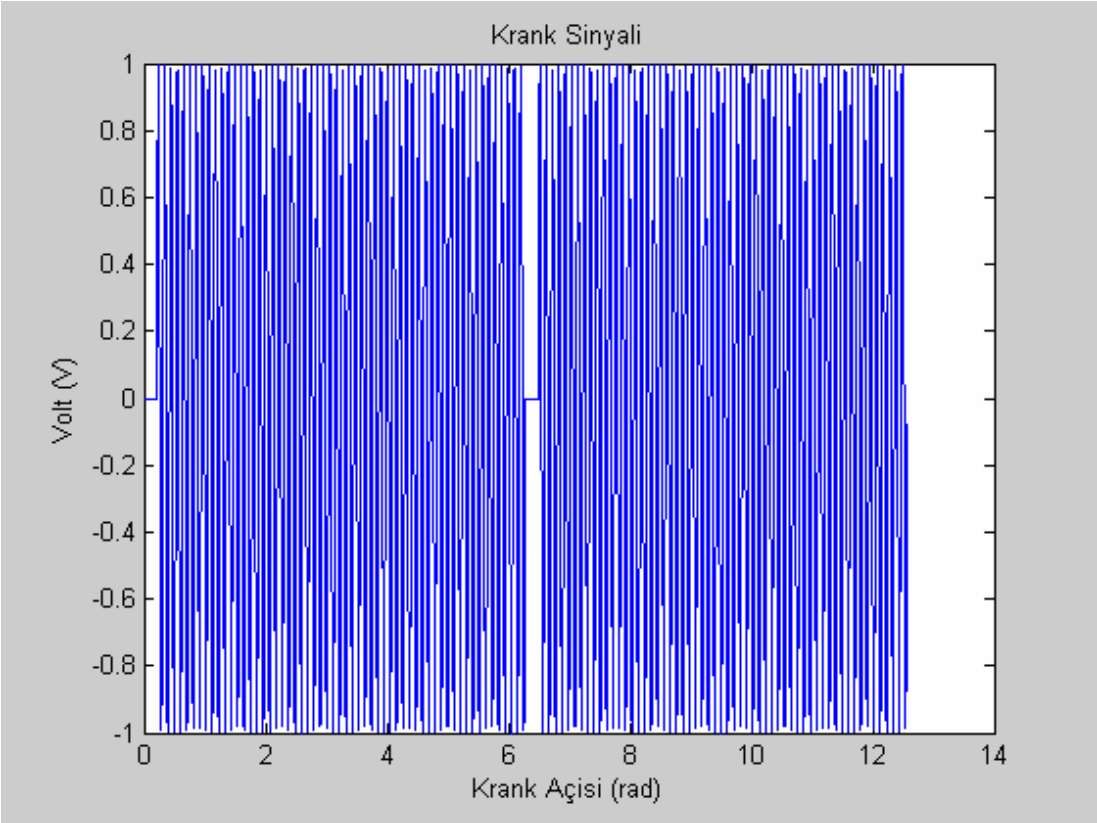


Şekil 2.4 Krank Mili ve Sensörü (Increment Speed Sensor)

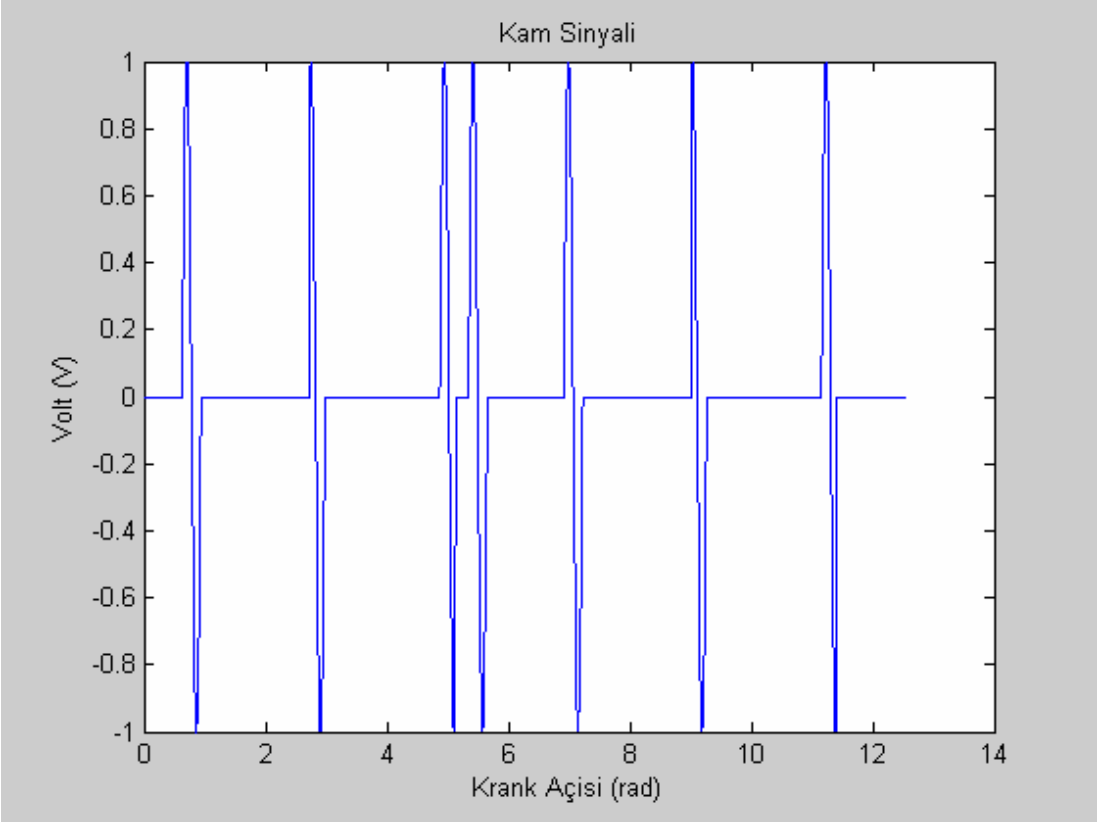
Şekil 2.4'de krank mili ve sensörü görülmektedir. Krank mili 58 diş ve 2 boşluktan oluşmaktadır. Krank mili hız sensörü 2 boşluğu referans alacak şekilde çalışmaktadır.

Bu tez çalışmasında gerekli olan krank ve kam sinyalleri yukarıdaki esaslara ve gerçek motor verisi gözönüne alınarak hazırlanan bir programla (Matlab m-file) oluşturuldu. Tez kapsamında oluşturulan modelde kullanılan krank ve kam sinyalleri verileri için 2 farklı program denendi ve test edildi. Programlardan biri krank ve kam sinyallerini krank açısı'na, diğeri ise motor devir sayısına (rpm) göre oluşturdu.

Şekil 2.5'de krank açısı'na bağlı krank sinyali, Şekil 2.6'da krank açısı'na bağlı kam sinyali gösterilmektedir. Benzer olarak 2500 rpm motor devri için zamana bağlı krank ve kam sinyalleri sırasıyla Şekil 2.7'de ve Şekil 2.8'de gösterilmiştir.



Şekil 2.5 Krank Açısı'na Bağlı Krank Sinyali



Şekil 2.6 Krank Açısı'na Bağlı Kam Sinyali

2.3 Motor Devrinden Dolayı Olan Gereksinimler

Bu tezde motor devrinin krank ve kam sinyalleri ile üretilmesi bazı gereksinimleri ortaya çıkarmıştır. Krank sinyali oldukça hızlı bir sinyaldir. Krank hızı sensörü her krank turu için 58 tepe noktası ve 2 boş nokta üretmektedir. Gerekli hızda kartı seçebilmek için bazı hesaplamalar gerekmektedir.

Motorun bir devrine karşılık gelen zaman Δt_{60}'ın birimi saniye/devir'dir ve

$$\Delta t_{60} = 60 / \omega \tag{2.3}$$

olarak hesaplanır. Burada ω motorun dev/dak birimindeki devir sayısıdır.

Şekil 2.7 ve Şekil 2.8'de de görüleceği gibi, eğer 2500 dev/dak için hesaplanırsa bir krank devrinin Δt_{60} = 0.024 sn, 2 krank turunun ise 0.048 sn süreceği bulunur.

Örnekleme frekansını ise 2.4 ve 2.5'deki denklemler ile hesaplamak mümkündür.

Δt : 1 krank sinyal zamanı

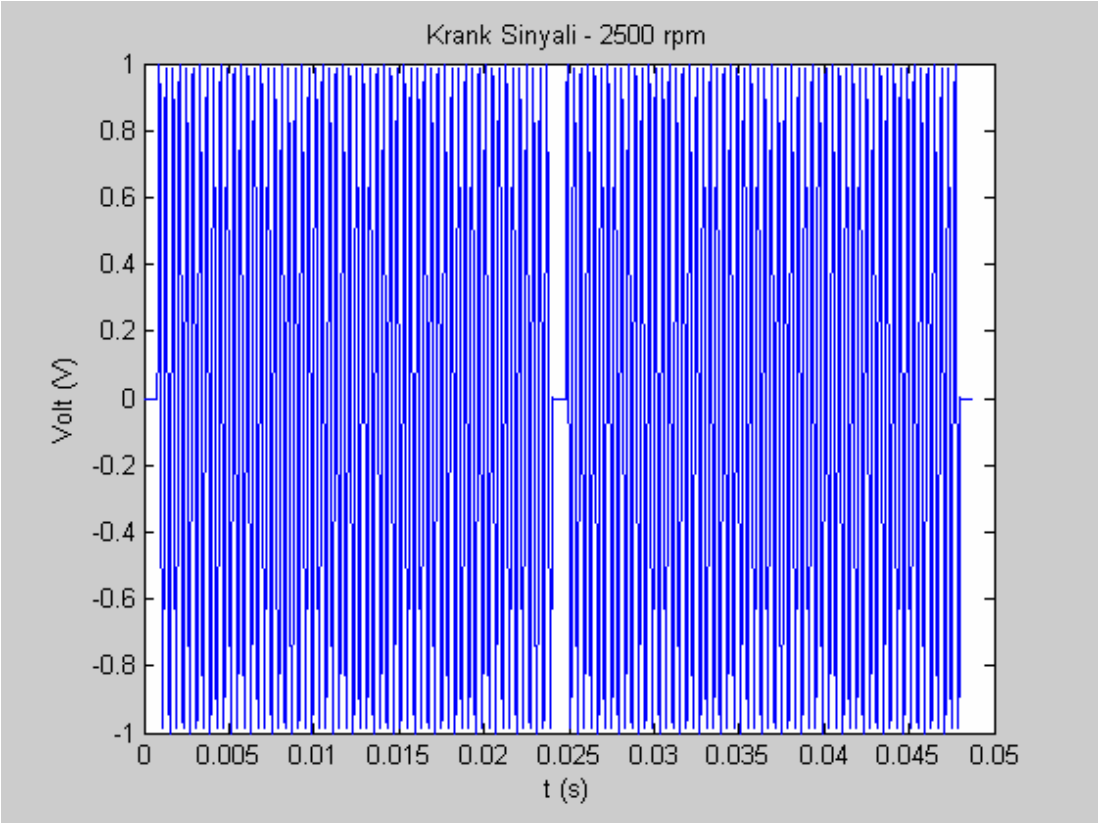
$$\Delta t = \Delta t_{60} / 60 \tag{2.4}$$

$$Örnekleme\ Frekansı\ (Hz) = 4 * (1 / \Delta t) \tag{2.5}$$

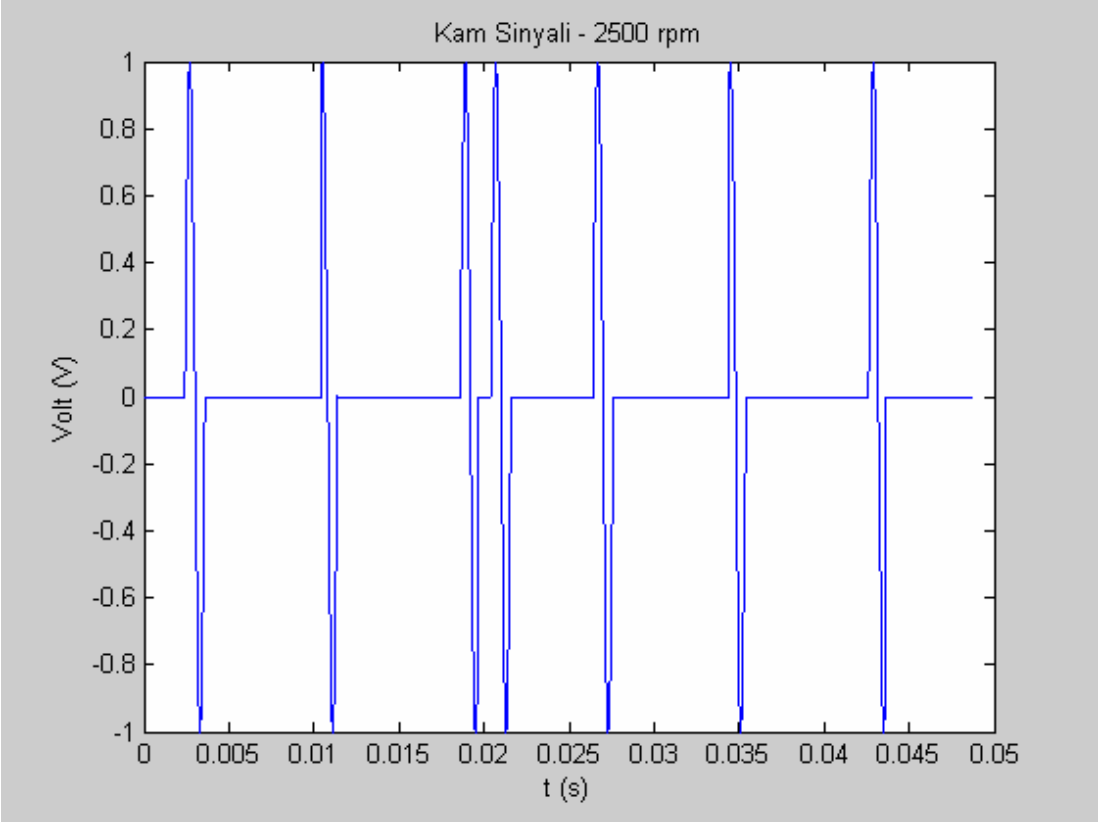
Örnekleme frekansı hesaplanırken kullanılan "4" çarpım faktörü sinüs sinyali oluşturulurken asgari 4 nokta kullanıldığını göstermektedir. Bu asgari bir yaklaşımdır ve bir sinüs tepeyi ve vadiyi yakalamayı amaçlar. Pratikte bu çarpım faktörünün daha yüksek değerlerinin kullanılması gereklidir.

Bulunan örnekleme frekansı kullanılacak olan uygun veri aktarım kartının seçilmesini sağlayan faktörlerden biridir.

Aşağıda bazı motor devirleri için yapılan hesaplar ve şekiller gösterilmektedir.



Şekil 2.7 RPM'e Bağlı Krank Sinyali (2500 rpm)



Şekil 2.8 RPM'e Bağlı Kam Sinyali (2500 rpm)

100 rpm için:

$$\Delta t_{60} = 60/100 = 0.6\ sn$$
$$\Delta t = 0.6/60 = 0.01\ sn$$
$$Örnekleme\ Frekansı = 4*(1/0.01) = 400\ Hz.$$

olarak bulunur.

Bir krank sinyali bir sinüs peryodunda 10 nokta kullanmaya karar verilirse

$$Örnekleme\ Frekansı = 10*(1/0.01) = 1000\ Hz = 1\ kHz$$

olarak bulunur. Görüldüğü gibi bu ikinci durumda 100 rpm'lik motor devrine ulaşmak için 1 kHz analog çıkış hızına sahip olan bir kart ile çalışmak gerekmektedir.

Maksimum motor devrinin 5000 rpm olduğu düşünülürse:

$$\Delta t_{60} = 60/5000 = 0.012\ sn$$
$$\Delta t = 0.012/60 = 0.0002\ sn$$
$$Örnekleme\ Frekansı = 4*(1/0.0002) = 20000\ Hz = 20\ kHz$$

Bir krank sinyali sinüs peryodunda 10 nokta kullanmaya karar verilirse

$$Örnekleme\ Frekansı = 10*(1/0.0002) = 50000\ Hz = 50\ kHz$$

olarak bulunur.

Yapılan hesapları tersten de yapmak mümkündür. Örnek olarak proje kapsamında testlerde kullanılan 10 kHz'lik analog çıkış hızına sahip olan NI PCI-6025E kartı verilebilir.

$$\Delta t = 4/Örnekleme\ Frekansı = 4/10000 = 0.0004\ sn$$
$$\Delta t_{60} = \Delta t*60 = 0.0004*60 = 0.024\ sn$$
$$\omega_{\max.} = 60/\Delta t_{60} = 60/0.024 = 2500\ rpm$$

olarak bulunur. Bu raporun ilerki bölümlerinde NI PCI-6025E kartı ile yapılan testlerde 2500 rpm'e çıkılabildiği gösterilmektedir.

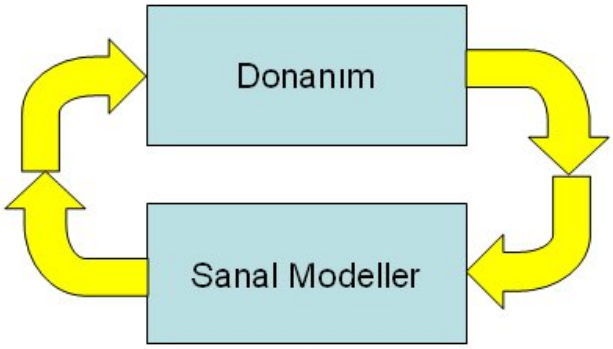
Bu bölümde ECU'nun motordan aldığı sinyaller hakkında bilgi verilmiş, gerçek zamanlı simülasyonlar için krank ve kam sinyallerinin oluşturulmasıyla ilgili gereksinimler anlatılmıştır.

Üçüncü bölümde Real-Time Windows Target ile National Instruments firmasının PCI 6025E kartı kullanılarak yapılan gerçek zamanlı simülasyonlar anlatılmaktadır.

3. REAL-TIME WINDOWS TARGET İLE YAPILAN GERÇEK ZAMANLI DONANIM İÇEREN SİMÜLASYONLAR

3.1 Giriş

Donanım içeren simülasyonlar (HILS, hardware-in-the-loop-simulations), araca ait bir donanımın, birlikte çalıştığı diğer parçalarının bilgisayarda hazırlanmış sanal modelleriyle gerçek zamanlı olarak çalıştırılıp, test edildiği simülasyonlardır (bkz. Şekil 3.1).



Şekil 3.1 Donanım İçeren Simülasyon Oluşum Şeması

Donanım içeren simülasyon sistemleri, günümüzde sürekli olarak gelişen ve mühendislerin sık olarak kullanmaya başladığı sistemlerdir. Günümüzde zaman kaybı ve maliyeti azaltmak için donanım içeren simülasyonların kullanımı sürekli artmaktadır. Bu simülasyon sistemleri gerçek donanımları kullanmadan yapılan tasarımları test etme imkanı sunar. Böylece yeni kontrol algoritmaları geliştirilebilir veya, örnek olarak, motorun kendisinde meydana gelebilecek bir hata gözlemlenebilir.

Otomotiv uygulamalarında donanım içeren simülasyonlara örnek olarak motora ait elektronik kontrol ünitesinin (ECU) sanal motor modeli ile bağlanarak beraber çalışması verilebilir.

Bu tez kapsamında bazı PC kartları (NI ve dSPACE) ve xPC Target donanımları denenmiştir. Bu bölümde donanım içeren simülasyonlarda Real-Time Windows Target simülasyon ayarları anlatılmış ve yapılan simülasyonların sonuçları gösterilmiştir.

3.2 RTWIN (Real-Time Windows Target)

Real Time Windows Target, gerçek zamanlı sistemlerin test edilebilmesi için kullanılabilen bir PC çözümüdür. Simulink bloklarıyla veya Stateflow diyagramlarıyla oluşturulan modellerin MATLAB ortamında gerçek zamanlı test edilmesini sağlar. Modeller Matlab/Simulink ortamında normal modda oluşturulduktan sonra build edilir ve external modda model çalıştırılır. Build işlemi ile modelin 'C' kodu üretilir (Real-Time Workshop) ve derlenir. Bu işlemlerden önce kullanılacak C derleyicinin yüklenmesi gerekir. Uygun olan C derleyici Matlab komut penceresinde mex –setup yazılarak seçilir ve yüklenir.

Real Time Windows Target, gerçek zamanda kurulan modeli çalıştırabilmek için Real-Time Kernel' i kullanır.

Real–Time simülasyon yapabilmek ve gerekli bağlantıları kurmak için bazı donanım (hardware) ve yazılımlara (software) ihtiyaç duyulmaktadır. İstenen kontrol algoritmasının ve dinamik sistemin tasarlanması için Matlab / Simulink gibi bir program kullanılabilir. Gerçek zamanlı yazılım arayüzleri (Software Interfaces), Simulink altında kullanılan Real–Time Windows Target Toolbox, xPC Target Toolbox veya alınan PCI (ISA) kartının firmasının sunduğu yazılımlar bu amaç için kullanılabilir. Daha sonra da Real–Time simülasyonu yapılacak sistemin manuel olarak veya mevcut programlar kullanılarak uygun C koduna çevrilmesi gerekmektedir. Bu sayede kartın üzerine sistem bilgileri aktarılabilir. Bu noktada kullanılacak kartın özellikleri simülasyon için oldukça büyük önem taşımaktadır. Kart işlemcisinin performansı, I/O kanal sayısı, çözünürlüğü (kanallara sunduğu bit

sayıları) ve bellek miktarı yapılacak bağlantılar ve simülasyon parametreleri için belirleyici rol oynamaktadır.

Bu tezde Real-Time Windows Target uygulamasında National Instruments firmasının NI PCI-6025E kartı kullanılmıştır (bkz. Şekil 3.2).

Family	Bus	Analog Inputs	Input Resolution	Max Sampling Rate	Input Range	Analog Outputs	Output Resolution	Output Rate	Output Range	Digital I/O	Counter/Timers	Triggers
NI 6025E	PCI, PXI	16 SE/8 DI	12 bits	200 kS/s	±0.05 to ±10 V	2	12 bits	10 kS/s	±10 V	32	2, 24-bit	Digital



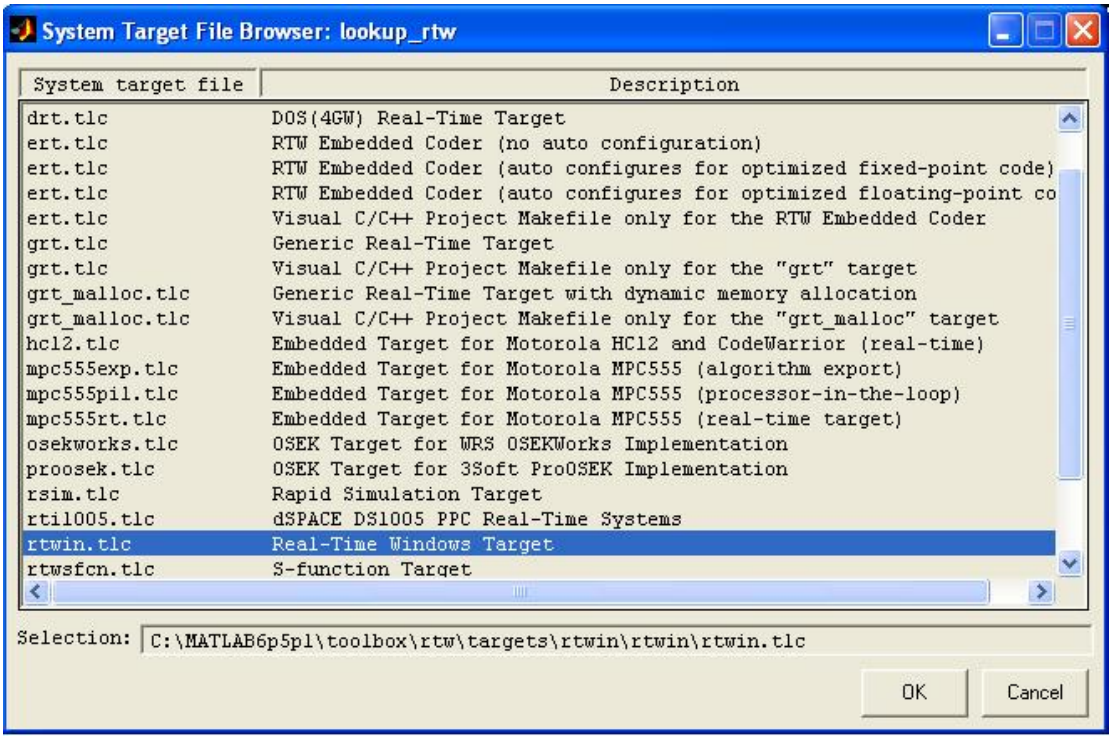
Şekil 3.2 NI PCI-6025E Kartı ve Özellikleri [31]

Şekil 3.2'de görülen NI PCI-6025E kartının pin ve blok diyagramları EK-A'da gösterilmektedir.

3.3 Real-Time Windows Target Simülasyon Ayarları

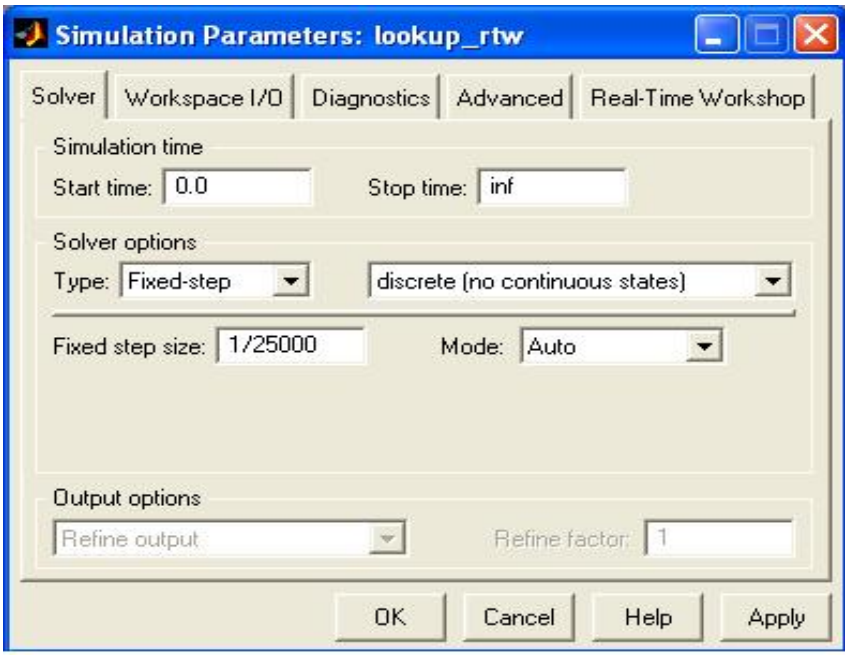
Matlab programının bir parçası olan "Real Time Workshop" bölümünde daha önceden belirlenmiş hedefler (target) için gerçek zamanlı çalışabilecek kodlar oluşturulabilir. Eğer yardımcı bir donanım kullanımı söz konusu ise, bu donanıma ait hedef seçilerek kod, bu donanıma ait dil için oluşturulmalıdır. Şekil 3.3'de Matlab'e ait Real Time Workshop bölümünde tanımlanmış hedeflerin bir kısmı görülmektedir.

