T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HAVALI SÜSPANSİYONLU AĞIR TİCARİ ARAÇLAR İÇİN YÜKSEKLİK KONTROL SİSTEMİ YAZILIMI GELİŞTİRİLMESİ VE DEVREYE ALINMASI

Cansu ŞAHBAZ BULUT

YÜKSEK LİSANS TEZİ

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Danışman

Dr. Öğretim Üyesi Hüseyin ÜVET

Şubat, 2019

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HAVALI SÜSPANSİYONLU AĞIR TİCARİ ARAÇLAR İÇİN YÜKSEKLİK KONTROL SİSTEMİ YAZILIMI GELİŞTİRİLMESİ VE DEVREYE ALINMASI

Cansu ŞAHBAZ BULUT tarafından hazırlanan tez çalışması 19.03.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı, Mekatronik Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğretim Üyesi Hüseyin ÜVET

Yıldız Teknik Üniversitesi

Danışman

Jüri Üyeleri

Dr. Öğretim Üyesi Hüseyin ÜVET, Danışman _____

Yıldız Teknik Üniversitesi

Dr. Öğretim Üyesi Kadir ERKAN, Üye _____

Yıldız Teknik Üniversitesi

Doç. Dr. Özgür Koray ŞAHİNGÖZ, Üye _____

İstanbul Kültür Üniversitesi

Danışmanım Dr. Öğretim Üyesi Hüseyin ÜVET sorumluluğunda tarafımca hazırlanan Havalı Süspansiyonlu Ağır Ticari Araçlar İçin Yükseklik Kontrol Sistemi Yazılımı Geliştirilmesi Ve Devreye Alınması başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim

Cansu ŞAHBAZ BULUT

İmza

Tez çalışması boyunca bütün imkanları ile destek olan Ford Otosan ailesine, başta şefim olmak üzere bütün yöneticilerime çok teşekkür ederim.

Desteği ve bilgisiyle hep yanımda olan; Taylan, Emre, Mesut ve bütün ekip arkadaşlarıma minnettarım.

Tez çalışmamın başından sonuna kadar desteği ve anlayışı ile yol gösteren danışman hocam Hüseyin Üvet'e teşekkür ederim.

Tezimi; kişisel olarak, bugünüme temel olan canım aileme ve herşeyiyle hep yanımda olan sevgili eşime adıyorum.

Saygılarımla.

İÇİNDEKİLER

KISALTMA LİSTESİ.....vii

ŞEKİL LİSTESİ.....viii

ÖZET.....xi

ABSTRACT.....xiii

1 Giriş 1

1.1. Literatür Özeti 1

1.1.1. Havalı (Pnömatik) Süspansiyon ve Hava Körükleri (Yayları).....3

1.2 Tezin Amacı 6

1.3 Hipotez 7

2 Elektronik Havalı Süspansiyon Kontrol Sistemi ve Sistemin Elektronik Parçaları 8

2.1 Solenoid Valf Bloğu.....8

2.2 Yükseklik ve Basınç Sensörleri 10

2.3 Kumanda ve Elektronik Kontrol Ünitesi 11

2.4 Havalı Süspansiyon Kontrol Sistemi Araç Konfigürasyonları.....12

2.4.1. İki Nokta Kontrollü Havalı Süspansiyon Sistemi 13

2.4.2. Tek Nokta Kontrollü Havalı Süspansiyon Sistemi 14

3 Elektronik Havalı Süspansiyon Kontrol Sistemi Fonksiyonları.....15

3.1 Sürüş Yüksekliğine getirme.....16

3.1.1. Otomatik Sürüş Yüksekliği 16

3.1.2. Manuel Sürüş Yüksekliği 16

3.2 Servis/Tümsek Yüksekliği.....16

3.3 Yüksek Hızda Yükseklik Ayarlama 17

3.4 Frenleme ve Viraj Alma Sırasında Seviye Ayarı Engelleme 17
3.5 Diyagnostik 18
3.6 Düşük Basınç Altında Seviye Ayarı Engelleme 18
3.7 Çekiş Yardımı 18
3.8 İlave Dingil Kaldırma 19
3.9 Araç Yükü Hesaplama ve Aşırı Yük Uyarısı 19
4 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılım Geliştirme Mimarisi 20
4.1 V Döngüsü(V-cycle) 20
4.2 Gereksinimlerin Belirlenmesi ve Yönetimi 22
4.2.1. Üst Seviye Gereksinimler 23
4.2.2. Alt Seviye Gereksinimler 24
4.2.3. Alt Seviye Test Gereksinimleri 25
4.2.4. Üst Seviye (Entegre) Test Gereksinimleri 25
4.3 Sistem Tasarımı 25
4.3.1. Üst Seviye Sistem Tasarımı 25
4.3.2. Alt Seviye Sistem Tasarımı – Veri Akış Diyagramları 26
4.4 Model Tabanlı Yazılım Geliştirme 28
4.4.1. EHSKS Model Tabanlı Yazılımı 30
4.4.1.1. AUTOSAR Uyumluluk 30
4.4.1.2. EHSKS Yazılımının Etkileşim/İletişim Ara yüzü 36
4.4.1.3. EHSKS Yazılım Geliştirme Kütüphaneleri 44
4.4.1.4. EHSKS Yazılımı Ana Fonksiyon Bloğu 47
4.4.1.5. EHSKS Yazılımı Kontrolcü Ara yüzü 60
5 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılım Geliştirme Testleri ve Validasyonu 67
5.1. Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Statik Kod Analizi 68

5.2 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Test Seviyeleri 69
5.2.1. Bileşen (Birim) Testi 70
5.2.2. Entegrasyon Testi 70
5.2.3. Sistem Testi 70
5.2.4. Kabul Testi 70
5.3 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Döngü İçinde Model Testi 71
5.4 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Döngü İçinde Donanım Testi 74
5.5 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Araç Testleri 86
5.5.1. Araç testi için yazılım implementasyonu 88
6 Sonuç ve Öneriler 90

KISALTMA LİSTESİ

AUTOSAR	Automotive Open System Architecture
ASM	Automotive Simulation Models
BCM	Body Control Module
BSW	Base Software
CAN	Controller Area Network
DID	Data Identifier
DTC	Diagnostic Trouble Code
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EES	Electrical Error Simulation
EHSKS	Elektronik Havalı Süspansiyon Kontrol Sistemi
EKÜ	Elektronik Kontrol Ünitesi
EOL	End Of Line
FIU	Failure Injection Unit
HIL	Hardware In the Loop
LIN	Local Interconnect Network
NVRAM	Non Volatile Random Access Memory
MBD	Model Based Development
MIL	Model In the Loop
OEM	Orijinal Ekipman Üreticisi
RCU	Remote Control Unit
RCP	Rapid Control Prototype
RM	Requirement Management
SIL	Software In the Loop
SUT	System Under Tests
UDS	Unified Diagnostic Services
VBF	Versatile Binary Format
VSEM	Vehicle Software and Electrical Management

ŞEKİL LİSTESİ

Şekil 1.1 Süspansiyon çelik yayları [7] 4
Şekil 1.2 Süspansiyon hava yayları [8] 4
Şekil 2.1 Sırasıyla Tek kanallı ve çift kanallı valf bloğu 9
Şekil 2.2 Valf bloğunun araç üzerine implementasyonu 9
Şekil 2.3 Yükseklik sensörü eşdeğer devresi ve sensörün genel görünümü 10
Şekil 2.4 Yükseklik sensörü ve sensör kullanın aksa bağlantısı 10
Şekil 2.5 Basınç sensörü ve araçta implementasyonu 11
Şekil 2.6 EHSKS Kumandası ve Kumandanın araç üzerinde konumu 12
Şekil 2.7 C4AS arka havalı süspansiyon yandan görünümü 13
Şekil 2.8 C2AS arka havalı süspansiyon yandan görünümü 13
Şekil 3.1 Elektronik Havalı Süspansiyon Kontrol Sistemi fonksiyonları 15
Şekil 4.1 V döngüsü akış şeması 21
Şekil 4.2 EHSKS yazılımı gereksinimlerinin, VSEM gereksinim yönetim sistemi üzerinden oluşturulması ve takibi 22
Şekil 4.3 Elektronik Havalı Süspansiyon Kontrol Sisteminin Dizayn Hata Modları Efekt ve Analizleri Dokümanı örnek madde Görünümü 24
Şekil 4.4 Elektronik Havalı Süspansiyon Kontrol Sistemi Fonksiyon Diyagramı 26
Şekil 4.5 "Driver Request/ Kullanıcı talepleri" Fonksiyonu Üst Seviye Veri Akış Diyagramı 28
Şekil 4.6 "Driver Request/ Kullanıcı talepleri" Fonksiyonu Alt Seviye Veri Akış Diyagramı 28
Şekil 4.7 AUTOSAR Katmanlı Yazılım Mimarisi [16] 31
Şekil 4.8 Arxml dosyasından üretilen örnek Simulink modeli 34
Şekil 4.9 Örnek bir DTC port konfigürasyonu 35
Şekil 4.10 EHSKS yazılımı; ana fonksiyon, giriş ve çıkış dönüşüm blokları 36
Şekil 4.11 EHSKS yazılımı; giriş dönüşüm bloğunda örnek bir CAN sinyali değerlendirme fonksiyonu 37
Şekil 4.12 EHSKS yazılımı; giriş dönüştürme bloğu NvRAM parametreleri okuma ve dönüştürme örnek fonksiyonu 38
Şekil 4.13 Ağır ticari bir araçta CAN hattının örnek dağılımı 39
Şekil 4.14 EHSKS yazılımı geliştirilen aracın CAN_BUS topolojisi 41
Şekil 4.15 EHSKS yazılımı; DTC (hata kodu) oluşturma fonksiyonları 43
Şekil 4.16 EHSKS yazılımı; EHSKS ortak kullanım fonksiyonları kütüphanesi 45
Şekil 4.17 EHSKS yazılımı; EHSKS diyagnostik fonksiyonları kütüphanesi 45
Şekil 4.18 EHSKS yazılımı; fonksiyonel sistem fonksiyonları kütüphanesi 46

Şekil 4.19 EHSKS yazılımı; giriş fonksiyonları kütüphanesi 46
Şekil 4.20 EHSKS yazılımı; çıkış fonksiyonları kütüphanesi 47
Şekil 4.21 EHSKS yazılımı; ana fonksiyon bloğu(RCP) genel görünümü 49
Şekil 4.22 EHSKS yazılımı; ana fonksiyon bloğu giriş odaklı çalışan fonksiyonlara genel bakış 51
Şekil 4.23 EHSKS yazılımı; pnömatik proses fonksiyon bloğuna genel bakış 52
Şekil 4.24 EHSKS yazılımı; ana fonksiyon bloğu, kullanıcı girişlerinin işlendiği fonksiyon grubuna genel bakış 53
Şekil 4.25 EHSKS yazılımı, dingil yükü hesaplama bloğu 55
Şekil 4.26 EHSKS yazılımı, dingil yükü değerinin gösterge üzerinden verilmesi 55
Şekil 4.27 EHSKS yazılımı, diyagnostik sistemler fonksiyon bloğu 56
Şekil 4.28 EHSKS yazılımı, diyagnostik sistemler fonksiyon bloğu kalibrasyon alt fonksiyonları 57
Şekil 4.29 EHSKS yazılımı, hata durumunun göstergede kullanıcıya iletilmesi 58
Şekil 4.30 EHSKS yazılımı, hata ara yüzü fonksiyon bloğu 59
Şekil 4.31 EHSKS yazılımı, kontrolcü çalışmasını kesme fonksiyonu 60
Şekil 4.32 EHSKS yazılımı, kontrolcü yapısı 61
Şekil 4.33 EHSKS yazılımı, kazanç ayarlamalı PD kontrolcü yapısı 61
Şekil 4.34 Yüksüz araçta kontrolcü testi verisi 62
Şekil 4.35 10ton yüklü araçta kontrolcü testi verisi 63
Şekil 4.36 EHSKS yazılımı, kontrolcü fonksiyonu 63
Şekil 4.37 EHSKS yazılımı, kontrolcü tolerans değeri ve tolerans dışı süre hesaplama grafiği 64
Şekil 4.38 EHSKS yazılımı, araç dayanım testi sırasında kontrolcü davranışı grafiği 66
Şekil 5.1 EHSKS yazılımı; Statik kod analizi için kullanılan PolySpace programı ara yüzü 69
Şekil 5.2 EHSKS yazılımı; döngü içinde model testi için yazılan örnek bir test kodu 72
Şekil 5.3 EHSKS yazılımı; döngü içinde model testi için yazılan örnek bir test kodu 73
Şekil 5.4 EHSKS yazılımı; döngü içinde model testi için kullanılan "Matlab Coverage Test" raporu 74
Şekil 5.5 HIL(Hardware In the Loop) testlerinin temel prensibi 76
Şekil 5.6 HIL(Hardware In the Loop) test platformlarının mimarisi 77
Şekil 5.7 HIL testlerinin V-döngüsündeki yeri 79
Şekil 5.8 EHSKS yazılımı HIL testlerinin gerçekleştirilme alanı 80

Şekil 5.9 EHSKS yazılımı HIL testleri için kullanılan D-Space Scalexio platformları ve kullanım yapısı 81

Şekil 5.10 HIL testi sırasında yükselme girişi verilen sistemin ilk durum değişimi 82

Şekil 5.11 HIL testi sırasında sürüş yüksekliği girişi verilen sistemin durum değişimi 83

Şekil 5.12 HIL testi sırasında araç statik konumda iken ani yük değişimi başlangıç durumu 83

Şekil 5.13 HIL testi sırasında araç statik konumda iken ani yük değişimi ile aracın yükselmesi 84

Şekil 5.14 HIL testi sırasında araç statik konumda iken ani yük değişimi sonrasında sürüş yüksekliğine oturması 85

Şekil 5.15 HIL testi sırasında düzensiz engebeli yolda dinamik araç tepkisi 85

Şekil 5.16 Devrilme (Roll) hareketi HIL testi 86

Şekil 5.17 Bozuk yol testi koşan bir araç 87

Şekil 5.18 Araç test parkurunun görünüşü 87

ÖZET

Havalı Süspansiyonlu Ağır Ticari Araçlar İçin Yükseklik Kontrol Sistemi Yazılımı Geliştirilmesi Ve Devreye Alınması

Cansu ŞAHBAZ BULUT

Mekatronik Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Dr. Öğretim Üyesi Hüseyin ÜVET

Tez konusu olan proje ile havalı süspansiyon sistemine sahip ağır ticari araçlardaki, Elektronik Havalı Süspansiyon Kontrol Sisteminin (EHSKS) gömülü yazılımının geliştirilmesi ve mevcut merkezi araç kontrol ünitesine gömülerek yerlileştirilmesi çalışmaları yapılmıştır. Çalışma kapsamında EHSKS yazılımı yerli olarak geliştirilmiş ve doğrulama testleri yapılmıştır. Geliştirme, doğrulama ve izlenebilirlik açısından; çalışmalar, yazılım geliştirme çalışmalarında kullanılan v-şeması takip edilerek devreye alınmıştır. Bu yazılım ve kontrolcü sayesinde; aracın, her yük ve yol durumunda seviye kontrolü sağlanarak, kullanıcının manuel girişlerine de adapte olacak bir sistem tasarımı yapılmıştır.

Bu tasarım ve yazılım geliştirme çalışması ile müdahale edilmesi oldukça zor olan ve her müşteri talebine cevap verilemeyen bir yazılım yerine esnek ve geliştirme

faaliyetlerinin sürekli olarak devam ettirilebileceği bir yazılım sistemi geliştirilmiştir.

Maliyetlerin çok kritik olduğu seri üretim prosesinde; bu proje ile, hazır olarak alınan bir elektronik kontrol ünitesi silinmiş, ekstra bir ünite gerektirmeden yazılım var olan bir elektronik kontrol ünitesine gömülü olacak şekilde geliştirilmiştir. Bu da araç başına yüksek miktarda maliyet azaltılması sağlamıştır.

Hava ile beslenen ve enerji tüketen havalı süspansiyonun hava tüketimi iyileştirilerek enerji verimliliği sağlanmıştır. Aktif sürüş durumları da değerlendirildiğinden yakıt tasarrufu sağlayacak sistemlerin geliştirilmesi için alt yapı yazılımı oluşturulmuştur.

Araç üzerinde bulunan yükler ile birlikte dingil başına düşen ağırlık hesaplanarak, sisteme ve sürücüye verilerek müşteri memnuniyeti ve sistem performans artışı sağlanmaktadır.

Araç dinamiği ve konfor açısından subjektif sürüşler açısından performansının ve süspansiyon hissiyatının iyileştirilmesi gerçekleştirilmiştir.

Bu çalışmalar sonucunda %100 oranda algoritma değişimin entegre edilebileceği bir yazılım geliştirilmiştir. % 30 oranında konfor arttırılmış, %20 oranda güvenlik anlamında daha gürbüz bir sistem tasarımı yapılmıştır.

Anahtar Kelimeler: Elektronik süspansiyon seviye kontrolü, gömülü yazılım geliştirme, havalı süspansiyon kontrolü

ABSTRACT

Development and Implementation of Electronic Level Control System for Air Suspended Heavy Commercial Vehicles

Cansu ŞAHBAZ BULUT

Department of Mechatronic Engineering

Doctor of Philosophy Thesis

Advisor: Asst. Professor Hüseyin ÜVET

Within the project, the handled subject in this paper, the embedded software of an Electronic Controlled Air Suspension (ECAS) System in heavy duty commercial vehicles with air suspension system has been developed, localized and embedded in the existing central vehicle control unit. Development of ECAS software and fulfilment of validation tests have been held locally. In terms of development, verification and traceability; the study has been carried out by following the v-scheme used in software development studies. Thanks to this software and controller, the system has been designed that can be adapted to user's manual inputs by providing vehicle height control for any loading and road conditions.

In the mass production process, where costs are very critical, high cost reduction has been provided with this project. Since developed software is embedded into the already existed ECU module instead of using external ECU for the ECAS System. Due to ECU change, higher cost reduction is provided. Air consumption has been

improved that satisfies energy efficiency since the system is supported with air and consumes energy. Since driving conditions are evaluated in the developed software. It could be used for a base software for future system developments which provide fuel saving.

Load per axle, which also includes payload of the vehicle, is calculated and supplied to system, as a result, customer satisfaction and system performance are increased. Vehicle dynamic performance and comfort are increased according to expert's subjective reviews. As a result of this study, 100% flexible and configurable software was developed. Vehicle comfort level is increased 30%. Software is developed according to robust control principals which increased software security about 20%.

Keywords: Electronic level control of air suspension, embedded software development, air suspension control

1
Giriş

1.1. Literatür Özeti

Bilgisayar teknolojisinin hızlı gelişimi ve özellikle mikroişlemci maliyetlerinin düşmesi, geçtiğimiz son yirmi yıllık süreçte araç üzerinde bulunun veri işleme ve karar verme ekipmanlarının çoğalmasına izin verdi [1]. Bu sayede araç performansının çeşitli yönlerini izlemek ve kontrol etmek için bilgisayar sistemleri kullanılmaya başlandı. Araç üzerinde bulunan elektronik sistemlerin yaygınlaşması ile bu sistemleri daha aktif kullanılması için, araştırmalar ve gelişmiş yazılım kontrol algoritmalarının geliştirilmesine ve araç üzerinde uygulanması oldukça yaygın hale gelmiştir [2]. Çok çeşitli çalışma koşullarında istenilen performansı sağlamak ve optimum sistem tepkisini elde etmek için çeşitli sistem parametrelerini sürekli izlenmesi ve sürekli olarak kontrolü günümüzde modern araçlar için olağan bir durum haline gelmiştir. Sistemler, özellikle yolcu ve sürücüyü korumak için aktif güvenlik önlemlerinin uygulanması için eklenen elektronik işlemciler ardından farklı fonksiyonları da içinde bulunduran karmaşık yapılara evrildiler.

Araçlar üzerine yerleştirilen kontrolcüler arasında taşıt süspansiyon sistemleri kontrolcüleri özel bir araştırma ve uygulama alanıdır [3]. Özellikle bilgisayar kontrolünün ileri yöntemlerinden, önemli ölçüde bu alanda aktif olarak yararlanılmaktadır. Bunun önemli sebeplerinden biri de sistemin doğası gereği belli bir denge noktasına oturma isteğidir. Özellikle konfor ve performans alanında sistem incelendiğinde, bu özelliklerden birinin iyileşmesi ile diğer özellik kötüleşmektedir.

Bilgisayar devriminden önce, araç süspansiyon sistemleri genellikle pasif olarak sınıflandırılabilirdi; özellikle sistemlerin pahalılığı ve tepki sürelerinin yetersizliğinden dolayı sistem kontrol algoritmaları bu alanda kullanılamadı. Bu sistemler üzerinde araştırma ve geliştirme genellikle, sistemin doğal frekansını ve yoldan gelen tepki frekanslarını karşılaştırarak daha iyi pasif sistem bileşenlerinin

geliştirilmesi üzerine yapılmıştır. Sistem bozucularının analizi ile araç içinde bulunan sürücü ve yolcunun rahatsızlıklarını minimize etmek hedeflenmektedir. Bilgisayar kontrol algoritmaları tipik olarak aktif veya yarı aktif olarak sınıflandırılan süspansiyon sistemlerinin kontrollünde kullanılmaktadır [4]. Bu sayede süspansiyon sistemi değişen rahatsızlık ortamlarına veya yol girişlerine sürekli olarak uyum sağlayabilmektedir. Elektronik sistemde kullanılan algoritmalar otomatik olarak sistem elemanlarını kontrol ederek askıya alınmış kütlenin hareketini azaltmaktadır. Bir anlamda yoldan gelen ve yolcu ya da sürücüye ulaşan titreşimi en aza indirmeye çalışmaktadır. Rahatsızlık verici ortam zamanla değişse de; sistem, çeşitli elektriksel sensörler ve sistem durumunu belirleyen algoritmaları kullanarak, istenen durumda tutacak şekilde sistem parametrelerini ayarlamakta ve / veya fiziksel aktüatörlere enerji vermektedir. Elektronik kontrol; sürücü veya yolcuya ulaşan titreşim seviyesini en aza indirerek, konfor seviyesi artırmakta ve insan yorgunluğu azaltmaktadır. Konfor seviyesinin artması ve sürücü yorgunluğunun azalması sonuç olarak daha güvenli bir sürüş deneyime fayda sağlamaktadır. Ek olarak, süspansiyondan araca aktarılan dinamik kuvvetleri azaltmasıyla, yapısal yorulma azalarak sistem üzerinde bulunan parçaların ömrü uzamaktadır. Aktif ve yarı aktif sistemler üzerine yapılan araştırmaların büyük çoğunluğu belirli tipte aktüatör üzerine odaklanılmaktadır [5]. Asılı kütlenin hareketini doğrudan etkileyen ayarlanabilir damperler veya hava aktüatörleri yarı aktif ve aktif kontrol sistemlerinin ana aktüatörlerindendir.

Araçlar üzerinde yarı aktif ve aktif süspansiyon sistemleri dışında, pasif sistemler içinde bulunan havalı süspansiyon sistemleri gün geçtikçe tercih edilme oranı artmaktadır [6]. Sürücü ve yolcu konforunu artırmak için özellikle tercih edilen, yarı aktif ve aktif süspansiyon sistemlerine göre fiyat ve enerji sarfiyatı açısından daha uygun olduğu için ve ayrıca yarı aktif ve aktif süspansiyonlara göre benzer konfor hissiyatını yakaladığı için tercih dilmektedir.