Ek olarak Şekil 3.3'de Real-Time Windows Target ile yapılacak simülasyonlarda Real Time Workshop tanımlı hedefler menüsünden Real-Time Windows Target seçildiği de gösterilmektedir.

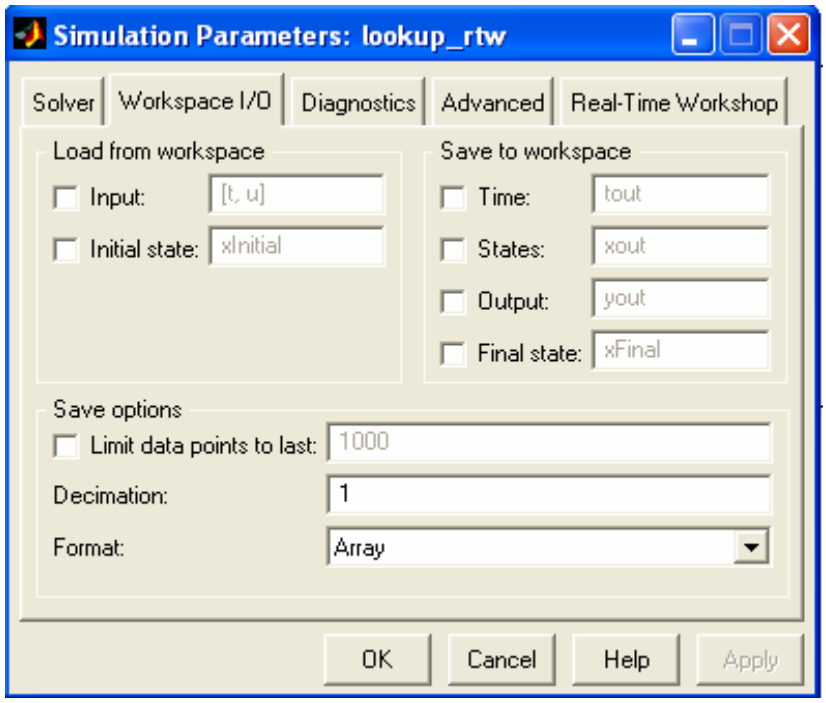


Şekil 3.3 Real Time Workshop Bölümünde Tanımlı Hedefler (target)

Simülasyon parametrelerinden Şekil 3.3'de görülen tanımlamayı yaptıktan sonra oluşturulan modelde istenilen diğer ayarlar da yapılır ve simülasyon başlatılır. Şekil 3.4 ve Şekil 3.5'de simülasyon parametrelerinde yapılan bazı ayarlar gösterilmektedir. Çözücü seçeneği için discrete seçmek zorunlu değildir.



Şekil 3.4 Simülasyon Parametrelerinin Ayarlanması-I

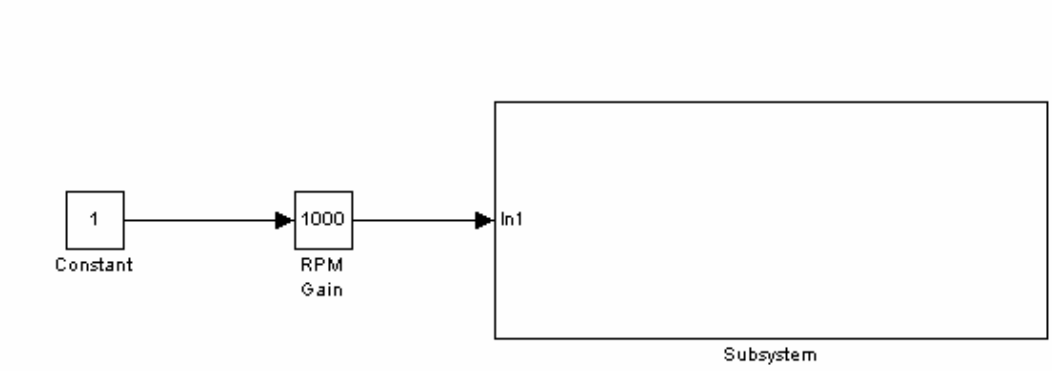


Şekil 3.5 Simülasyon Parametrelerinin Ayarlanması-II

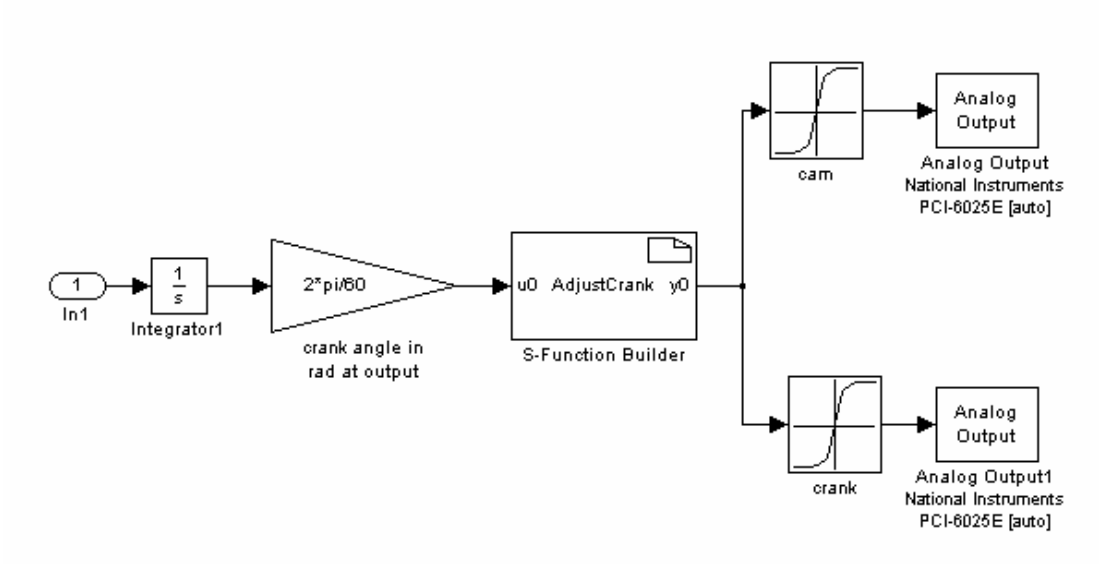
Yukarıda gösterilen simülasyon parametreleri ve diğer ayarlar yapıldıktan sonra model gerçek zamanlı olarak çalıştırılır.

3.4 NI PCI-6025E Kartı ile Yapılan Testler

Bu tezde, ilk olarak NI firmasının PCI-6025E kartı kam ve krank sinyalleri gönderimi için kullanılmıştır. Kart'ın 2 analog çıkışı olması nedeniyle testler sadece 2 analog sinyalle yapılmıştır. Şekil 3.6 ve Şekil 3.7'de krank ve kam sinyallerinin oluşturulduğu model ve alt sistemi görülmektedir.



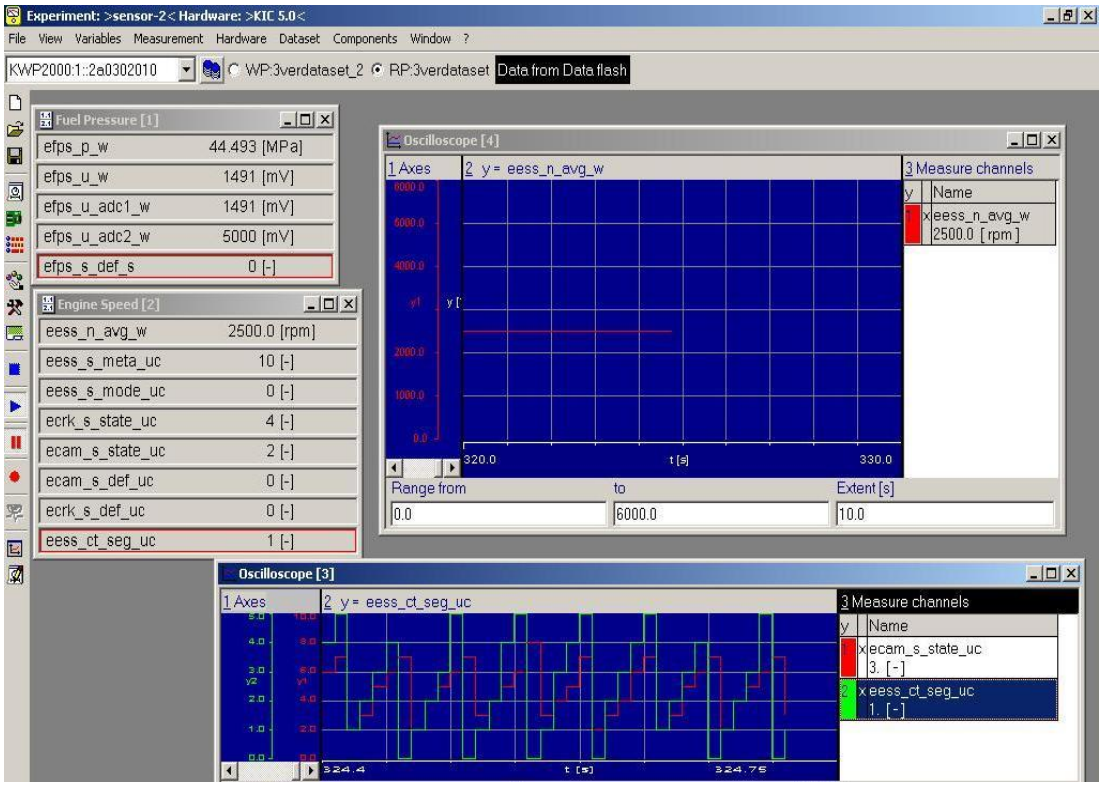
Şekil 3.6 Modelin Genel Görünüşü



Şekil 3.7 Alt sistem

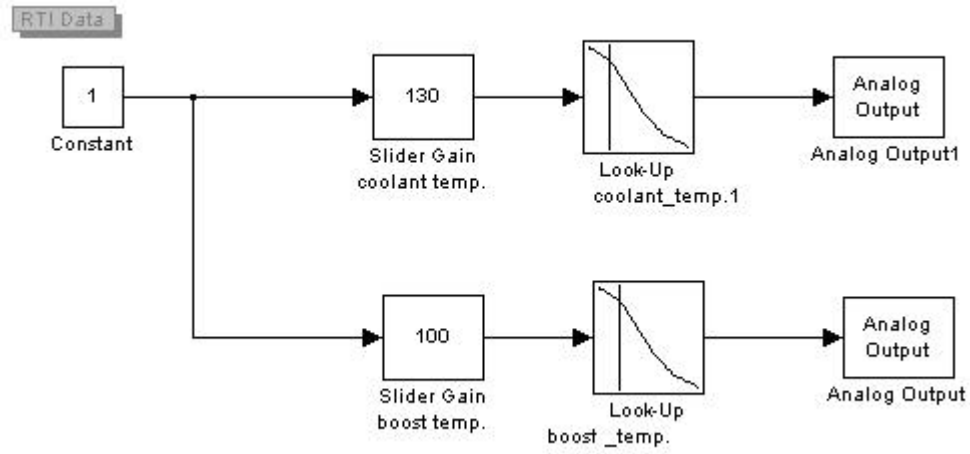
Model, krank açısına bağlı olarak oluşturulan kam ve krank sinyal verilerinin senkronize olarak çalışması amacıyla kurulmuş ve modelin gerçek zamanda istenilen 'motor devri' (rpm) değerleri ile çalışması amaçlanmıştır.

Şekil 3.8'de modelin gerçek zamanda çalıştırıldığı anda ölçüm ekranının aldığı değerler gösterilmektedir.



Şekil 3.8 Ölçüm Ekranı (2500 rpm)

Motor devrinin istenen şekilde değiştirildiği model ölçüm ekranı ile kontrol edilmiştir. National Instruments kartının 2 analog çıkışa sahip olması nedeniyle motor devri ve diğer sensör sinyalleri ayrı ayrı kontrol edilmiştir. Şekil 3.9'da soğutma suyu (coolant) ve aşırı doldurma sistemi (boost) sıcaklık sinyalinin oluşturulduğu model gösterilmektedir.



Şekil 3.9 Sıcaklık Test Modeli

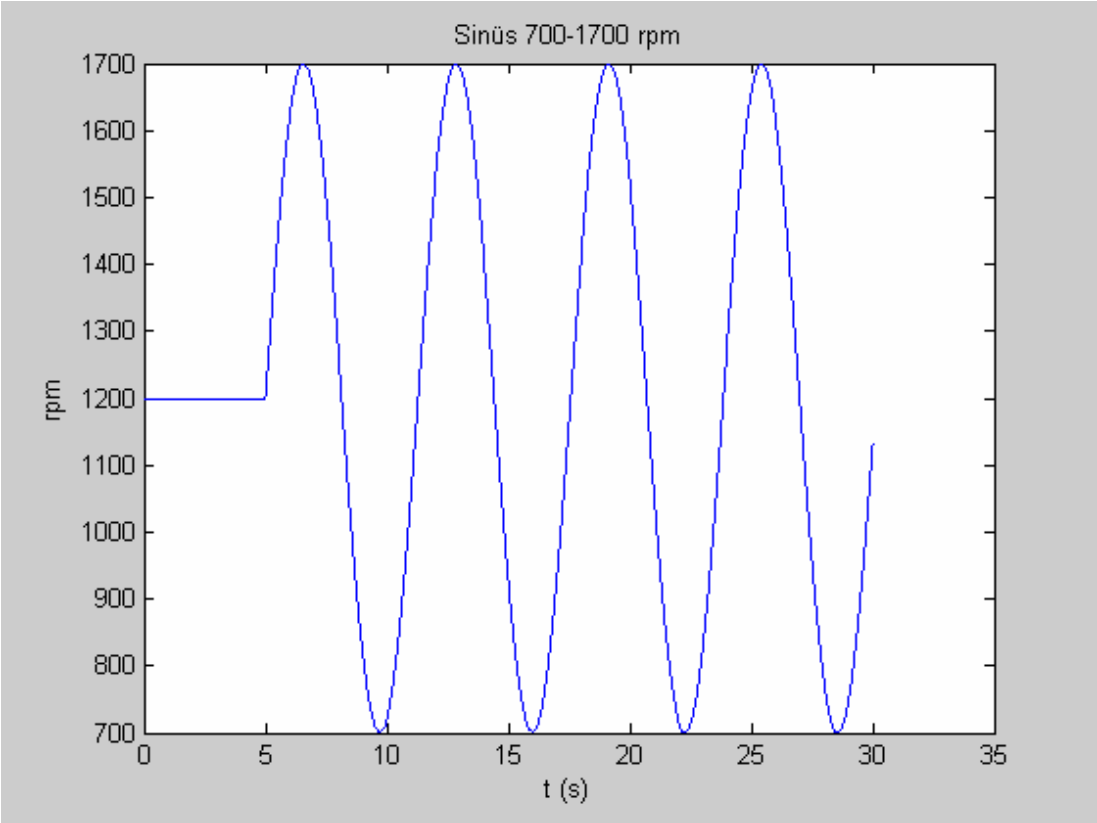
Şekil 3.9'daki model NI PCI-6025E kartı kullanılarak Real-Time Windows Target'da çalıştırılıp test edilmiştir.

Yapılan testler ile çalışma voltaj aralığı tespit edildikten sonra gerekli sıcaklık-voltaj tabloları look-up table'larda oluşturulmuştur. Böylece istenilen sıcaklık değerlerinin girilebildiği tablolar hazırlanmıştır.

Yukarıda bahsedilenlere ek olarak NI PCI-6025E kartı ile motor devrinin sabit olmadığı testler de yapılmıştır. Bu testler "sinüs, rampa, gerçek veri, MVEM (mean value engine model)" den oluşmaktadır.

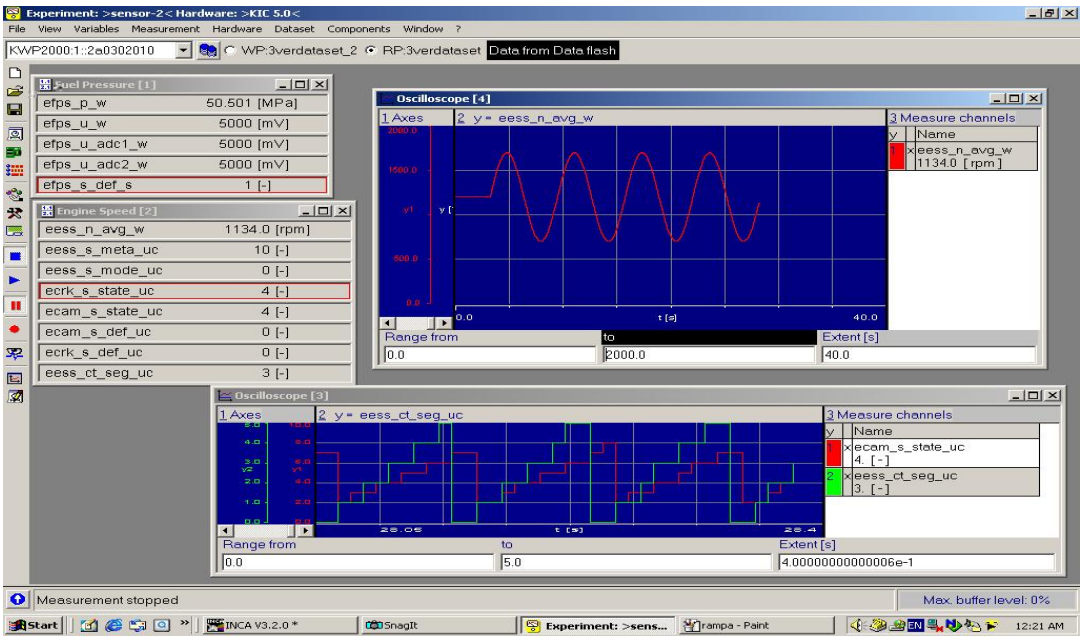
- Sinüs (700 – 1700 RPM)

Motor devri girişi için Şekil 3.10'da görülen veri oluşturulmuş ve motor devrinin ölçüm ekranından istenilen veri şeklinde sonuç verdiği gözlenmiştir.



Şekil 3.10 Oluşturulan Sinüs Verisi

Şekil 3.10'da görüldüğü gibi motor devrinin 1200 rpm ile başlayıp bir süre sabit devam ettikten sonra 700-1700 rpm genlikli sinüs oluşturması amaçlanmıştır. İstenilen sonucun alındığı ölçüm ekranı Şekil 3.11'de gösterilmektedir.



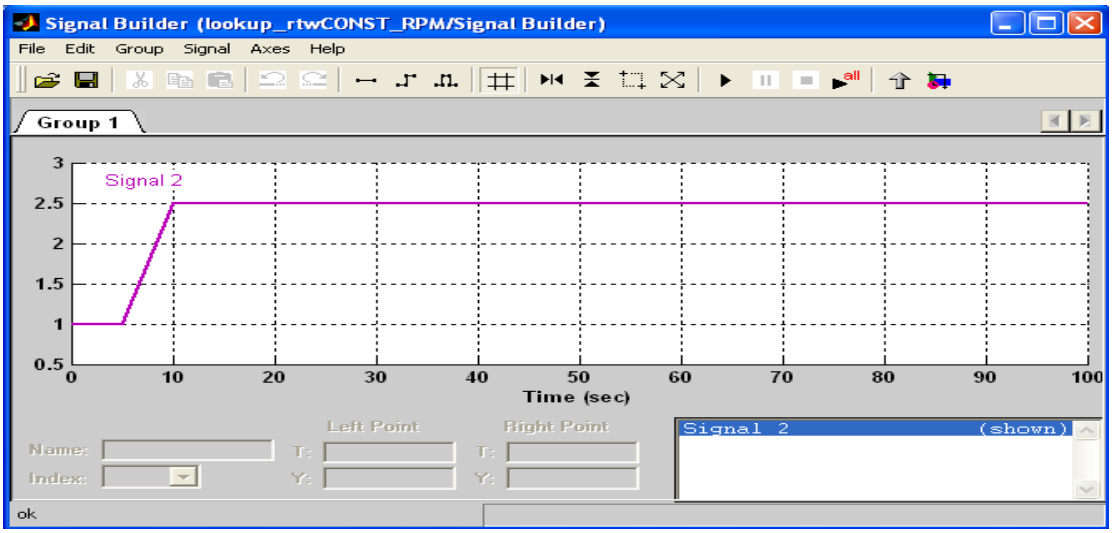
Şekil 3.11 Ölçüm Ekranından Sinüs Veri Görüntüsü

- Rampa (1000 – 2500 RPM)

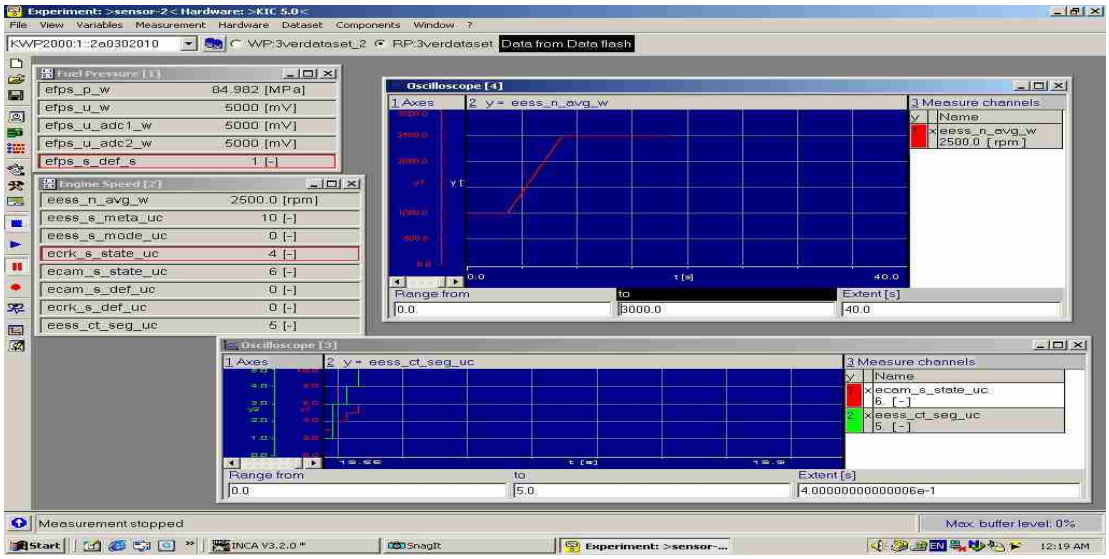
Motor devri girişi için Şekil 3.12'de görülen rampa verisi oluşturulmuş ve motor devrinin ölçüm ekranından istenilen veri şeklinde sonuç verdiği gözlenmiştir.

Rampa verisi, motor devrinin 5 saniye 1000 rpm'de kaldıktan sonra 10. saniyede 2500 rpm'e ulaşması ve bu rpm'de sabit olarak devam etmesi şeklinde hazırlanmıştır.

Şekil 3.13'te rampa verisi ile çalıştırılan modelin ölçüm ekranında gözlenen rampa verisi sonucu gösterilmiştir.



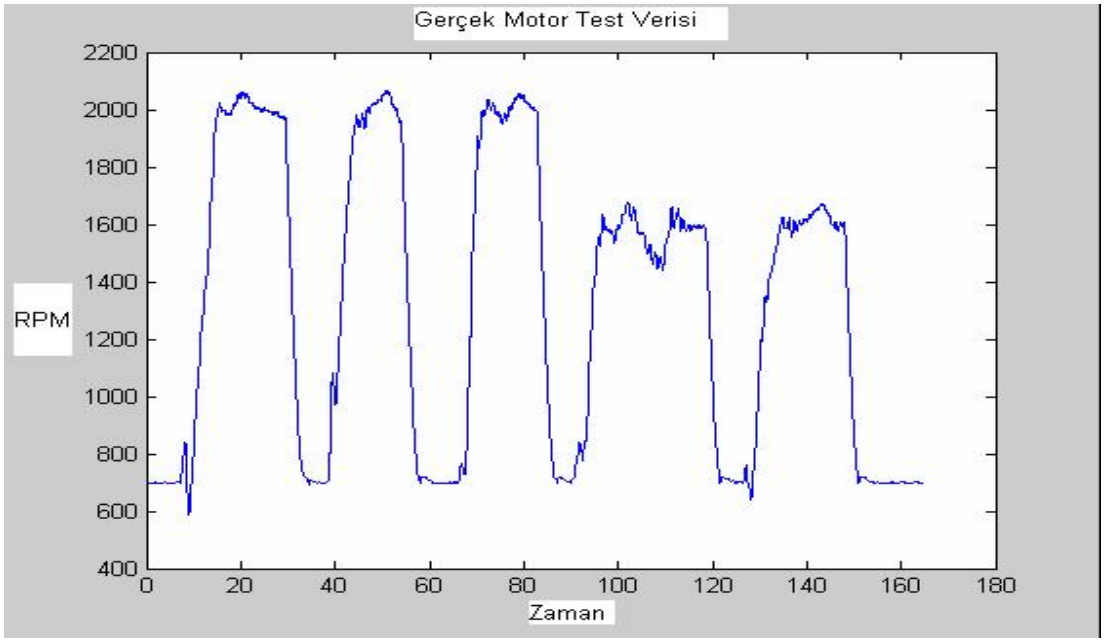
Şekil 3.12 Sinyal Oluşturucu ile Oluşturulan Rampa Verisi



Şekil 3.13 Ölçüm Ekranından Rampa Görüntüsü

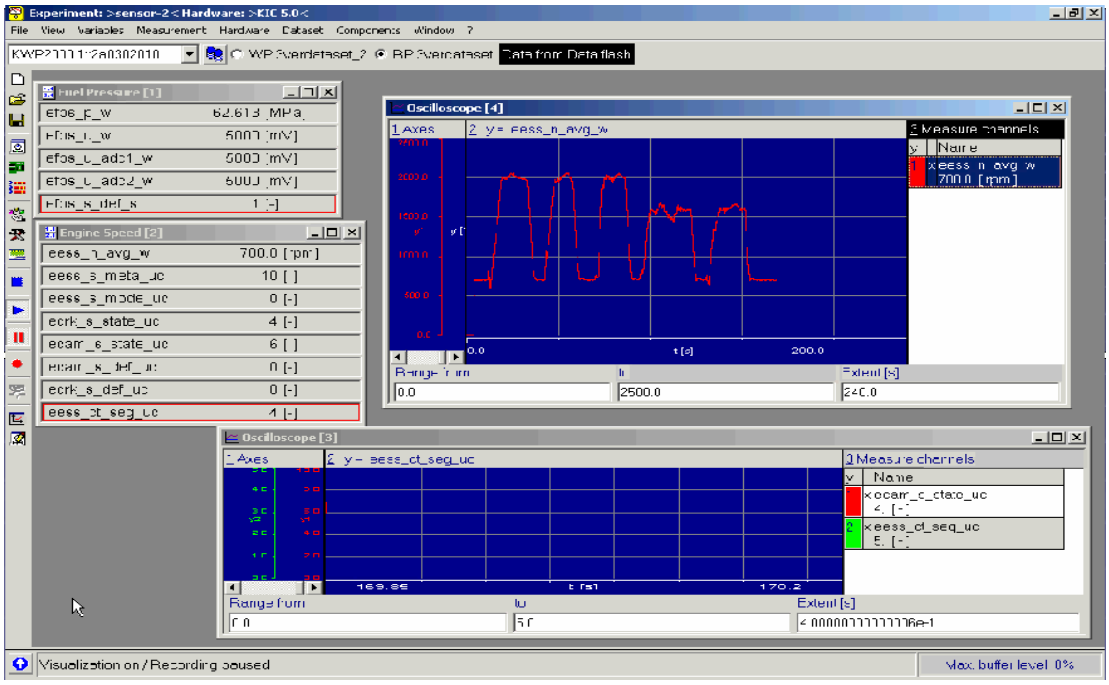
- Gerçek Veri İle Yapılan Test

Bu testte gerçek motordan alınan veri kullanılmıştır. Alınan veri Şekil 3.14'de gösterilmektedir. Burada tip-in, tip-out testleri yapılmıştır.



Şekil 3.14 Gerçek Motor Test Verisi

Önceki testlerde olduğu gibi motor devrinin davranışı ölçüm ekranı aracılığı ile gözlenmiştir. Bu ekran Şekil 3.15'de gösterilmektedir.

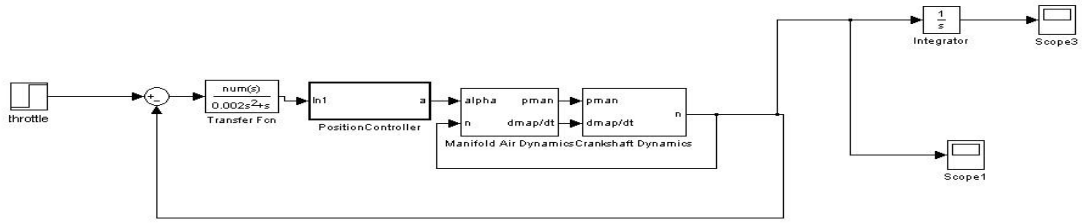


Şekil 3.15 Ölçüm Ekranı Gerçek Veri Test Görüntüsü

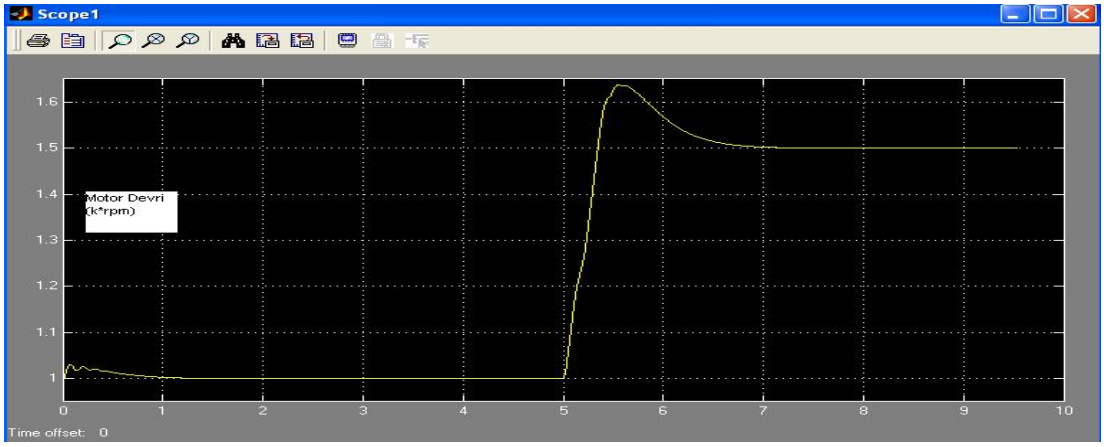
- MVEM (Mean Value Engine Model) ile Yapılan Test

Bu tezde oluşturulan modelin karmaşık bir motor modeli ile beraber çalışabildiği göstermek amacı ile bir MVEM kullanılmıştır. MVEM "spark ignition" dediğimiz benzinli motorlar için kullanılırlar. Dizel motorların modellenmesinde de kullanılabilmektedirler. Bir MVEM modelin ECU ile test edilmesinin ana nedeni geliştirilen ECU test sisteminin pratikte kullanılan motor modelleriyle sorunsuz çalışabildiğini göstermek içindir. Daha ileriki çalışmalarda sisteme jenerik bir dizel motor modeli eklenmesi planlanmaktadır. En önemli motor alt sistemlerinin fiziksel modellerinden oluşan basitleştirilmiş dinamik motor modeli olan MVEM, temel olarak ana 3 alt sistemden oluşmaktadır. Bunlar; yakıt dinamiği, krank-mili dinamiği, manifold hava dinamiğidir. Bu MVEM modeli Aksun Güvenç vd. [32, 33] kaynaklarından alınmıştır. [33]'de bu MVEM modeli için bir hız kontrolcüsü de tasarlanmıştır.

Elde edilen hız kontrolü gerçek zamanda (Real–Time) da denenerek aslında konvansiyonel bir aracın Motor Kontrol Ünitesinde (ECU) davrandığı gibi davranması sağlanmış ve gözlenmiştir. Şekil 3.16'da MVEM modeli, Şekil 3.17'de ise simülasyonda elde edilen motor devri değişimi gösterilmiştir.

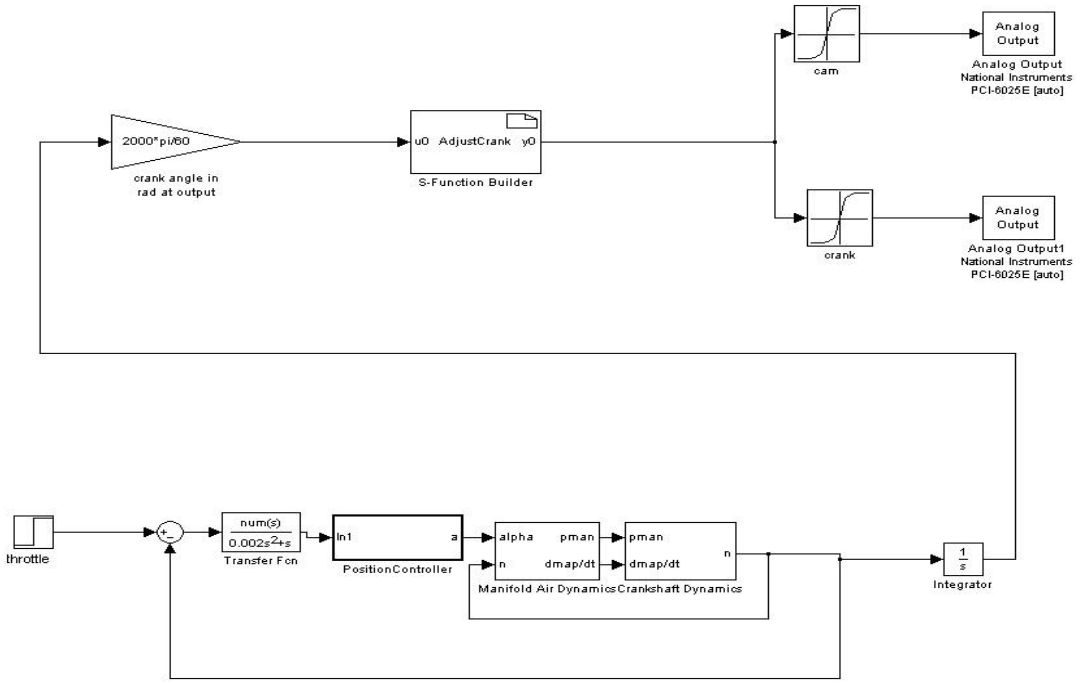


Şekil 3.16 MVEM (Mean Value Engine Model)

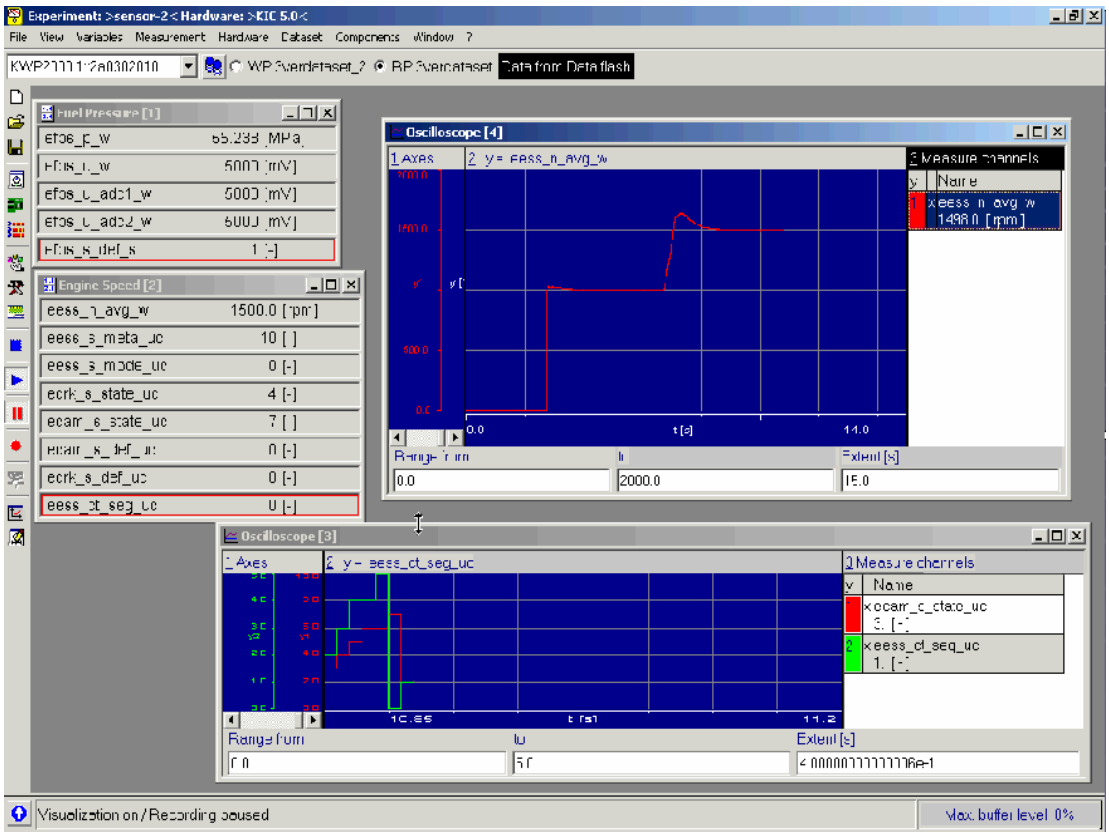


Şekil 3.17 MVEM ile Elde Edilen Motor Devri Cevabı

Şekil 3.18'de oluşturulan motor devri modeli ile MVEM, Şekil 3.19'da ise ölçüm ekranından alınan sonuç gösterilmektedir



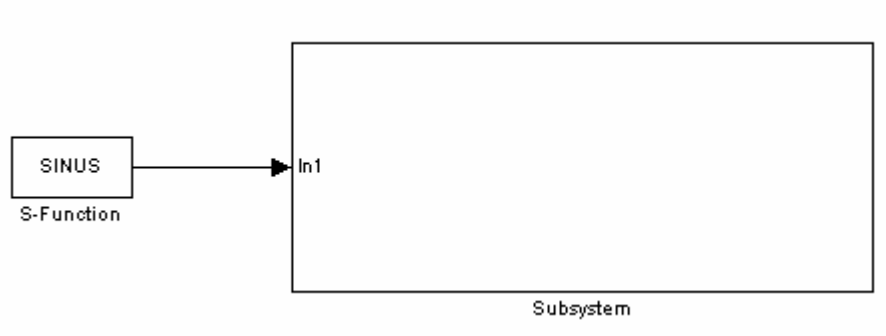
Şekil 3.18 MVEM ile Oluşturulan Motor Devri Test Modeli



Şekil 3.19 Ölçüm Ekranından MVEM Motor Devri Sonucu

Yukarıda anlatılan testlere ek olarak Matlab'de istenilen motor devrinin oluşturulduğu S-function'lar yazılarak testler yapılmıştır.

Şekil 3.20'de motor devri verisinin S-function ile oluşturulduğu model gösterilmektedir.



Şekil 3.20 S-Function ile Motor Devri Oluşturulan Model

Tez kapsamında yapılan testlerde kullanılan S-function'lardan biri EK-B'de verilmiştir.

Bu bölümde NI PCI-6025E kartı ve Real-Time Windows Target ile yapılan çalışmalar anlatılmıştır. Bu kart, ECU test sistem gereksinimleri için yeterli analog girişe sahip değildir ve yüksek motor devir sayılarını simüle edememektedir.

Dördüncü bölümde ise daha hızlı bir kart olan dSPACE firmasının DS1103 kartı ile yapılan çalışmalar anlatılmaktadır.

4. DSPACE SİSTEMLERİ VE YAPILAN ÇALIŞMALAR

4.1 Giriş

dSPACE, gerçek zamanlı donanım içeren simülasyonlarda kullanılan kartları, panelleri üreten firmalardan bir tanesidir. Günümüzde ECU için hazırlanan sistemler sürekli gelişmekte olup, dSPACE firması da ECU testleri için özel simülatörler üretmektedir. Bu testler için özel olarak hazırlanan kartları, çeşitli boyutlarda, ihtiyaçlara bağlı olarak değişen simülatörleri üretmektedirler. Bunlara örnek olarak DS1005 ve DS2210 kartları verilebilir. Ayrıca bu kartların geliştirilen modelleri DS1006 ve DS2211 kartları da ECU testleri için üretilmektedir. Motor elektronik kontrol ünitesi testleri için hazırlanan bu çift kartlı sistemlerde model, esas işlemci kartı olan DS1005 (DS1006) üzerinde çalışır. dSPACE simülatörü ise ürettiği I/O sinyallerini DS2210 (DS2211) HIL I/O kartı yoluyla ECU'ya gönderir. ECU bağlantısı ayrı bir bağlantı paneli yoluyla yapılır. dSPACE simülator donanımı PC link kartları yoluyla bağlıdır (ISA, PCMCIA, veya PCI). Şekil 4.1'de dSPACE'in sunmuş olduğu bazı kartları, bağlantı panelini ve simülatörü gösterilmektedir.



Şekil 4.1 dSPACE Simülatörü, Kartları ve Bağlantı Panelleri [34, 35]

Bu tez kapsamındaki çalışmalarda dSPACE firmasının DS1103 PPC kontrolcü kartı kullanılmıştır. Bu bölümde bu kart ile yapılan simülasyonlar ve sonuçları anlatılmıştır.

4.2 DS 1103 ile Yapılan Çalışmalar

dSPACE DS1103 PC kartı, 400MHZ işlemci hızına sahip olup 16 bitlik 16 A/D kanalı, 12 bitlik 4 A/D kanalı ve 14 bitlik 8 D/A kanalına sahiptir. Şekil 4.2'de görülen DS 1103 PPC kontrolcü kartının pin ve blok diyagramları EK-C'de gösterilmektedir.



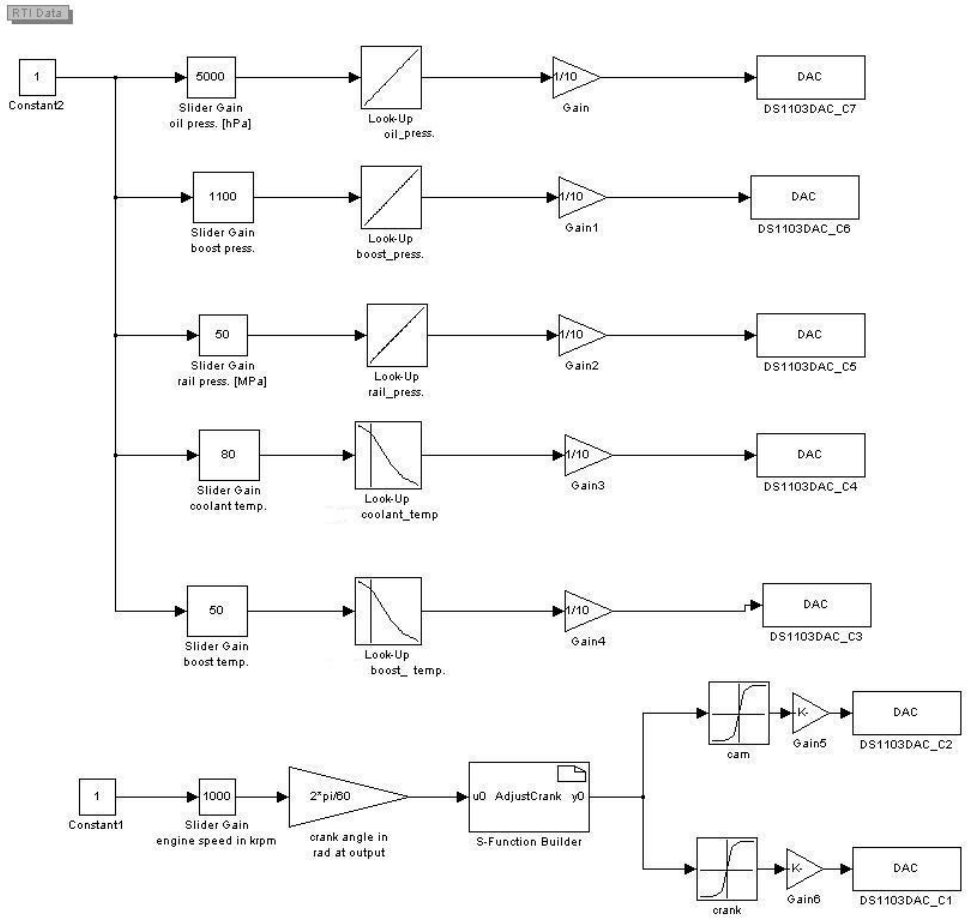
Şekil 4.2 dSPACE DS1103 PPC Kontrolcü Kartı [36]

Ayrıca gerekli I/O bağlantılarını yapabilmek için dSPACE firmasının LED'li bağlantı paneli de kullanılmıştır. Şekil 4.2 ve Şekil 4.3'de DS1103 kartı ve bağlantı paneli gösterilmektedir.



Şekil 4.3 dSPACE LED'li Bağlantı Paneli [37]

Bu tez kapsamında DS1103 ile yapılan testler, DS1103 kartının 8 analog çıkışa ve yeterli hıza sahip olması nedeniyle tüm analog sinyallerin bir arada yollanabileceği bir model oluşturularak yapılmıştır. Bu model Şekil 4.4'de gösterilmektedir.



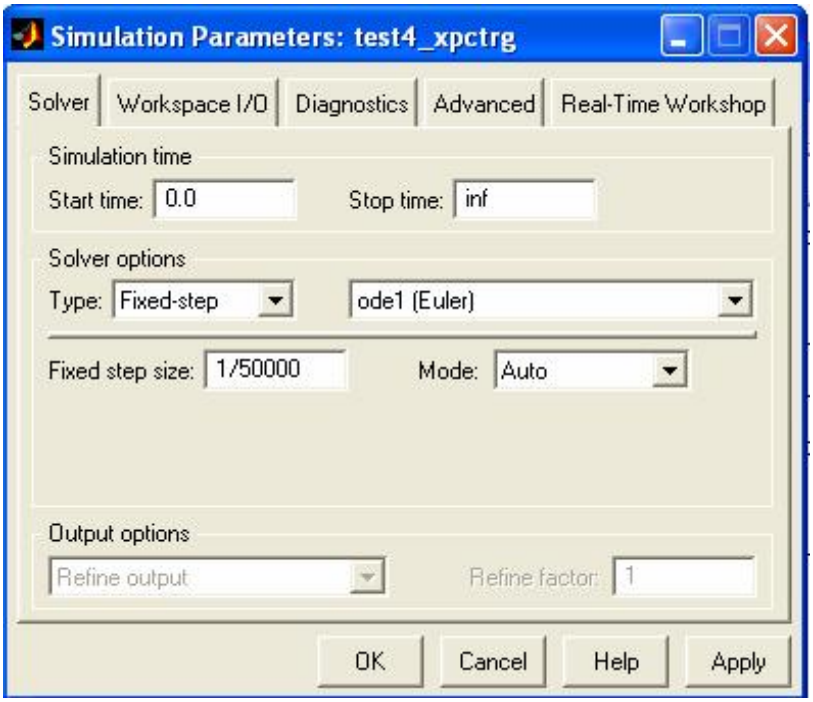
Şekil 4.4 Basınç, Sıcaklık ve Motor Devrinin Birlikte Oluşturulduğu Model

Şekil 4.4'de görüldüğü gibi dSPACE kartının analog çıkış sayısının yeterli olması sonucu basınç, sıcaklık ve motor devri için istenilen değerlerin girilebildiği model bir kerede oluşturulabilmektedir.

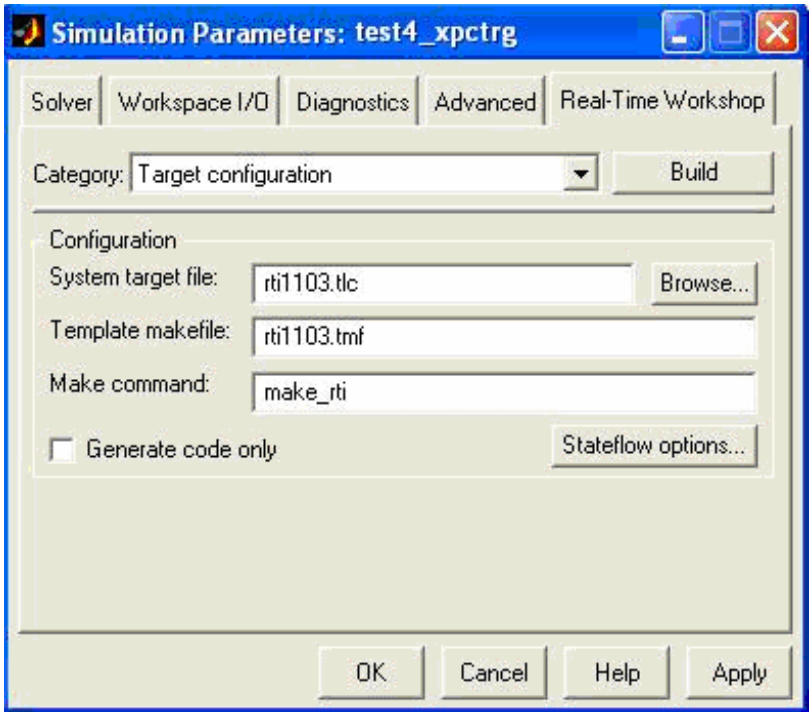
Oluşturulan modelin dSPACE kartı üzerinde çalışması için öncelikle dSPACE kartının Matlab programına tanıtılması gerekmektedir. Bu nedenle programın gerekli sürücüleri yüklenerek Matlab Programının kartı tanıması sağlanır. Böylece Matlab / Simulink altında dSPACE kartının blokları, kart üzerinde çalışacak C kodunun oluşması için alt menüler ve özellikler Simulink ortamına taşınmış olur. Bu işlemler gerçekleştikten sonra Simulink Modeli açılır ve buradan gerekli Real–Time simülasyon parametreleri ayarlanabilir.

Simülasyon parametreleri kartlar arasında farklılıklar göstermektedir. Özellikle çözücü metodu, kart için dikkatle seçilmelidir. Simülasyon zamanı da genellikle sonsuz olarak belirtilir. Ayrıca en önemli menülerden biri de Real–Time Workshop menüsüdür. Şekil 4.5 ve Şekil 4.6'da simülasyon parametreleri ayar menüsü ile

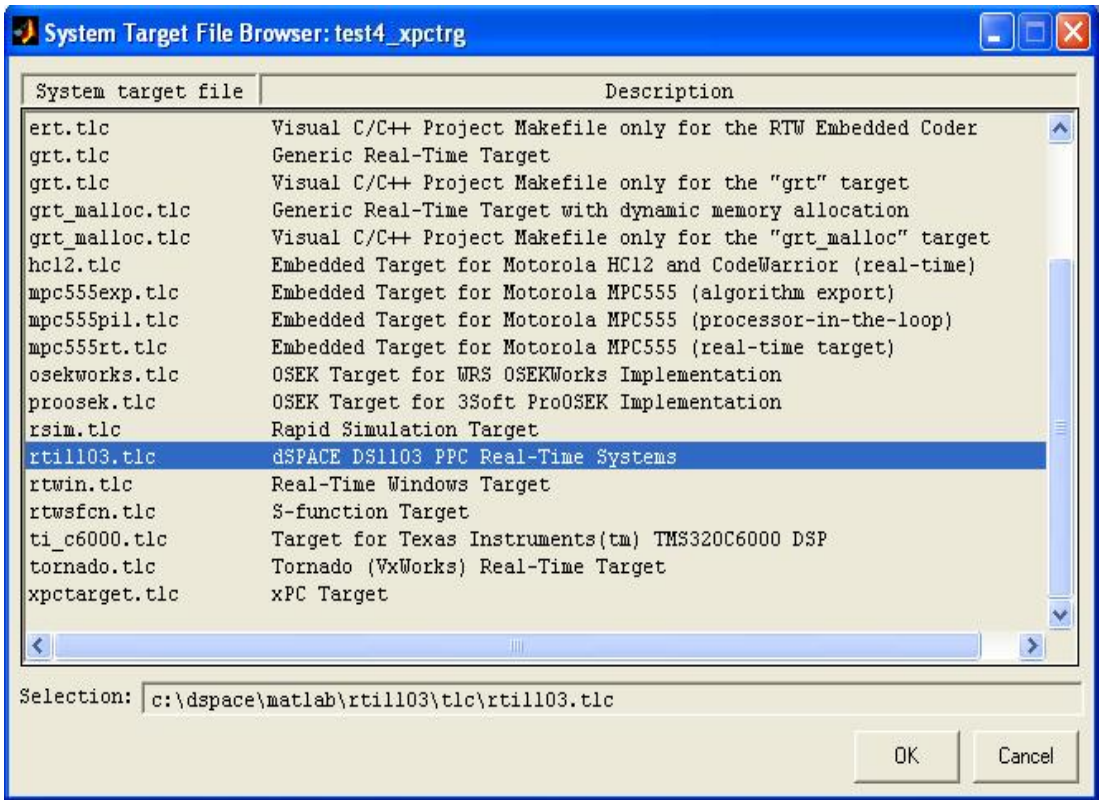
Real-Time Workshop menü ayarları gösterilmektedir. Ek olarak Şekil 4.7'de dSPACE sistemi ile yapılacak simülasyonlarda Real Time Workshop tanımlı hedefler menüsünden DS 1103 PPC kontrolcü kartının hedef olarak seçildiği de gösterilmektedir.