Yük taşımacılığını yapıldığı büyük kamyon ya da çekici gibi araçlarda, hem yükü korumada mevcut yaprak yaylı sistemlere göre daha iyi olması hem de sürücünün konforunu olumlu anlamda artırmasından dolayı özellikle tercih edilmektedir. Ayrıca ülkelerin yaptığı çeşitli araştırmalarda da belirlendiği üzere havalı

süspansiyonlu sistemler üzerinde gittiği yollara mevcut yaprak yaylı sistemlere göre daha az hasar vermektedir. Bu yüzden havalı süspansiyonlu sistemlerin artırılması için havalı süspansiyonlu araçları teşvik eden yasal düzenlemeler yapılmaktadır. Örneğin, Havalı süspansiyona sahip araçlarda taşıma kapasitenin benzer boyutlarda ve aynı dingil sayısına sahip araçlara göre bir tona yakın miktarda artırılmasını konu alan yasa yayını yapılmıştır (85/3/EEC 19 Aralık 1984) [6]. Bu yasanın da etkisiyle havalı süspansiyon uygulamaları ve araç varyantları genişlemiş, çoğu üretici havalı süspansiyon ve havalı süspansiyon üzerinde çalışmalarını yoğunlaştırmıştır.

1.1.1 Havalı (Pnömatik) Süspansiyon ve Hava Körükleri (Yayları)

Hava körükleri gibi pnömatik komponentler uzun zamandır titreşim yalıtımı için değişik sistemlerde kullanılmaktadır. Hava körükleri yapısı gereği kauçuk ve metal birleşimi bir yapıda olup, içlerine basılan hava sayesinde kuvvet oluşturulmaktadır. Genel olarak, eşdeğer bir çelik yaya göre daha küçük bir paket alanında daha yüksek bir yay oranına sahip olmanın istendiği yerlerde kullanılırlar.

Sistemin içine hava ilavesi ile istenilen yük ve yükseklik seviyesi sağlanabilmektedir. Hava yayları, süspansiyon sistemlerinde sertlik elemanı olarak geniş bir yelpazede kullanılabilmektedir. Özellikle lüks sınıfı otomobillerin süspansiyonunun dışında, büyük kamyonlar, otobüsler, demiryolu araçları, inşaat ve tarım araçlarının süspansiyon sistemlerinde havalı süspansiyon tercih edilmektedir. Büyük kamyonda araç süspansiyonu uygulamalarının ek olarak kabin ve koltuk süspansiyonlarında da havalı körükler kullanılmaktadır. Ayrıca, fabrika operasyonlarında özellikle ağır makinaların oluşturduğu titreşimin azaltılması için hava yayları kullanılmaktadır.



Şekil 1.1 Süspansiyon çelik yayları [7]



Şekil 1.2 Süspansiyon hava yayları [8]

Bir hava yayı (körük) temel olarak kauçuk ve plastik ipliklerle örülmüş kumaş içine hapsolmuş bir hava sütunudur. Hava yayları farklı şekillerde ve ebatlarda tasarlanarak ve üretilebilmektedir. Konvansiyonel yayların şekil değişimine karşı tepki kuvveti oluştururken, hava yayları hacimsel değişikliklere tepki olarak kuvvet üretmektedirler.

Hava Yayların çelik yaylara göre bazı avantajları aşağıda sıralanmaktadır.

1) Hava yayları isteğe bağlı olarak ayarlanabilen yük taşıma kapasitesine sahiptir. Sistem daha yüksek oranda bir yük taşıması gerektiği durumlarda yerleştirildiği bölgede bulunan alana bağlı olarak, yayı taşıma kapasitesini arttırmak için (otomatik veya manuel) hava eklenebilir ya da bağlama yüksekliği azaltılarak istenen kuvvet elde edilebilmektedir. Hava yaylarının taşıma büyüme eğrilerinde aynı kuvvet değeri için farklı basınç ve yükseklik noktaları bulunabilmektedir. Hava

yaylarının bu özelliği sayesinde süspansiyon sistemleri basınç ya da yüksekliğini koruyacak şekilde kontrol edilebilmektedir.

2) Hava yaylarının yapısal özelliği gereği üzerine uygulanan yükün artığı durumlarda basıncı artırılarak sistem denge noktası aynı olacak şekilde ayarlanabilmektedir. Bunun sonucu olarak sistemin sertliği de yüke doğru orantılı olarak artmaktadır. Bu yüzden sistemin doğal frekansı neredeyse aynı olacak şekilde değişmeden kalmaktadır.

3) Yükün yüksekliği gerektiğinde veya kullanıcının isteğine bağlı olarak, hava yayının içinde bulunan hava miktarının artırılıp veya azaltılmasıyla ayarlanabilmektedir. Yük seviyesinin ayarlanması işlemi özellikle kamyon araçlarının yükleme alanlarına yanaşması ve dorse bağlama işlemleri için oldukça önemlidir. Ayrıca otobüs gibi yolcu taşıma araçlarında "diz çökme" olarak adlandırılan aracın bir tarafını yere yakınlaştırılması ile aracın içine binen yolculara kolaylık sağlanmasının hedeflendiği özellikle içinde kullanılmaktadır.

4) Hava yayları benzer özellikteki çelik yaylar göre daha düşük sürtünme dinamiğine sahiptir. Bu özelliği nedeniyle özellikle yüksek yükler için kullanılan yaprak yaylara göre histerezisleri daha düşüktür. Ayrıca mevcut çelik süspansiyon karşılıklarına göre yoldan araca iletilen yüksek frekans titreşimlerin sönümlemesi konusunda, yolcu konforunu iyileştirecek düzeyde hissedilir sonuçlar vermektedir.

Genel otomotiv mühendisliği penceresinden bakıldığında pasif titreşimlerin sönümlenmesi için hava yayınının mühendislik işlevi çelik yay ile aynıdır. Sistemin serbestlik derecesinden dolayı yukarı yönlü yer değiştirmeleri minimize etmek için hava yayları benzerleri çelik yaylar gibi tepki kuvveti üretirler. Hem hava yayları hem de çelik yaylar nispeten az miktarda sönüm içerdiğinden yeterli sönümüm elde edilebilmesi için genellikle sönüm elemanı olan amortisörlerle birlikte kullanılırlar. Genellikle uygulama olarak hava yayları ve amortisörler aracın üzerinde ayrı lokasyonlara konumlandırılsa da, süspansiyonun bulunduğu bölgede oluşabilecek paket alanı sıkıntısına göre birlikte çalışacak şekilde de kullanılabilirler.

90'lı yıllardan sonra aktif olarak kullanılmaya başlayan elektronik havalı süspansiyon sistemleri büyük otomotiv tedarikçilerinin geliştirdiği ürünler ile daha

kolay ulaşılabilir ve uygulanabilir hale gelmiştir. Bu tedarikçiler çoğunlukla sistemin tamamını bir paket olarak vererek, araç üreticilerinin ürettiği araç tiplerine göre konfigüre edilmiş yazılım ve donanımlar sağlamaktadır. Günümüzde de halen bu yapı kullanılmakta, sistem tedarikçilerinden araç tipi konfigürasyonundan satış sonrasına kadar destekler alınmaktadır.

Başta pnömatik valflerin mekanik şekilde sürülmesi ile çalışılmaya başlanan süspansiyon kontrol sistemleri, uzun zamandır elektronik olarak kontrol edilmektedir.

1.2 Tezin Amacı

Bu tezin amacı, Elektronik Havalı Süspansiyon Kontrol Sistemi yazılımını yerli olarak geliştirmek ve doğrulama testlerini yaparak seri üretime hazır olacak şekilde yayınlamaktır. Geliştirme, doğrulama ve izlenebilirlik açısından; çalışmalar, model tabanlı yazılım geliştirme çalışmalarında kullanılan v-şeması takip edilerek devreye alınması amaçlanmaktadır. Bu yazılım ve kontrolcü sayesinde; aracın, her yük ve yol durumunda seviye kontrolü sağlanarak, kullanıcının manuel girişlerine de adapte olacak bir sistem tasarımı yapılması amaçlanmıştır.

Oluşturulan yazılım ile sistem isteklerinin hızlıca cevaplanması amaçlanmıştır. Hızlı adaptasyonu sağlayabilmek amacıyla model tabanlı yazılım mimarisi kullanılmıştır.

Var olan sistemlere oranla dinamik kontrolcü performansı arttırılmış bir yazılım oluşturulması amaçlanmış. Bu sayede sürüş konforu ve araç dinamiği etkileri iyileştirilmiştir.

Var olan sistemlerin yükten bağımsız kontrolcü algoritmaları kullanmasının sistem üzerindeki negatif etkilerini en aza indirmek amaçlanmış, bunun için yük kazanç katsayıları kullanılmıştır. Kontrolcü, bu katsayılar ile adaptif olarak kontrol edilmiştir.

Adetlerin oldukça yüksek, maliyetlerin çok kritik olduğu seri üretim prosesinde, bu proje ile araç başına yüksek miktarda maliyet azaltılması sağlanması en önemli hedeflerden biridir. Hava ile beslenen ve enerji tüketen havalı süspansiyonun hava tüketimi iyileştirilerek enerji verimliliği; araç yükü sisteme verilerek müşteri memnuniyeti ve sistem performans artışı sağlamak amaçlanmaktadır.

1.3 Hipotez

Kazanç ayarlamalı bir PD kontrolcü kullanılarak, bütün araç fonksiyonları göz önünde bulundurularak oluşturulmuş bir elektronik havalı süspansiyon sistemi tasarımı yapılmıştır. Mevcut hali ile eyleyici ve sensörleri ile birlikte yurtdışındaki imalatçılardan paket halinde alınan bir sistem, Matlab Simulink ortamında model tabanlı yazılım geliştirme ve validasyonu ile birlikte bütün halinde tasarlanmıştır.

Sistem tasarımı aşamasında, sensör ve eyleyici seçiminden yazılım testlerinin raporlanması aşamasına kadar geliştirilmiş bir bilgi birikimi kullanılmıştır. Yurtdışından alınan sistemlerden daha az hava tüketen, çok daha ucuz ve bilgi birikimini arttıran bir çalışma oluşturulmuştur.

Mevcut sistemlerin müdahale edilemez yapısı ortadan kaldırılmış, her sistem ve kullanıcıya göre adapte edilebilir bir çalışma geliştirilmiştir.

Ayrıca mevcut sistemlerin dinamik olarak sürüş sırasında kontrol yapısının oldukça yavaş olması sebebiyle negatif etkilenen konforun iyileştirilmesi sağlanmıştır. Mevcut sistemlerin dinamik kontrolcüleri sistem bozulma kontrolünü 60sn'de yaparken geliştirilen EHSKS yazılımı ile bu süre 5sn 'ye kadar indirilmiş ve kontrolcü tepkisi hızlandırılmıştır.

Mevcut sistemlerde besleme basıncı 10bar iken geliştirilen EHSKS ile besleme basıncı 12bar olarak düzenlenmiş ve kontrolcünün oturma zamanı büyük oranda hızlandırılmıştır.

Aynı zamanda, mevcut sistemler kontrolcü çalışmasını yükten bağımsız olarak tasarladığından, yük olmadığında sistemin hedef aşımı (overshoot) daha yüksek olmaktadır. Geliştirilen çalışmada kontrolcü katsayıları yük kazanç katsayısı ile düzenlendiğinden sistemin minimum aşım ile hedefe oturtulması çalışılmıştır.

2
Elektronik Havalı Süspansiyon Kontrol Sistemi ve Sistemin Elektronik Parçaları

Elektronik Havalı Süspansiyon Kontrol Sitemi temel olarak şasi yüksekliğinin sabit her yük ve bozucuda sabit kalması için yükseklik sensörlerinden alınan verilen işlenmesi ile hava körüklerindeki havanın kontrol edilmesini sağlayan sistemlerdir.

Körüklerin içerisinde bulunan havanın arttırılıp azaltılması ile aracın yüksekliği değiştirilebilmektedir.

Aynı zamanda kullanıcı isterlerini de sisteme alarak, dorse takma-çıkarma, yükleme yapmak için belirli yüksekliğe gelme gibi fonksiyonları yükseklik kontrolü yaparak sağlar.

Elektronik Havalı Süspansiyon Kontrol Sistemi aşağıdaki temel elemanlardan oluşmaktadır.

- Solenoid Valf Bloğu
- Yükseklik Sensörleri
- Basınç Sensörleri
- Kumanda
- Elektronik Kontrol Ünitesi

2.1 Solenoid Valf Bloğu

Havalı süspansiyon sisteminde körüklerdeki havanın kontrolünü sağlamak amacıyla pnömatik valfler kullanılır. Valfler bu sistemler için eyleyici görevini üstlenmektedir. Bu bir veya iki kanallı valf blokları içinde bulunan solenoidler elektriksel olarak tetiklenerek körükler içindeki havanın doldurulması veya boşaltılması sağlanır.

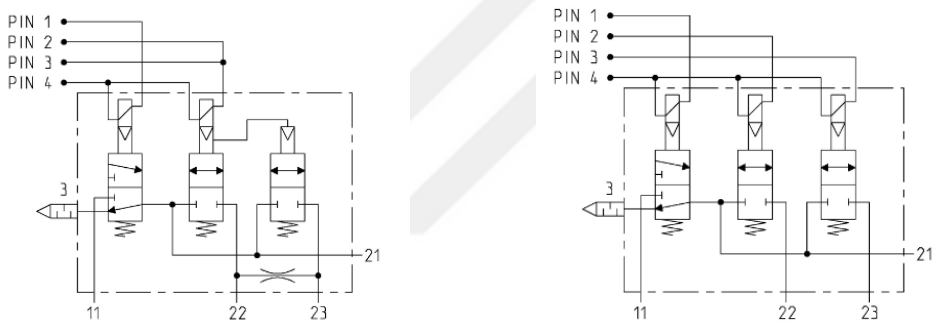
Valflerde bir adet merkez solenoid denilen 3/2 valf bulunur. Pin 4 ile tetiklenen bu solenoid, normalde boşaltma tetiklendiğinde körükleri doldurma yönünde çalışır.

Çift kanallı valflerde iki adet birbirinden bağımsız kontrol edilen 2/2 valf bulunur. Bu sayede tek valf ile iki ayrı körük ve/veya körük grubunu ayrı basınçlar ile kontrol etmek mümkün olmaktadır.

Tek kanallı valflerde bu iki 2/2lik valf ufak bir orifis ile mekanik olarak birbirine bağlıdır. Bu bağlantı bu iki valf çıkışının aynı basınç ile kontrol edilmesini sağlamaktadır.

Valf şemalarında gösterilen; 3 numaralı port, egzoz (hava çıkışı) hattını, 11 numaralı port, hava giriş hattını, 22 ve 23 numaralı portlar ise körüklere dağıtılan basınçlı hava çıkışlarını ifade etmektedir.

Buna göre, merkezi 3/2 valfi ve herhangi bir 2/2 valf tetiklendiğinde bu 2/2nin bağlı olduğu körük doldurma moduna alınmış olur. Yalnızca 2/2 valfin tetiklenmesi durumunda bu tetiklenen 2/2nin bağlı olduğu körük boşaltma moduna alınmış olur.



Şekil 2.1 Sırasıyla Tek kanallı ve çift kanallı valf bloğu

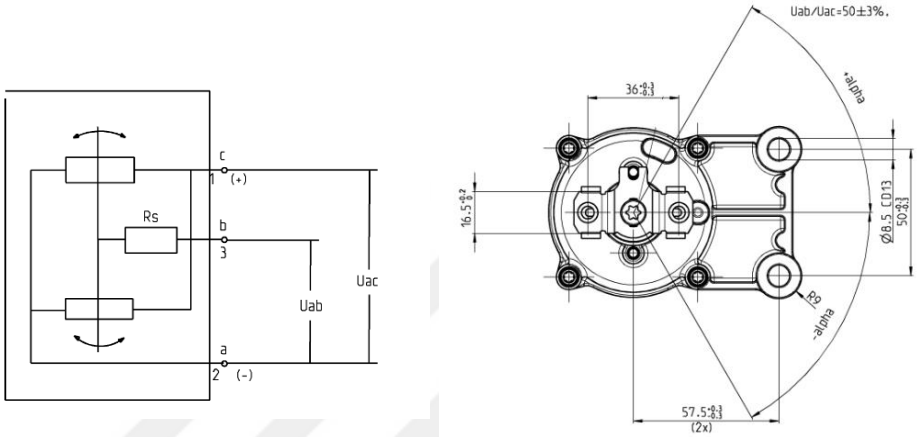


Şekil 2.2 Valf bloğunun araç üzerine implementasyonu

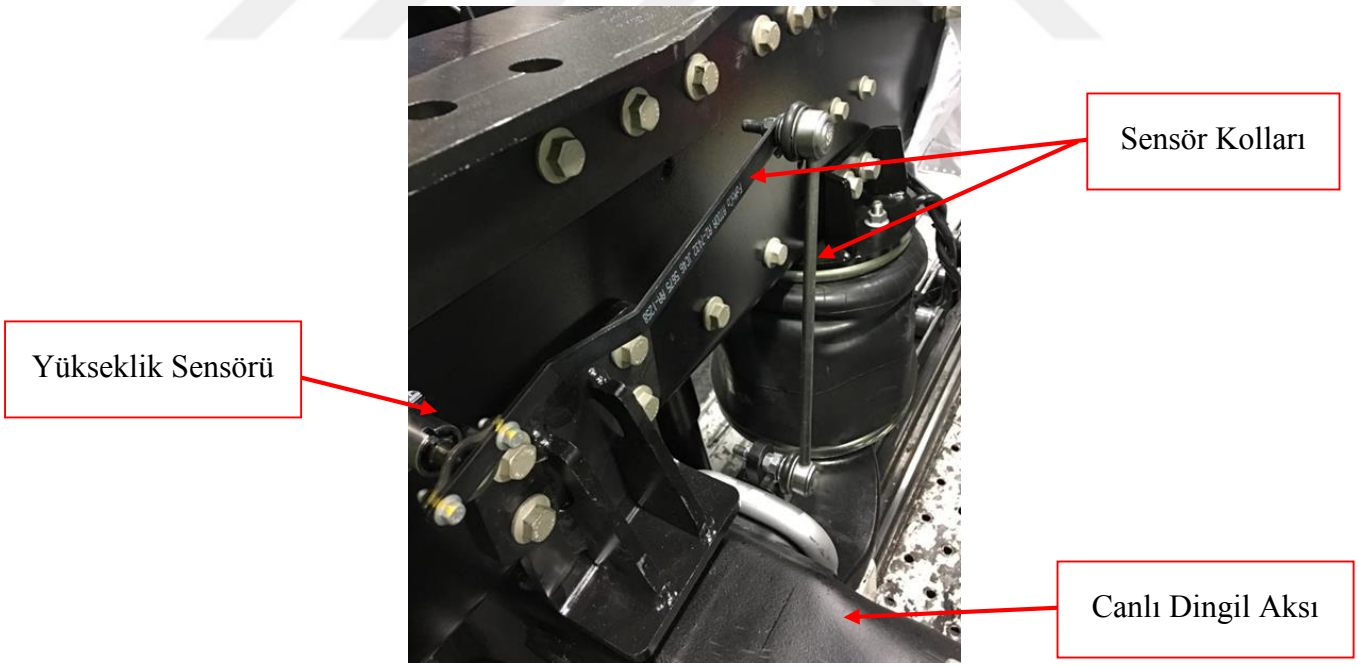
2.2 Yükseklik ve Basınç Sensörleri

EHSKS inputlarını yükseklik (deplasman) ve basınç sensörlerinden alır. Yükseklik sensörleri, şasi ve dingil arasındaki deplasman farkının ölçülmesinde kullanılır ve Şekil 2.4'de görüldüğü gibi konumlandırılır.

Sensör kolu, dingil ve şasinin deplasman farkına göre değişken bir açıya gelerek, analog bir çıkış verir. Bu analog çıkış işlemci tarafından sensörün karakteristik grafiğine göre işlenerek yükseklik verisine dönüştürülür.



Şekil 2.3 Yükseklik sensörü eşdeğer devresi ve sensörün genel görünümü



Şekil 2.4 Yükseklik sensörü ve sensör kullanın aksa bağlantısı

Aynı zamanda körüklerdeki hava basıncını ölçmek için körüklerin hava girişlerinin bulunduğu noktalarda basınç sensörleri kullanılmaktadır (Şekil 2.5).

Basınç Sensörleri göreceli olarak hava basıncını ölçer ve analog olarak sisteme sağlar. Basınç sensörlerinden alınan bu veriler, karakteristik grafik ve transfer fonksiyonu yardımıyla basın birimleri olarak KPa ve bar'a dönüştürülerek dingil üzerine düşen yük hesaplamalarında ve güvenlik fonksiyonlarında kullanılır.

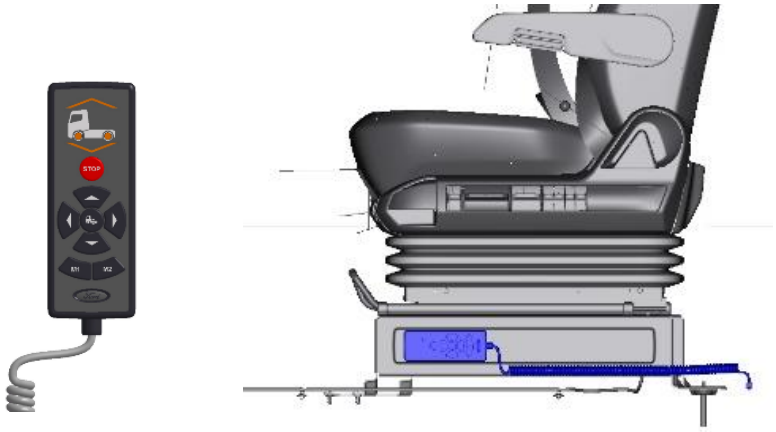
Sensörde herhangi bir sorun olursa grafikte de görüldüğü gibi tanısal aralıkta bir çıkış vererek sorun tespitinde referans oluşturur.



Şekil 2.5 Basınç sensörü ve araçta implementasyonu

2.3 Kumanda ve Elektronik Kontrol Ünitesi

Kullanıcının yükleme boşaltma aşamalarında ya da servis sırasında dışarıdan kontrol edebilmesi için kullanılan bir kumanda modülü bulunmaktadır. Bu parça, tez konusu olan çalışma ile yeniden tasarlanmış ve seri üretim için devreye alınmıştır.