Şekil 4.5 Simülasyon Parametreleri Ayarları



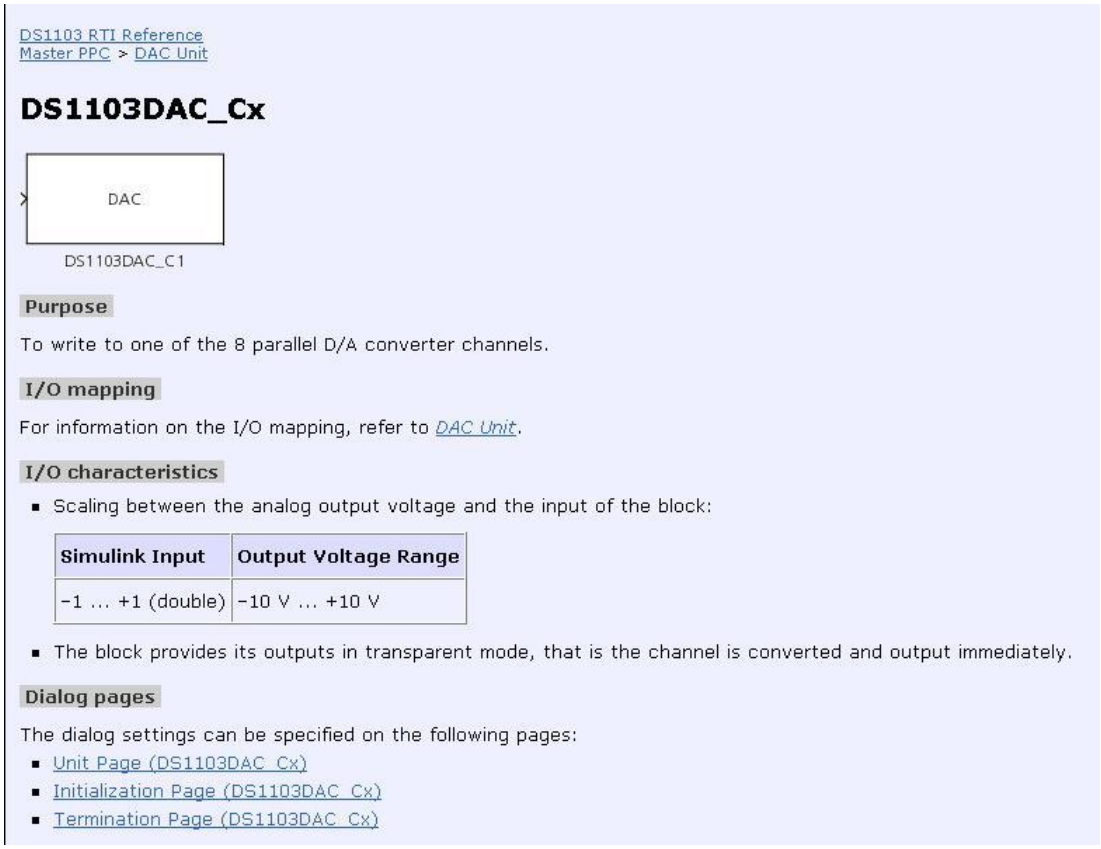
Şekil 4.6 Real-Time Workshop Menüsü



Şekil 4.7 Real Time Workshop Bölümünde Tanımlı Hedefler (target)

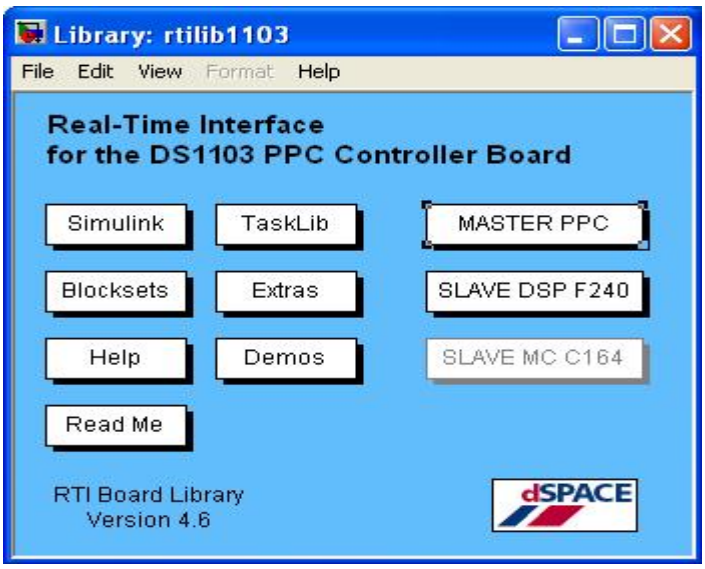
Bu kısımda simülasyonun doğru çalışması için kurulmuş olan konfigürasyonun düzgün ayarlanması gereklidir. Amaç, yaratılan C kodunun karta uygun olan optimizasyona tabi tutulmasıdır. Bu sebeple “System Target” dosyası şekildeki gibi ayarlanır. Simülasyon parametrelerinde gerekli görülen ayarlar için de mutlaka dSPACE implementasyon kitapçıklarına başvurulmalıdır.

Daha sonraki aşama sisteme uygun Dijital – Analog çeviricilerin yerleştirilmesidir. Eğer sistemden bir çıkış veya sisteme herhangi bir giriş söz konusu ise mutlaka DAC ve ADC bloklarının eklenip ayarlanması gereklidir. dSPACE kütüphanesi içinde yer almakta olan bu DAC çeviricisi kullanılırken yardım dosyaları okunarak kullanım aralıkları belirlenmelidir. Yapılan çalışmalar esnasında kullanılan çeviricilere değinirsek; Dijital – Analog (DAC) çevirici olarak kullanılan blok -10 ... +10 volt arasında giriş almakta ve buna karşın aldığı sinyali -1 ... +1 aralığında Simulink ortamına aktarmaktadır. Bu yüzden 1/10 değerinde kazançlar (gain) analog çıkışlara yerleştirilmiştir. Şekil 4.8’de DAC blok açıklama menüsü gösterilmektedir.



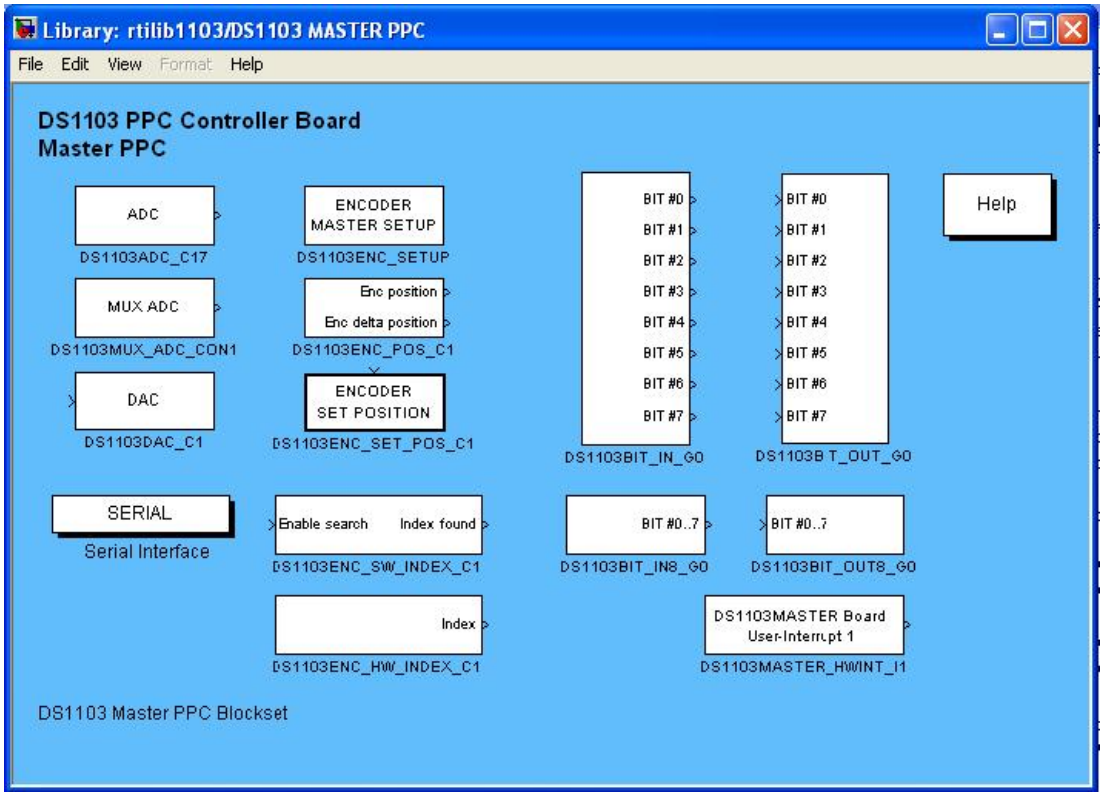
Şekil 4.8 DAC Blok Açıklama Menüsü

Yukarıda bahsedilen DAC Simulink bloklarını elde etmek için dSPACE firmasının sunduğu kütüphane kullanılmalıdır. Bloklara ulaşmak için Matlab "Command Window" üzerinde "rtilib" komutu yazılır. Komutun yazılmasıyla dSPACE bloklarını barındıran bir kütüphane kullanıma sunulur. Şekil 4.9'da bu kütüphane gösterilmektedir.



Şekil 4.9 dSPACE Blok Kütüphanesi

Buradaki kütüphaneden “Master PPC” alt kütüphanesi ve buradan da DAC çeviricileri seçilebilmektedir (bkz. Şekil 4.10).

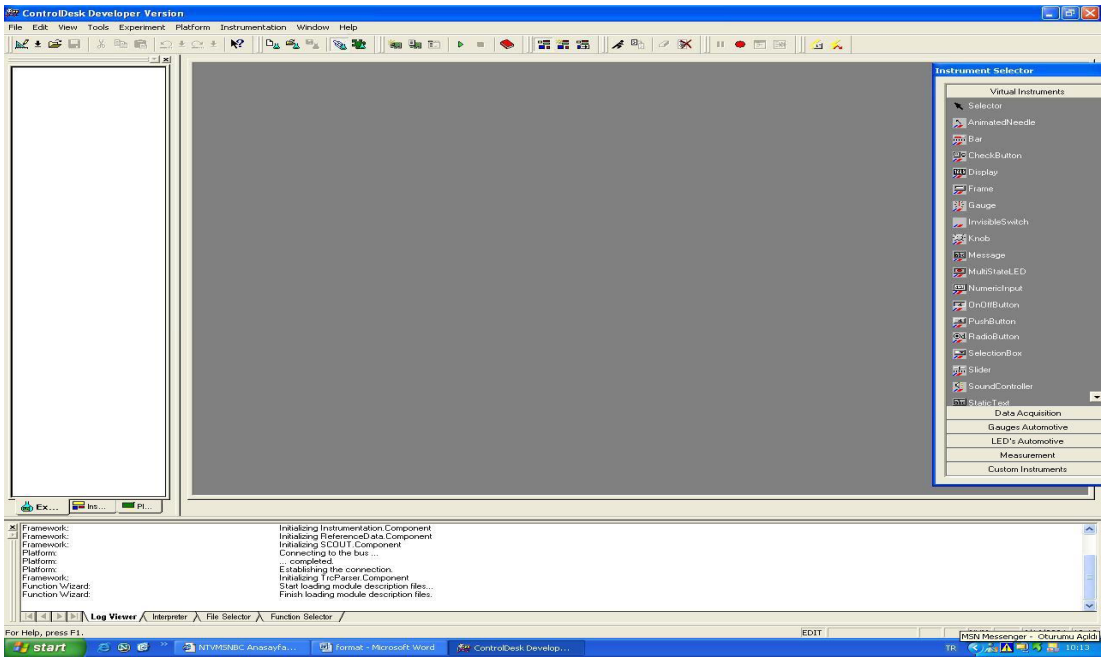


Şekil 4.10 Master PPC Alt Kütüphanesi

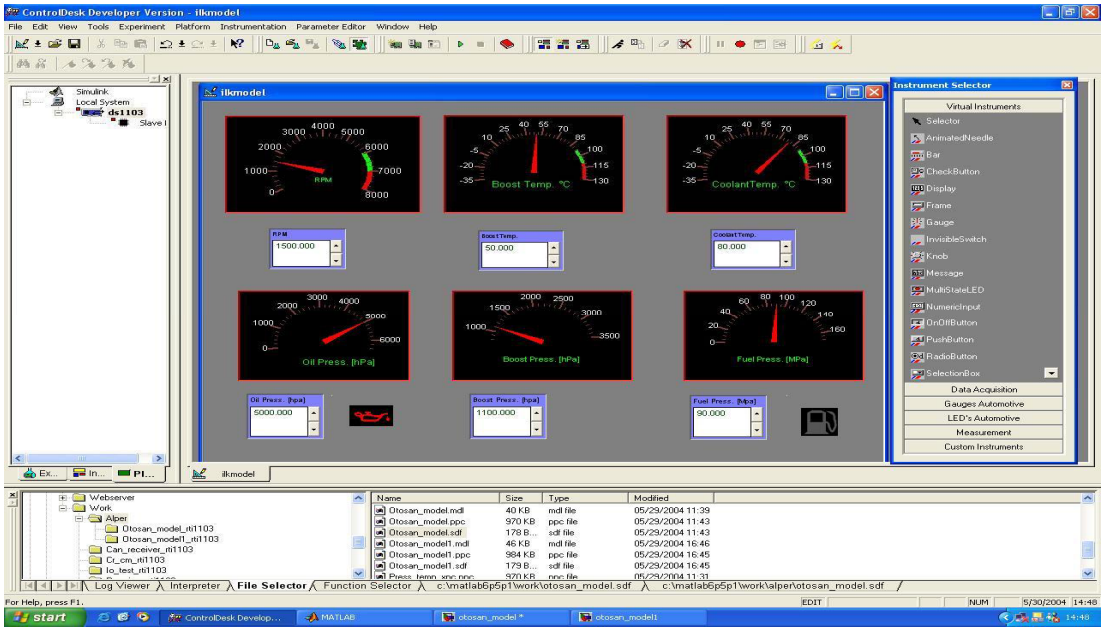
Model için genel ayarlar yapıldıktan sonra dSPACE firmasının sunmuş olduğu görsel açıdan parametreleri değiştirebilme ve görme imkanını sunan “Control Desk” programının ayarları yapılır. “Control Desk” programı dSPACE firmasının sunduğu, gerçek zamanlı (Real-Time) simülasyonlarda parametrelerin okunması ve arayüzlerin oluşturulması için yaratılan bir programdır. Program sayesinde ana modelin çalıştığı kart üzerindeki bütün parametreler özel göstergelerle sunulabilmektedir. Ayrıca simülasyon sırasında birçok parametrenin gerçek zamanda (Real-Time) değiştirilmesi sonucunda sistemin cevabını gözlemlemeyi sağlamaktadır. Ek olarak; program sayesinde dSPACE kartı üzerinde istenen modellerin Simulink programı olmadan da çalışması sağlanmaktadır.

Program, oluşturulan modelin daha kullanışlı ve verimli bir şekilde kullanılması için ayarlanmıştır. Motor sinyal parametrelerinin nasıl değiştiğini ve hız ayarlamasının nasıl yapıldığını göstermek için kullanılmıştır. Programda motor üzerinde devir sayısı, basınç ve sıcaklık verileri ayarlanabilmekte ve göstergeleri görülmektedir. Ayrıca istenen veriler gerçek zamanda değiştirilebilmektedir.

Öncelikle, programda yeni bir deney açılmalı ve bu deney karta aktarılmalıdır. Bu işlemler program üzerindeki menüler yardımıyla yapılır. Daha sonra istenen simülasyon için bir arayüz oluşturulabilir. Oluşturulan arayüzde gerekli parametrelerin izlenmeleri sağlanabilmektedir. Farklı anahtarlar konularak parametrelerin gerçek zamanda simülasyon esnasında değiştirilmesi sağlanır. Şekil 4.11 ve Şekil 4.12'de "Control Desk" programı arayüzü ve oluşturulan modelin "Control Desk" programındaki görüntüsü bulunmaktadır.

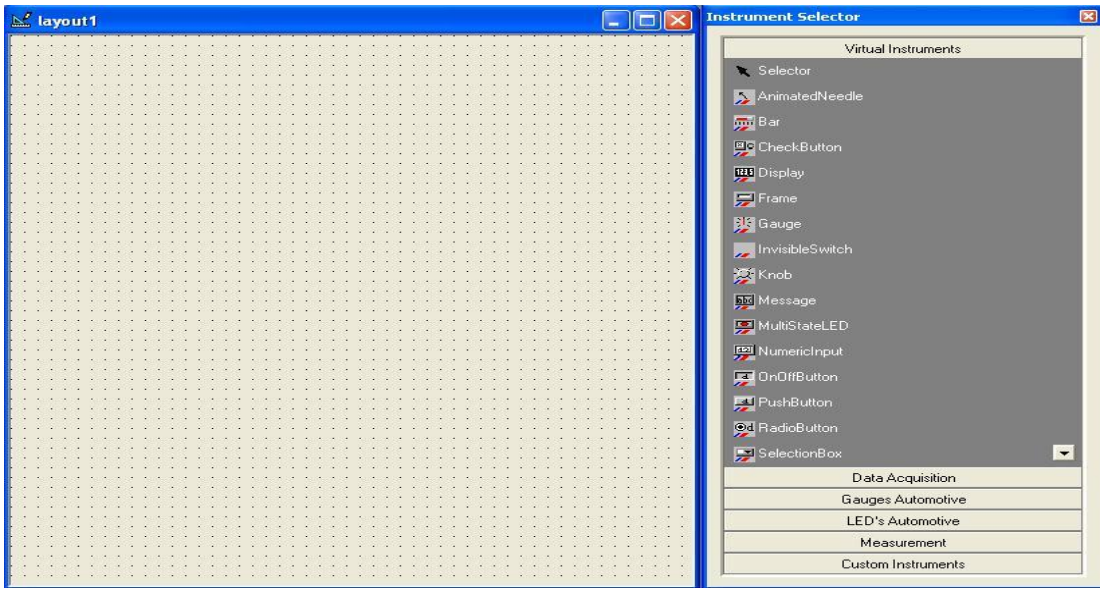


Şekil 4.11 Control Desk Programı Genel Görünümü



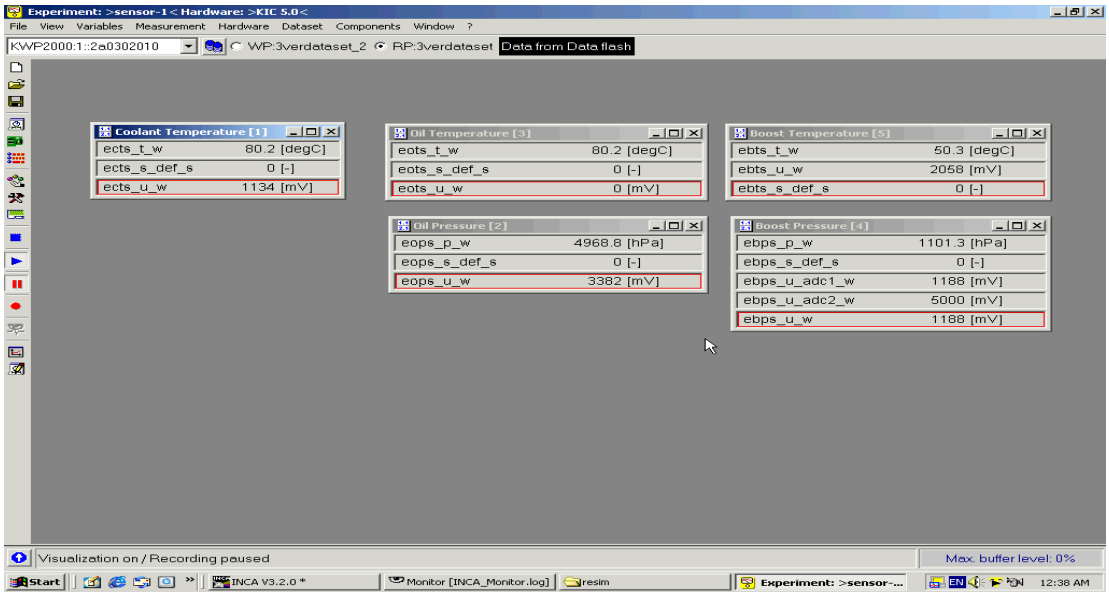
Şekil 4.12 Control Desk Programında Oluşturulan Model

Matlab / Simulink ortamında oluşturulan model ‘build’ edildikten sonra, “Control Desk” platformunda modelin ‘sdf’ (system description file) uzantılı dosyası kullanılmakta olan karta yüklenir. Böylece “Control Desk” ortamında oluşturulan görsel modellere veri aktarımı yapılabilir. Şekil 4.13’te boş bir ‘layout’ ve ‘gauge’ araçları gösterilmektedir.



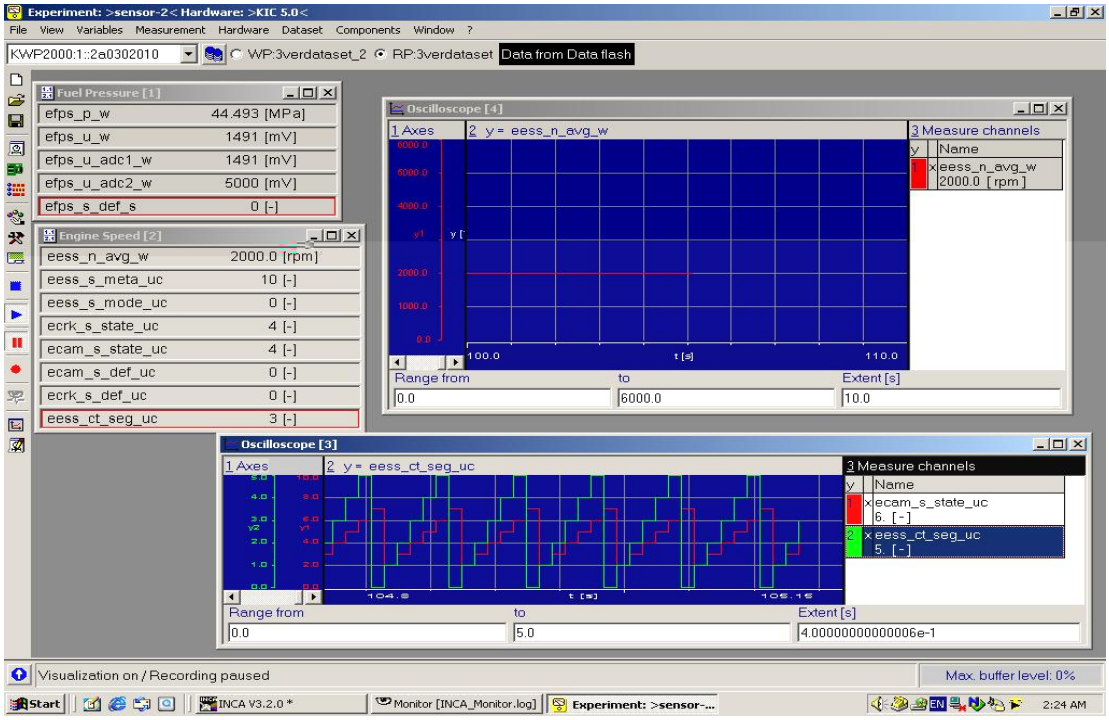
Şekil 4.13 Control Desk Programı Layout ve Gauge Görüntüsü

Şekil 4.14’de sensörlerin ölçüldüğü ekranın, yakıt basıncı (rail pressure) hariç, görüntüsü bulunmaktadır. Görüldüğü gibi hata kodları sıfır okunmakta ve istenilen değerler girilebilmektedir.



Şekil 4.14 Sensör Verileri Ölçüm Ekranı

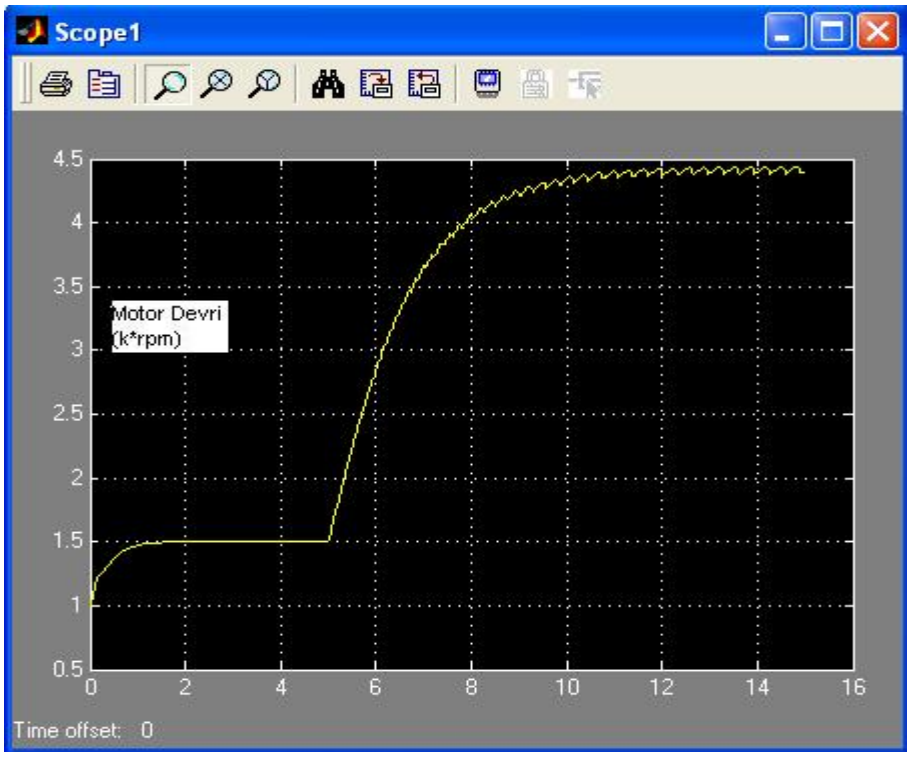
Benzer olarak Şekil 4.15'de yakıt basıncı ve motor devrinin hatasız olarak ayarlandığı ölçüm ekran penceresi gösterilmektedir.



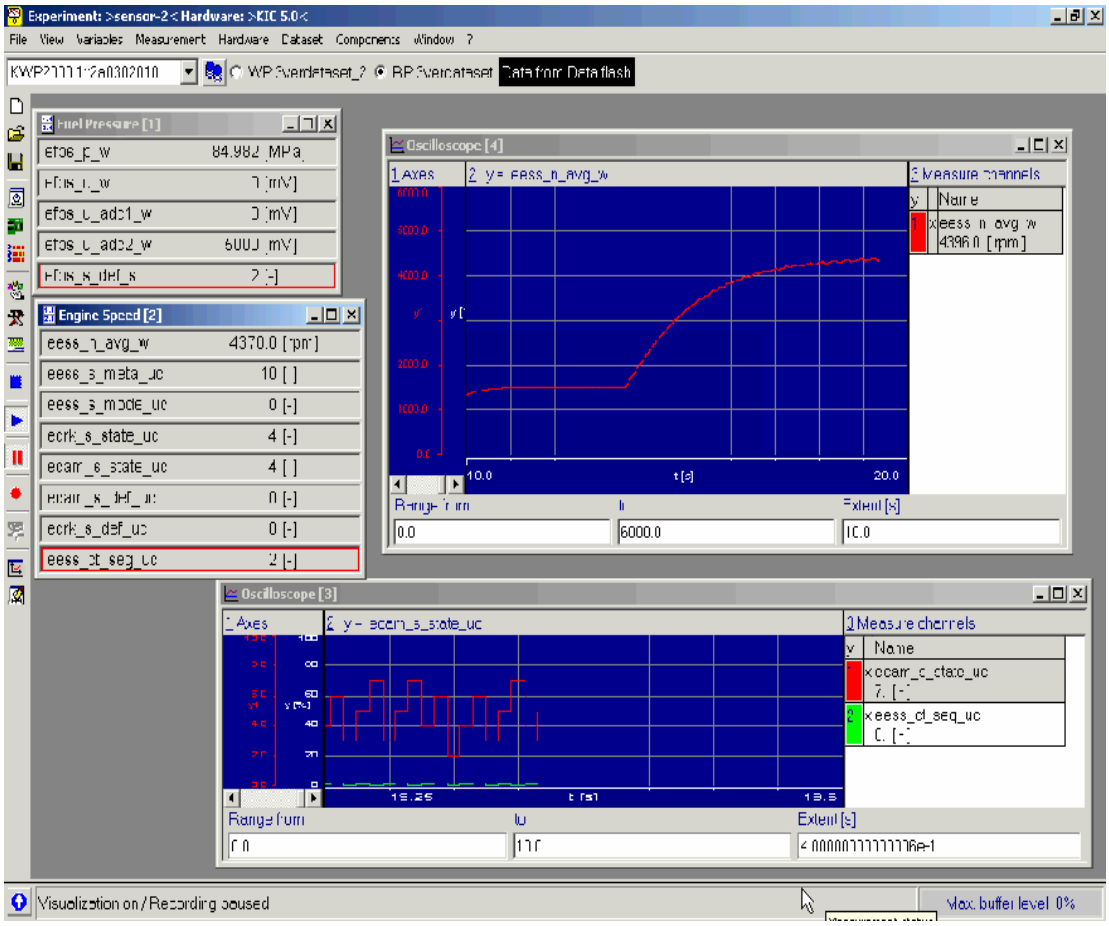
Şekil 4.15 Motor Devri ve Yakıt Basıncı Ölçüm Ekranı

dSPACE firmasının DS1103 kartı ile yukarıda anlatılan genel model kurulup istenilen şekilde çalıştığı gözlendikten sonra, National Instruments PCI 6025E kartı ile denenen testler bu kart ile tekrarlanmıştır. Ek olarak, motor devri için farklı veriler de hazırlanıp test edilmiştir. Kart hızının daha yüksek olması nedeni ile NI PCI 6025E kartı ile ulaşılan 2500 rpm sınırı DS1103 kartı ile 5400 rpm'lere kadar çıkarılmıştır. Aslında bu kart ile çok daha yüksek rpm'lere (8000 rpm) ulaşmak mümkün olmaktadır. Fakat kullanılan ECU' daki dataset ile mümkün olan maksimum motor devri 5400 olarak gözlenmiştir.

Ulaşılacak maksimum motor devri (rpm) testi dışında daha önce anlatılan MVEM ile de model çalıştırılmıştır. NI kartı ile yapılan testlere benzer olarak; Şekil 4.16'da MVEM ile uygulanacak olan veri, Şekil 4.17'de ise ölçüm ekranından alınan sonuçlar gösterilmektedir.



Şekil 4.16 MVEM ile Oluşturulan Veri



Şekil 4.17 Ölçüm Ekran Görüntüsü

Şekil 4.16 ve Şekil 4.17'de görülmekte olduğu gibi motor devri için MVEM ile oluşturulan veri ölçüm ekranında istenilen şekilde gözlenmektedir.

Bu bölümde dSPACE sistemleri ve dSPACE firmasının DS1103 kartı ile yapılan çalışmalar anlatılmıştır. Bu kart ECU test sistem gereksinimleri için yeterli analog girişe sahiptir ve yüksek motor devir sayılarını simüle edebilmektedir.

Motor ECU Test Sistemi Geliştirme Tezi kapsamında beşinci bölümde xPC Target Donanımı hakkında bilgi verilmiş ve yapılan testler anlatılmıştır.

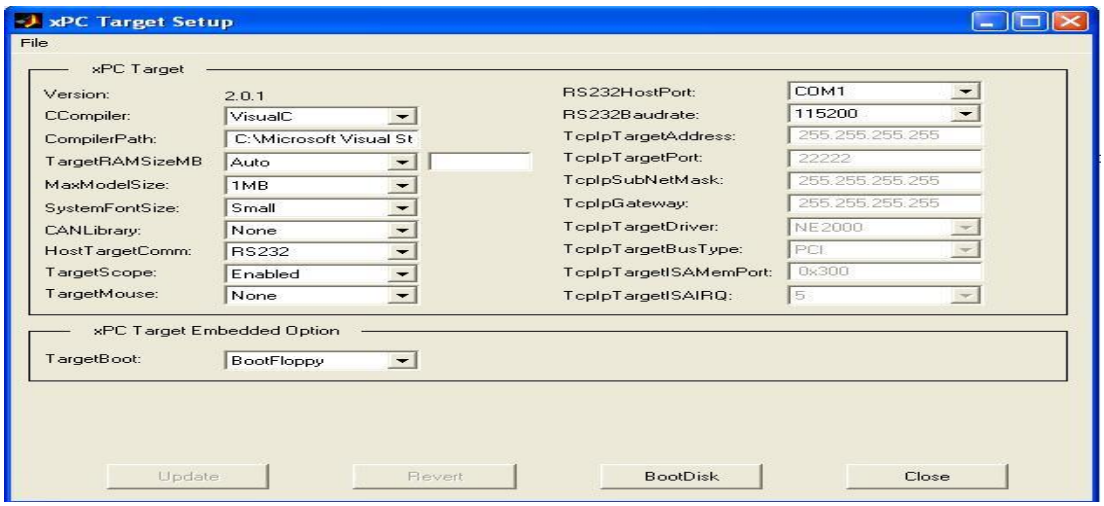
5. XPC TARGET DONANIMI

5.1 Giriş

xPC Target ve xPC TargetBox, Real-Time Windows Target gibi gerçek zamanlı simülasyonların yapılabilmesi için kullanılan bir başka çözümdür. Bu bölümde xPC Target ve xPC TargetBox hakkında bilgi verilecek ve bu sistemlerle yapılan çalışmalar anlatılacaktır.

5.2 xPC Target

xPC Target, Real-Time Windows Target'dan farklı olarak Ana Sistem-Hedef Sistem (Host-Target) opsiyonuna sahiptir. Model hedef bilgisayarda bir 'Kernel' ile çalışır. Hedef (target) opsiyonu işletim sisteminde meydana gelecek olan kesilmeleri önler ve modelin uzun süreli çalıştırılabilmesini sağlar. xPC Target'ın Ana-Hedef bağlantısı için Seri ve Network olmak üzere iki farklı seçeneği vardır. İki farklı seçenek için de ayarlar 'xpcsetup' komutu ile Matlab ortamında açılan 'xPC Target Setup' menüsünden yapılır (bkz. Şekil 5.1) ve ayarlar hedef PC' nin çalıştırılabilmesi için bir diskete kaydedilir.



Şekil 5.1 xPC Target Setup Menüsü

xPC Target Setup menüsünde diskete kaydedilen ayarlar ‘hedef’ (target) bilgisayara bağlanabilmek için önemlidir. Bağlantı yapılırken hedef bilgisayar bu disket ile açılır. Aşağıda seri ve network bağlantıları açıklanmıştır.

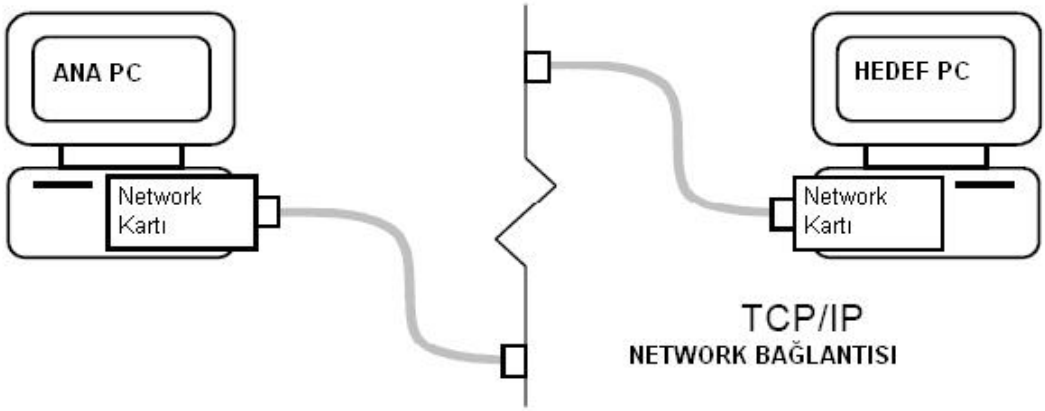
Seri Bağlantı: İki bilgisayar arasında RS 232 bağlantı portlarıyla yapılır.

Network Bağlantısı: LAN, internet veya doğrudan eternet kartları kullanılarak çapraz network kablolarıyla yapılır.

Her bağlantı şekli için de xPC Target Setup menüsündeki ayarlar dikkatlice yapılmalıdır. Şekil 5.2 ve Şekil 5.3’de bu bağlantılar gösterilmektedir.



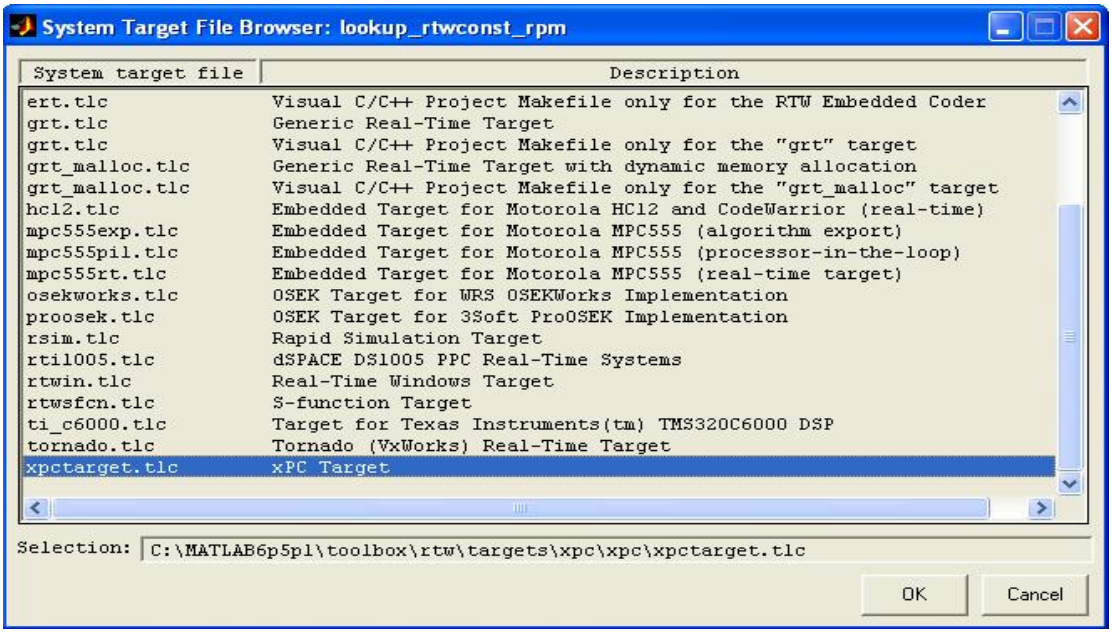
Şekil 5.2 Seri Bağlantı Gösterimi



Şekil 5.3 Network Bağlantı Gösterimi

İstenilen bağlantı seçeneği seçildikten ve hedef bilgisayar setup disketi ile açılır ve modelde gereken ayarlar yapılır. Sistem Real-Time simülasyon için hazır hale getirilir. Ayrıca hedef bilgisayar açıldıktan sonra ana bilgisayardaki Matlab komut penceresinde ‘xpctargetping’ yazılarak bağlantının başarılı olup olmadığı kontrol edilir. Daha sonra da ‘xpctest’ komutu ile kapsamlı testler yapılır.

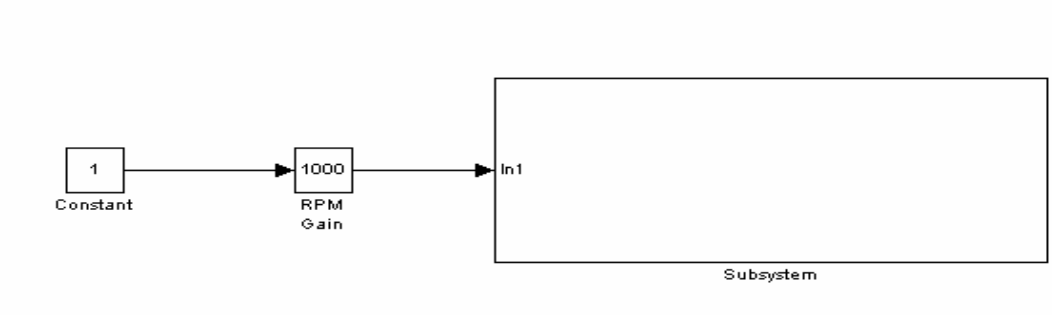
xPC Target ile yapılan testlerde analog çıkışlar ilk olarak daha önce olduğu gibi NI PCI-6025E kartı ile sağlanmıştır. Real-Time Windows Target'tan farklı olarak; simülasyon parametrelerindeki ayarlar yapıldıktan sonra model build edilip hedef bilgisayara model yüklenir ve çalıştırılır. Şekil 5.4'de Real-Time Workshop hedef tanımlaması gösterilmektedir.



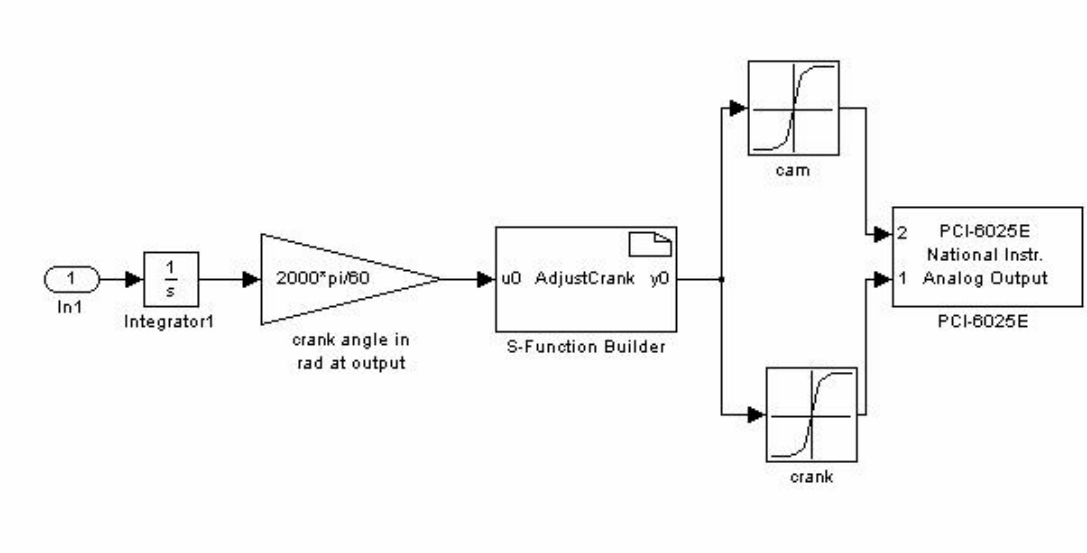
Şekil 5.4 Real-Time Workshop Hedef Tanımlama Menüsü

Sistem farklı olduğundan dolayı model oluştururken seçilen bloklara da dikkat edilmesi gerekir. Örnek olarak seçilen analog çıkış bloğu Real-Time Windows Target için seçilen bloktan farklı olacaktır. Modelin hatasız çalışması Matlab / Simulink kütüphanesi altındaki uygun blokların seçilmesi ile mümkündür.

Şekil 5.5 ve Şekil 5.6'da testlerde kullanılacak olan model ve altsistemi gösterilmektedir.



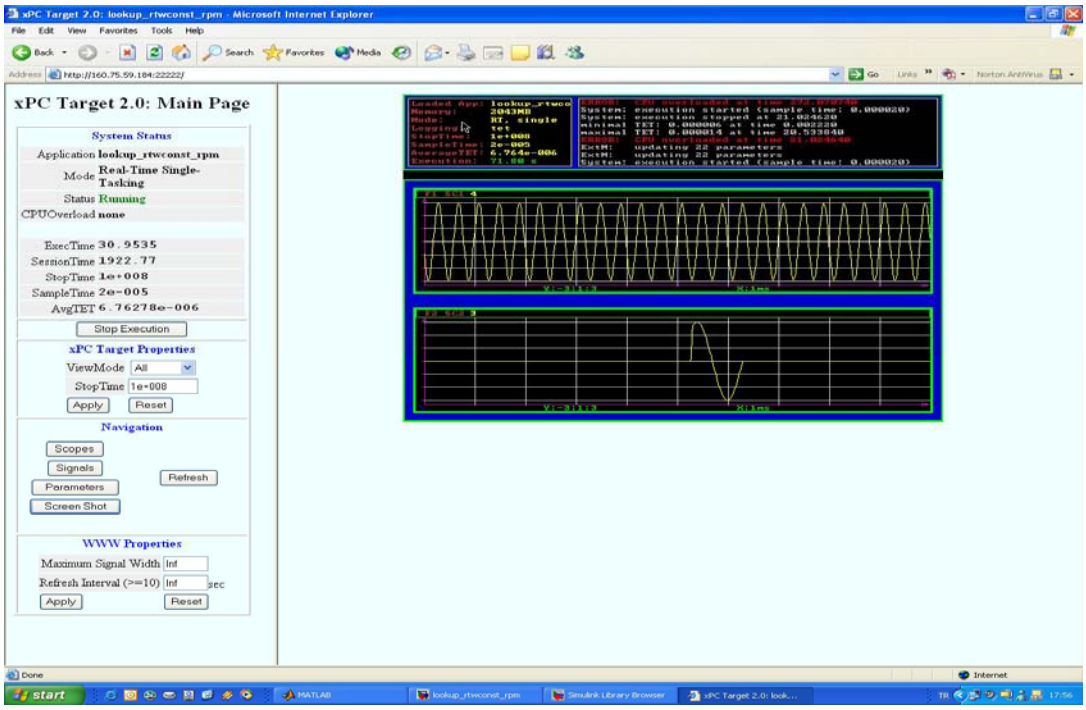
Şekil 5.5 Model Genel Görünümü



Şekil 5.6 Motor Devri Modeli Alt Sistemi

Şekil 5.6'da görüleceği gibi krank ve kam sinyallerinin gönderildiği analog çıkış bloğu Real-Time Windows Target kütüphanesinden farklı olarak xPC Target kütüphanesinin altında National Instruments' ın D/A seçeneğinin içinden alınmıştır.

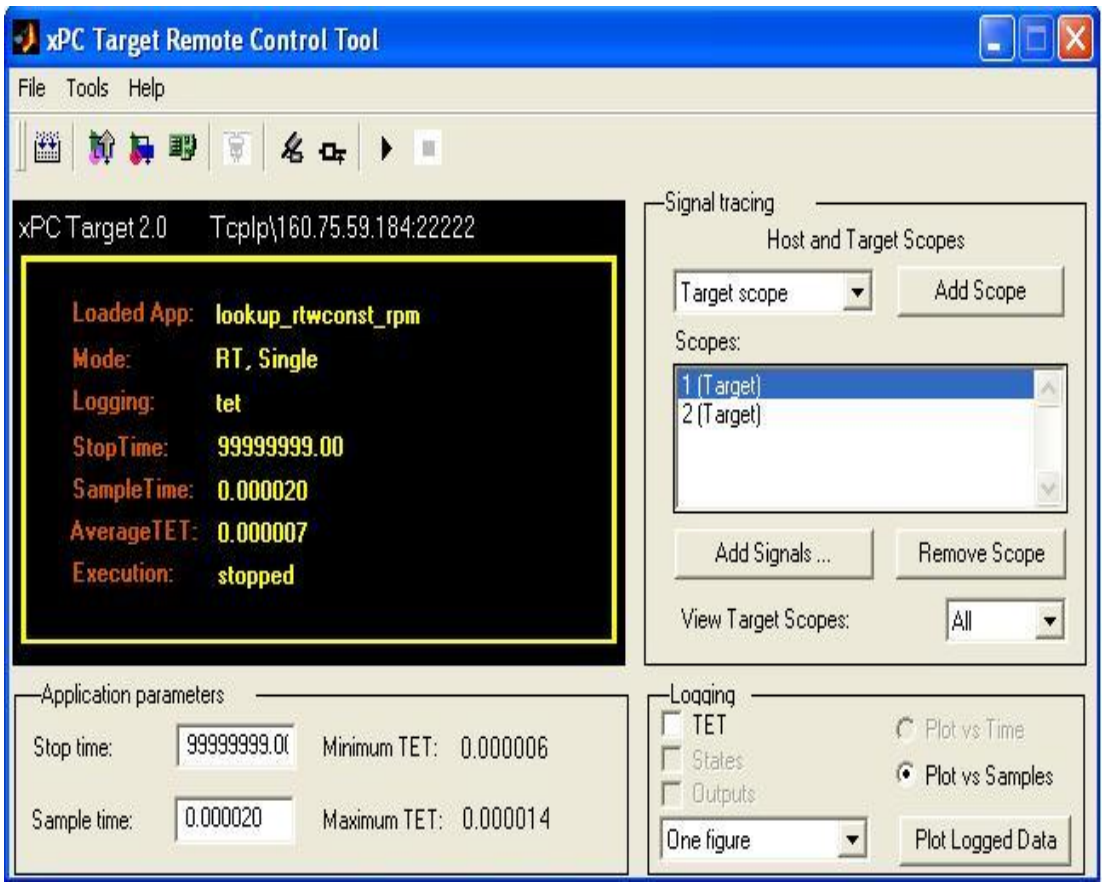
Matlab'de 'help xpc' komutu ile xPC Target ile yapılabilecek diğer seçeneklerin komutlarına ulaşılmaktadır. Testlerde kullanılan başka bir komut 'xpcwwwenable'dır. Bu komut ile xPC Target 'www' arayüzünü kullanmak mümkün olmaktadır. Şekil 5.7'de bu arayüz gösterilmektedir.



Şekil 5.7 xPC Target 'www' Arayüzü

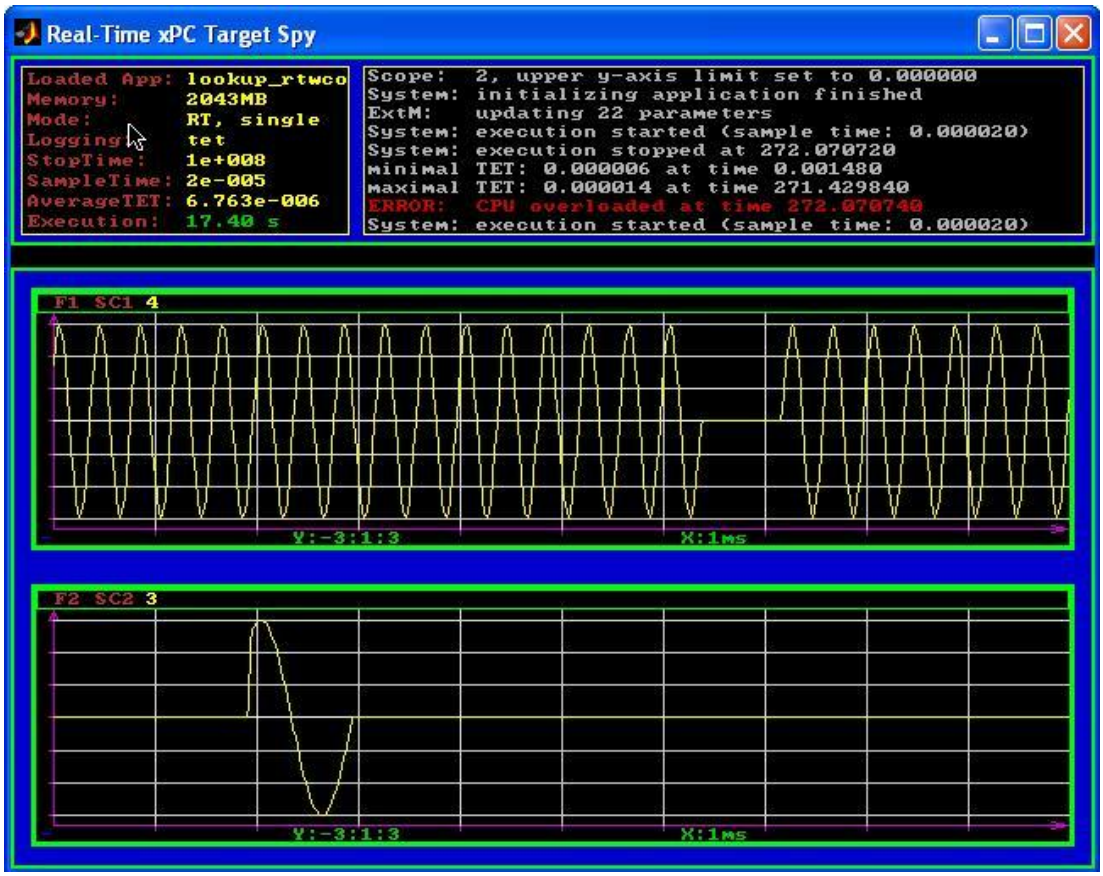
Matlab komut penceresinde bu komutu yazıp xPC Target www arayüzünü aktif hale getirdikten sonra kullanılan göstericide hedef bilgisayarın IP ve port numarası yazılarak bu arayüze ulaşmak mümkündür. Örnek olarak; Matlab komut penceresinde: xpcwwwenable, internet göstericisinde: http://160.75.59.184:22222/ yazılmaktadır.