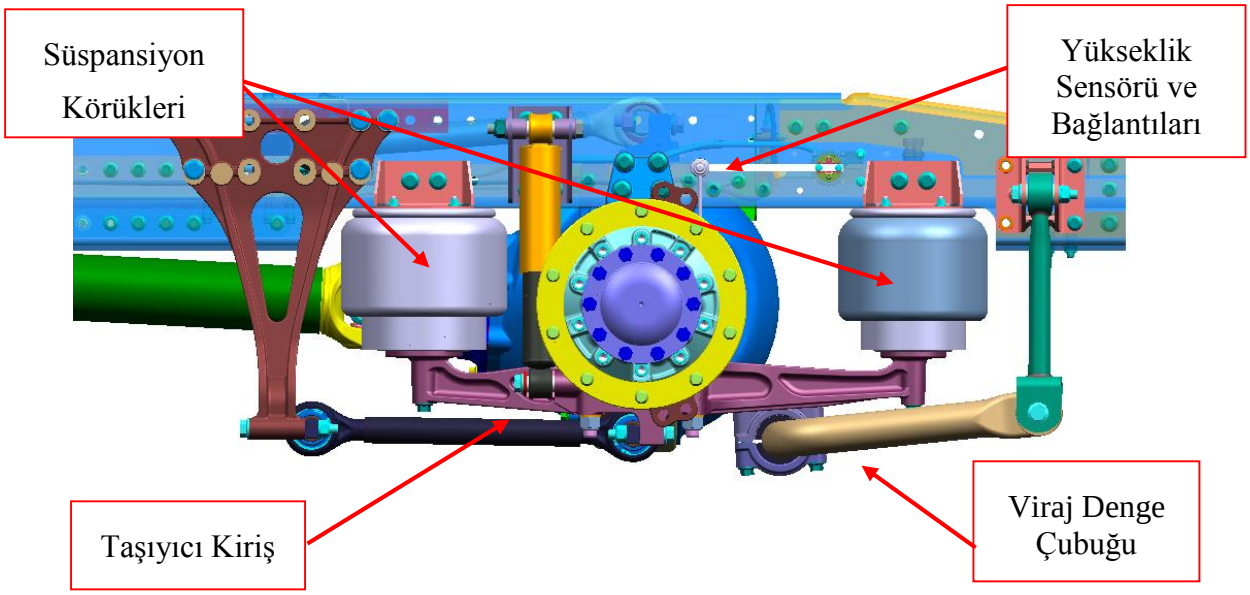
Şekil 2.6 EHSKS Kumandası ve Kumandanın araç üzerinde konumu

Kullanılan giriş sinyallerinin işlenip kontrol algoritmalarından geçirilerek valflerin sürülmesi araç üzerinde bulunan Elektronik Kontrol Ünitesi (EKÜ) tarafından sağlanır. Bu tez kapsamında sisteme özel olarak alınan EKÜ silinmiş, araçta var olan Gövde Kontrol Modülü kullanılarak sistem kontrol edilmiştir.

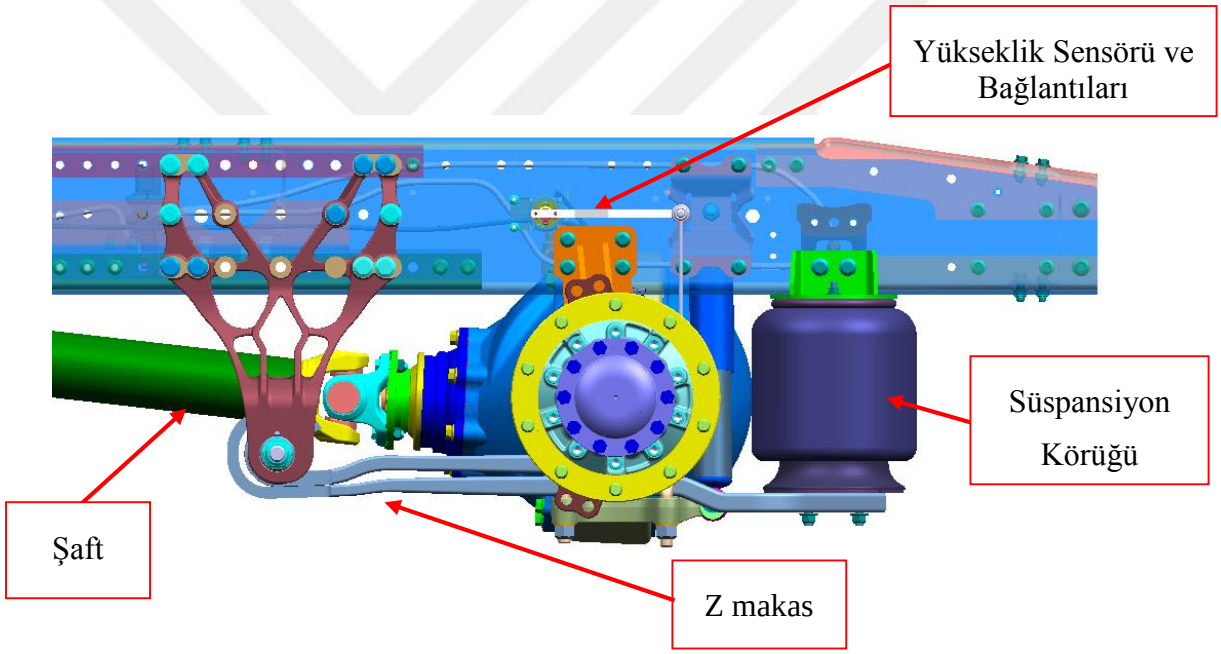
2.4 Havalı Süspansiyon Kontrol Sistemi Araç Konfigürasyonları

Havalı süspansiyon kontrol sisteminde kontrol metodu, dingil başına düşen sensör sayısı ya da kontrol edilen bölge sayısına göre adlandırılır.

Şekil 2.7 ile dört körüklü arka havalı süspansiyon sisteminin kullanıldığı C4AS, Şekil 2.8 ile iki körüklü arka havalı süspansiyon sisteminin kullanıldığı C2AS örnekleri verilmiştir. Körük veya mekanik tasarım değişikliği ile aracın devrilme (roll), doğal frekansı, titreşim ve darbe sönümleme karakteristikleri yüksek oranda değişebilmektedir.



Şekil 2.7 C4AS arka havalı süspansiyon yandan görünümü



Şekil 2.8 C2AS arka havalı süspansiyon yandan görünümü

2.4.1. İki Nokta Kontrollü Havalı Süspansiyon Sistemi

Havalı süspansiyonlu dört körüklü 4x2 çekici araçlarda (C4AS) (Şekil 2.7) olduğu gibi; iki nokta kontrollü sistemlerde, iki adet deplasman sensörü kullanılmıştır. Bu iki sensörden araç şasisinin sağ ve sol tarafının ayrı ayrı yükseklikleri hesaplanır ve çift kanallı valf kullanılarak sağ ve sol hava körükleri farklı basınçlarda doldurulup

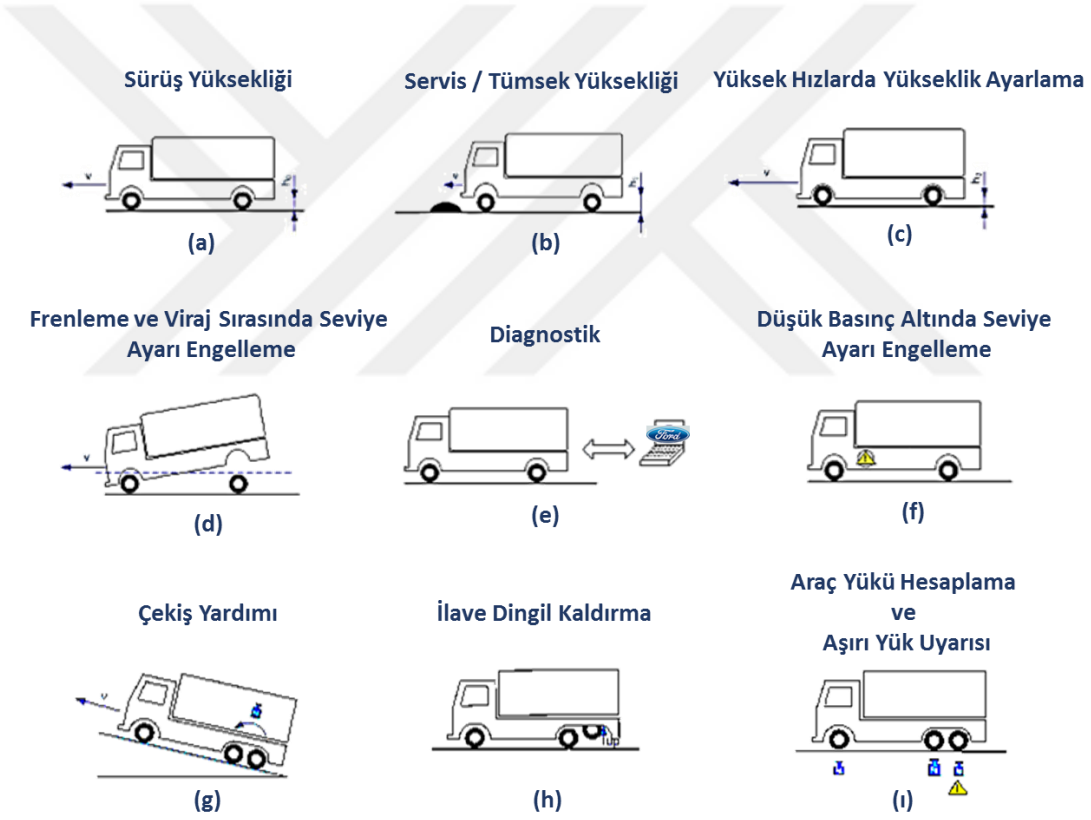
boşaltılabilir. Aynı şekilde iki körüğe bağlı olarak iki ayrı basınç sensörü kullanılmıştır. Bu yöntem ile birlikte kontrol edilen çekici araçlar mekanik tasarım ile birlikte devrilme (roll) hareketine olan direnci daha yüksek hale gelmektedir. Bu şekilde kontrol edilen ve bu mekanik yapıya sahip olan C4AS araçların yük taşıma kapasitesi daha yüksektir.

2.4.2. Tek Nokta Kontrollü Havalı Süspansiyon Sistemi

Havalı süspansiyonlu iki körüklü 4x2 çekici araçlarda (C2AS) (Şekil 2.8) kullanıldığı gibi; tek nokta kontrollü sistemlerde, bir adet deplasman sensörü kullanılmıştır. Bu sensörden araç şasisinin yüksekliği, tek bir veri üzerinden hesaplanır ve tek kanallı valf kullanılarak hava körükleri doldurulup boşaltılabilir hale gelmiştir. Tek kanallı bu valflerin çıkış portları (Şekil 3) arasında; oldukça küçük ancak bu iki portu birbirine bağlayarak hava geçişine izin veren bir kanal vardır. Bu kanal sayesinde çıkış portlarının bağlı olduğu iki körüğün de hava basıncı yaklaşık olarak aynı olarak okunur ve buna göre aracın şasisi dengeye getirme prensibi çalışılmıştır. Bu yönetim ile birlikte ve buna göre düzenlenmiş mekanik tasarım ile kontrol edilen C2AS araçların birlikte devrilme (roll) hareketine olan direnci daha yüksektir.

3
Elektronik Havalı Süspansiyon Kontrol Sistemi Fonksiyonları

Elektronik Havalı Süspansiyon Kontrol Sisteminin temel fonksiyonu; her yükleme ve hız durumunda şasi yüksekliğini belirli bir seviyede tutarak sistemin kullanıcı tarafına ilettiği bozucuların etkisini azaltmak ve araç dinamiği ve sürüş hissini olabildiğince doğrulamaktır. Bunu sağlarken birtakım ek fonksiyonları da içinde barındırır. Bunlar, Şekil 3.1'de görüldüğü üzere tek tek açıklanacaktır.



Şekil 3.1 Elektronik Havalı Süspansiyon Kontrol Sistemi fonksiyonları

3.1 Sürüş Yüksekliğine getirme

Sürüş yüksekliğine getirme fonksiyonu Elektronik Havalı Süspansiyon Kontrol Sisteminin temel fonksiyonudur (Şekil 3.1-a). Araç fabrikada seri üretim hattından çıktıktan sonra belirli parametrelerin yüklendiği kalibrasyon aşamasına gelir. Kalibrasyon adımında, havalı süspansiyon sisteminin mekanik olarak tanımlanan çalışma yüksekliğinin bu aşamada araca tanıtılması sağlanır. Sağlanan bu değer Elektronik Havalı Süspansiyon Kontrol Sistemi için "sürüş yüksekliği" olarak tanımlanır.

Sürüş yüksekliği, araçta otomatik ve manuel olarak sağlanabilir şekilde geliştirilmiştir.

3.1.1 Otomatik Sürüş Yüksekliği

Otomatik sürüş yüksekliği aracı hareketli iken sabit yükseklikte tutmak için kullanılmıştır. Bu fonksiyon ile araç dinamiği ciddi şekilde iyileştirilirken sürücü konforu da sağlanmıştır. Araç dinamik olarak belirli bir hızın üzerinde ise otomatik olarak sürüş yüksekliğine getirilir. Bu hız, yazılımda 30 km/saat olarak belirlenmiştir. Bu ve bunun gibi belirlenen değerler, üst seviye sistem gereksinimlerinden ve/veya araç dinamiği gereksinimlerinden gelmektedir. Gereksinimler ve yapıları ilerleyen başlıklarda detaylandırılmıştır.

3.1.2 Manuel Sürüş Yüksekliği

Manuel sürüş yüksekliği yükleme boşaltma yaptıktan sonra yola çıkmadan önce kullanılabildiği gibi, araç açık konumdayken yükleme ve boşaltma yapılması sırasında araç yüksekliğinin değişmemesi için de kullanılabilir. Bu talep sürücü tarafından Elektronik Havalı Süspansiyon Kontrol Sisteminin kumandası aracılığıyla Elektronik Kontrol Ünitesine (EKÜ) iletilir.

3.2 Servis/Tümsek Yüksekliği

Alçak yüksekliğe sahip araçlarda sürüş yüksekliğine ek olarak tümsek/servis yüksekliği tanımlanmıştır ve kullanıcının bunun gibi durumlarda kullanabilmesi için kabin içinde bir buton sağlanmıştır (Şekil 3.1-b). Bu butondan EKÜ'ne gelen sinyale göre tümsek/servis yüksekliği sağlanacak şekilde geliştirme yapılmıştır.

3.3 Yüksek Hızda Yükseklik Ayarlama

Araç yüksek hızlara çıktığında hava direncini en aza indirebilmek için aracın yere yaklaştırılması amacı ile sürüş yüksekliği algoritma tarafından güncellenmektedir (Şekil 3.1-c). Böylece yakıt tasarrufu ciddi anlamda arttırılır.

3.4 Frenleme ve Viraj Alma Sırasında Seviye Ayarı Engelleme

Frenleme ve/veya hızlanma sırasında sürücü konforunu en yüksek dereceye çıkarabilmek ve frenleme sırasında fren hava basınçlarını etkilememek adına, Elektronik Havalı Süspansiyon Kontrol Sisteminin kontrolcüsü çalıştırılmaz(Şekil 3.1-d).

Aynı zamanda aracın viraja girme durumu ve derecesi yazılım tarafından hesaplanarak bir skala içinde ağırlıklandırılmıştır. Bu değere göre belirlenen sınırlar dahilinde Elektronik Havalı Süspansiyon Kontrol Sisteminin kontrolcüsü çalıştırılmaz. Bu limit değerleri araç dinamiği uzmanları ile sürüş değerlendirmelerine göre devreye alınır. Aracın viraj gibi dönüş durumlarında, kontrolcü sisteminin devrilmeyi (yana yatmayı) kompanse etmek için çalışması, hava tüketimini arttırdığı gibi oldukça tehlikeli durumlara yol açabilir. Sistem çalışıp, devrilme (yana yatma) durumunda sağ ve soldaki farkı kapatmak için bir tarafa fazla hava basarken diğer tarafı indirmeye çalışacaktır. Ancak bu dönüşün ters yöne doğru başka bir dönüş ile devam etmesi durumunda araç tamamen devrilme riski ile karşı karşıya kalabilir.

Bu süreçte en optimum yöntemi kullanmak hem açılma adedine göre ömrünü dolduran valflerin ömrüne etki edecek hem tüketilen hava miktarını belirleyecek hem de aracın devrilmeye karşı korunmasını sağlayacaktır.

3.5 Diyagnostik

Araç hattan indiğinde ya da servise geldiğinde bir tanılama sistemi ile bağlanıp yapılandırma ve kalibrasyon işlemlerinin tamamlanabiliyor olması gerekmektedir (Şekil 3.1-e). Bunları sağlayabilmek adına yazılım ve tanılama sistemi için UDS (Unified Diagnostic Services) protokolü kullanılmıştır. UDS, ISO 14229-1'de belirtilen otomotiv elektroniği içindeki Elektronik Kontrol Üniteleri (EKÜ) için kullanılan bir iletişim protokolüdür. ISO 14230-3 (KWP2000) ve ISO 15765-3 (DoCAN) üzerinden türetilmiştir. Bu bağlamda UDS, şirkete özgü olmayan, uluslararası bir standarttır. Şimdiye kadar bu iletişim protokolü, Orijinal Ekipman Üreticisi (OEM) tedarikçilerinin yaptığı hemen hemen tüm yeni EKÜ'lerde kullanılmaktadır. Bu EKÜ'ler, elektronik yakıt enjeksiyonu (EFI), motor kontrol ünitesi (ECU), şanzıman, anti-kilit fren sistemi, kapı kilitleri, frenleme ve daha pek çok fonksiyonu kapsar.

Algoritmanın bu protokole uygun olması gerektiği gibi bütün kalibrasyon rutinlerini de sağlayacak şekilde oluşturulması gerekmektedir.

3.6 Düşük Basınç Altında Seviye Ayarı Engelleme

Elektronik Havalı Süspansiyon Kontrol Sisteminin aksiyon elemanları olan valflerin ventillerini doğru şekilde çalıştırabilmek için sağlanan hava basıncının belirli bir basıncın üstünde olması gerekmektedir (Şekil 3.1-f). Kullanılan valflerin ventil açma kuvvetine göre bu basınç limiti değişebildiği gibi, araç içindeki basıncın fren hava basıncının da belirli bir limitte tutulması gerekmektedir. Bu sebeple sistem içindeki toplam hava basıncı hesaplanarak ve/veya kullanılarak Elektronik Havalı Süspansiyon Kontrol Sistemi kontrolcüsünün belirlenmiş bir basıncın altında çalışmaması sağlanmıştır.

3.7 Çekiş Yardımı

Arka havalı süspansiyon sistemine sahip kaldırılabilir dingili bulunan 6x2 araçlarda, yere basan kaldırılabilir dingil, araç üzerinde bulunan yükün bir kısmını üstüne alarak canlı (çeker) dingil üzerine düşen yükü azaltmaktadır (Şekil 3.1-g). Bu yük dağılımı, mekanik parçaların ömrü ve konfor açısından avantajlar sağlamakta iken kuvvetli çekiş gerektiren durumlarda dezavantaja sebep olmaktadır. Bu durumu

ortadan kaldırabilmek için Elektronik Havalı Süspansiyon Kontrol Sistemi, kaldırılabilir dingil hava körüklerindeki havayı belirlenen basınç katsayısı oranında düşürerek bu dingilin aldığı yükün bir kısmını canlı dingil üzerine transfer eder.

3.8 İlave Dingil Kaldırma

Elektronik Havalı Süspansiyon Kontrol Sistemi kaldırılabilir dingile sahip 6x2 araçlarda, dingilin hava basıncı ile kontrol edilerek kaldırılmasını sağlar (Şekil 3.1-h). Bu sayede, kullanıcı daha fazla manevra kabiliyetine sahip olabilirken, yükleme durumuna göre doğru yük dağılımını sağlamak için dingili indirerek de kullanabilir.

3.9 Araç Yükü Hesaplama ve Aşırı Yük Uyarısı

Elektronik Havalı Süspansiyon Kontrol Sisteminin kritik fonksiyonlarından biri dingil yükü hesaplamadır. Hesaplanan dingil yükü CAN_BUS iletişimi ile fren modüllerine basılmaktadır. Basılan bu dingil yüklerine göre fren sistemi, hava basıncı ve fren kuvveti arasındaki ilişki fonksiyonu çıktılarıyla frenlerin kontrolünü sağlar (Şekil 3.1-ı).

Dingil yükleri, dingilleri süspanse eden hava körükleri içindeki basıncın değerlendirilmesi ile hesaplanmıştır. Hava körüklerine bulunan hava basıncı, göreceli olarak değer basan basınç sensörleri ile ölçülmüştür. Ölçülen basınç ve yükseklik değerleri bir doğrusallaştırma fonksiyonu ile dingil başına düşen yükün grafiğini verir. Bu sayede aracın bulunduğu yüksekliğe göre verdiği hava basıncından yük bilgisine ulaşılmıştır.

Aks yükleri toplamı, aracın kapasitesinin ve/veya limitinin üzerindeyse, kullanıcıya aşırı yük bilgisi verilir ve/veya regülasyonlar gereği bütün dingillerin yere indirilmesi sağlanır.

4
Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılım Geliştirme Mimarisi

Yazılım Geliştirme Yaşam Döngüsü (Software Development Life Cycle - SDLC) kavramı bir döngü yaklaşımdır [9]. Bir sarmal hareket halinde olan bir yaklaşımdır. Yazılım Geliştirme Yaşam Döngüsü aşamaları aşağıdaki şekilde sıralanabilir.

1. Planlama
2. Tanımlama
3. Tasarım
4. Geliştirme
5. Entegrasyon ve testler
6. Uygulama
7. Bakım

Kaynaklara göre döngünün basamakları farklı adlandırılabilir ya da sayısı artabilir. Ancak yazılım geliştirirken temel olan; aşağıdaki adımlarını takip etmek, doğru dokümantasyon ve birbirini besleyen adımlardır.

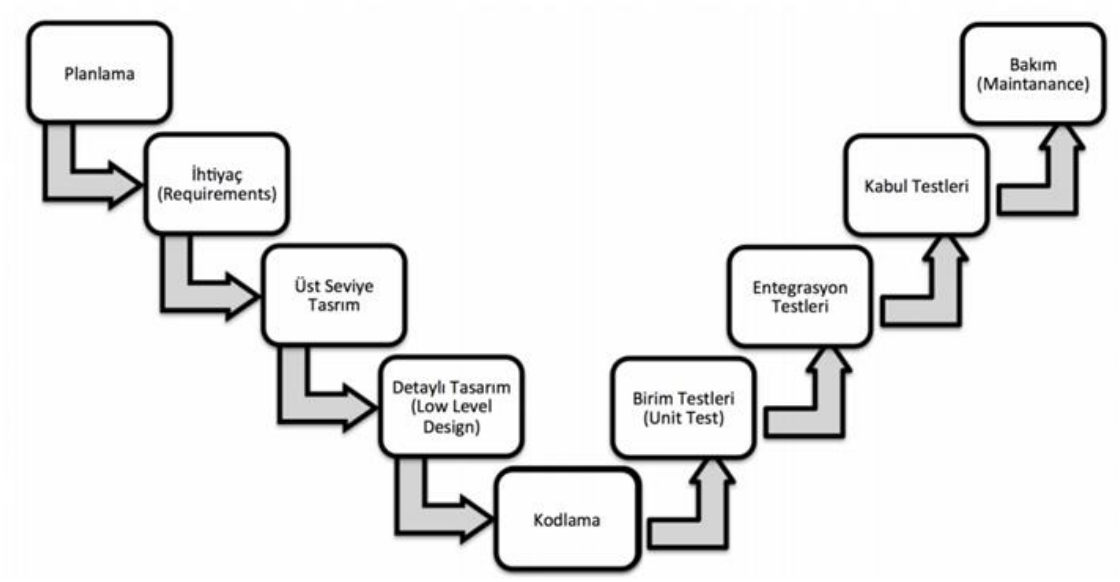
1. Problem, fırsatlar ve hedeflerin belirlenmesi
2. İnsan seviyesi enformasyon ihtiyaçlarının belirlenmesi
3. Sistem ihtiyaçlarının (isterlerinin) belirlenmesi
4. Şimdiye kadar olan adımlar ışında elde edilen isteklere cevap veren bir sistemin tasarlanması
5. Yazılımın geliştirilmesi ve dokümantasyonu.
6. Sistemin testleri ve bakım adımlarının belirlenmesi.
7. Sistemin gerçeğe alınması ve değerlendirilmesi.