Ayrıca, hedef bilgisayardaki model ana bilgisayardan xPC Target Remote Control Tool penceresiyle (bkz. Şekil 5.8) kontrol edilebilmektedir. Bu kontrol arayüzü ile hedef bilgisayardan anlık görüntüler almak, sinyal ve monitor (scope) eklemek ve veri kaydetmek mümkün olmaktadır.

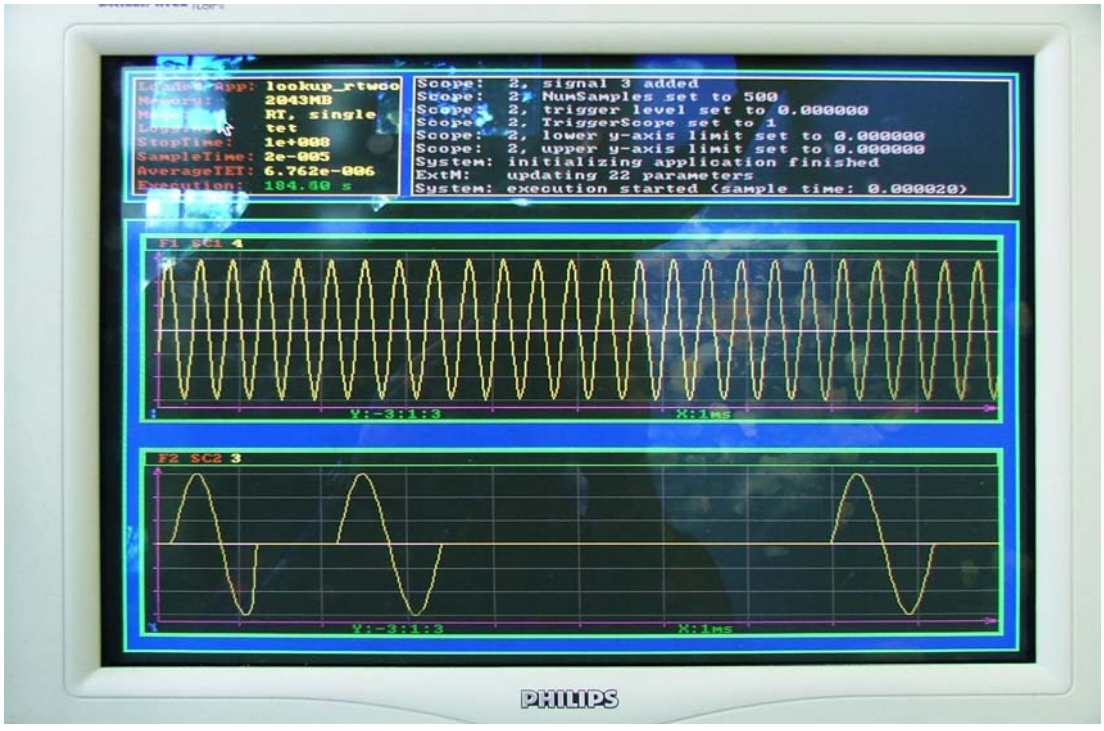


Şekil 5.8 xPC Target Remote Control Tool Penceresi

Şekil 5.9' da hedef bilgisayardan model çalışırken xPC Target Control Tool'daki xPC Target Spy seçeneğiyle alınan krank ve kam sinyallerinden bir görüntü, Şekil 5.10'da da aynı sinyaller dışarıdan kamerayla görüntülenmiş şekilde gösterilmektedir.

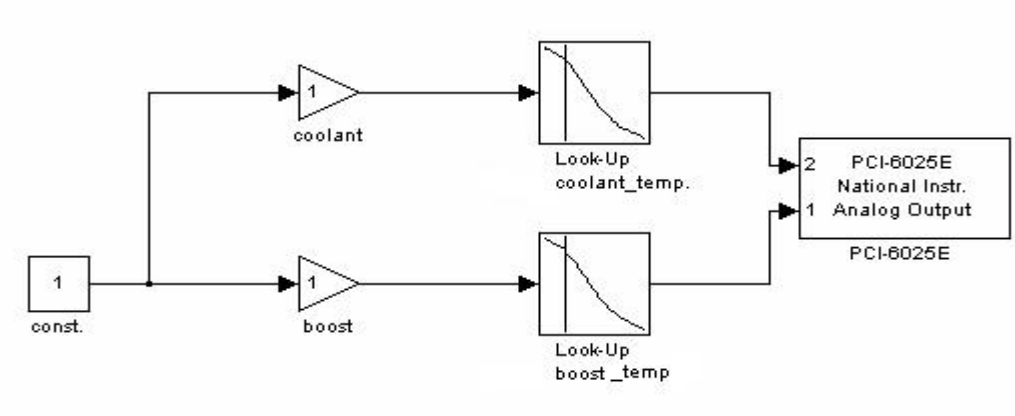


Şekil 5.9 xPC Target Spy Seçeneği ile Hedef Bilgisayardan Alınan Görüntü



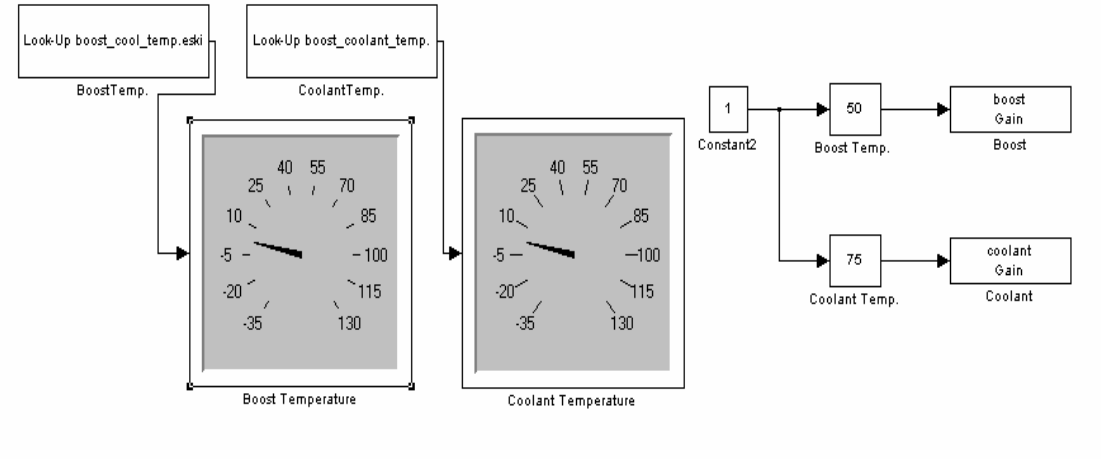
Şekil 5.10 Monitör Görüntüsü

xPC Target ile yapılan testlerde veri aktarım kartı olarak hedef bilgisayarda daha önceden Real-Time Windows Target için kullanılan NI PCI-6025E kartının kullanıldığı belirtilmişti. Kartın 2 analog çıkışa sahip olması yapılan testlerde sensörlerin çiftler halinde test edilmesini gerektirmiştir. Şekil 5.11'de sıcaklık sensörleri için Simulink'te oluşturulan model gösterilmektedir.



Şekil 5.11 Sıcaklık Sensörleri (xPC Target Sistemi için)

xPC Target ve daha sonra anlatılacak olan xPC TargetBox, Matlab/Simulink kütüphanesinde yer alan Dials&Gauges Blockset alt kütüphanesindeki göstergeleri gerçek zamanlı olarak kullanabilme imkanı sunmuştur. Şekil 5.12'de yukarıdaki model için sıcaklık değerlerinin girilebildiği ve görsel olarak izlenebildiği Simulink'te hazırlanan model gösterilmektedir. 2 ayrı model ile hazırlanan sistemde, analog çıkışın bulunduğu model gerçek zamanlı çalıştırıldıktan sonra göstergelerin bulunduğu model 'normal' modda çalıştırılır. Böylece istenilen sıcaklık değerlerini girip göstergelerden okumak mümkündür. Aşağıda görülen modelin nasıl oluşturulduğu, komutları xPC TargetBox anlatılırken verilecektir.



Şekil 5.12 Sıcaklık Sensörleri için Göstergelerin Olduğu Model (xPC Target)

5.3 xPC TargetBox

xPC TargetBox (bkz. Şekil 5.13), oluşturulan modellerin Real-Time Workshop ve xPC Target tarafından yürütülen uygulamaların çalıştırıldığı yüksek performanslı endüstriyel bir PC'dir. Bu sistem harici disket sürücü, şasi, I/O kabloları, dış güç kaynağı, konnektör panelleri, terminal kartları ve diğer yan ürünleri içerir. xPC TargetBox, alanın dar olduğu, seyyar uygulamalarda rahatça kullanılabilir. xPC TargetBox, 6 µs'lik örnekleme hızına ulaşabilir. Bu performans xPC TargetBox' daki işlemciye ve kullanılan modele bağlıdır.



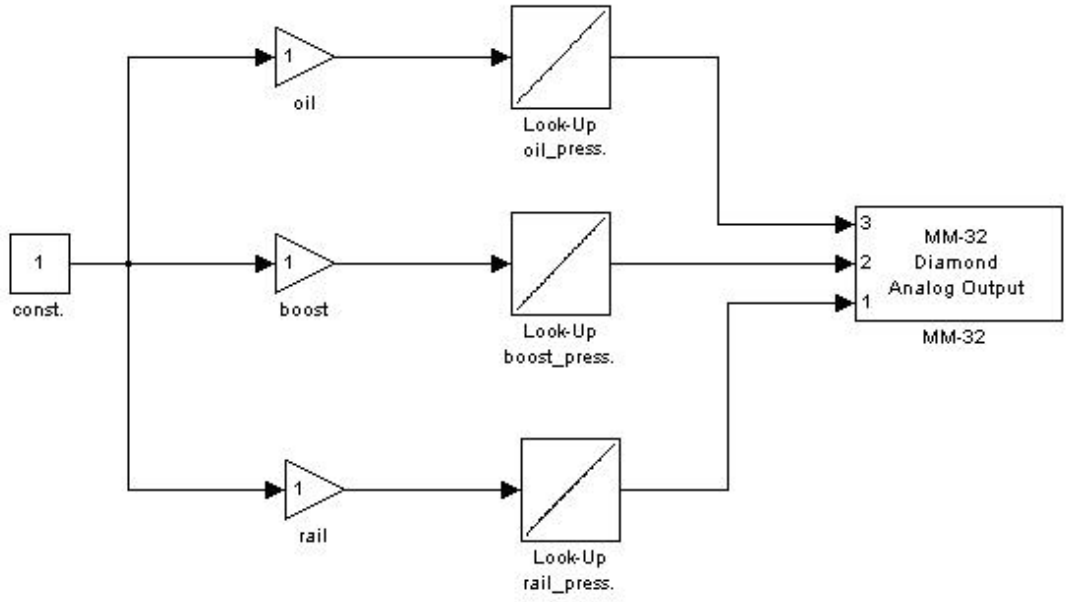
Şekil 5.13 xPC TargetBox [38]

xPC TargetBox ile yapılan çalışmalarda analog çıkışa sahip olan IO 301 portu kullanılmıştır. xPC TargetBox kendi içinde, 32 analog (16-bit) giriş kanallı, 4 analog (12-bit) çıkış kanallı (D/A) ve 24 digital I/O hattına sahip Diamond'un MM-32-AT analog-digital kartını bulundurmaktadır.

xPC TargetBox ile kam, krank, sıcaklık ve basınç sinyalleri daha önceki testlerde kullanılan modellere bazı eklemeler ve gerekli ayarlar yapılarak oluşturulmuştur.

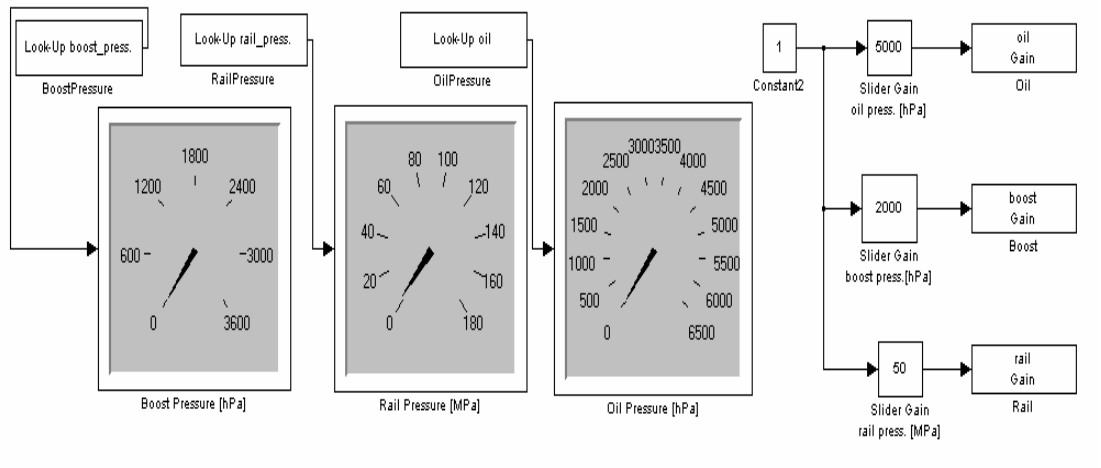
4 analog çıkışa sahip olunduğu için testlerde kullanılan modellerden biri 3 basınç sensörü ile diğeri ise 2 sıcaklık sensörü ve motor devri (bir krank, bir kam sinyali (2)) ile oluşturulmuştur. EK-D'de DIAMOND kartının blok diyagramı ve pin numaralandırması gösterilmektedir.

xPC TargetBox için hazırlanan modellerden basınç sensörleri ile oluşturulanı Şekil 5.14'de gösterilmektedir.



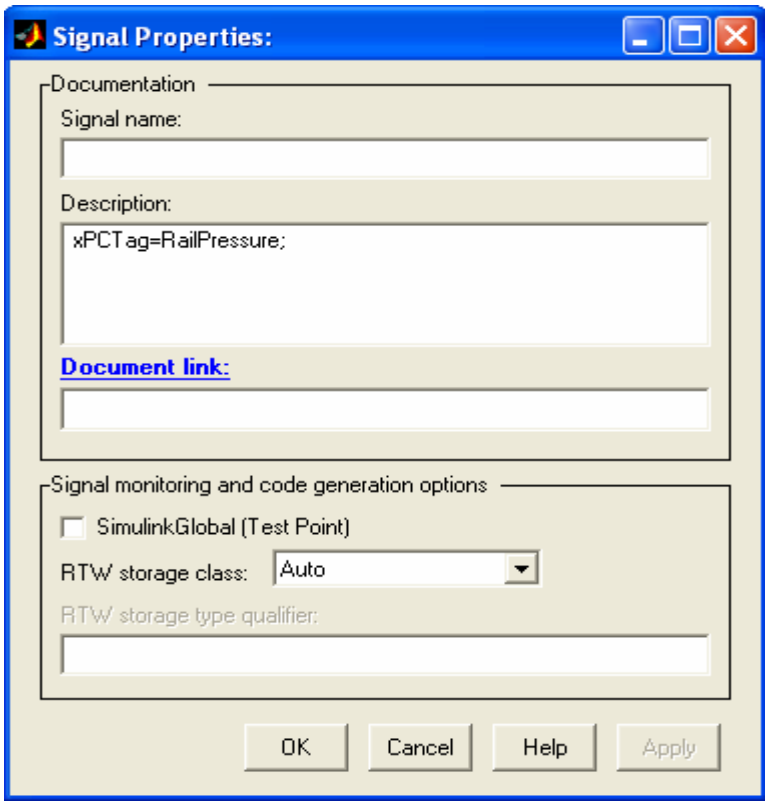
Şekil 5.14 Basınç Sensörlerinin Oluşturulduğu Model (xPC TargetBox)

Şekil 5.14'teki modelde görüldüğü gibi 3 basınç sensörü bir aradadır. Görsel açıdan kolaylık sağlamak için daha önce xPC Target'ta oluşturulan görsel model kullanılmıştır. Şekil 5.15'de göstergelerin olduğu model gösterilmektedir.

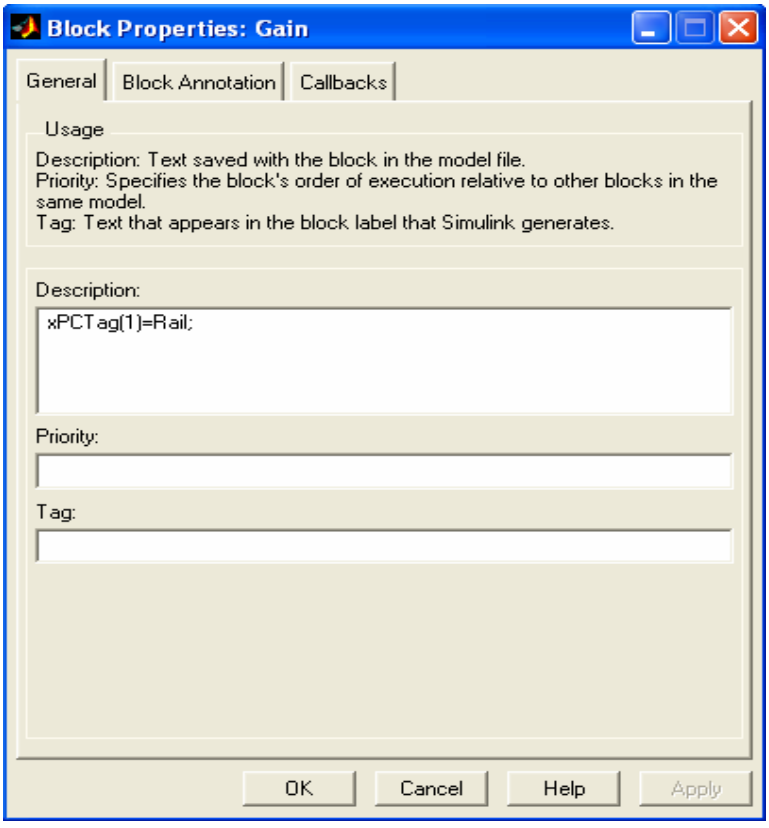


Şekil 5.15 Basınç Göstergelerinin Olduğu Model (xPC TargetBox)

Şekil 5.15'de gösterilen modeldeki göstergelerin çalışabilmesi için gerekli olan 'to workspace' ve 'from workspace' blokları için gerekli olan bazı ayarlar vardır. Bu blokları oluşturmak için Şekil 5.14'de gösterilen modelin sinyal ve blok özellikleri menülerine girilir ve tanımlama kısmına gerekli komutlar yazılır. Böylece bu blokları oluşturmak mümkün olur. Şekil 5.16 ve Şekil 5.17'de bu menülerden birer tanesi gösterilmektedir.



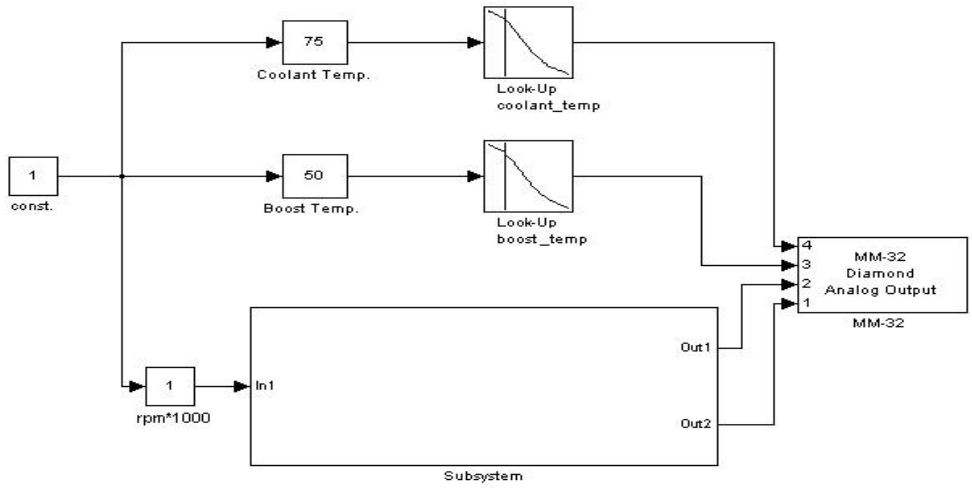
Şekil 5.16 Sinyal Özellikleri Menüsü



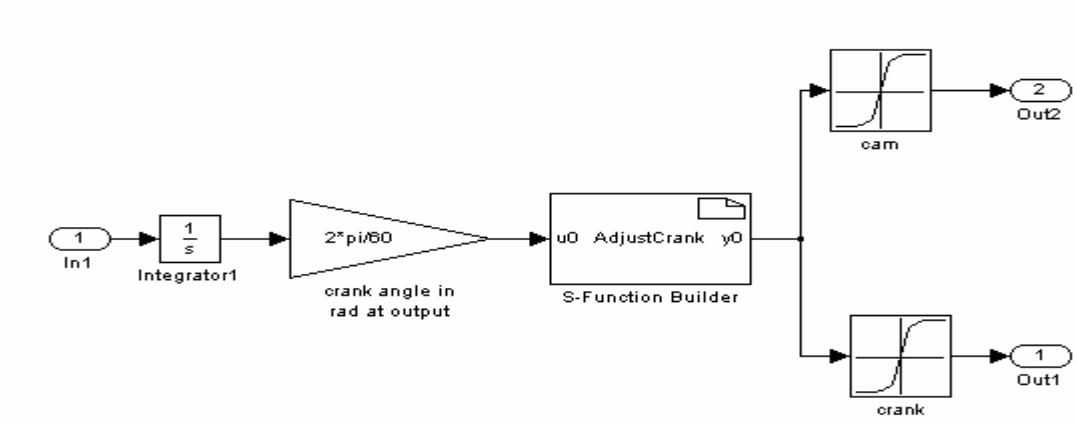
Şekil 5.17 Blok Özellikleri Menüsü

Şekil 5.16 ve Şekil 5.17'de gösterildiği gibi her bir gain çıkışının sinyal özellikleri (signal properties) menüsüne xPCTag=RailPressure (her bir basınç sensörü için farklı isim), ve kazanç bloklarının özellik (gain block properties) menülerine xPCTag(1)=Rail yazılıp her bir sensör için bu yazımı tamamladıktan sonra Matlab komut sayfasında xpcsliface('model ismi') yazılarak görsel modeldeki göstergelerin yerleştirileceği model sayfasında gerekli "to ve from" blokları otomatik olarak oluşturulmuş olur. Modeldeki slider'lara istenilen değerler girilip simülasyonlar gerçekleştirilir.

xPC TargetBox ile yapılan diğer test sıcaklık sensörleri ve motor devrinin bir arada olduğu model ile yapılmıştır. 4 analog çıkışa sahip olunduğu için 2 sıcaklık, kam ve krank sinyallerini birlikte gönderilmiştir. Şekil 5.18 ve Şekil 5.19'da bu modeller gösterilmektedir.



Şekil 5.18 Motor Devri ve Sıcaklık Kontrollü Model



Şekil 5.19 Motor Devri Alt Sistemi

xPC Target ve xPC TargetBox sistemleri ile yapılan testlerde dSPACE sistemlerinden alınan verim alınamamıştır. xPC Target sistemi, xPC TargetBox'a göre oluşturulan modelin daha hızlı çalışabildiği bir sistemdir. Fakat testlerde veri aktarım kartı olarak NI PCI-6025E kartının kullanılması analog çıkış sayısını ve aktarım hızını düşürmektedir. xPC TargetBox'ta istenilen motor devir hızına ulaşılamamıştır. Bunun sebebi elde bulunan DIAMOND veri aktarım kartı ve krank-kam verisinin büyük olmasındandır. xPC TargetBox için daha hızlı ve daha çok analog çıkışa sahip olan kart seçeneği mevcuttur. Bu kartlardan biri kullanılarak daha iyi sonuçlar elde etmek mümkün olabilir.

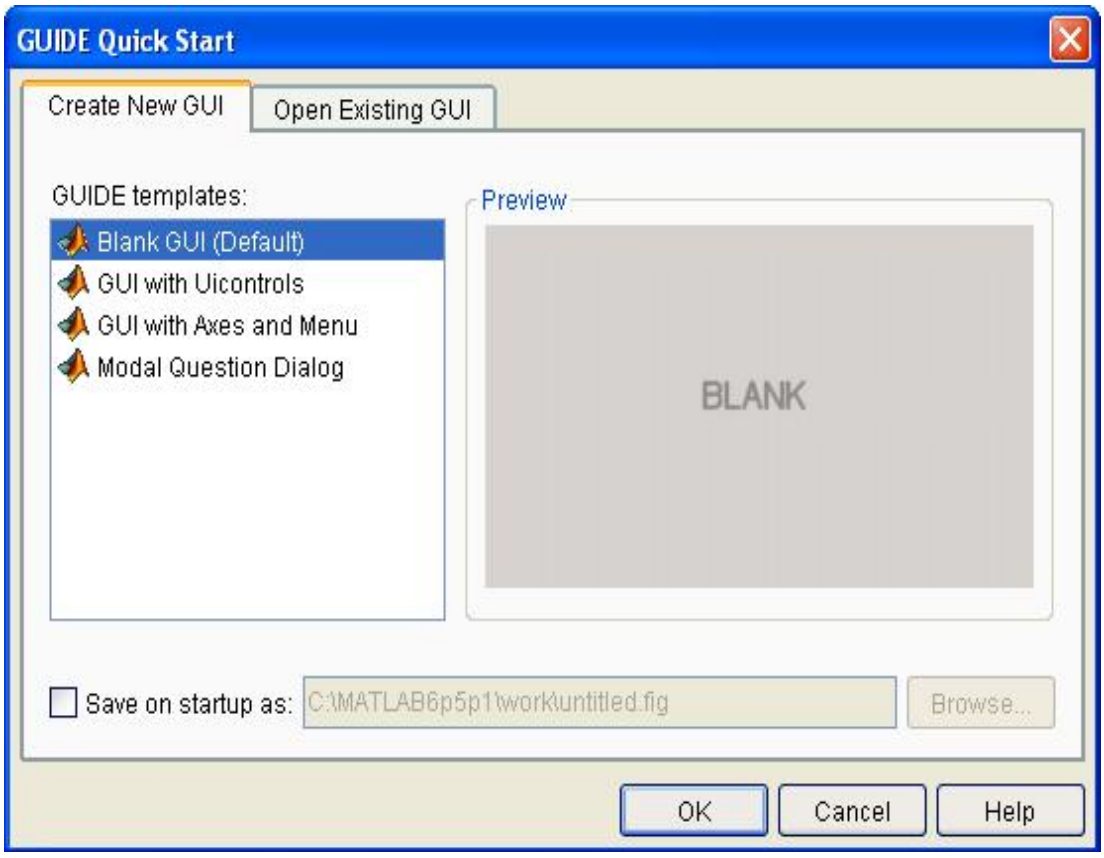
Bu bölümde xPC Target yazılımı ve xPC TargetBox donanımının tezde kullanımı anlatılmıştır.

Altıncı bölümde, Matlab GUI (graphical user interface) ile ilgili bilgi verilmiş ve simülasyonlarda kullanılan hatalı sinyaller gösterilmiştir.

6. MATLAB GUI VE HATALI SİNYALLERLE YAPILAN SİMÜLASYONLAR

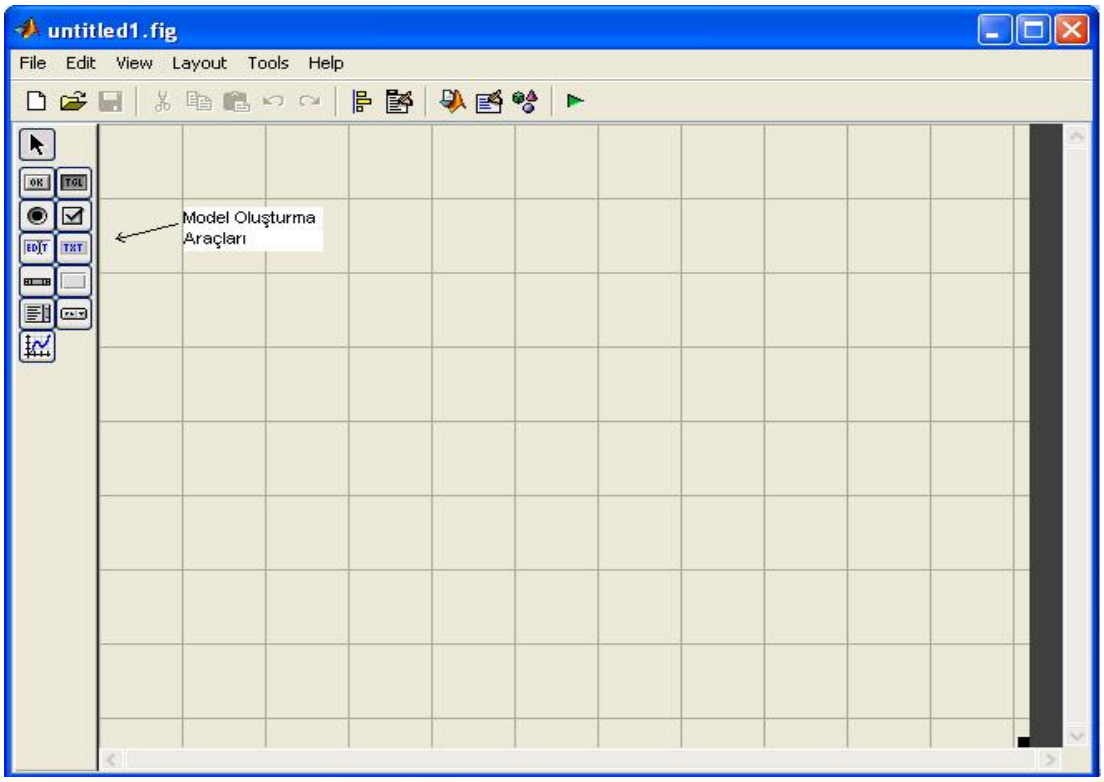
6.1 Giriş

Matlab GUI (graphical user interface) dSPACE ControlDesk yazılımına benzer şekilde, görsel ortamda istenilen modelin çalıştırılıp kontrol edilmesini sağlayan bir arayüzdür. Şekil 6.1'deki hızlı başlangıç penceresini açabilmek için Matlab komut penceresinde 'guide' yazılır.



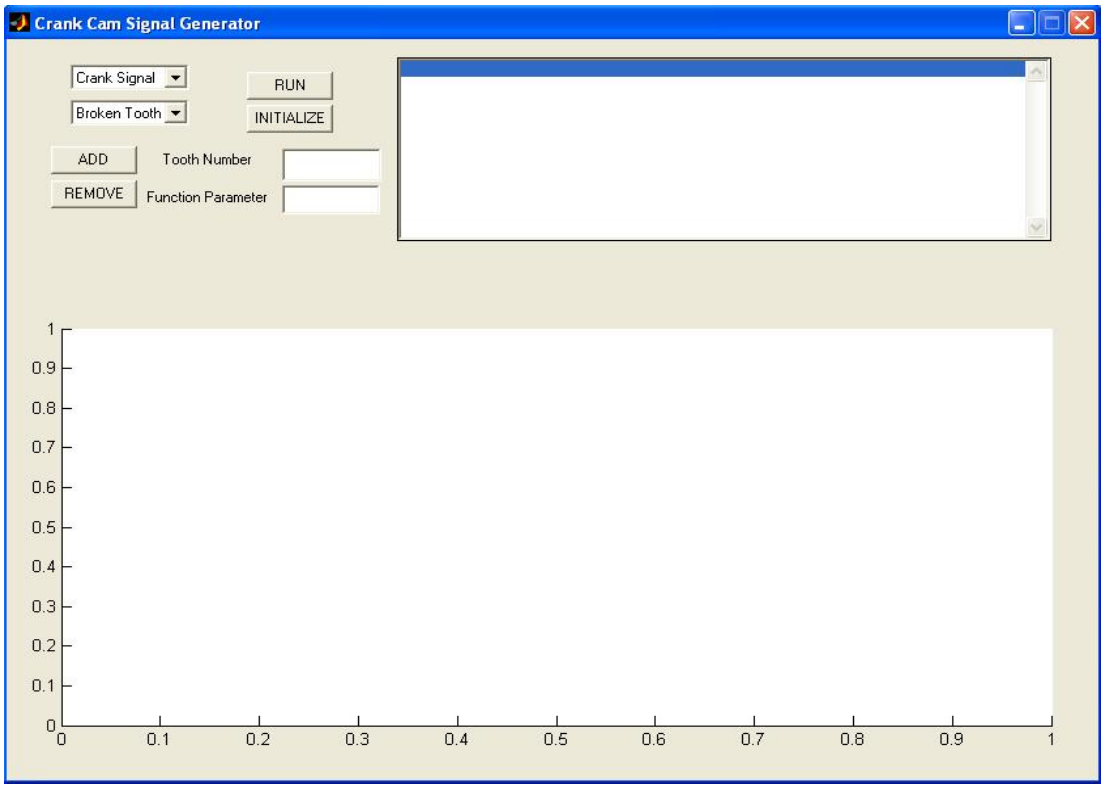
Şekil 6.1 GUIDE Hızlı Başlangıç Penceresi

Bu pencere yardımı ile yeni bir GUI modeli oluşturmak veya var olan bir modeli açmak mümkündür. Şekil 6.2'de arzu edilen modelin oluşturulabileceği boş bir GUI model oluşturma penceresi gösterilmektedir. İstenilen model bu boş GUI penceresinde oluşturulduktan sonra gerekli fonsiyonların yazılacağı m-file (bkz. EK-E) Matlab GUI tarafından otomatik olarak oluşturulur.



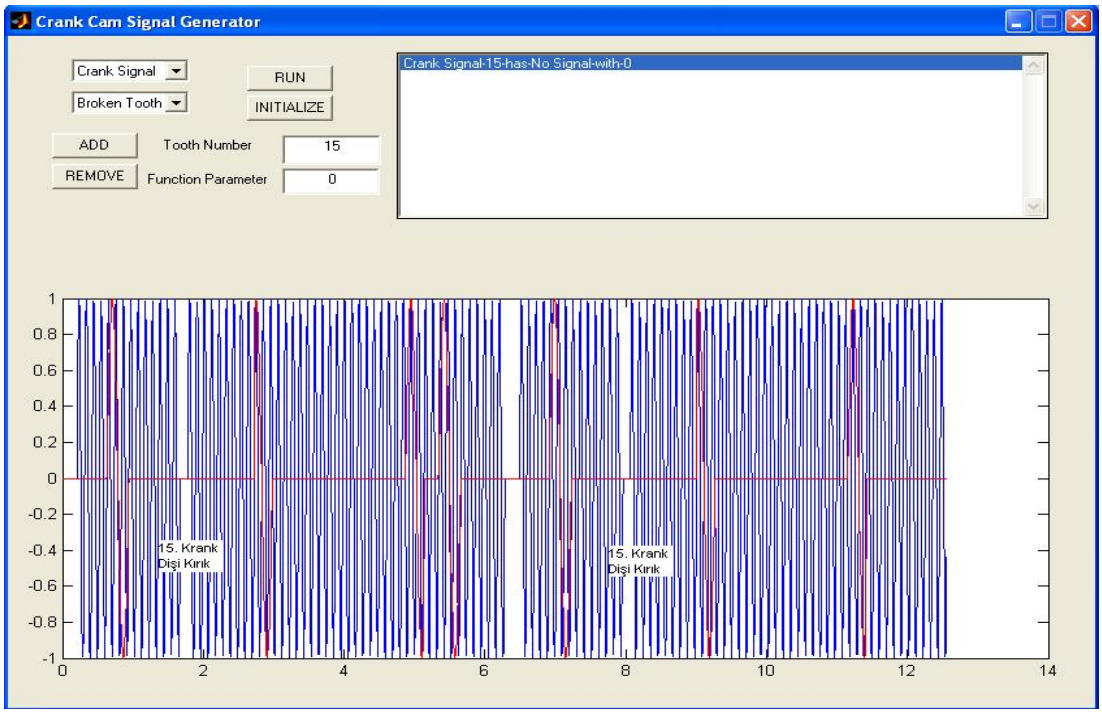
Şekil 6.2 Matlab GUI Model Oluşturma Penceresi

Şekil 6.3'de ise motor devri için krank ve kam sinyallerinin düzgün veya hatalı olarak oluşturulmasını sağlayan GUI modeli gösterilmektedir.

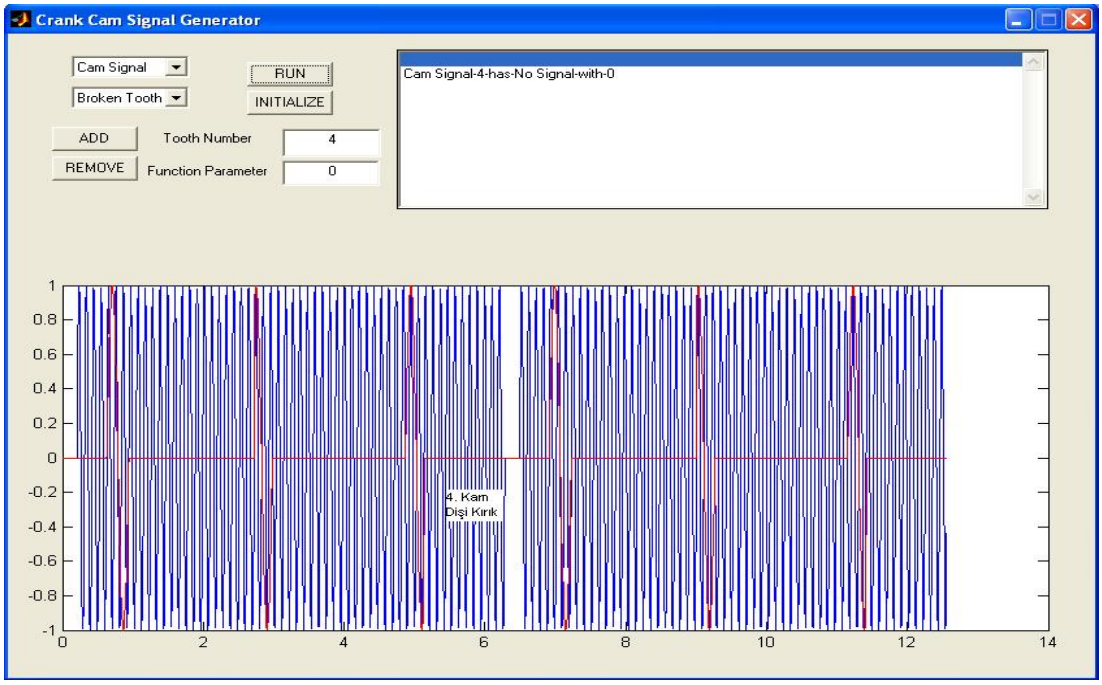


Şekil 6.3 Krank ve Kam Sinyali Bozma GUI Modeli

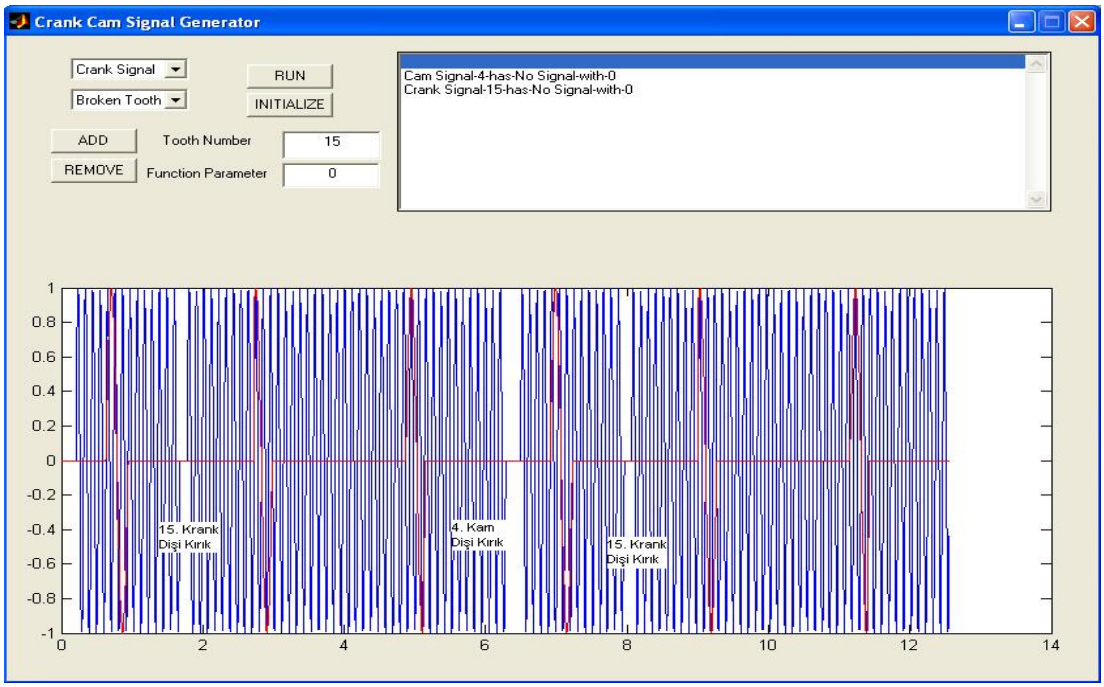
Şekil 6.3'de gösterilen GUI modeli ile krank ve kam sinyalleri bozulup motor devri sinyali ECU'ya gönderilebilir. Çeşitli hata senaryoları denenebilir. Bu senoryalardan bir krank dişi kırık (15. diş), bir kam dişi kırık (4. diş) ve hem krank hem de kam dişinin kırık olduğu (15. krank, 4. kam) durumlar sırasıyla Şekil 6.4, Şekil 6.5 ve Şekil 6.6'da gösterilmektedir.



Şekil 6.4 15. Krank Dişinin Kırık Olduğu Motor Devri Sinyali

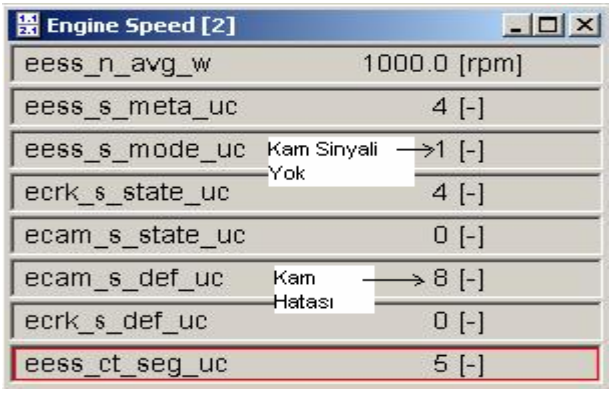


Şekil 6.5 4. Kam Dişinin Kırık Olduğu Motor Devri Sinyali



Şekil 6.6 15. Krank ve 4. Kam Dişinin Kırık Olduğu Motor Devri Sinyali

Donanım içeren simülasyonlar aracılığı ile tüm senoryalar test edilip sonuçları incelemek mümkün olmaktadır. Bu hata senoryalarını çoğaltmak mümkündür. Gerçek zamanlı donanım içeren ECU testi simülasyonlarında ölçüm ekranından hata alındığı gözlenmiştir. Şekil 6.7'de, Şekil 6.5'te gösterilen 4. kam dişinin olmadığı durumdaki hatalı sinyalin görüldüğü ölçüm ekranı gösterilmektedir. 1000 rpm motor devrinde yapılan simülasyonda kam sinyalinde hata olduğu ölçüm ekranından anlaşılmaktadır.



Şekil 6.7 Ölçüm Ekranı

Bu bölümde Matlab GUI ile oluşturulan model gösterilmiş ve simülasyonlarda kullanılan bozulmuş sinyallerden örnekler verilmiştir.

Yedinci bölümde sonuçlar ve öneriler verilerek kullanılan platformlar arasında karşılaştırma yapılmıştır.

7. SONUÇLAR VE ÖNERİLER

Bu tezde ana amaç bir motor elektronik kontrol ünitesinin donanım içeren testlerini yapmak için kullanılacak kullanımı kolay, yazılımla tüm işlemleri kontrol edilebilen bir motor simülatörünün geliştirilmesidir.

Bu tezde yapılan testler İTÜ Makina Fakültesi Otomasyon Laboratuvarındaki sistemlerle yapılmıştır. Bu sistemler National Instrument firmasının NI PCI-6025E kartı, dSPACE'in DS1103 kartı ve xPC Target donanımından oluşmaktadır.

Bu tez kapsamında ilk olarak oluşturulması en zor olan motor devri sinyalleri üzerinde çalışılmıştır. Tezde ana kriter olan motor devrinin oluşturulması için gerekli olan krank ve kam sinyallerinin üretilmesi üzerine araştırma yapılmıştır. Yapılan araştırmalarda gerçek motordan alınan veriler ve sensör çalışma prensiplerinden yararlanarak bir Matlab m-dosyası yazılıp krank ve kam verileri oluşturulmuştur. Böylece kurulan modelde gerekli olan krank ve kam verileri sağlanmıştır. Motor devri için gerekli işlemler yapıldıktan sonra model ilk olarak tezin üçüncü bölümünde ayrıntılı olarak anlatılan Real-Time Windows Target'ta NI PCI-6025E kartı ile test edilmiştir. İkinci bölümde anlatılan motor devrinden dolayı olan gereksinimlerdeki hesaplamalardan da görüldüğü gibi 10 kHz'lik analog çıkış hızına sahip olan NI PCI-6025E kartı ile maksimum 2500 rpm'e ulaşılmıştır. Ayrıca kartın 2 analog çıkışa sahip olmasından dolayı testler sınırlı düzeyde yapılmıştır (aynı anda 2 sıcaklık sensörü, 2 basınç sensörü, krank ve kam sinyallerinin beraber gönderimi gibi...).

Tezde yapılan diğer testler, dördüncü bölümde gösterilmekte olan dSPACE firmasının DS1103 kartı ile gerçekleştirilmiştir. Kartın 200 kHz'lik yüksek hıza sahip olması nedeniyle motor devrinde oldukça yüksek devirlere çıkılmıştır. Ölçüm ekranında 5400 dev/dak görülmesine rağmen osiloskop ile yapılan incelemede dakikada 8000 devire kadar çıkılabildiği görülmüştür. Ayrıca 8 analog çıkışa sahip olması sayesinde bu kart tüm sensörleri ve motor devrini bir arada test edebilme imkanını vermiştir. Ek olarak dSPACE firmasının sunmuş olduğu görsel açıdan parametreleri değiştirebilme ve görme imkanını sunan "Control Desk" programı kullanılmıştır. Program sayesinde ana modelin çalıştığı kart üzerindeki bütün parametreler özel göstergelerle sunulabilmektedir. Ayrıca simülasyon sırasında

birçok parametrenin gerçek zamanda (Real–Time) değiştirilmesi sonucunda sistemin cevabını gözlemlemek mümkün olmaktadır. Program sayesinde dSPACE kartı üzerine istenen modellerin Simulink programı olmadan da çalışması sağlanmaktadır. Proje kapsamında DS1103 kartı ile test edilen model aynı zamanda 'control desk' programında görsel olarak oluşturulup test edilmiştir.

Bu tezde yapılan bir diğer test te xPC Target yazılım ve donanımı ile yapılmıştır. xPC Target ve xPC TargetBox sistemleri ile yapılan testlerde dSPACE sistemlerinden alınan verim alınamamıştır. xPC Target sistemi, xPC TargetBox'a göre oluşturulan modelin daha hızlı çalışabildiği bir sistemdir. Fakat xPC Target ile yapılan testlerde veri aktarım kartı olarak NI PCI-6025E kartının kullanılması analog çıkış sayısını ve aktarım hızını düşürmüştür. xPC TargetBox kullanımında ise istenilen motor devri hızına ulaşılamamıştır. Bunun sebebi elde bulunan DIAMOND MM-32-AT veri aktarım kartının yetersizliği ve krank-kam verisinin büyük olmasıdır. xPC TargetBox için daha hızlı ve daha çok analog çıkışa sahip olan kart seçeneği mevcuttur. Bu kartlardan biri kullanılarak daha iyi sonuçlar elde etmek mümkün olabilir. xPC Target için ise örnek olarak National Instruments firmasının NI PCI-6713 kartı verilebilir. xPC Target donanımı ile yapılan testler ve ayarlar beşinci bölümde anlatılmıştır.

Tablo 7.1 ve Tablo 7.2'de sırasıyla kullanılan kartların özellikleri ve kullanılan yazılım bilgisi ile bu kartların karşılaştırması verilmiştir.

Tablo 7.1 Kullanılan Kartların Özellikleri ve Kullanılan Yazılım Bilgisi

Kullanılan Kart	Analog Çıkış Sayısı	Maksimum Örnekleme Hızı	Kullanılan Yazılım
DS1103	8	200 kHz	Matlab/Simulink/RTWIN, dSPACE ControlDesk
NI PCI-6025E	2	10 kHz	Matlab/Simulink/RTWIN xPC Target
Diamond MM-32-AT	4	6 µsn (~166.7 kHz)	Matlab/xPC TargetBox

Tablo 7.2 Kullanılan Platformlarda Alınan Test Sonuçlarının Karşılaştırılması

Kullanılan Kart	Anolog Çıkış Sayısı	Motor Devri (rpm)
DS1103	8	~5400
NI PCI-6025E	2	2500
Diamond MM-32-AT	4	2000

Sonuç olarak, tez kapsamında Otomasyon Laboratuvarında bulunan sistemlerle yapılan testlerde en çok verim dSPACE sistemlerinden alınmıştır. Tezde motor devri ve gerekli sensör sinyalleri oluşturulmuş ve ölçüm programı aracılığı ile izlenmiştir. ECU ile yapılan gerçek zamanlı donanım içeren simülasyonlarda dSPACE firmasının sunmuş olduğu kartlar veya simülatörleri kullanmanın avantaj sağladığı görülmektedir. Bu tezde oluşturulan motor sinyalleri ile istenilen senaryolar oluşturulup istenilen hata kodları ile incelemeler yapmak mümkün olmuştur. Bu hata senoryalarından bir kısmı altıncı bölümde gösterilmiştir. İleride istenildiği taktirde enjeksiyon sinyallerinin okunması, CAN uygulamaları ve ECU prototiplendirme üzerine kontrol algoritmaları çalışmalarının yapılması önerilir.

KAYNAKLAR

[1] **Bonnick, A.,** 2001. Automotive Computer Controlled Systems, Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford, 78-83.

[2] **Softing,** 2004. http://www.softing.com/en/ae/standards.htm, 4 Haziran.

[3] **Kimura, A., & Maeda, I.,** 1996. Development of engine control system using real time simulator, *Proceedings of the 1996 IEEE International Symposium on CACSD*, Michigan, USA, 157–163.

[4] **Isermann, R., Schaffnit, J., Sinsel, S.,** 1999. Hardware-in-the-loop simulation for the design and testing of engine-control systems, C*ontrol Engineering Practice 7*, 643-653.

[5] **Schaffnit, J., Sinsel, S., Isermann, R.,** 1998. Hardware-in-the-loop simulation for the investigation of truck diesel injection systems, *Proceedings of the American Control Conference*. Philadelphia, June.

[6] **Woermann, R. J., Theuerkauf, H. J, Heinrich, A.,** 1999. A Real-Time Model of a Common Rail Diesel Engine, SAE Paper No. 01-0862.

[7] **Sunwoo, M., Sim, H., Lee, K.,** 1999. Design and Development of an ECU and its Air-Fuel Ratio Control Scheme for an LPG Engine with a Bypass Injector, SAE Paper No. 0-7803-5296-3.

[8] **Jurak, M., Tuma, J.,** 2002. Testing of A Car Engine Control Unit Using the Hardware-in-the-loop Simulation, *International Carpathian Control Conference ICCC.* Malenovice, Czech Republic, May 27-30.

[9] **Weeks, R. W., Moskwa, J. J.,** 1995. Automotive Engine Modelling for Real-Time Control Using Matlab/Simulink, SAE Paper No. 950417.

[10] **Bosch,** 2004. http://archive.bosch.com/en/archive/theme_11_2003.htm, November 11.

[11] **D. Kempf, L. Bonderson, and L. Slafer,** 1987. Real Time Simulations for Application to ABS Development, SAE Paper No. 870336.

[12] **Lee, J. C., Suh, M. W.,** 1999. Hardware-in-the Loop Simulator for ABS/TCS, *Proceedings of the 1999 IEEE International Symposium on Control Applications*, Kohala Coast-Island of Hawai'i, Hawai'i, USA, August 22-27.