4.1 V Döngüsü(V-cycle)

Yazılım geliştirme yaşam döngüsü stratejisinin tez konusu açısından kullanılan en önemli çalışması v-döngüsüdür. Adımlar V şeklini oluşturduğu için V-döngüsü denilmiştir. Üst seviye de daha genel bakışa göre tasarım yapılır. Üst seviye tasarımların daha sonra daha detaylı alt seviye tasarımlarının yapılır. Daha alt

seviyelerinin açıldığı, bunların girdilerini-çıktılarını ve beklentilerinin yazıldığını söylenebilir. Sonrasında kodlamaya geçilir. Aşağı doğru akan bir proje yönetimi söz konusudur. Ancak buradan sonra yukarı doğru çıkılmaya başlanır. Birim testleri yapılır. Her üretilen modülün testi yapılır. Daha sonra bu modüllerin/alt seviye tasarım ürünlerinin birbiri ile çalışma durumu, entegrasyon testleri yapılır ve kabul testleri aşamasında müşterinin kabul edip etmemesi ile ilgili müşteriye yani araca gidilir. Araçta uygulama test edilir ve bakım başlar.

Bu yaklaşımın diğer bir özelliği de, Şekil 4.1'den görüleceği üzere aynı hizada olan aşamaların birbirini karşılamasıdır. Detaylı tasarım yapılması ile, bunun hemen karşısında kodlama bittikten sonra birim testleri yapılır. Yapmış olduğumuz tasarıma ait testler yapılır. Üst seviye tasarıma ait entegrasyon testleri yapılır. Her bir alt modülün birbiri ile uyumlu çalışıp çalışmadığı test edilir. İhtiyaç analizleri ile sistem açısından nelere ihtiyaç olunduğu çıkarılır. İhtiyaçların ise kabul testleri yapılır. V-Şeklinde model aşağı inip yukarı çıkan ve her aşamasının karşılığının olduğu bir modeldir. "Bir şey yapıldı ama bunun karşılığı doğru mudur?" sorgulamasının yapıldığı, en aşağıya, kodlamaya kadar inip sonrasında yukarıya doğru sistemi taşıyan bir yapıya sahiptir.

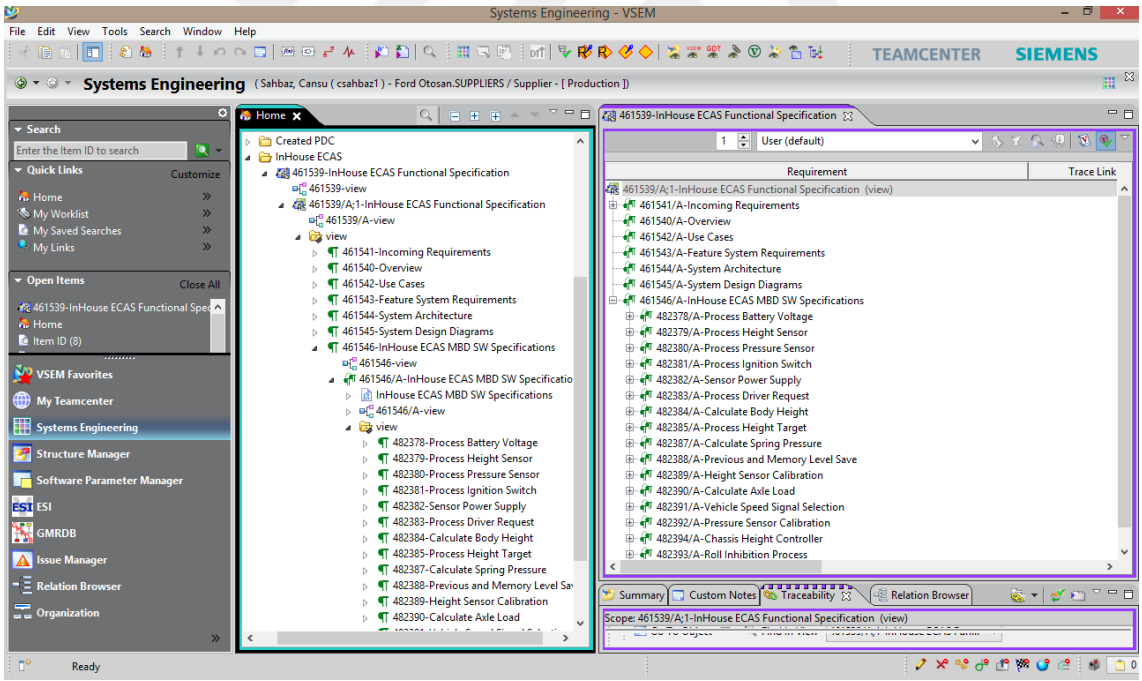


Şekil 4.1 V döngüsü akış şeması

4.2 Gereksinimlerin Belirlenmesi ve Yönetimi

Yazılım geliştirme basamaklarının temelini gereksinimlerin (requirement) belirlenmesi ve doğru şekilde yönetilmesi oluşturur. Gereksinim oluşturmanın ardından yazılım geliştirme aşamaları gelir. Ancak bu, sarmal döngü şeklinde ilerleyen bir süreçtir. Yazılımı ve testi aşamasında bulunan sorun ve/veya eksikler, tanımlanması gerekiyorsa gereksinimlerin güncellenmesine sebep olabilir. Dolayısıyla gereksinim dokümanları yaşayan dokümanlar olarak değerlendirilmeli, sürekli güncel tutulmalı ve versiyonlama ya da ilişkilendirme metotları ile doğru şekilde yönetilmelidir.

Bu tez kapsamında, gereksinimlerin yönetimi için Siemens tarafından sunulan VSEM (Vehicle Software and Electrical Management) programının RM (Requirement Management) ara yüzü kullanılmıştır. Bu araç sayesinde, ekip olarak gereksinimler üzerinde çalışabilme, doküman revizyonlarının takibi, gereksinim hiyerarşisine ve/veya bağlantısına göre ilişkilendirmeler kurabilme gibi özellikler elde edilmiştir.



Şekil 4.2 EHSKS yazılımı gereksinimlerinin, VSEM gereksinim yönetim sistemi üzerinden oluşturulması ve takibi

Elektronik Havalı Süspansiyon Kontrol Sisteminin üst seviye gereksinimleri (high level), sistem fonksiyonlarını ve hangi durumda bu fonksiyonları sağlamasını

gerektiğini belirler. Dışarıdan gelen (Incoming) gereksinimler, kanuni olarak tanımlanmış (regülasyon ve/veya patentler gibi) ya da araç içindeki ortak çalışılan fonksiyon gereksinimleri ve güvenlik gereksinimleri için kullanılır. Gelen gereksinimler aynı zamanda birer üst seviye gereksinimlerdir, bu üst seviye gereksinimlere göre sistemin üst seviye tasarımı oluşturulur.

Oluşturulan üst seviye gereksinimlerinin testleri için de üst seviye test gereksinimleri oluşturulur. Bu program ile üst seviye ve alt seviye (low level) gereksinimler arasında bağlantı ilişkisi kurularak değişiklikler revizyon ve bağlantılar ile yönetilebilmektedir. Bu bağlantılar fonksiyon bazında gereksinimler ve test gereksinimleri arasında da sağlanabildiği için fonksiyon gereksinimi değiştiğinde test prosedürünün ve çıktılarının güncellenmesi de bu şekilde yönetilebilmektedir.

4.2.1 Üst Seviye Gereksinimler

Üst seviye gereksinimler, bahsedilen fonksiyonun temel olarak yapması gereken görevleri tanımlar.

Dışarıdan gelen, (diğer araç fonksiyonları ve girdileri gibi) ya da kanuni olarak tanımlanan (patent, regülasyon gibi) gereksinimler, fonksiyon için birer üst seviye gereksinimleridir.

Örnek olarak vermek gerekirse;

1. EHSKS dışarıdan kullanıcı girişlerine izin vermeli, kullanıcı istediği yüksekliğe aracı getirebilmelidir.
2. EHKS aracın sürüşte olduğunu algılamalı, belli bir hızın üstünde otomatik olarak sürüş yüksekliğine gelmelidir.
3. EHKS dingil üzerine düşen yükü hesaplamalı ve CAN_BUS üzerinden EBS sistemine göndermelidir.
4. Araç maksimum yüksekliğe geldikten sonra körüklere hava basılmamalıdır.

İfadelerinin tamamı üst seviye gereksinimdir. 1 ve 2 numaralı gereksinim fonksiyonun çalışması ve kullanıcı ara yüzü ile ilgili ihtiyaçları belirlerken, 3 ve 4

numaralı gereksinimler diğer araç fonksiyonları için de gerekli olan, bir anlamda dışarıdan gelen/etkilenen/etkileyen gereksinimi örneklemektedir.

4 numaralı gereksinim, sistem tasarımı aşamasında yapılan hata analiz çalışması (DFMEA - Design Failure Mode Effects Analysis- Dizayn Hata Modları Efekt ve Analizleri) sonucunda ortaya çıkmıştır. Bu çalışma, sırasıyla fonksiyonda oluşabilecek hata modlarının bulunması, karşılaşma sıklığının belirlenmesi, bunun önlenmesi için alınacak aksiyonlar gibi bütün bir değerlendirmeyi içermektedir. Aşağıda örnek bir madde gösterilmektedir.

Function Description			Failure Mode					*Potential Cause(s)/ Mechanism(s) of Failure	*Prevention			Current Desig
Input/ Assumption(s)	*Verb Noun	*Desired Output /Engg. Measurable	*Potential Failure Mode	*Potential Effects	*Sev	Cls			Design Methods	Product Characteristics	Occ	*Detection and/or <x[Virtual Analysis
[1] Automatic Ride Height for ECAS (Electronically Controlled Air Suspension)	1. Vehicle gets ride height on defined speed.	C2AS XX mm C4AS XX mm Lowliner Vehicle speed 30 km/h Requirement ve parameter set'ler yazılacak. Parameter set'lerde ride height seviyeleri yazıyor. Overshoot / Undershoot requirement no	[1.1] [NO FUNCTION] Vehicle level is not set to ride height automatically.	Basic design specified ride height level target is not met (low/high). (Sev: 5) Fuel consumption increase. (Sev: 5) Exceed maximum trailer height (Seviye 2 lowliner) (Sev: 10) Vehicle does not meet ground clearance. (Sev: 5)	#	YC		[1.1.1] Control module does not maintain target height: | -incorrect parametrization - 3 | -incorrect calibration - 2 | -limited operating strategy -1 | -continuously in diagnostic mode - 1 | - etc.	-DID (data identifier) are not written. DIDs' must be given as MFD and with Excel to Calibration team and must be inserted to diagnostic tool. At 18MY FODP is used as diagnostic tool and BCM is used as target ECU.	Check MFD file.	2	Parameter Checksum for incorrect parametrization.

Şekil 4.3 Elektronik Havalı Süspansiyon Kontrol Sisteminin Dizayn Hata Modları Efekt ve Analizleri Dokümanı örnek madde Görünümü

4.2.2 Alt Seviye Gereksinimler

EHSKS alt seviye gereksinimleri, 4.3.1 ve 4.3.2 başlıklarında detaylandırılan üst seviye sistem tasarımı ve akış şemalarının ardından oluşturulur. Alt seviye gereksinimlerde, üst seviye gereksinimlerin teknik detayları ile tanımlanması söz konusudur. Yazılım mühendisi bu iki gereksinimi inceleyerek başka bir girdiye ihtiyaç duymadan fonksiyonu oluşturabilecek sınırlara ulaşabiliyor olmalıdır. Bu sebeple alt sistemde kullanılacak girdiler, fonksiyonun ne yapması gerektiği, çıkış olarak vermesi beklenen veriler bu gereksinimde detaylandırılmalıdır.

Örnek olarak 4.2.1 başlığında incelenen 2 numaralı "EHKS aracın sürüşte olduğunu algılamalı, belli bir hızın üstünde otomatik olarak sürüş yüksekliğine gelmelidir. "üst seviye gereksinimini alacak olursak;

Bu gereksinim altında birden fazla alt fonksiyon (subsystem) oluşturulabilir, ancak genel itibariyle alt sistem gereksinimleri aşağıdaki gibi detaylandırılabilir.

1. Araç hızı 30km/saat hıza eşit veya fazla ise, araç yükseklik hedefi sürüş yüksekliğine eşitlenmelidir.
2. Başka bir kullanıcı girişi olup, otomatik sürüş yüksekliği bozulana kadar araç yüksekliğini sürüş yüksekliğinde tutulacak şekilde hedef yüksekliği sabit tutulmalıdır.
3. Araç diyagnostik adımında ise veya yeterli basınç yok ise sürüş yüksekliği hedefi bozulmalıdır.

4.2.3 Alt Seviye Test Gereksinimleri

Oluşturulan her bir alt sistem için bir test gereksinimi dokümanı oluşturulur. Bu test gereksinimi alt seviye gereksinimlerinin sağlanıp sağlanmadığının testi yapılırken kullanılacak yöntem ve uygulamaları içerir. Bir test mühendisi; alt seviye sistem modeli, gereksinimi ve test gereksinimi referansı ile sistem testlerini herhangi bir açıklamaya gerek duymadan yapabilmelidir. Bu sebeple test gereksinimleri olası bütün girişlere karşılık beklenen çıktıları doğru şekilde tanımlamalı, sistemin yapısına uygun test metodunu içermelidir.

4.2.4 Üst Seviye (Entegre) Test Gereksinimleri

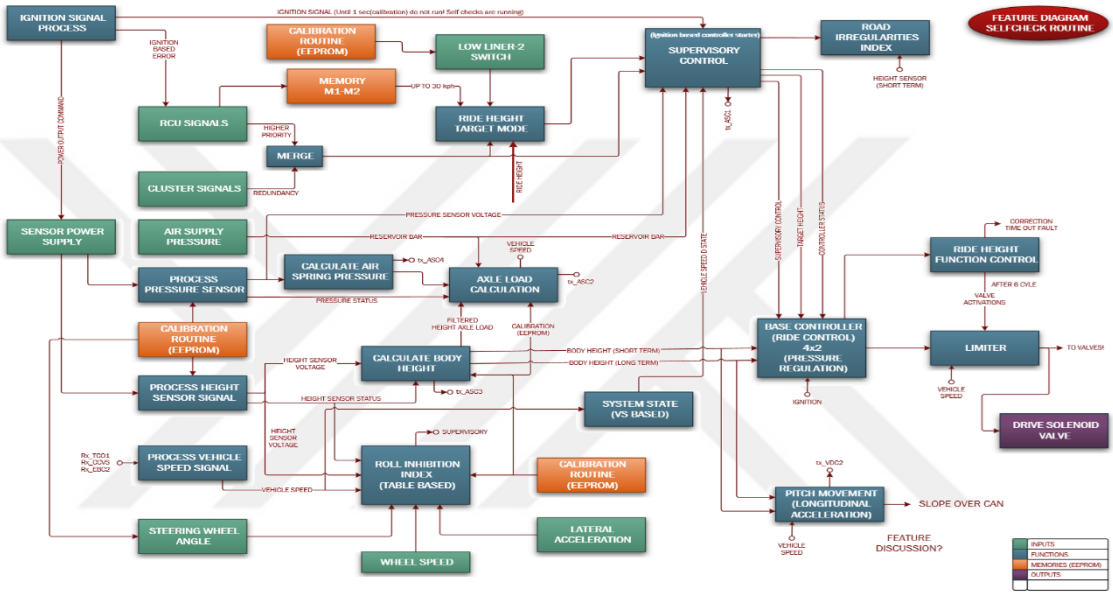
Oluşturulmuş alt fonksiyonların birbiri ile çalışması kritik aşamalardan bir diğeridir. Bir alt fonksiyonun çıkışı diğer bir alt fonksiyonda işleme alındığından, birlikte çalışan bu alt fonksiyonların tek tek test edilmesinin yanında entegre olarak birlikte de test edilmesi gerekmektedir. Bu entegre testleri ve test edilecek alt fonksiyonların belirlendiği gereksinimler üst seviye test gereksinimleridir.

4.3 Sistem Tasarımı

4.3.1 Üst Seviye Sistem Tasarımı

Elektronik Havalı Süspansiyon Kontrol Sisteminin üst seviye gereksinimleri doğrultusunda sistemin temel olarak gerçekleştirmesi gereken fonksiyonlar

belirlenmiştir. Bu temel fonksiyonların birbiri ile bağlantısının gösterildiği ve oluşturulduğu sistem fonksiyon diyagramı oluşturulmuştur. Sistem için gerekli olan giriş ve çıkışlar ve genel hatlarıyla üst seviye sistem tasarımı yapılır. Böylece her bir fonksiyona bir mühendis atanabilir ve fonksiyonel geliştirme – doğrulama aşamaları izlenebilirliği artar. Şekil4.4'te EHSKS için oluşturulmuş ilk fonksiyon diyagramı görülmektedir. Bu diyagram sistemin genel olarak giriş ve çıkış aksiyonlarını ve temel üst seviye sistem gereksinimine karşılık gelen fonksiyonel bloklarını göstermektedir.



Şekil 4.4 Elektronik Havalı Süspansiyon Kontrol Sistemi Fonksiyon Diyagramı

4.3.2 Alt Seviye Sistem Tasarımı – Veri Akış Diyagramları

Üst seviye fonksiyon diyagramları alt sistemleri ile birlikte değerlendirilir. Üst seviye fonksiyonların altında, birden fazla derinlikte ve uzunlukta alt seviye fonksiyon blokları bulunmaktadır. Bu blokların ve fonksiyonların birbiri ile ve hiyerarşik olarak etkileşimleri değerlendirilmelidir. Bu değerlendirme alt seviye veri akış diyagramı olarak değerlendirilir. Alt Seviye Veri Akış Diyagramı ise her bir fonksiyon içindeki alt-fonksiyon ve tanımlanmış giriş-çıkış parametreleri ile sistemin bütün ayrıntılarını gösterir. Böylece fonksiyonların sistem entegrasyonu

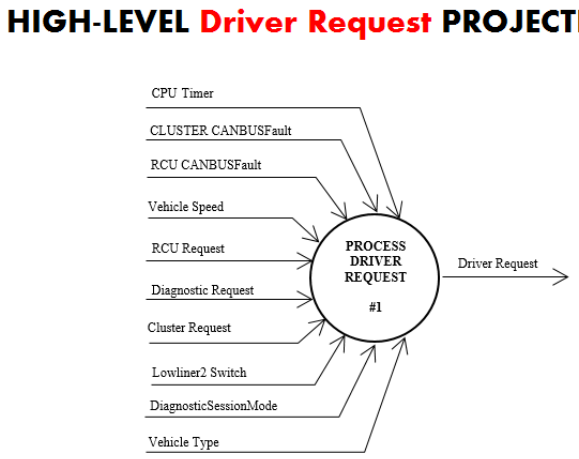
kontrollü yapılabilir. Aynı zamanda model tabanlı yazılımın temeli oluşturulmuş olur.

2015'te yayınlanan bir araştırma raporunda da incelendiği üzere, [10] model tabanlı yazılım geliştirme çalışmalarında akış diyagramları adını verdiğimiz metodolojinin kullanılması oldukça verimli ve paralel çalışmayı en mümkün kılan yöntemlerden biridir.

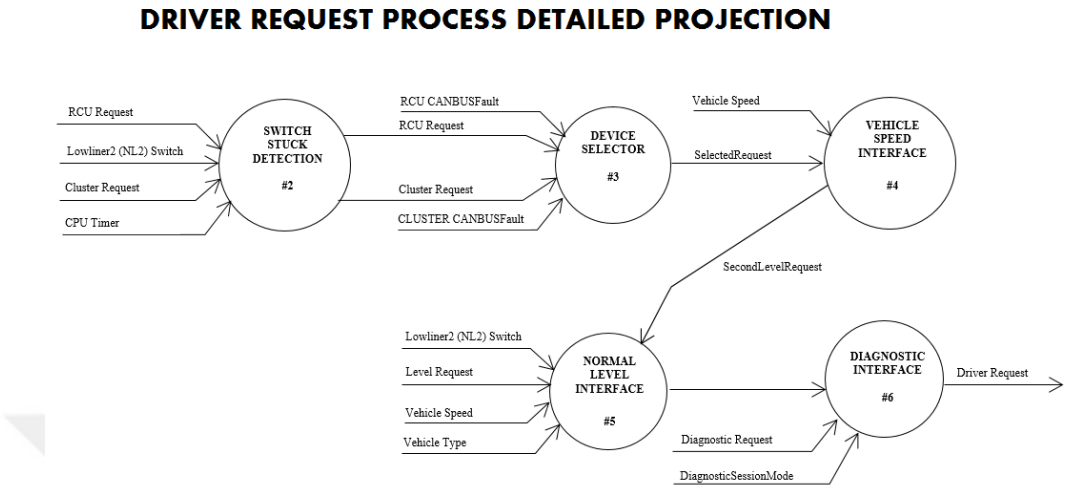
Bu metodolojinin kullanılmasının ardından referans araştırma raporu ile tanışıldığından, çalışma metodolojisinin de sağlaması yapılmıştır denilebilir.

Örnek olarak; 4.2.1 başlığında incelenen üst seviye gereksinim 1 "*EHSKS dışarıdan kullanıcı girişlerine izin vermeli, kullanıcı istediği yüksekliğe aracı getirebilmelidir. "* için birden fazla alt sistem ve gereksinim oluşturulmuştur. Bu gereksinimler doğrultusunda veri akış diyagramları oluşturulmuştur. "Driver Request/ Kullanıcı talepleri" fonksiyonu bu gereksinimlerden yola çıkılarak oluşturulmuş, kullanıcı taleplerini önceliklendiren, işleyen, sistemin geneline dağıtan bir fonksiyondur. Şekil 4.5 ve Şekil 4.6'de "Driver Request/ Kullanıcı talepleri" fonksiyonunun sırasıyla üste seviye ve alt seviye veri akış diyagramları gösterilmiştir.

Bu akış diyagramları, önce üst seviye olacak şekilde tasarlanır, sistemin giriş ve çıkışları temel anlamda oluşturulur. Ardından sistemin alt fonksiyonları oluşturulmuş ve her bir alt fonksiyon için veri akış diyagramı detaylandırılmıştır.



Şekil 4.5 "Driver Request/ Kullanıcı talepleri" Fonksiyonu Üst Seviye Veri Akış Diyagramı



Şekil 4.6 "Driver Request/ Kullanıcı talepleri" Fonksiyonu Alt Seviye Veri Akış Diyagramı

Fonksiyonlar ve alt sistemleri için oluşturulan veri akış diyagramlarının tamamı birbiri ile ilişkilidir. Bu ilişkileri kurarak ve bütün sistemi görmek için, veri akış diyagramlarının entegrasyonu sağlanmıştır. Sistem bütün ilişki ve alt sistemleri ile incelenebilir hale gelmiştir.

4.4 Model Tabanlı Yazılım Geliştirme

Yazılım projelerinde, iyi tanımlanmış gereksinimlerden ve mimari tasarımlardan satır bazında kod geliştirmek kolaydır. Bu durum, daha karmaşık, milyonlarca satır kod içeren sistemlerde bir ekip tarafından geliştirilebilir. Bu noktada tasarım ve gereksinim arasındaki tutarlılık ön plana çıkmaktadır. Yapılacak olan değişikliklerdeki herhangi bir yanlış anlama tasarımın değişmesine ve sonunda kodun da binlerce satırında değişikliğine sebep olabilir. Bu değişim sırasında geliştirici, doğru olan satırlarda da yanlışlıkla değişiklik yapabilir. Yapılacak olan değişiklik, değişiklik gerektirmeyen yerlere de etkisi sonucunda beklenmeyen

hatalara sebep olabilir ve testlerde bu hatanın farkına varılamayabilir. Sonuçta, bütün bu durumlar, üretilecek olan yazılımın maliyetini beklenene göre daha fazla arttırmaktadır. Bu tip karmaşaların önüne geçmek ve maliyeti düşürebilmek amacıyla yazılım sektöründe çeşitli yazılım araçları geliştirilmiştir. Tasarımı daha kolay ifade etmek ve yazılımla eşleştirmek bu problem için bir çözüm olabilir, ancak sistemin büyüklüğü ve birçok alt sistemler mevcut ise bu araçlar bazı noktalarda yetersiz kalabilmektedir.

Bu tez kapsamında, belirtilmiş olan problemlere çözüm bulabilmek için model dili ve bu dili şekillendiren model tabanlı yazılım geliştirme ana çatısı uygulanmıştır. Bu ana çatıda, bu tür yapılmış çalışmalarda olmayan grafiksel ifade edilmiş hem grafik ara yüzü hem de mantıksal modellerden, belirlenmiş olan programlama diline özgü kod üretim işlemi gerçekleştirilmiştir.

Geliştirilecek olan yazılımın daha kolay idame ettirilebilmesi ve test edilebilmesi model tabanlı yazılım geliştirmenin en büyük avantajlarındandır. İlk bakışta model tabanlı yazılım geliştirme yöntemi, geliştirme hızını artırmaktadır. Bu artış kodun tasarlanmış modellerin bir ya da birkaç dönüşümü ile birlikte otomatik üretilmesi ile sağlanmaktadır. Aynı zamanda geliştiricilerin, grafiksel ifade edilmiş modelleri metinsel kod parçalarına göre daha kolay ve hızlı anlayabilmesi geliştirme hızına önemli bir katkı sağlamaktadır [11]. Otomatik dönüşüm işlemleri ve biçimsel olarak tanımlanmış modelleme dilleri, geliştirilen kodun kalitesini artırır ve bu şekilde tasarlanan mimariler başka projelerde tekrar kullanılabilir. Tasarlanan modeller platformdan bağımsız olacağı için, hedef platformlar değişse bile yazılım hızlıca uygulanabilirdir.

Bu sayede herhangi bir yeni geliştirme, eklentiler veya yeni teknoloji ile ilgili değişikliklerin bu şekilde yapılması daha kolay olacaktır [12]. Geliştirilmiş modellerin başka platformlara taşınabilirliği model tabanlı yazılım geliştirme yöntemi sayesinde daha kolaylaşmış olacaktır [13]. Model tabanlı yazılım araçlarındaki kod üreteçleri kodun üretilmesini sağlamasının yanı sıra aynı zamanda test kodunun da otomatik üretme kabiliyetinin bulunması, yazılımın test edilme çabalarını önemli miktarda azaltır [14].

Model parçaları ve gereksinimler arasında ilişkiler kurulabilmesi de desteklenen özelliklerden bir diğerdir. Bu özellik ile, gereksinime bağlanmış olan model blokları kişilerden bağımsız ve hızlıca takip ve kontrol edilebilmektedir [15].

4.4.1 EHSKS Model Tabanlı Yazılımı

4.4.1.1 AUTOSAR Uyumluluk

Tez kapsamında Elektronik Havalı Süspansiyon Kontrol Sistemi, sisteme özel olarak alınan bir ECU silinerek, araçta var olan bir ECU olan Gövde Kontrol Modül (GKM/ BCM)'üne entegre edilmiştir. Bu ECU, detayı aşağıda verilen otomotiv ECU'ları yazılım ve iletişim geliştirme standardı olan AUTOSAR'a uyumlu olduğundan, yazılım geliştirme ve entegrasyon çalışmaları da bu standart uygun olarak geliştirilmiştir.

AUTOSAR, aynı ya da benzer yazılım birleşenlerinin tekrar tekrar geliştirilmesini engellemek, bu birleşenleri belli kurallar altında standartlaşmak amacıyla bir araya gelmiş bir konsorsiyumdur. Bu konsorsiyum farklı ortaklık çerçevesinde gelişmiştir. Konsorsiyum BMW, Daimler, VW, Ford, GM, Toyota, PSA, Continental ve Bosch gibi büyük otomotiv devleri tarafından 2003 yılında kurulmuştur.

AUTOSAR(AUTomotive Open System ARchitecture), otomotiv sektöründe kullanılan bir çeşit yazılım mimarisi standardıdır. Giderek karmaşıklaşan sistemleri, basitleştirmek ve bağımsızlaştırmak üzere kullanılmaktadır. Diyagnostik ve Flexray haberleşme protokolleri gibi sistemden bağımsız yazılım parçalarının tek bir standartta toplanmasına olanak sağlar [16].

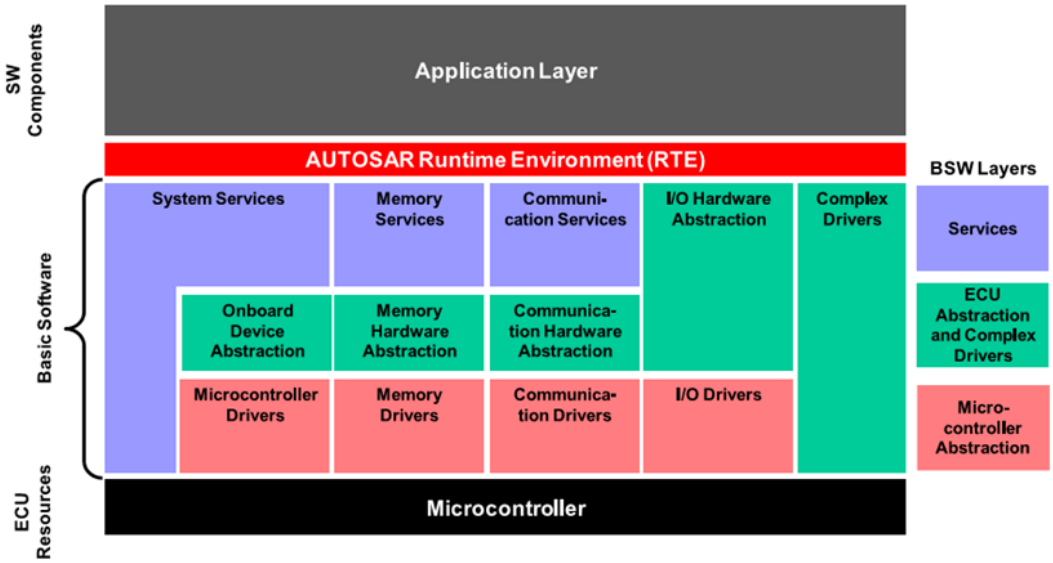
Böylelikle, tek yazılım parçası hiçbir değişen işlemci/ECU/EKÜ'ye göre güncelleme gerektirmeksizin yıllarca kullanılabilir. Yeni fonksiyonlar konfigürasyondaki küçük değişiklikler ile kolaylıkla uygulanabilir. AUTOSAR olmayan gömülü sistemlerin en büyük dezavantajı donanıma yani EKÜ'ye bağımlı olmaları ve farklı bir EKÜ için yazılımın neredeyse sıfırdan geliştirilme zorunluluğudur. AUTOSAR ile donanımdan bağımsız bir şekilde yazılım geliştirilebilir ve bu yazılımı istenilen herhangi bir sistemde/EKÜ'de kullanılabilir. Temel yazılım, farklı EKÜ'lerde ve alanlarda kullanılabilir. Farklı araç, platformlara ve EKÜ donanımlarına adapte edilebilir.

En önemli avantajları, yazılım modüllerinin birbirinden bağımsız olarak geliştirilmesini, entegrasyonunu, zaman içinde güncellenebilmesini ve tekrar kullanımını sağlamasıdır. Böylece maliyetli bir iş olan yazılım geliştirmede, zaman ve gider yükünü oldukça hafifletmektedir.

AUTOSAR, otomotiv sektöründeki EKÜ'ler için tasarlanmıştır. Bu tür EKÜ'ler şu özelliklere sahiptir:

- Donanım ile güçlü bir iletişime, etkileşime sahiptir. (Sensör ve eyleyiciler gibi)
- CAN (Controller Area Network), LIN (Local Interconnect Network), Flexray, Ethernet gibi araç ağlarına bağlantı
- Tipik olarak 16 Bit ya da 32 Bit olan mikro denetleyiciye sahip
- Gerçek zamanlı sistem
- İç ve dış geçici (flash) hafızadan program yürütmesi

Bu tez kapsamında geliştirilen EHSKS yazılımı Vector firmasının sağladığı AUTOSAR uyumlu bir modül üzerinden geliştirilmiş ve devreye alınmıştır.



Şekil 4.7 AUTOSAR Katmanlı Yazılım Mimarisi [16]

AUTOSAR, gömülü sistem yazılımını 5 ana katmana ayırır. İlk bakışta OSI Modeli'ni andıran, mantık olarak oldukça benzeyen fakat kendine has yapısı olan bir katman modelidir. Her katmanda belli yazılım modülleri diğer katmanlardan soyutlanır ve katmanlar birbirleri arasında AUTOSAR ara yüzleri üzerinden iletişime geçer.

AUTOSAR yazılımın hiyerarşik yapısını yukarıdan aşağıya bir yaklaşımla tanımlar. Temel yazılım modüllerini yazılım katmanlarıyla eşleştirir ve bunlar arasındaki ilişkiyi belirler.

Mikro denetleyici Soyutlama Katmanı (Microcontroller Abstraction Layer—MCAL)

Temel yazılımın en alt yazılım katmanını oluşturur. Sürücüler içerir ki bunlar mikro denetleyiciye direkt erişimi olan yazılım modülleridir. Görevi, üst katmanları mikro denetleyiciden bağımsız hale getirecek ayrımı oluşturmaktır.

EKÜ Soyutlama Katmanı (ECU Abstraction Layer—ECAL)

Mikro denetleyici Soyutlama Katmanının sürücülerine ara yüzüdür. Ara yüz kısaca donanım fonksiyonlarına erişimi sağlar. Ara yüz, verinin içeriğini değiştirmez. Örneğin, kaç tane CAN_BUS denetleyicisi olduğundan bağımsız olarak CAN_BUS iletişimine bu ara yüzle erişebilir.

Bunun haricinde bu katman enjeksiyon kontrolü, elektrik vana kontrolü gibi AUTOSAR'da standartlaşmamış, harici cihazlar için sürücüler içerir.

Görevi, üst katmanları EKÜ'den bağımsız hale getirecek ayrımı oluşturmaktır.

Servis Katmanı (Services Layer)

Uygulama ve diğer temel yazılımlar için gerekli temel yazılım modülleri içeren katmandır. Örneğin; işletim sistemi, ağ iletişimi, hafıza yönetimi (NvRAM), tanı servisleri (Diagnostics, UDS) ve hata tanımı, Watchdog, vb. Modülleri içeren katmandır.

Görevi, üst katmanları EKÜ ve Mikrodenetleyiciden bağımsız hale getirmektir.

AUTOSAR Çalıştırma Ortamı—AUTOSAR Runtime Environment (RTE)

AUTOSAR'ın ana bölgesi diyebileceğimiz katmandır. Uygulama yazılım(Application Software)modülleri arasındaki iletişimi ve bu yazılım modüllerinin temel yazılım modülleriyle iletişimini sağlar. Veri ve bilgi akışını sağlar. Uygulama yazılımı (SW-C, Software Component, Yazılım bileşeni) ile temel yazılım modülleri arasında ara yazılım olarak çalışır.

Uygulama yazılımındaki olayların/fonksiyonların, RTE etkinlikleriyle (Belirli bir zamana ya da olaya bağlı olarak fonksiyonun düzenli ya da düzensiz tetiklenmesi) çalıştırır.

Uygulama Katmanı—Application Layer

Bir EKÜ'nün ya da bir otomobilin uygulama yazılımı birbirine bağlı birden fazla yazılım birleşeninden oluşan bir sistem olarak tasarlanabilir. Bu tasarlanan yazılımın bulunduğu katman uygulama katmanıdır. Sistem ve araç yazılım mühendislerinin yazılım geliştirme ile ilgilendikleri, fonksiyonların bulunduğu katmandır.

Tez kapsamında geliştirilen EHSKS yazılımı bu katmanda geliştirilmiştir ve diğer yazılımlar ile etkileşimi bu katmanda yapılmıştır.

Bir yazılım geliştirme süreci sırasında çeşitli bilgilerin belirlenmesi gerektiği gibi AUTOSAR uyumlu EHSKS yazılımı geliştirirken de aynı bilgiler oluşturulmuştur. Bu bilgiler, donanımsal gereksinimler de dahil olmak üzere birçok alt maddeden oluşur. Bu bilgiler mikro denetleyici (donanım) katmanı ile uygulama katmanı arasındaki yapılar bütününü (BSW: Temel yazılım) oluşturmak için kullanılır. Genel hatlarıyla aşağıdaki şekilde sıralanabilir.

- Giriş ve çıkış pinleri
- EHSKS içerisinde kullanılacak olan J1939 standardına uygun CAN mesaj listesi
- Uygulama katmanında kullanılan her bir sistem yazılımının geliştirilen yazılıma gönderecekleri bilgiler. (Inter-runnable variables)
- Yazılımın çalışması için gerekli kalıcı hafıza gözleri sayısı, isimleri, boyutları ve ilk durum (üretici) değerlerinin belirlenmesi (NvRAM ve DID'ler)
- Yazılımın farklı özelliklere sahip araçlarda araca uygun olarak konfigure edilebilmesi için gerekli kalıcı hafıza gözleri sayısı, isimleri, boyutları ve ilk durum değerlerinin belirlenmesi
- Yazılımın tanılayıcı hata kodlarının (DTC: Diagnostic Trouble Code), sayısı, isimlerinin belirlenmesi

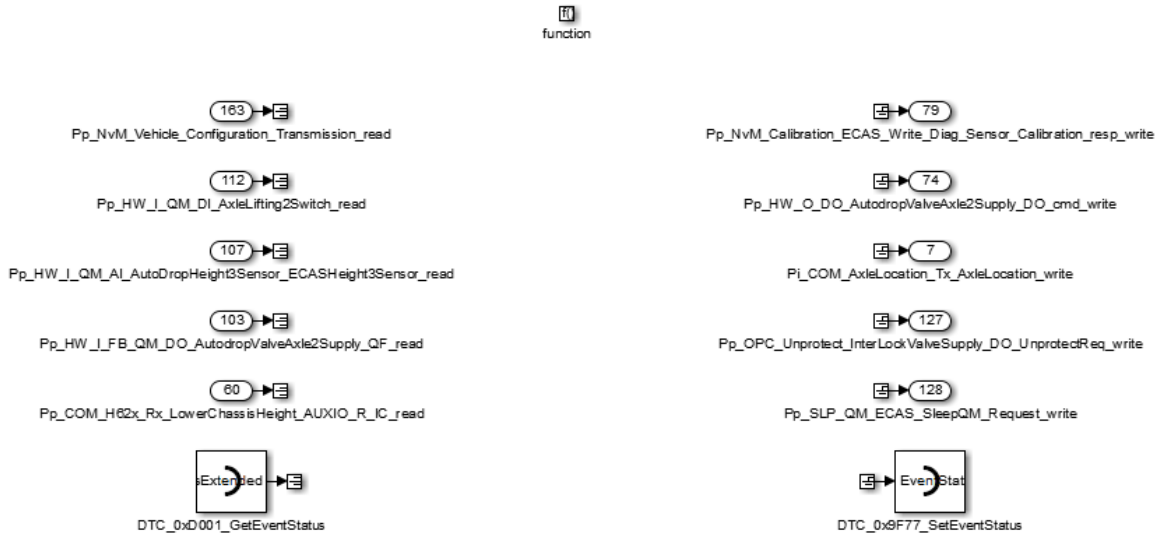
Tüm bu tanımlamalar yapıldıktan sonra temel yazılımını tasarım aşamasına geçilmiştir. Bu aşamada Da Vinci programı kullanılmaktadır. Da Vinci, AUTOSAR

uyumlu BSW yazabilmek için gerekli programlar bütününün adı olup, Developer ve Configurator olarak 2 ayrı alt programdan oluşur.

Developer programı ile BSW için daha önce belirlenen bilgiler kullanılarak aşağıdaki gibi bir tasarım mimarisi oluşturulur.

Bu mimaride her bir port için uygun port tipinin de seçilmesi gereklidir. Bu aşamalar yazılım gömülen ECU'nun üreticisi tarafından sağlanmıştır. Temel fonksiyon bloklarını içeren EHSKS temel bloğu bu katmandan gelen giriş ve çıkışlarla bağlanır.

Temel yazılımın geliştirme adımı tamamlandıktan sonra CAN sinyallerinin tanımlanması ve ilgili konfigürasyonların tamamlanması yapılmıştır. Bunun için de Da Vinci Konfigurator programı kullanılmaktadır. Bu program ile yazılım mimarisinin genel ağaç görünümü aşağıdaki gibi elde edilir. Bu aşamada BSW tamamlanmış olur ve aplikasyon katmanları oluşturulur. Bunun için "armxl" dosya uzantılı bir çıktı elde edilir. Bu çıktı, içerisinde BSW'de dahil olmak üzere tüm yazılım bilgilerini barındırır (Şekil 4.10).

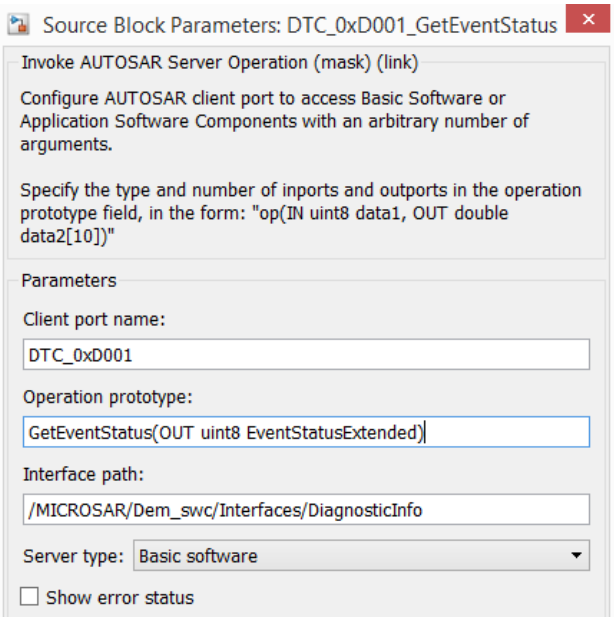


Şekil 4.8 Arxml dosyasından üretilen örnek Simulink modeli

Tüm bu temel yazılım adımlarının tamamlanmasının ardından çıktı olarak alınan arxml dosyası ile aplikasyon katmanlarının üretimi ve implementasyon adımları başlamış olur. Bu dosya Simulink platformunda üretilen EHSKS yazılımı gibi diğer yazılımların direkt olarak temel yazılıma entegre edilebilmesi için bir Simulink platformu sunar. Bu platformun üretilmesi için her bir aplikasyon için bu

platformların teker teker üretilmesi gerektiğinden EHSKS yazılımı için oluşturulan her yazılım versiyonunda bu işlem tekrar edilmiştir.

Arxml dosyası içerisinde her bir aplikasyon katmanı bir obje olarak tanımlanmış olup Matlab programı ile Simulink model dosyasına çevrilebilirler. Bu çevrim işleminin yapılması ile aşağıdaki gibi bir model dosyası oluşur. Bu model dosyası aplikasyon yazımının implemente edilebilmesi için gerekli tüm giriş ve çıkış portlarını içerir (Şekil 4.9). Bu portlar aynı zamanda temel yazılımın tanıyabileceği bir konfigürasyona sahiptir. Böylece tüm yazılım kolayca bir araya getirilebilir. Bu sayede sistem içinde bulunan yazılımla EHSKS yazılımı ile bir araya getirilerek yayınlanmıştır.



Şekil 4.9 Örnek bir DTC port konfigürasyonu

Aplikasyon yazılımının araçtaki güvenlik özelliklerine göre değişen çalışma sıklıkları bulunur. Kritik fonksiyonlar için bu değer daha düşük, konforu etkileyen fonksiyonlar için daha yüksektir. Yani kritik fonksiyonlar konforu etkileyen fonksiyonlara göre daha sık aralıklarla çalıştırılır. Bu özellikler de yine arxml dosyasında tanımlı olup Simulink modeline otomatik olarak aktarılmaktadır. EHSKS yazılımı güvenliği kritik bir fonksiyon olmadığından diğer fonksiyonların da çalışma hızını etkilememek adına bu süre 50 ms olarak belirlenmiştir.

4.4.1.2. EHSKS Yazılımının Etkileşim/İletişim Arayüzü

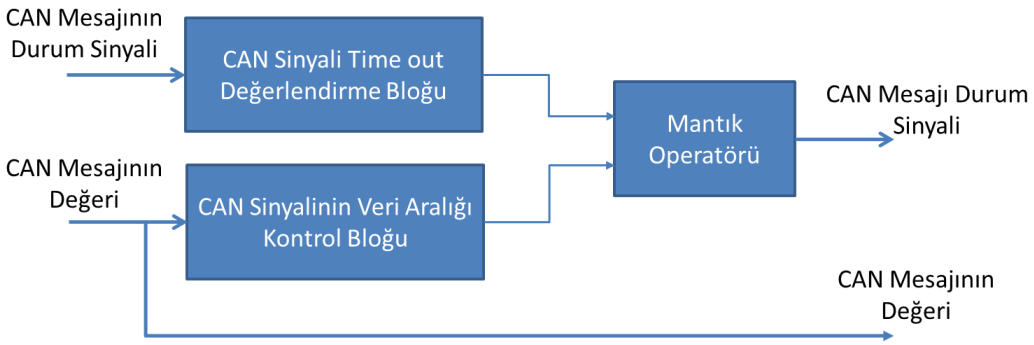
Tez konusu olan EHSKS yazılımının Şekil 4.10'de de gösterildiği gibi giriş ve çıkışlar ile diğer fonksiyonlar ve araç ile etkileşim /iletişim halindedir. Giriş ve çıkışların sisteme uygun şekilde ayarlandığı ve iletildiği bloklar giriş ve çıkış dönüştürme bloklarıdır.