[13] **Park, S. T., Yang, S. Y., Seol, I. S., J.M. Cho,** 1999. Development of ABS Using HILS, *IEEE Information Systems and Technologies*, Korus' 99, 219-221.

[14] **Park, K., Heo, S. J.,** 2003. Design and Implementation of Hardware-in-the-loop Simulator for Vehicle Dynamics Control Systems, *12th International Pacific Conference on Automotive Engineering*.

[15] **Cho, J. M., Hwang, D. H., Lee, K. C., Jeon, J. W., Park, D. Y., Kim, Y. J., Joh, J. S.,** 2001. Design and Implementation of HILS System for ABS ECU of Commercial Vehicles, *ISIE*, Pusan, Korea.

[16] **Dugoff, H., Fancher, P. S., Segel, L.,** 1970. An Analysis of Tire Traction Properties and Their Influence on the Vehicle Dynamics performance, SAE Paper No. 700377.

[17] **Lee, K. C., Jeon, J. W., Hwang, D. H., Kim, Y. J.,** 2003. Performance Evaluation of Antilock Brake Controller for Pneumatic Brake System, *Industry Applications Conference*, 38th IAS Annual Meeting. Conference Record of the, **1**, October 12-16, 301-307.

[18] **Lee, W., Park, S., Sunwoo, M.,** 2004. Towards a seamless development process for automotive engine-control system, *Control Engineering Practice 12*, 977-986.

[19] **Weber, M., Isermann, R.,** 2004. Rapid Prototyping of Model Based Control Algorithms for Diesel-Engines with Turbocharger, *XXI ICTAM*. Warsaw, Poland, August 15-21.

[20] **Hafner, M., Jost, O., Isermann, R.,** 2002. Mechatronic Design Approach for Engine Management Systems, *Mechatronics 12*, 1035-1046.

[21] **Ackermann, J.,** 1998. Active steering for better safety, handling and comfort, *International Conference on Advances in Vehicle Control and Safety*, Amiens, France, July 1-9.

[22] **Ackermann, J., Bünte, T., Sienel W., Jeebe H., Naab K.,** 1996. Driving safety by robust steering control, *Int. Symposium on Advanced Vehicle Control,* Aachen, Germany, June.

[23] **Jang, S. H., Park, T. J., Han, C. S.,** 2003. A Control of Vehicle Using Steer-by-Wire System with Hardware-in-the-loop-Simulation System, *Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*.

[24] **Hanselmann, H.,** 1996. Hardware-in-the-Loop Simulation Testing and its Integration into a CACSD Toolset, *The IEEE International Symposium on Computer-Aided Control System Design*. Dearborn, Michigan USA, September 15-18.

[25] **Boot, R., Richert, J., Rükgauer, A.,** 1999. Automated Test of ECUs in a Hardware-in-the-Loop Simulation Environment, *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design*, Kohala Coast-Island of Hawai'i, Hawai'i, USA, August 22-27.

[26] **Caraceni, A., De Cristofaro, F., Ferrara, F., Scala, S., Philipp, O.,** 2003. Benefits of Using A Real-Time Engine Model During Engine ECU development, SAE Paper No. 01-1049.

[27] **Lefarth, U., Baum, U., Beck, T., Werther, K., Zurawka, T.,** 1998. An integrated approach to rapid product development for embedded automotive control systems, *Control Engineering Practice*, **6**, 529–540.

[28] **Bortolazzi, J. Hirth, T., Raith, T.,** 1996. Specification and Design of Electronic Control Units, *Design Automation Conference, with EURO-VHDL '96 and Exhibition, Proceedings EURO-DAC '96*, Europan, September 16-20, 36-41.

[29] **Müller-Glaser, K. D., Frick, G., Sax, E., Kühl, M.,** 2004. Multiparadigm Modelling in Embedded System Design, *IEEE Transactions on Control Systems Technology*, **12**, No. 2, March.

[30] **Giusto, P., Demmeler, T.,** 2003. Rapid Design Exploration of Safety-Critical Distributed Automotive Applications via Virtual Integration Platforms, *The Journal of Systems and Software 70*, 245-262.

[31] **NI,** 2004. http://sine.ni.com/apps/we/nioc.vp?cid=10971&lang=US, 15 Ağustos.

[32] **Aksun Güvenç, B., Sencer, B., Giray, M., Güvenç, L**., 2004. Use of a Simulink Engine Blockset in Real Time Hardware in the Loop Simulations, *IEEE International Conference on Mechatronics*, İstanbul, June 3-5.

[33] **Aksun Güvenç, B., Sencer, B., Giray, M., Güvenç, L.,** 2004. Matlab/Simulink Ortamında Çalışan Kontrol Amaçlı Bir İçten Yanmalı Motor Kütüphanesi, *OTEKON'04, Otomotiv Teknolojileri Kongresi*, Bursa, 21-23 Haziran.

[34] **dSPACE**, 2004. http://www.dspace.de/shared/data/pdf/flyer2004/dspacesimulator_mid-size_flyer_2004.pdf, 15 Ağustos.

[35] **dSPACE**, 2004. http://www.dspace.de/ww/en/pub/products/ecutest/hardwarde_concepts/dspace_simulator_compact.htm, 15 Ağustos.

[36] **dSPACE**, 2004. http://www.dspaceinc.com/shared/data/pdf/katalog2004/dspace_ds1103_flyer2004.pdf, 15 Ağustos.

[37] **dSPACE**, 2004. http://www.dspace.de/ww/en/pub/products/prodover/singbord/conledpanels.htm, 15 Ağustos.

[38] **The MathWorks**, 2004. https://tagteamdbserver.mathworks.com/ttserverroot/Download/20612_91044v01_xPC_TrgBx.pdf, 16 Ağustos.

[39] **NI**, 2004. http://www.ni.com/pdf/manuals/322072b.pdf, 15 Ağustos.

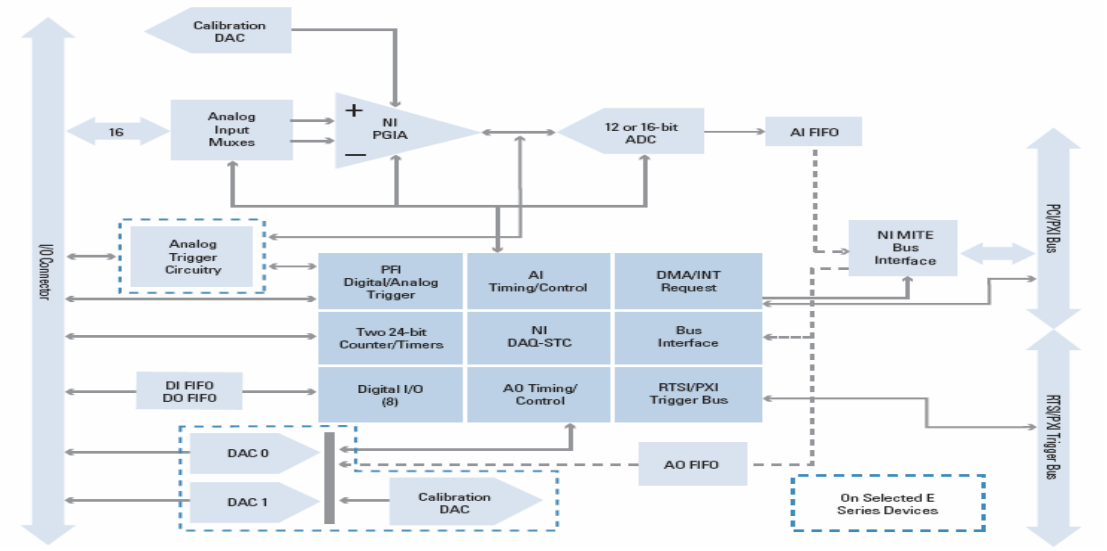
[40] **Diamond Systems Corporation,** 2004. http://www.diamondsystems.com/files/binaries/DMM32v2.64.pdf, 16 Ağustos.

EKLER

EK-A NI PCI-6025E Kartının Blok ve Pin Diyagramı
EK-B S-Function ile Motor Devri Verisi Oluşturma
EK-C DS 1103 PPC Kontrolcü Kartının Blok Diyagramı
EK-D DIAMOND MM-32 AT Kartının Blok ve Pin Diyagramı
EK-E Matlab GUI ile Oluşturulan m-file

EK-A NI PCI-6025E KARTININ BLOK VE PİN DİYAGRAMI

Tez kapsamında Real-Time Windows Target ve xPC Target sistemleri ile yapılan testlerde kullanılan NI (National Instruments) PCI-6025E kartının blok ve pin diyagramı Şekil A.1 ve Şekil A.2'de gösterilmektedir.



Şekil A.1 NI PCI-6025E Kartı Blok Diyagramı [39]

AI GND	1	51	P2.7
AI GND	2	52	GND
AI 0	3	53	P2.6
AI 8	4	54	GND
AI 1	5	55	P2.5
AI 9	6	56	GND
AI 2	7	57	P2.4
AI 10	8	58	GND
AI 3	9	59	P2.3
AI 11	10	60	GND
AI 4	11	61	P2.2
AI 12	12	62	GND
AI 5	13	63	P2.1
AI 13	14	64	GND
AI 6	15	65	P2.0
AI 14	16	66	GND
AI 7	17	67	P1.7
AI 15	18	68	GND
AI SENSE	19	69	P1.6
AO 0	20	70	GND
AO 1	21	71	P1.5
NC	22	72	GND
AO GND	23	73	P1.4
D GND	24	74	GND
P0.0	25	75	P1.3
P0.4	26	76	GND
P0.1	27	77	P1.2
P0.5	28	78	GND
P0.2	29	79	P1.1
P0.6	30	80	GND
P0.3	31	81	P1.0
P0.7	32	82	GND
D GND	33	83	P0.7
+5 V	34	84	GND
+5 V	35	85	P0.6
AI HOLD	36	86	GND
EXT STROBE	37	87	P0.5
PFI 0/AI START	38	88	GND
PFI 1/REF TRIG	39	89	P0.4
PFI 2/AI CONV	40	90	GND
PFI 3/CTR 1 SRC	41	91	P0.3
PFI 4/CTR 1 GATE	42	92	GND
CTR 1 OUT	43	93	P0.2
PFI 5/AO SAMP	44	94	GND
PFI 6/AO START	45	95	P0.1
PFI 7/AI SAMP	46	96	GND
PFI 8/CTR 0 SRC	47	97	P0.0
PFI 9/CTR 0 GATE	48	98	GND
CTR 0 OUT	49	99	+5 V
F OUT	50	100	GND

Şekil A.2 NI PCI-6025E Kartı Pin Diyagramı [39]

EK-B S-FUNCTİON İLE MOTOR DEVRİ VERİSİ OLUŞTURMA

Tez kapsamında simulink kütüphanesindeki bloklar yardımıyla da oluşturulabilecek (signal builder gibi) istenilen herhangi bir motor devri verisi Matlab programında s-function ile hazırlandı. Aşağıda s-function ile oluşturulan 'sinüs' verisi gösterilmektedir.

```
%% File : sinüs.tlc
%%
%% Abstract:
%%      sfunction sinüs block target file
%%      See sinüs.c
%%
%% Alper Çebi,
%implements "SINUS" "C"
%function BlockTypeSetup(block,system) void
   %openfile buffer
   #define pi 3.14
   %closefile buffer
   %<LibCacheDefine(buffer)>
%endfunction
%%Function:Outputs
================================================================
%function Outputs(block, system) Output
real_T sltime;
  %assign rollVars = ["U", "Y"]
  %roll idx = RollRegions, lcv = RollThreshold, block, "Roller", rollVars
```

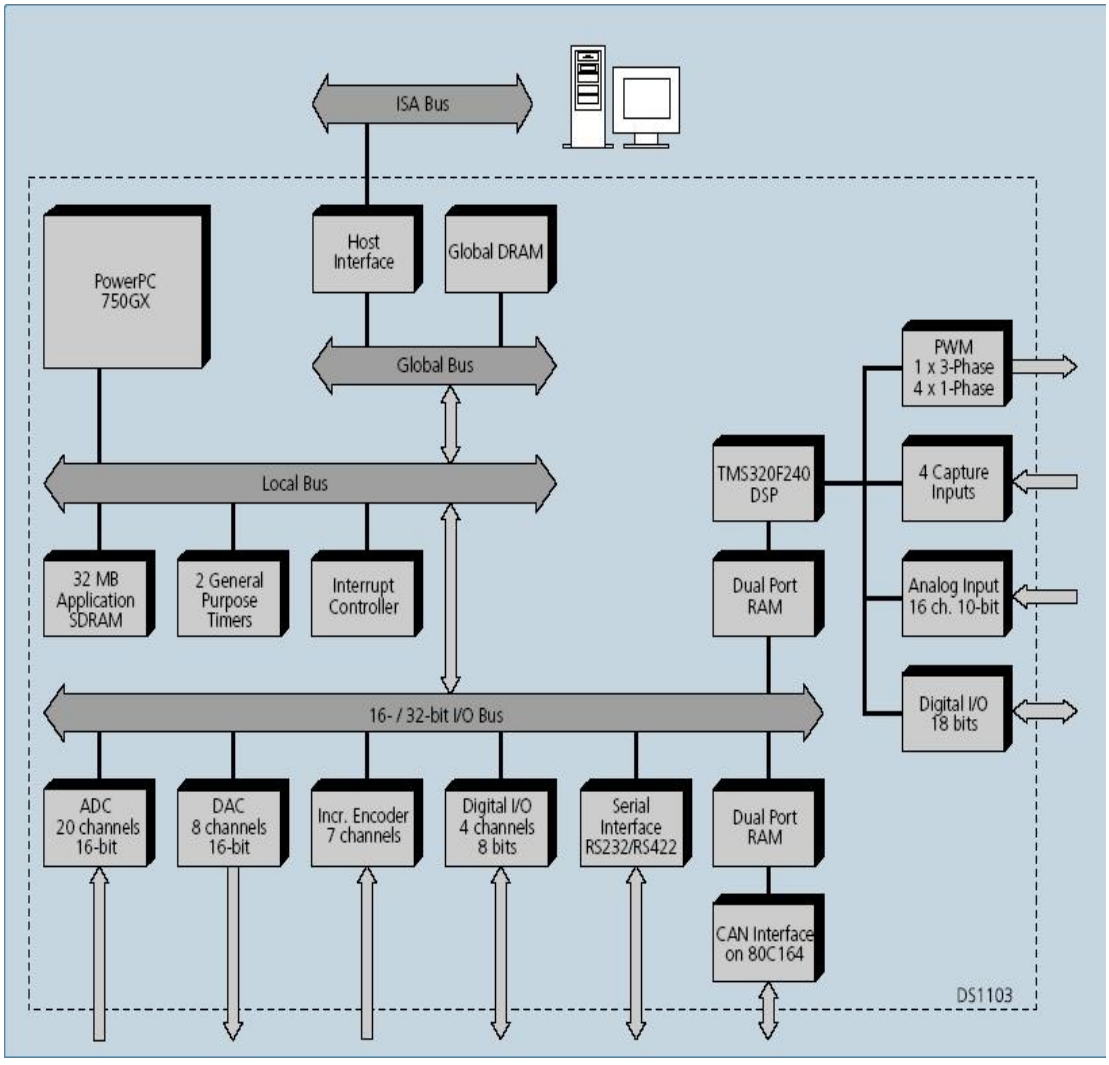
```
        %assign sinüs = LibBlockOutputSignal(0, "", lcv, idx)
        %assign u = LibBlockInputSignal(0, "", lcv, idx)
        if (%<u> < 6*22/7)
          %<sinüs>=1200;
        else
          %< sinüs>=1200+500*sin(2*22/7*sltime/4);
  %endroll
%endfunction
```

==========

Bu s-function ile motor devrinin, bir süre 1200 rpm'de kalıp sonra bir sinus dalgası şeklinde (700 rpm-1700 rpm arası) hareket etmesi sağlanmıştır.

EK-C DS 1103 PPC KONTROLCÜ KARTININ BLOK DİYAGRAMI

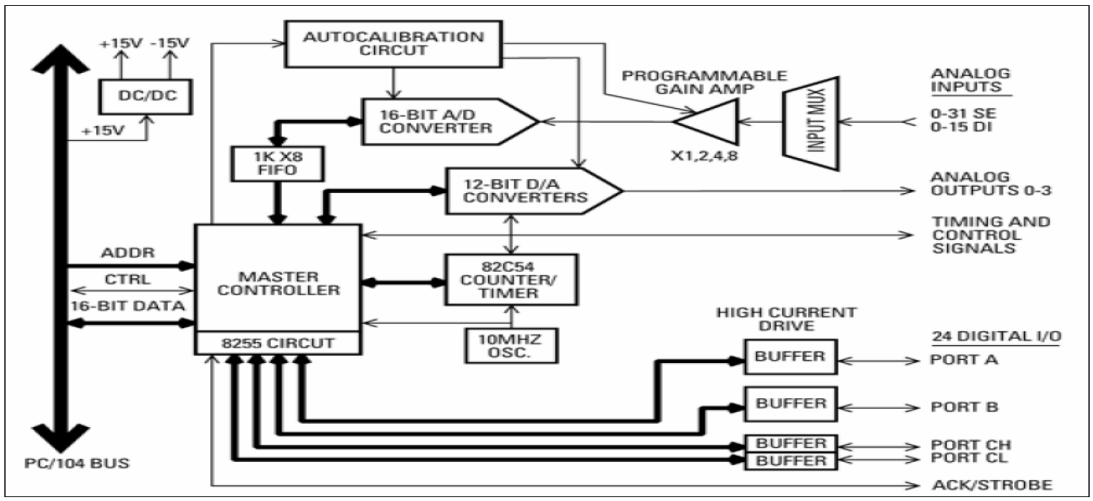
Tez kapsamında dSPACE sistemleri ile yapılan testlerde kullanılan DS 1103 PPC kontrolcü kartının blok diyagramı Şekil C.1'de gösterilmektedir.



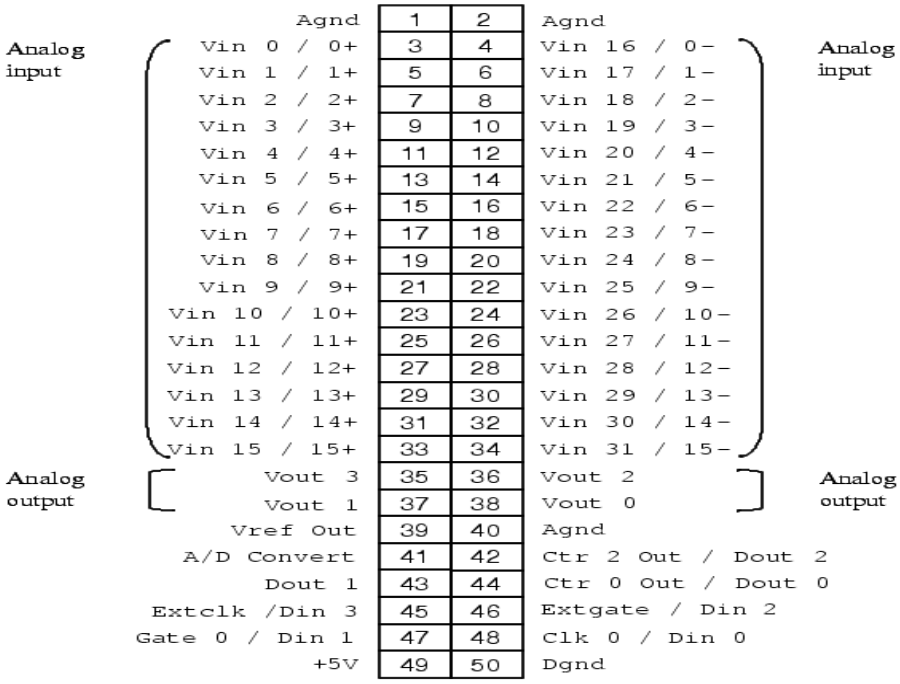
Şekil C.1 DS 1103 Kontrolcü Kartı Blok Diyagramı [36]

EK-D DIAMOND MM-32 AT KARTININ BLOK VE PİN DİYAGRAMI

Tez kapsamında xPC TargetBox sistemi ile yapılan testlerde kullanılan DIAMOND MM-32 AT kartının blok ve pin diyagramı Şekil D.1 ve Şekil D.2'de gösterilmektedir.



Şekil D.1 DIAMOND MM-32 AT Kartının Blok Diyagramı [40]



Şekil D.2 DIAMOND MM-32 AT Kartının Pin Numaralandırılması [40]

EK-E MATLAB GUI İLE OLUŞTURULAN M-FILE

Bu tez çalışmasında GUI'de oluşturulan modelin Matlab m-file'ı aşağıda gösterilmektedir.

```
function varargout = test(varargin)
% TEST M-file for test.fig
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @test_OpeningFcn, ...
                   'gui_OutputFcn',  @test_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin & isstr(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
function test_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
evalin('base','dn=[];');
     evalin('base','Cam =[];');
```

```
    evalin('base','A = [0 0 0 0 0 0 0];');
    evalin('base','kA= [0 0; 0 0; 0 0; 0 0; 0 0; 0 0; 0 0];');
    evalin('base','j_GUI=1;');
function varargout = test_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function popupmenu2_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function popupmenu2_Callback(hObject, eventdata, handles)
function popupmenu3_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function popupmenu3_Callback(hObject, eventdata, handles)
function pushbutton1_Callback(hObject, eventdata, handles)
d2=str2double(get(handles.edit1,'String'));
d1 = get(handles.popupmenu2,'Value');
if (d1 == 1)
   if ((d2 <= 0) | (d2 > 58))
   msgbox('Change Crank Tooth Number!','Crank Tooth Number Error','error');
   else
     data1 = 'Crank Signal';
     data2 = get(handles.edit1,'String');
```

```
      data3_t = get(handles.popupmenu3,'Value');
      if (data3_t == 1)
         data3 = 'No Signal';
      elseif (data3_t == 2)
         data3 = 'Random tooth';
      elseif (data3_t == 3)
         data3 = 'Disturbed Sine Signal';
      end
      data4 = get(handles.edit2,'String');
      datalb = get(handles.listbox2,'String');
      datalb{length(datalb) + 1} = strcat(data1,'-', data2 ,'-has-', data3 ,'-with-', data4);
      set(handles.listbox2,'String',datalb);
      evalin('base','i = 1;');
      assignin('base','temp',str2double(get(handles.edit1,'String')));
      evalin('base','dn(j_GUI,i) = temp;');
      evalin('base','i = i + 1;');
      assignin('base','temp',get(handles.popupmenu3,'Value'));
      evalin('base','dn(j_GUI,i) = temp;');
      evalin('base','i = i + 1;');
      assignin('base','temp',str2double(get(handles.edit2,'String')));
      evalin('base','dn(j_GUI,i) = temp;');
      evalin('base','j_GUI = j_GUI + 1;');
    end
  elseif (d1 == 2)
     if ((d2 <= 0) | (d2 > 7))
    msgbox('Change Cam Tooth Number!','Cam Tooth Number Error','error');
    else
      data1 = 'Cam Signal';
```

```
    data2 = get(handles.edit1,'String');
    data3_t = get(handles.popupmenu3,'Value');
    if (data3_t == 1)
       data3 = 'No Signal';
    elseif (data3_t == 2)
       data3 = 'Random tooth';
    elseif (data3_t == 3)
       data3 = 'Disturbed Sine Signal';
    end
    data4 = get(handles.edit2,'String');
    datalb = get(handles.listbox2,'String');
    datalb{length(datalb) + 1} = strcat(data1,'-', data2 ,'-has-', data3 ,'-with-', data4);
    set(handles.listbox2,'String',datalb);
    assignin('base','temp',d2);
    assignin('base','temp1',get(handles.popupmenu3,'Value'));
    assignin('base','temp2',str2double(get(handles.edit2,'String')));
    evalin('base','A(temp) = 1;');
    evalin('base','kA(temp,1) = temp1;');
    evalin('base','kA(temp,2) = temp2;');
   end
end
function pushbutton2_Callback(hObject, eventdata, handles)
d_lb = get(handles.listbox2,'String');
d_vl = get(handles.listbox2,'Value');
a_GUI = length(d_lb);
b_GUI = strfind(d_lb{d_vl},'Crank');
if (length(b_GUI) == 0)
   %Cam Delete.
```

```
    b_GUI = findstr(d_lb{d_vl}, '-');
    c_GUI = d_lb{d_vl}((b_GUI(1)+1):(b_GUI(2)-1));
    c_GUI_dbl = str2double(c_GUI);
    assignin('base','temp_GUI_1',c_GUI_dbl);
    evalin('base','A(temp_GUI_1) = 0');
else
    b_GUI = findstr(d_lb{d_vl}, '-');
    c_GUI = d_lb{d_vl}((b_GUI(1)+1):(b_GUI(2)-1))
    c_GUI_dbl = str2double(c_GUI);
    assignin('base','temp_GUI_1',c_GUI_dbl);
    evalin('base','I = find(dn(:,1,1)==temp_GUI_1)');
    evalin('base','dn(I,:) = []');
end
 d_lb(d_vl) = [];
set(handles.listbox2,'String',d_lb);
function pushbutton3_Callback(hObject, eventdata, handles)
evalin('base','Crank_Cam');
function pushbutton4_Callback(hObject, eventdata, handles)
evalin('base','clc');
evalin('base','clear');
evalin('base','dn=[];');
      evalin('base','Cam =[];');
      evalin('base','A = [0 0 0 0 0 0 0];');
      evalin('base','kA= [0 0; 0 0; 0 0; 0 0; 0 0; 0 0; 0 0];');
      evalin('base','j_GUI=1;');
      data = [' '];
      celldata = cellstr(data);
set(handles.listbox2,'String',celldata);
```

```
function listbox1_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function listbox1_Callback(hObject, eventdata, handles)
function edit10_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function edit10_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function edit2_Callback(hObject, eventdata, handles)
function Untitled_1_Callback(hObject, eventdata, handles)
function pushbutton2_ButtonDownFcn(hObject, eventdata, handles)
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

```
end
function edit1_Callback(hObject, eventdata, handles)
function pushbutton5_Callback(hObject, eventdata, handles)
function listbox2_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
function listbox2_Callback(hObject, eventdata, handles)
```

ÖZGEÇMİŞ

Alper ÇEBİ 1980 yılında İzmir'de doğdu. İlk ve orta öğrenimini tamamladıktan sonra lise tahsilini İzmir Selma Yiğitalp Lisesi'nde gerçekleştirdi. 1997 yılında İzmir Dokuz Eylül Üniversitesi Makina Mühendisliği Bölümünü kazandı ve bir sene dil eğitimi için İngilizce hazırlık okuduktan sonra 2002 yılında mezun oldu.

2002 yılında İTÜ Fen Bilimleri Enstitüsü Disiplinlerarası Mekatronik Mühendisliği Programı'nda Yüksek Lisans öğrenimine başladı.