Şekil 4.10 EHSKS yazılımı; ana fonksiyon, giriş ve çıkış dönüşüm blokları

Giriş Dönüştürme Bloğu

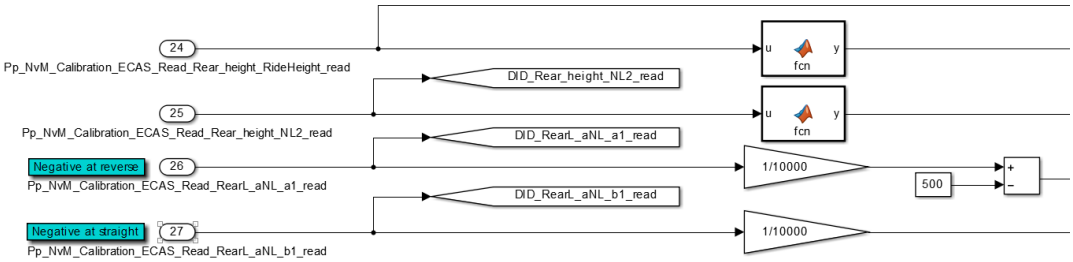
Giriş dönüştürme bloğunda, CAN_BUS üzerinden okunan sinyallerin ayrılması ve hatalarının analizi yapılmaktadır. Her bir CAN sinyalinin gelmesi beklenen bir data aralığı vardır. Giriş dönüştürme bloğunda sinyallerin bu aralıkta olup olmadığı tespit edilmektedir. Aynı zamanda sinyallerin 'time-out' denilen zaman aşım hatasına girip girmediği de bu blokta değerlendirilmektedir. Buna göre bu bloklar yazılım içinde gruplama metodu olarak kullanılan Bus'lara dönüştürülerek, sinyal değeri ve durumu olarak ana işlem bloğuna aktarılır. Bir fonksiyon içinde sinyalden beklenen veri değerlendirilirken mutlaka durumu da göz önünde bulundurulmuştur. Şekil.4.11'deki örnekte de görülebileceği gibi, CAN mesajları iki ayrı değerlendirme sonucunda mantık operatörleri ile durum değerlendirilmesi yapılmıştır. Sinyali kullanan her bir fonksiyonda sinyalin durumu da ayrı ayrı değerlendirilmiştir.



Şekil 4.11 EHSKS yazılımı; giriş dönüşüm bloğunda örnek bir CAN sinyali değerlendirme fonksiyonu

Aynı zamanda giriş dönüştürme bloğu doğrudan bağlı olan donanımların testinin yapıldığı bloktur. Yazılımın içinde bulunduğu ECU ile de bağlantılı olan bu özelliğe göre, her bir donanım bağlantısının aynı zaman 'sense' denilen ölçüm girişleri de mevcuttur ve bu ölçümlere karşılık 'QF- Quality Factor' denilen durumları giriş olarak yazılıma sunar. Bu girişler, bağlı olan donanımların açık devre ya da kapalı devre olması ve yük çekmemesi gibi bilgileri değerlendirip durum bilgisini hatalı veya değil gibi kullanılmak üzere yazılıma sağlar.

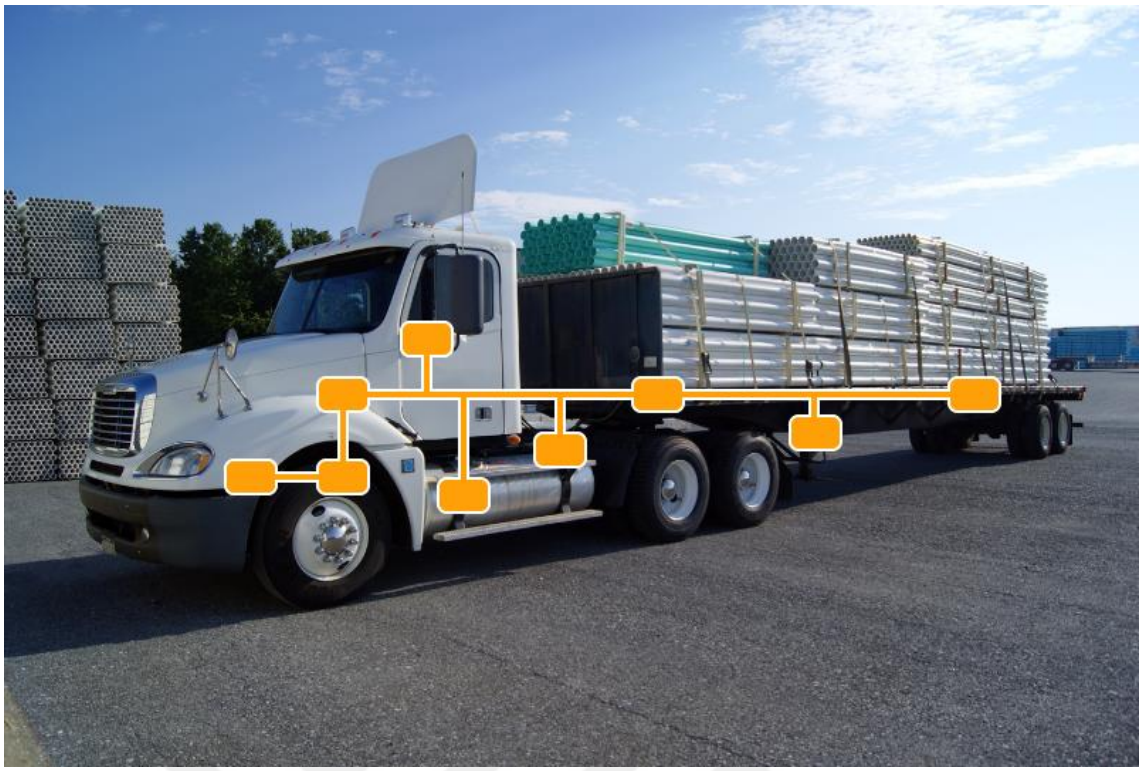
Giriş değerlendirme bloğunun, çalıştığı diğer bir fonksiyon da kalıcı hafıza birimlerinin okunması ve işlenmesidir. Bu birimler tek tek okunmuş, belirli dönüştürme işlemlerinden geçirilmiştir. Örneğin bir hafıza gözüne yazılan 16 byte'dan oluşan bir verinin her bir byte'ı başka bir parametre olabilir. Çoğunlukla da öyledir. Dolayısıyla bu uzun verinin tek tek ayrılarak doğru parametreler atanması işlemi giriş değerlendirme bloğunda yapılmaktadır. Çoğu zaman veri dönüşümü yapılması gerekmiştir. Parametrenin kullanılabilmesi için belli katsayılar ile çarpılması veya bölünmesi gerekmektedir. NvRAM adını alan bu hafıza alanı ve DID adı verilen hafıza gözlerine bit ve byte cinsinden veriler yazılabildiğinden noktalı, ondalık veya işareti olan, negatif gibi, değerler yazılıp okunamadığından bu veriler belli katsayılar veya ön işlemler ile değerlendirilip yazılmıştır ve okunduktan sonra da yine bu işlemler ile dönüştürülerek kullanılmıştır. Okuma ve dönüşüm işlemi giriş dönüştürme bloğunda yapılmıştır. (Şekil 4.12)



Şekil 4.12 EHSKS yazılımı; giriş dönüştürme bloğu NvRAM parametreleri okuma ve dönüştürme örnek fonksiyonu

CAN_BUS iletişimi

CAN BUS, Bosch ve Intel firmaları tarafından 1987 yılında geliştirilen bir iletişim protokolüdür. CAN BUS hattının görevi, elektronik kontrol üniteleri arasındaki iletişimi (veri alış verişini) sağlamaktır. CAN High ve CAN Low denilen 2 hattan oluşmaktadır ve tüm kontrol üniteleri bu hat üzerinden haberleşmekte, bilgileri paylaşmaktadır. CAN BUS hattını oluşturan iki adet can bus kablosu, parazitin azaltılması için birbiri üzerine dolanmış (örgü) şeklindedir. CAN açılımı: Controller Area Network: "Kontrol Alan Ağı". BUS: "veri yolu" anlamına gelmektedir. CAN BUS: "Kontrol alan ağı veri yolu" anlamına gelir. Buna "kontrol üniteleri ağı" da denebilir. Burada İngilizce "otobüs" anlamına gelen "BUS" kelimesinin kullanılma sebebi, tüm bir bilgilerin tek bir hat üzerinden yolculuk yapmasına, verilerin bir elektronik kontrol ünitesinden alınıp, diğerine bırakılmasına atıf yapmaktır. Bu ağ yapısı Şekil 4.13'te de görüldüğü gibi aracın tamamında yayılmış halde bulunmaktadır.



Şekil 4.13 Ağır ticari bir araçta CAN hattının örnek dağılımı

Bugün modern otomobillerdeki elektronik kontrol ünitesi (ECU) sayısı neredeyse 50-70 adete yaklaşmıştır. Her elektronik kontrol ünitesinin birbirine bilgi göndermesi gerekir. Ayrıca bir sensörden gelen bilgiyi bir çok elektronik kontrol ünitesinin kullanması gerekmektedir. Bu durumda her biri için ayrı kablo çekilmesi, maliyet, karmaşa, ağırlık, arıza olasılığının artması, yer kaplama gibi olumsuz sonuçlar doğurur. CAN_BUS ile bu karmaşa ortadan kalkar.

CAN_BUS sisteminin sahip olduğu özellikler şöyle sıralanabilir:

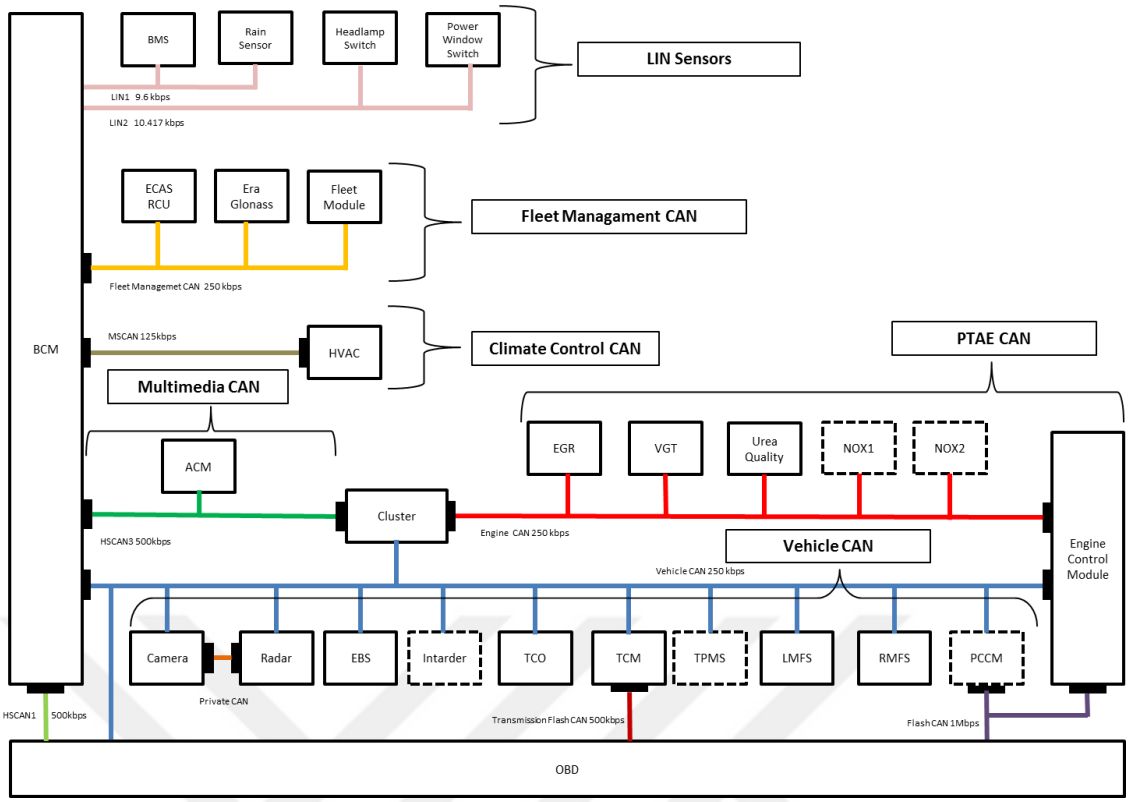
- Mesaj önceliği
- Kayıp zaman güvenliği – Yapılandırma esnekliği
- Senkronizasyonlu çoklu kabul: Aynı veri birçok ünite tarafından alınabilir
- Sistemdeki veri yoğunluğunu kaldırabilme
- Çok yöneticili (Multimaster) çalışma
- Hata tespiti ve hataya ilişkin sinyalleri üretme
- Mesajın yollanmasında hata oluşması halinde mesajın iletim hattının (BUS) boş olduğu bir anda mesajı otomatik tekrar yollama

- Ünitelerde oluşan hataları ayırt edebilme ve özerk olarak kalıcı hatalı üniteleri kapatabilme

Ağır ticari araçlarda SAE tarafından standartlaştırılmış J1939 protokolü kullanılır. Aşağıda SAE J1939 protokolünün özellikleri maddeler halinde sıralanmıştır [17]:

- Society of Automotive Engineers (SAE) tarafından geliştirilmiştir.
- Ağır görev araçlarındaki haberleşme sistemini tanımlar.
- Fiziksel katman olarak CAN sistemini kullanan yüksek düzeyli haberleşme protokolüdür.
- Çift burgulu kablo kullanılır.
- Maksimum 40 metre uzunluğundaki bağlantıları destekler.
- 250 KByte/sec bilgi iletişim hızında çalışır.
- ECU'da maksimum 30 node'u destekler.
- Maksimum 253 kontrol uygulamasını destekler.
- 1785 byte'a kadar mesajı destekler.
- Parameter Group Numbers (PGN) setini destekler.

Tez konusu çalışmanın yapıldığı araç CAN_BUS topolojisi Şekil 4.14'da görüldüğü gibidir. Bu topolojiye göre BCM modülü içinde bulunan tez konusu EHSKS yazılımı CAN üzerinden birçok sistem ile bağlı ve veri alışverişi halindedir.



Şekil 4.14 EHSKS yazılımı geliştirilen aracın CAN_BUS topolojisi

NvRAM & DID

NVRAM (Non Volatile Random Access Memory) güç kaynağı beslemesine bağlı olmaksızın veri depolayabilen bellektir. Araçlarda her yeniden çalıştırma sırasında kullanılması gereken bilgilerin hafızada tutulması için kullanılmaktadır. NVRAM, verileri RAM bellek dizisinden EEPROM bellek dizisine kopyalayarak veri içeriğini korur. Güç kesildiğinde, EEPROM belleği verileri tutar. Güç geri geldiğinde, RAM belleği EEPROM dizisindeki verileri çeker, böylece EEPROM belleğinde tutulan veriler tekrar kullanılabilir.

Güce bağlı olmadan veri tutabilmesi haricinde, NVRAM kullanımının sağladığı başka avantajlar da vardır. NVRAM'ler oynar parçaya sahip olmadıkları için düşük enerji tüketirler. Bunun yanında, okuma ve yazma hızları genellikle volatile RAM'lere göre daha yüksektir.

EEPROM'lar (Electrically Erasable Programmable Read-Only Memory) FGT-Floating Gate Transistor (kayar nokta kapılı transistör) içeren salt okunur bellek

türüdür. Bu transistorların kapısına belli bir değerden yüksek voltaj uygulanarak EEPROM'un içerisindeki veriler silinebilir veya yeni veriler yazılabilir.

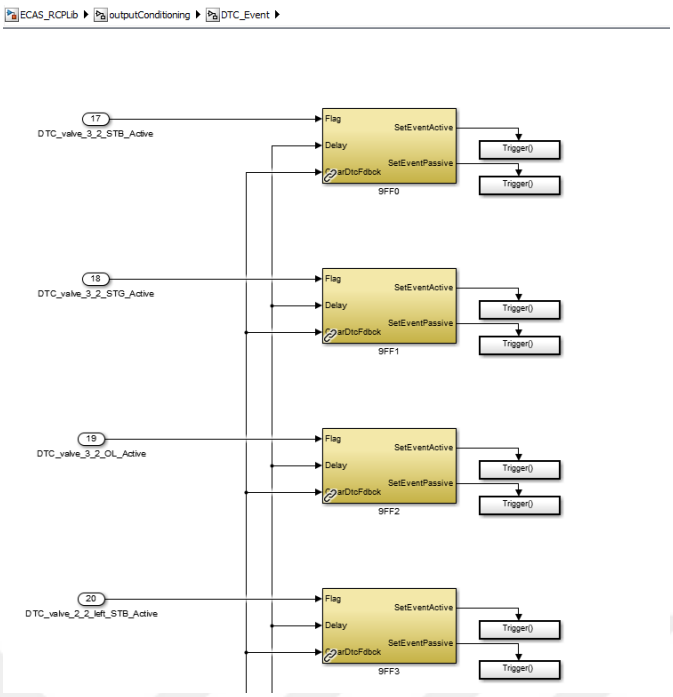
DID (Data Identifier), özgün bir veri kümesini diğerlerinden ayıran tanımlayıcılardır. Araç diyagnostik yazılımında, DID'ler Service ID'lerin altına tanımlanmıştır. EEPROM'lara yazılan DID'ler vasıtasıyla EKSKS sistem parametreleri araç yazılımına tanımlanmıştır.

Çıkış Dönüştürme Bloğu

Çıkış dönüştürme bloğunda, DTC (Diagnostic Trouble Code) adı verilen hata mesajlarının basılması işlemi yapılmıştır. DTC'ler sistemde oluşan hata durumlarının tanımlanması ve belli kodlara atanması ile oluşturulur. Oluşturulan bu kodlar, DTC şeklinde araçta uyarı ikazlarına ve serviste de hata mesajına dönüştürülerek kullanılır. Bu sayede hatanın tanımı, kullanıcı ve servise aktarılmış olmuştur. DTC tanımları servisler için ayrıca yayınlanmış ve tamir metotları belirlenmiştir.

DTC'ler fonksiyonda tespit edilen hata durumuna göre oluşturulmuştur. Hata aktif olduğu sürece durumu aktif olmalıdır. Ancak hatanın ortadan kalkması durumunda DTC durumu pasif olarak kalmalıdır. Pasif DTC'ler kullanıcıya ikaz oluşturmamalı ancak servis bağlandığında görebilmelidir. Böylece araçta ve fonksiyonda ara ara oluşmuş ancak giderilmiş ya da devreden çıkmış hataların görünmesi sağlanmıştır. Bu, fonksiyon ve araç geçmişi açısından oldukça önemlidir.

Hatanın pasife çekilmesi için sürekli olarak kontrol edilmesi gerekmektedir. Araç her açıldığında ya da sistem her uyandığında sistemin kedini kontrol etmesi var olan DTC'leri uyarı olarak basması, artık var olmayan hatalar için ise DTC'lerin pasife geçmesi gerekmektedir. Şekil 4.15'de yazılım içinde bu işlemlerin yapıldığı bloklar gösterilmiştir. DTC'lerin silinmesi ancak servis bağlandığında özel bir bilgisayar ve yazılım ile sağlanabilir. Bu şekilde yapılan bütün işlemler de server üzerinden kaydedilmekte ve gerektiğinde incelenmek üzere saklanmaktadır.



Şekil 4.15 EHSKS yazılımı; DTC (hata kodu) oluşturma fonksiyonları

Aynı zamanda çıkış dönüştürme bloğu doğrudan bağlı olan donanımların çıkış sinyallerinin basıldığı bloktur. Bu çıkışlar fonksiyona göre aktif edilerek, uygulama ve donanım alanlarında fiziksel sinyaller oluşturulur. Bu sinyaller ile sistemin eyleyicileri olan valfler kontrol edilmiştir.

Çıkış dönüştürme bloğunda, CAN_BUS üzerinden gönderilen sinyallerin çıkış dönüşüm işlemleri yapılmıştır. Her bir CAN sinyalinin gönderilmesi gereken bir data aralığı vardır. Bu aralıkta sinyaller EHSKS yazılımı ana fonksiyon bloğunda oluşturulmuş ve çıkış dönüştürme bloğuna yazılım içinde gruplama metodu olarak kullanılan Bus'lara ile gönderilmiştir. Çıkış dönüştürme bloğu içinde ise uygulama katmanına iletilmesi sağlanmıştır.

Çıkış değerlendirme bloğunun, çalıştığı diğer bir fonksiyon da kalıcı hafıza birimlerinin yazılmasıdır. Bu birimler tek tek yazılır, belirli dönüştürme işlemlerinden geçirilir. Örneğin bir hafıza gözüne yazılan 16 byte'dan oluşan bir verinin her bir byte'ı başka bir parametre olabilir. Çoğunlukla da öyledir. Dolayısıyla bu uzun verinin tek tek ayrılarak hafıza gözüne yazılması işlemi giriş değerlendirme bloğunda yapılır. Çoğu zaman veri dönüşümü yapılması gereklidir. Parametrenin yazılabilmesi için belli katsayılar ile çarpılması veya bölünmesi gerekmektedir.

NvRAM adını alan bu hafıza alanı ve DID adı verilen hafıza gözlerine bit ve byte cinsinden veriler yazılabildiğinden noktalı, ondalık veya işareti olan, negatif gibi, değerler yazılıp okunamadığından bu veriler belli katsayılar veya ön işlemler ile değerlendirilip yazılır ve okunduktan sonra da yine bu işlemler ile dönüştürülerek kullanılır. Yazma işlemi çıkış dönüştürme bloğunda yapılır. (Şekil 4.10)

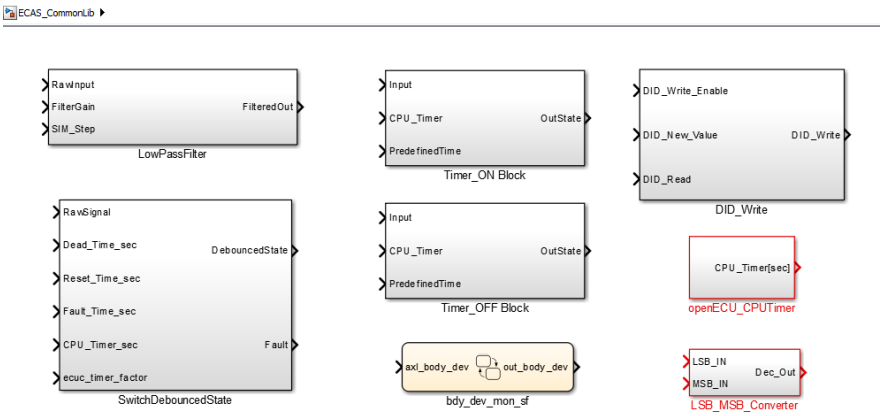
4.4.1.3. EHSKS Yazılım Geliştirme Kütüphaneleri

Yazılım geliştirme sürecinin grup çalışmasına uygun bir şekilde devam edebilmesi için çoklu erişimli bir ortak alan ve kütüphane mantığı uygulanmıştır. Geliştirilen fonksiyonlar ilgili kütüphanelerde kaydedilmiştir. Kaydedilen bu fonksiyonlar RCP(Rapid Control Prototype) adı verilen birleştirme kütüphanesinde birbiri ile etkileşimleri tanımlanarak bağlanmıştır.

Böylece, kütüphane ortamında bir yazılım ve blok güncellemesi yapıldığında, sistem içinde nerede kullanıldığını bulmaya çalışmaksızın otomatik olarak güncelleme yapılması sağlanmış olmuştur. Özellikle birden fazla alt fonksiyon içinde kullanılan filtreleme fonksiyonları gibi genel uygulama fonksiyonları bu yöntem ile oldukça verimli ve hızlı güncellenebilmiş ve kullanılabilmiştir.

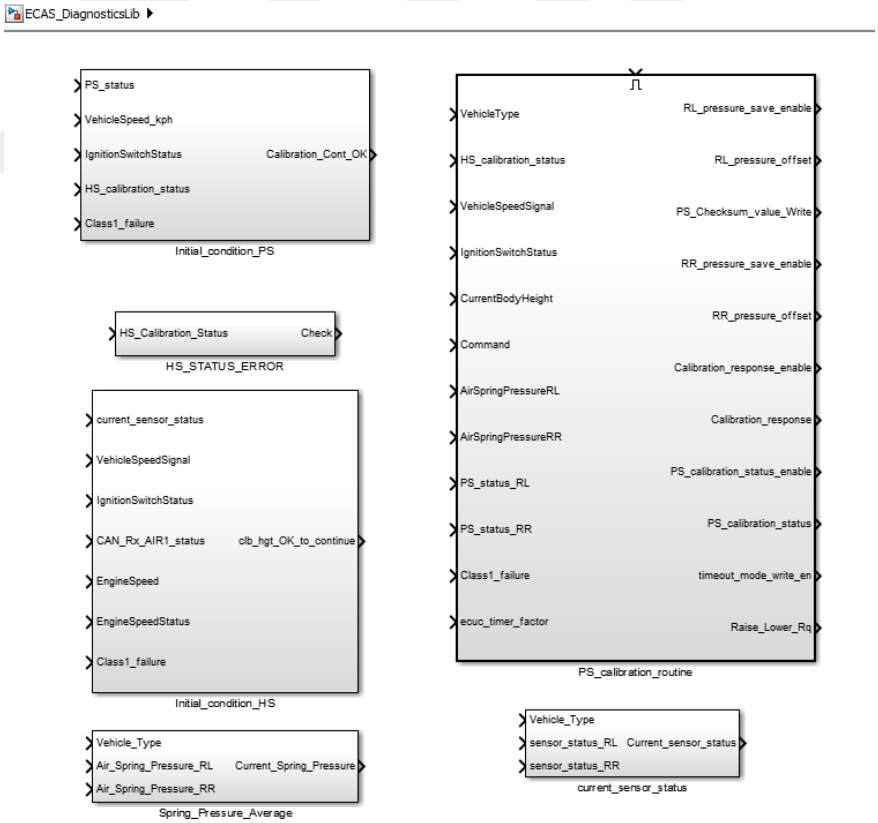
EHSKS geliştirilirken kullanılan kütüphaneleri şöyle sıralayabiliriz;

- **EHSKS ortak kullanım fonksiyonları kütüphanesi:** bu kütüphane giriş verileri için kullanılan filtreler, zamanlama blokları, giriş butonları için basılma durumu değerlendirme gibi blokları içerir. Bu bloklar (Şekil 4.16) kullanırken değişmeden kullanılmak üzere tasarlanmıştır.



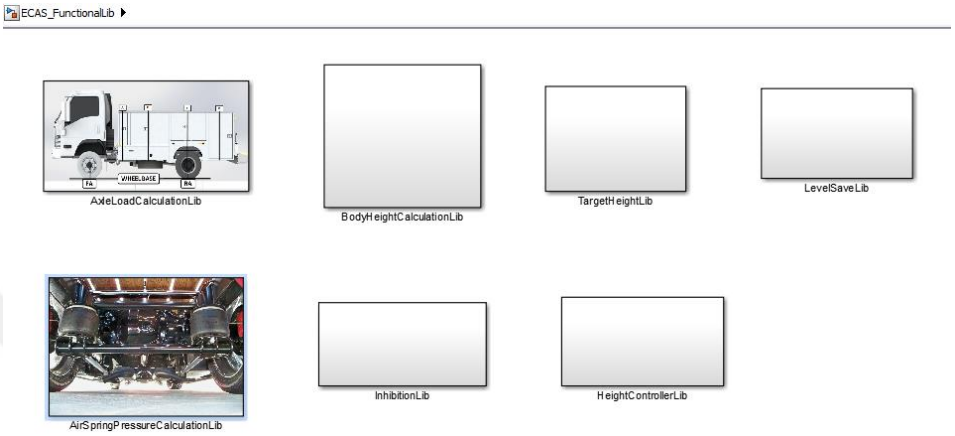
Şekil 4.16 EHSKS yazılımı; EHSKS ortak kullanım fonksiyonları kütüphanesi

- **EHSKS diyagnostik fonksiyonları kütüphanesi:** bu kütüphane (Şekil 4.19) sistemin diyagnostik fonksiyonlarını içerir. Bu fonksiyonları sensör kalibrasyonları, parametre yazma işlemleri olarak kısaca tanımlayabiliriz.



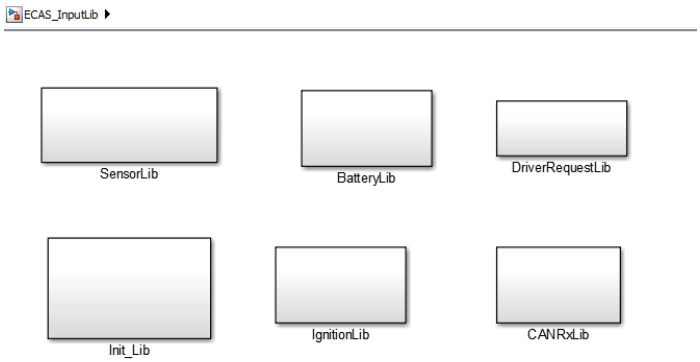
Şekil 4.17 EHSKS yazılımı; EHSKS diyagnostik fonksiyonları kütüphanesi

- **EHSKS güvenlik fonksiyonları kütüphanesi:** bu kütüphane, sistemin hata ile ilgili olan iki en temel fonksiyonunu içerir. Sağlama yapma ve DTC oluşturma ve kontrol etme fonksiyonları bu iki temel fonksiyondur.
- **EHSKS fonksiyonel sistem fonksiyonları kütüphanesi:** bu kütüphane (Şekil 4.18) EHSKS'nin temel fonksiyonlarını içerir. Örneğin dingil yükü hesaplama bloğu bu kütüphanenin içindedir. Bu fonksiyon sistemin temel fonksiyonudur.



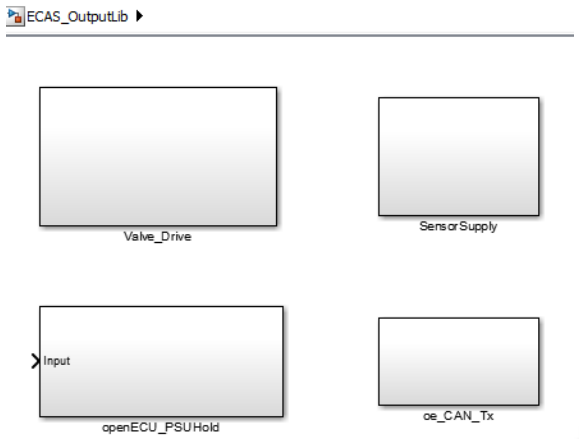
Şekil 4.18 EHSKS yazılımı; fonksiyonel sistem fonksiyonları kütüphanesi

- **EHSKS giriş fonksiyonları kütüphanesi:** bu kütüphane (Şekil 4.19) sistemin girişlerinin işlendiği blokları içerir. Örneğin, sensör girişleri sistem için giriş olduğundan sensör girişlerinin değerlendirildiği fonksiyon blokları bu kütüphanenin içinde yer alır.



Şekil 4.19 EHSKS yazılımı; giriş fonksiyonları kütüphanesi

- **EHSKS çıkış fonksiyonları kütüphanesi:** bu kütüphane sistemin çıkışlarının işlendiği blokları içerir. Örneğin, valf sürme blokları sistem için fiziksel çıkışlar olduğundan bu fonksiyon blokları (Şekil 4.20) bu kütüphanenin içinde yer alır.



Şekil 4.20 EHSKS yazılımı; çıkış fonksiyonları kütüphanesi

- **EHSKS RCP (hızlı prototip) kütüphanesi:** Bu kütüphane, bütün bu kütüphaneler içinde bulunan fonksiyonların bir araya gelmiş ve ilgili olanların etkileşimde olduğu kütüphanedir. Temel olarak 3 bloktan oluşur. Bu bloklar, Şekil 4.10'de de gösterilmiş olan Giriş dönüşüm bloğu, ana fonksiyon bloğu ve çıkış dönüşüm bloğudur.

4.4.1.4. EHSKS Yazılımı Ana Fonksiyon Bloğu

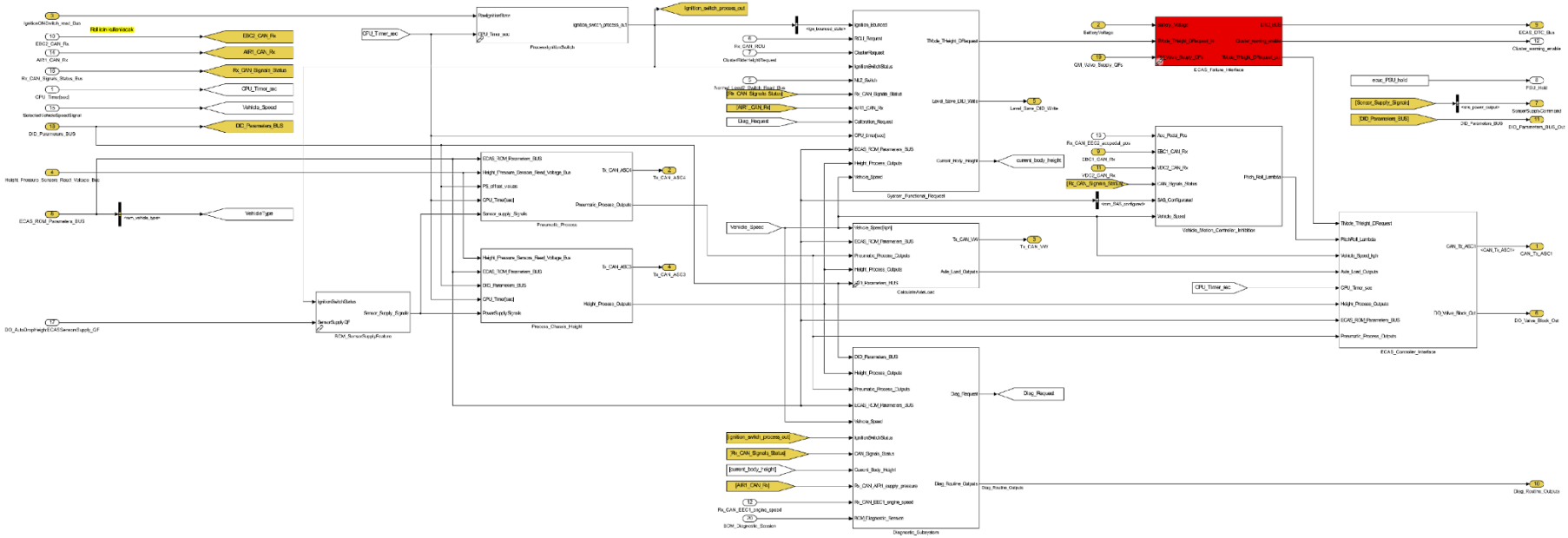
Tez kapsamında, yazılım fonksiyonları geliştirirken V döngüsü temel alınmıştır. Daha önce de bahsedildiği ve V döngüsü referansının da önerdiği gibi, alt sistem yazılım geliştirme süreci öncelikli olarak gereksinimlerin belirlenmesi ve buna uygun olarak veri akış diyagramlarının oluşturulması ile başlamıştır.

Üst seviye sistem tasarımında ortaya çıkan üst fonksiyonların her biri için alt fonksiyonlar belirlenmiş bunlar için yazılım geliştirme süreci bağımsız olarak başlamıştır. Bu süreç 4.4.1.3 başlığında anlatılmış olan kütüphanelerin oluşturulması ile başlamıştır. Oluşturulan kütüphaneler kullanılarak etkileşimler tanımlandığında yazılımın ana fonksiyon bloğu oluşturulmuş olur.

Ana fonksiyon bloğunun giriş ve çıkış dönüştürme bloklarından bağımsız olmasındaki temel amaç, giriş ve çıkış bloklarının kullanılan ECU'ya göre hızlıca değişmesi ile ana fonksiyon bloğunun değiştirilmeden kullanılabilmesidir. Bu şekilde, kullanılan ECU ve özellikleri değiştiğinde; giriş ve çıkış ara yüzlerine uygun dönüştürme blokları oluşturularak, temel fonksiyon bloğu doğrudan kullanılabilir olacaktır.

Bu yazılım geliştirme stratejisi, EHSKS yazılımının donanımdan bağımsız ve değişen donanıma hızla adapte olabilen bir yazılım haline gelmesini sağlamıştır.

Ana fonksiyon bloğu içerisinde sistemin temel özelliklerini içeren fonksiyonlar birbiri ile bağlanmış halde Şekil 4.21 'deki gibi görülebilir.



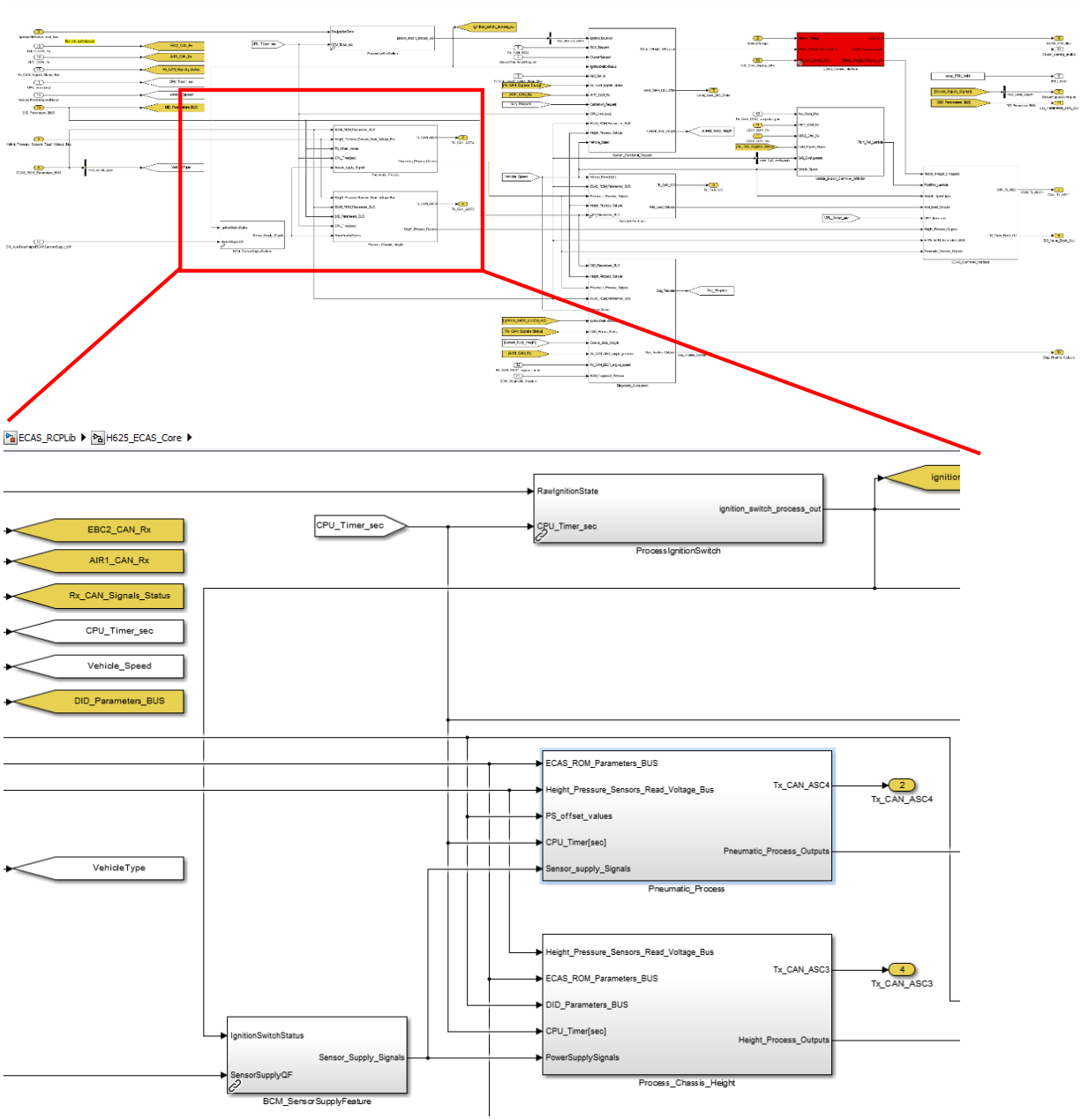
Şekil 4.21 EHSKS yazılımı; ana fonksiyon bloğu(RCP) genel görünümü

Fonksiyonlara yakından bakacak olursak;

Ana fonksiyon bloğu soldan sağa doğru girişlerin değerlendirilmesi ve sonuçlandırılması, karar ve çalışma durumlarının oluşturulması ve çıkışların basılması gibi fonksiyonların tamamını kapsamaktadır.

Kontak sinyalinin durumu “Process Ignition Swtich” bloğunda hesaplanır (Şekil 4.24). Tez konusu olan EHSKS yazılımı kontak açılması ile birlikte aktive olduğu için sistem için önemli olan girişlerin başında kontak sinyali ve bu sinyalin durumu gelmektedir.

Aynı şekilde sensörlerin akım çekip çekmemesinden yola çıkarak sensör besleme pinlerinin durumu ve dolayısıyla sensörlerin bağlantı durumu test edilir. Bu değerlendirme bloğu da EHSKS yazılımı için oldukça kritiktir, sensör besleme pininde oluştuğu fark edilen bir sinyal sistemin çalışmasını doğrudan kapatabilmektedir. Bu blok, “Sensor Supply Failure ” ismi ile tasarlanmıştır.



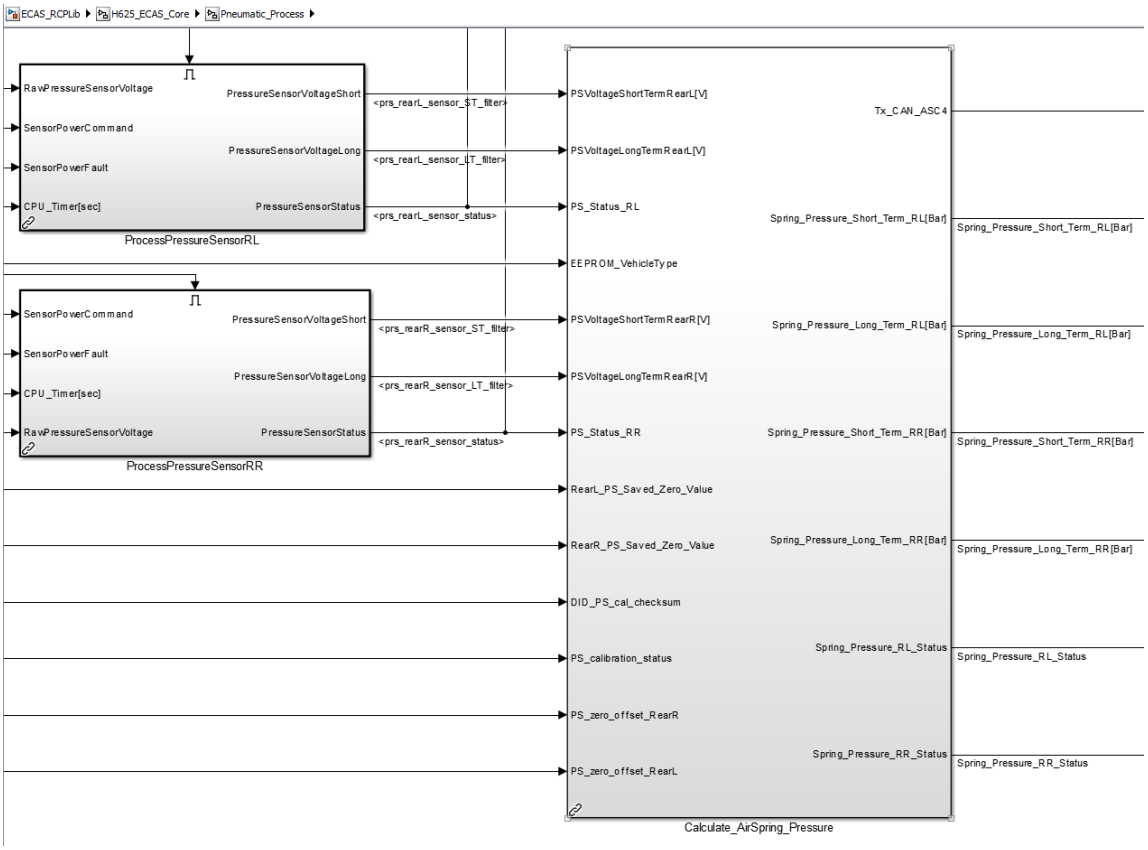
Şekil 4.22 EHSKS yazılımı; ana fonksiyon bloğu giriş odaklı çalışan fonksiyonlara genel bakış

Pnömatik ve yükseklik değerlendirme blokları, basınç ve yükseklik sensörlerinden gelen ham sensör voltajlarını işleyerek sistem içinde kullanılabilir bar cinsinden basınç ve mm cinsinden yükseklik değerlerine dönüştürmek üzerine tasarlanmıştır.

Şekil 4.23'le gösterilen, bu bloğun altında "Process pressure sensor rear left" ve "Process presure sensor rear right" olarak tanımlanmış iki sensör tarafı için gelen voltajı değerlendirme blokları bulunur. Bu bloklar, gelen ham sensör voltajlarının

doğru aralıkta olup olmadığını kontrol ederek farklı frekanslarda filtreleme işlemlerini gerçekleştirilmiştir.

Filtreleme ve durum değerlendirmesi yapılan sensör voltajları “Calculate air spring pressure” bloğunda sensör transfer fonksiyonu kullanılarak mV değerinden KPa ve bar değerine dönüştürülür ve her bir sensör için çıkış olarak diğer fonksiyonlara sağlanır.



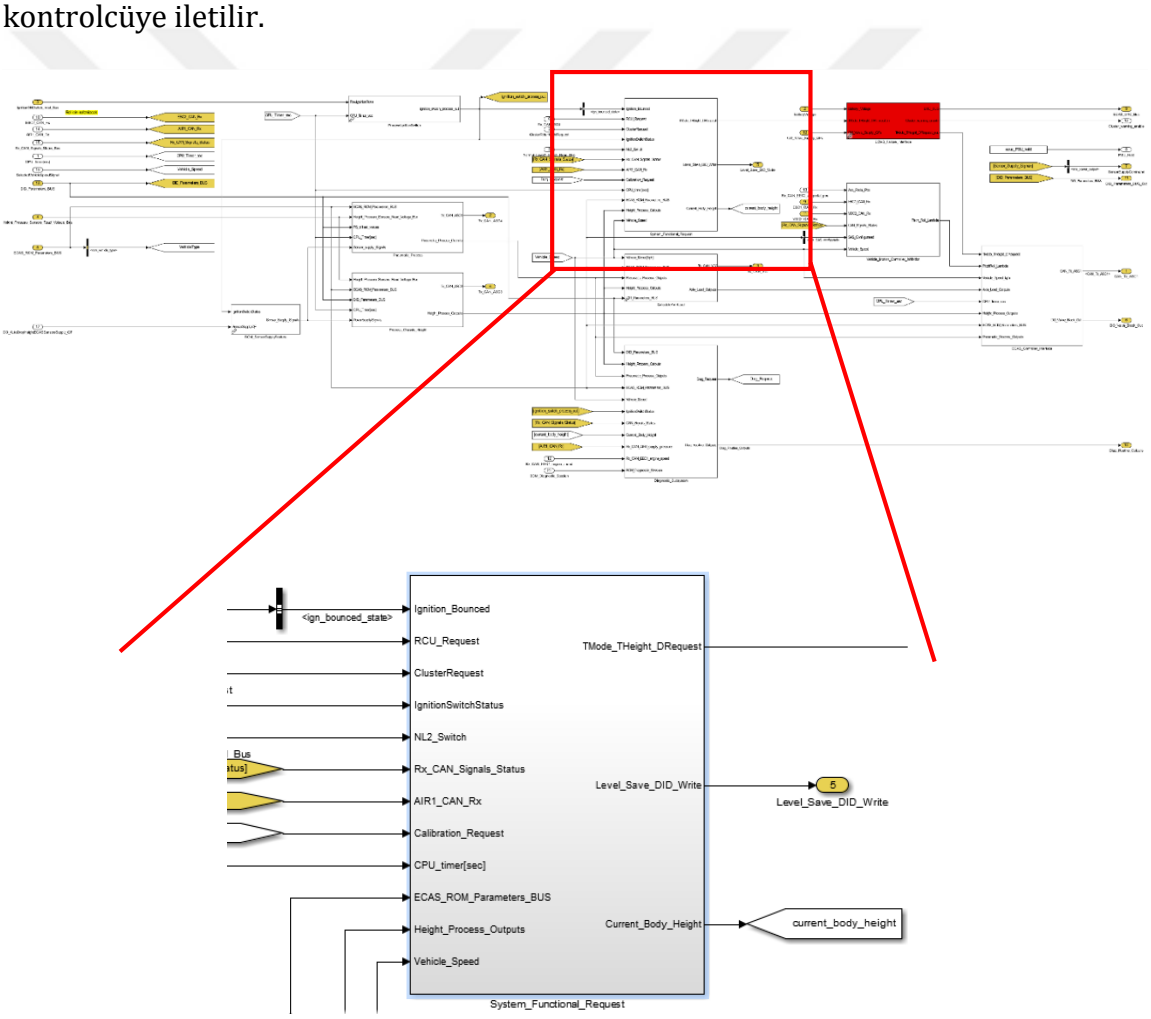
Şekil 4.23 EHSKS yazılımı; pnömatik proses fonksiyon bloğuna genel bakış

Aynı durum yükseklik değerlendirme bloğu için de geçerlidir. Bu blokta da gelen ham voltaj değerleri filtrelenir ve durumları değerlendirilir. Filtrelenmiş voltaj değerleri ve durumlar kullanılarak yükseklik sensörlerinin transfer fonksiyonlarına sokulur. Çıkış olarak ise mm cinsinden yükseklik değeri hesaplanır ve her bir sensör için ayrı olarak sistem içinde diğer fonksiyonlarda kullanılmak üzere dışarı basılır.

Daha önceki başlıklarda sistemin kullanıcı girişlerine göre manuel olarak kontrol edilebilmesi gereksiniminden bahsedilmiştir. Bu girişlerin işlenmesi birbiri

üzerinde öncelik ve etki durumlarının değerlendirilmesi için ise Şekil 4.24'da da gösterilen, "System functional Request" adı verilen fonksiyonel sistem isterleri bloğu tasarlanmıştır. Bu fonksiyon temel olarak; göstergeden, butonlardan, kumandadan veya siste içindeki herhangi bir talepten kaynaklanan isterleri karşılaştırıp kontrolcüye bir ister olarak iletecek şekilde çalışmaktadır. İsterin devreye alınabilmesi için gereken azami koşulları değerlendirir. Birden fazla isterin önceliklendirilmesini sağlar. İsterlerin aktif veya deaktif olma koşullarını belirler.

Aynı zamanda kontrolcüye bir hedef noktası ve bu hedef noktasının yükseklik bakımından değerini iletir. Bazı hedefler için yükseklikler sistem çalışırken kaydedilir ve ister durumuna göre bu değerler geri okunarak hedef yükseklik olarak kontrolcüye iletilir.



Şekil 4.24 EHSKS yazılımı; ana fonksiyon bloğu, kullanıcı girişlerinin işlendiği fonksiyon grubuna genel bakış

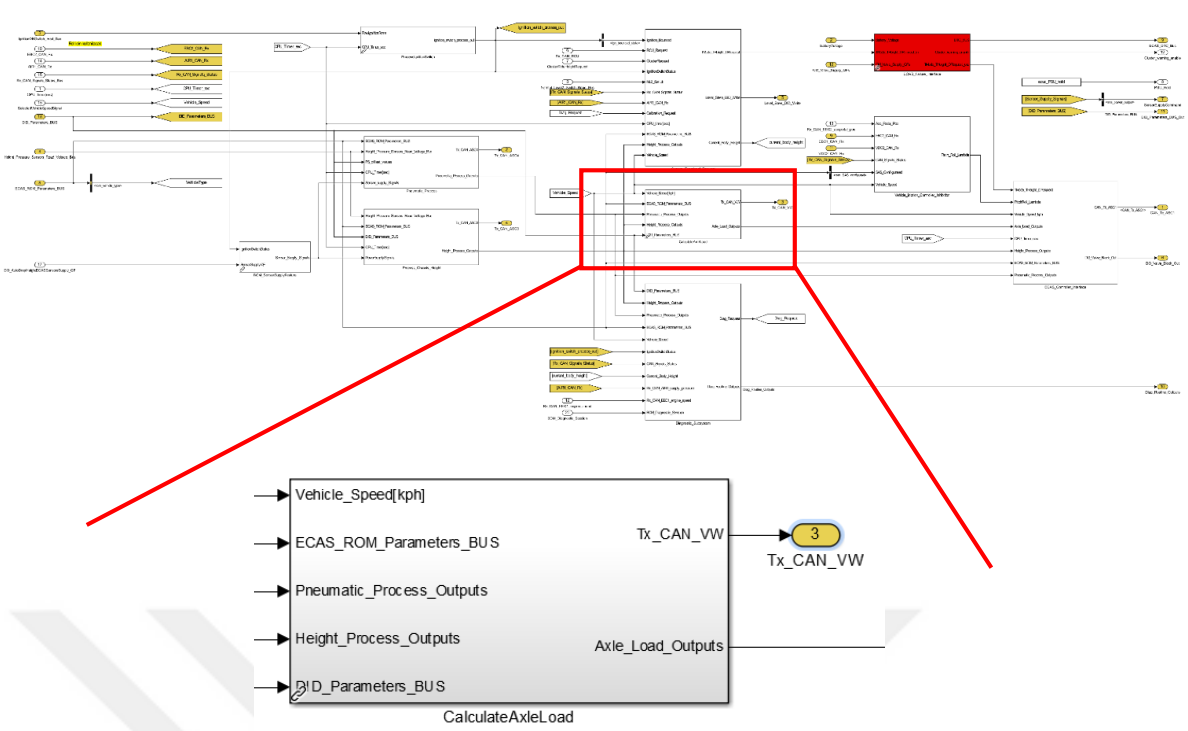
Kullanıcı girişlerinin işlendiği fonksiyon bloğu birden çok katmanlı alt fonksiyonlardan oluşur. Kısaca bahsetmek gerekirse, buton girişlerinde kitlenme olup olmadığının kontrolü bu alt fonksiyonlarda kontrol edilir.

Birden fazla kullanıcı giriş alanı mevcuttur, bunlar kumanda ve gösterge paneli üzerinden kontrol olarak sıralanabilir. Bu iki giriş alanından aynı anda ya da birbirine zıt isterlerin gelme durumu da bu alt fonksiyon bloklarında gerçekleştirilir.

Aynı zamanda, kullanıcının kaydetmek istediği ve gerektiğinde çağırmak istediği yükseklik değerleri bulunmaktadır. Bunlar kumanda üzerinden kısa yollar vasıtası ile kullanıcıya kolaylık olması amacıyla tanımlanmıştır (Memory1 ve Memory2 butonları). Bu fonksiyonlar bulunulan yükseklik değerini NvRAM gibi kalıcı hafıza gözlerine kaydedebilmeyi ve ister girildiğinde geri çağırabilmeyi sağlamaktadır.

EHSKS diyagnostik bağlantı sırasında iken kullanıcı isterleri dikkate alınmaz. Yalnızca diyagnostik sistemi üzerinden iletilen isterler değerlendirilerek kontrolcü üzerinden çıkışa aktarılır. Sistemin diyagnostik durumda olup olmadığı ve buna göre isterlerin değerlendirilmesi de bu fonksiyon bloğunda gerçekleştirilir.

Bunlar gibi birçok alt fonksiyonun çalıştırılması ile fonksiyonel sistem isterleri bloğunda birtakım çıkışlar oluşturulur. Bu çıkışlar, sistemin tanımlı ise bir hedefi, bu hedefin yükseklik değerini ve eğer hedef tanımlı değilse yalnızca işlem talebini kontrolcü bloğuna iletir



Şekil 4.25 EHSKS yazılımı, dingil yükü hesaplama bloğu

Şekil 4.25'de de gösterilmiş olan dingil yükü hesaplama bloğu, aracın canlı dingiline düşen yükü yükseklik ve basınç verilerini değerlendirerek hesaplar. Hesaplanan aks yükü bilgisi CAN_BUS aracılığı ile araç içindeki diğer modüllerin, kullanımına iletilir. Bu aks yükü verisi EBS kontrol modülü tarafından kullanılır. EBS, CAN_BUS üzerinden okuduğu aks yükü bilgisinden yaklaşık olarak araç ağırlığını hesaplamak üzere kullanır. Bu hesaplanan veriye göre fren kuvvetlerinin kontrolü yapılır. Dolayısıyla aks yükü bilgisi doğru basılamıyor ise, araç fren kuvvetleri beklenenden az veya çok olarak uygulanabilir. Bu da öncelikli olarak güvenlik, ikincil olarak da konfor sorunlarına sebebiyet verebilir.

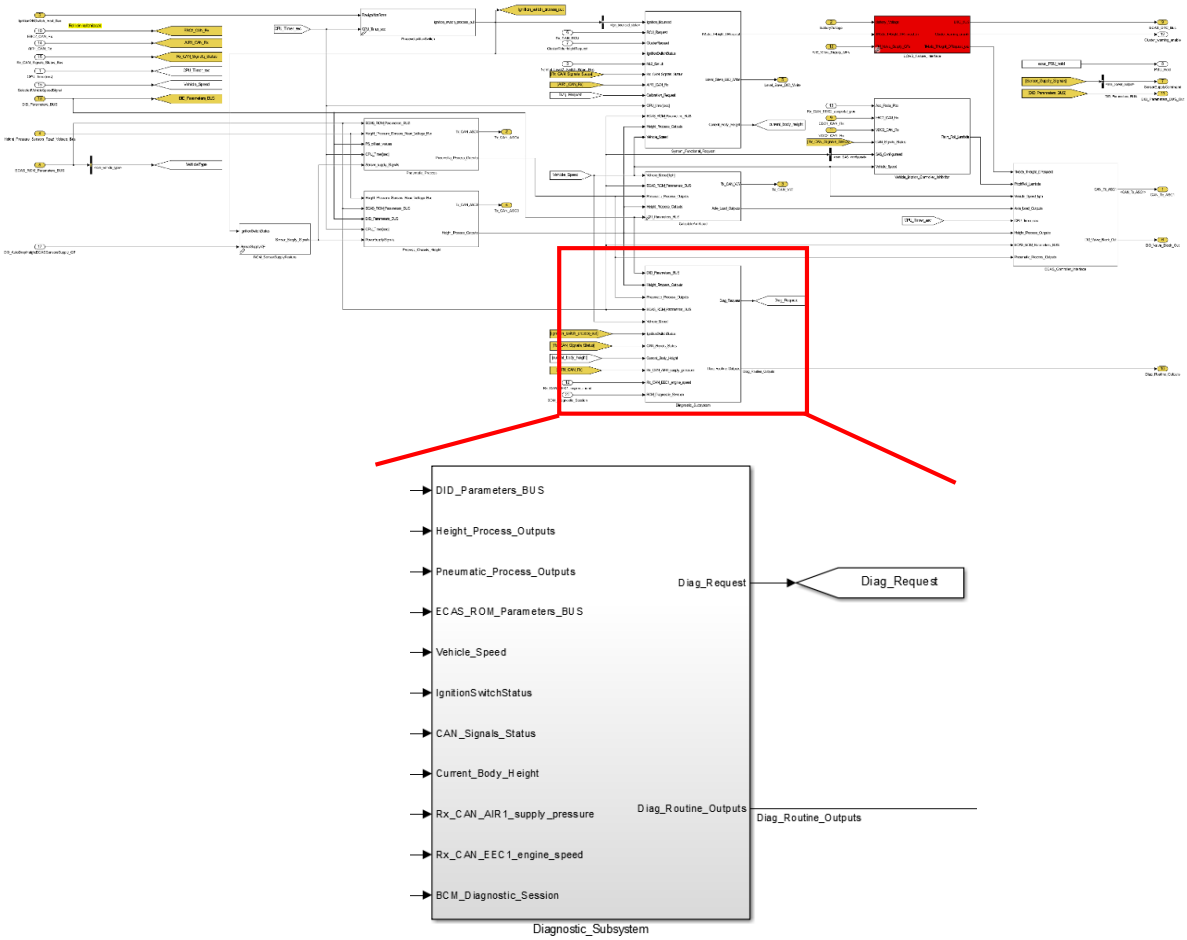


Şekil 4.26 EHSKS yazılımı, dingil yükü değerinin gösterge üzerinden verilmesi

Tez konusu olan EHSKS yazılımı, fabrikada hat sonu denilen noktada araçlara yüklenir. Bu yükleme işlemi diyagnostik sistemler aracılığı ile yapılır. Yazılım yüklemenin ardından sistem için gerekli ve araçtan araca değişiklik gösteren parametrelerin araca yazılması, kısaca konfigürasyon yazılması işlemi de bu tanılama sistemleri ile gerçekleştirilir.

Hat sonuna bu işlemlerinin yapılmasını yanında, araç servise gittiğinde ve herhangi bir sorun ve bakım aşamasında da bu sistem kullanılır. Diyagnostik sistemler aynı zamanda araçta var olan hataların tespiti, tanısı ve çözümü için de kullanılır.

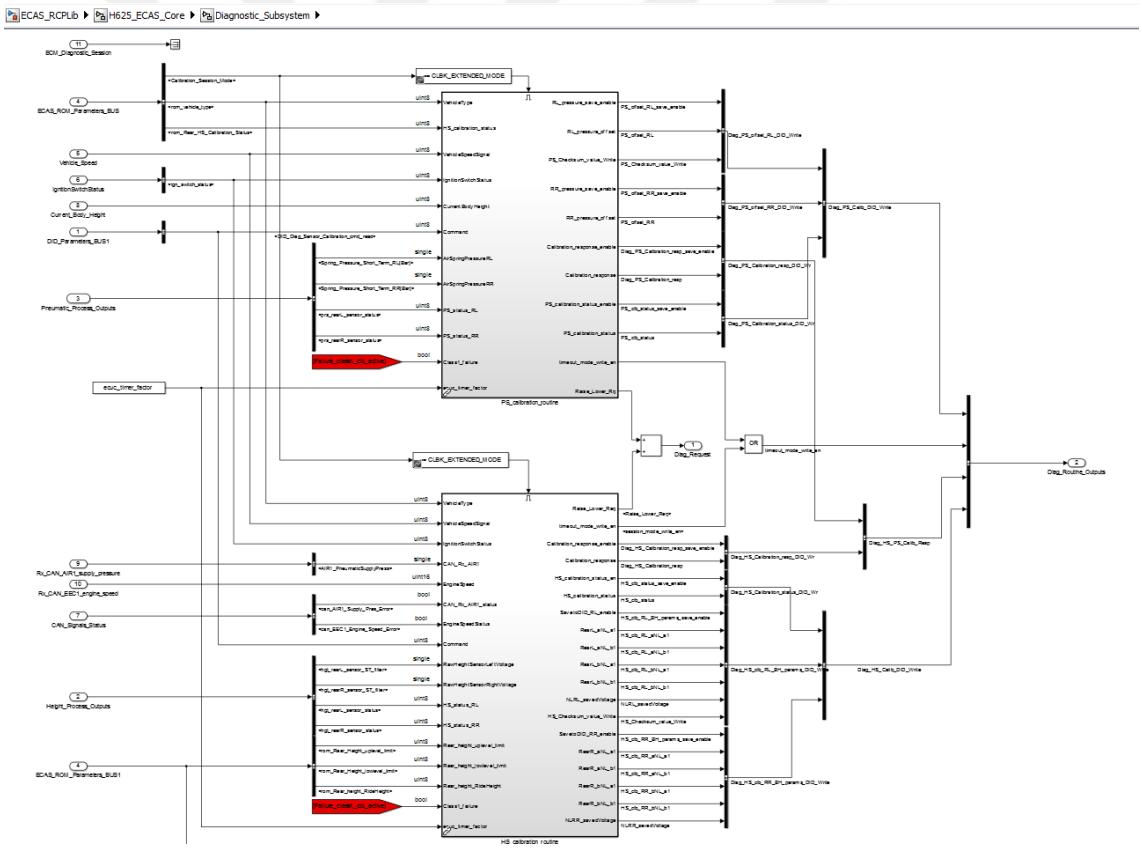
EHSKS yazılımının diyagnostik sistemler ile iletişiminin sağlanması amacıyla diyagnostik fonksiyonlar bloğu oluşturulmuştur (Şekil 4.27). Bu blok diyagnostik sistemler ve yazılım arasındaki ara yüzü ve iletişimi sağlar.



Şekil 4.27 EHSKS yazılımı, diyagnostik sistemler fonksiyon bloğu

Diyagnostik fonksiyonların kalibrasyon ara yüzünü ele alacak olursak Şekil 4.41'de görülen iki ana blok kullanılmıştır. Bu bloklardan biri basınç sensörü kalibrasyonunu diğeri yükseklik sensörü kalibrasyonunu sağlamaktadır.

Basınç sensörleri daha önce de bahsedildiği gibi, üzerine gelen hava basıncı ile diyaframın hareket etmesi sonucu volt değeri ile sinyal gönderen bir elektronik devredir. Bu sensör, göreceli olarak ölçüm yaptığından, kalibrasyon sırasında ofset değeri belirlenir. Körükler içindeki hava tamamen boşaltılarak, sensör ölçümündeki göreceli sıfır noktası belirlenmiş olur. Bundan sonra okunan değerler üzerinden yazılım içinde belirlenen bu sıfır değeri çıkarılarak asıl voltaj aralığı ve dolayısıyla basınç değeri bulunmuş olur. Bu göreceli sıfır değeri; kalibrasyon sırasında, daha önce de bahsedilen NvRAM'lere (kalıcı hafıza gözlerine) kaydedilir. Şekil 4.29 bu işlemlerin yazılım içindeki fonksiyon bloklarını ifade etmektedir.



Şekil 4.28 EHSKS yazılımı, diyagnostik sistemler fonksiyon bloğu kalibrasyon alt fonksiyonları

Yükseklik sensörü kalibrasyonu üç noktadan yapılır. Aracın en üst en alt ve sürüş yüksekliği noktalarındaki voltaj değerleri NvRAM'lere kalıcı olarak kaydedilir.

Sensör voltajı ve yükseklik bağlantılı olarak çıkarılan bir doğrusal grafiğin eğimi ile kalibrasyon doğrusu çizilmiş olur. Buna göre herhangi bir noktada gelen voltaj değerine göre aracın yüksekliği hesaplanabilir hale gelmiş olur.

Diyagnostik fonksiyon içinde yalnızca kalibrasyonlar değer parametre yazma işlemleri de yapılır. Bu parametrelerin araç tipine göre doğru yazılıp yazılmadığının kontrolünü sağlayan sağlama fonksiyonları vardır. Bu fonksiyonlar yazılması gereken değerleri belirli oran ve modlar ile toplayarak doğru araç tipine beklenen parametreler dışında yazılan bir değer olup olmadığını karşılaştırarak tespit eder.

Diyagnostik sistem ile yazılım ve ECU arasındaki iletişim prosedürü UDS'dir. Bu protokol ile UDS standardını sağlayan bütün diyagnostik sistemler ile ECU iletişim geçebilir ve yata kodu okuma, parametre yazma gibi işlemler gerçekleştirilebilir.

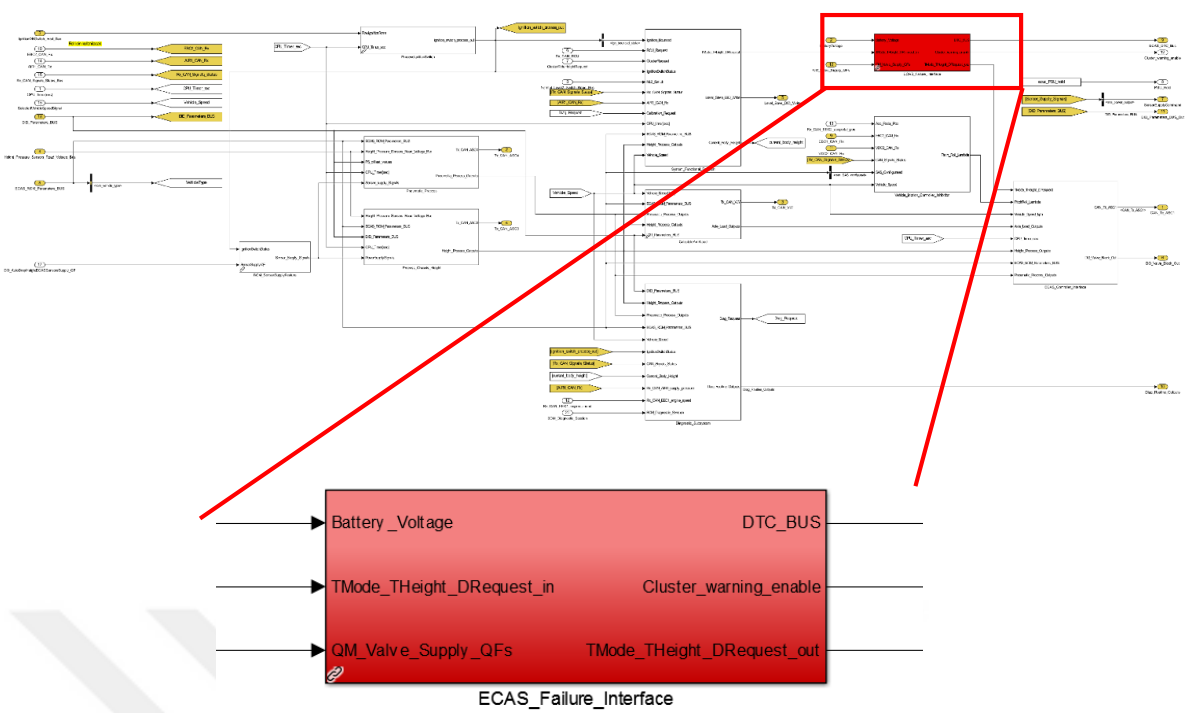
EHSKS yazılımı içinde, yazılımda ve/veya sistemde var olan problemlerin tespit edildiği, tespit edilen problemlerin birbirine göre önceliklendirildiği ve kullanıcı uyarı mesajı oluşturma ve/veya hata kodu oluşturma gibi aksiyonların alındığı hata ara yüz fonksiyon bloğu bulunur.

Bu blok ile hatalar birkaç gruba ayrılır, bazı hata durumlarında sistemin çalışması tamamen engellenirken; bazı hata durumlarında yalnızca kullanıcıya uyarı verilerek (Şekil 4.29) sistem çalışmasına devam edilir. Bazı durumlarda ise sistem güvenlik sebebi ile kısıtlı olarak çalıştırılır.



Şekil 4.29 EHSKS yazılımı, hata durumunun göstergede kullanıcıya iletilmesi

Bu sınıflandırma hata ara yüz fonksiyon bloğu içinde yapılır. Daha kritik olan hata bir öncelikli olarak değerlendirilir. Sistem buna göre kısıtlanır veya hata kodu(DTC) oluşturulur (Şekil 4.30).



Şekil 4.30 EHSKS yazılımı, hata ara yüzü fonksiyon bloğu

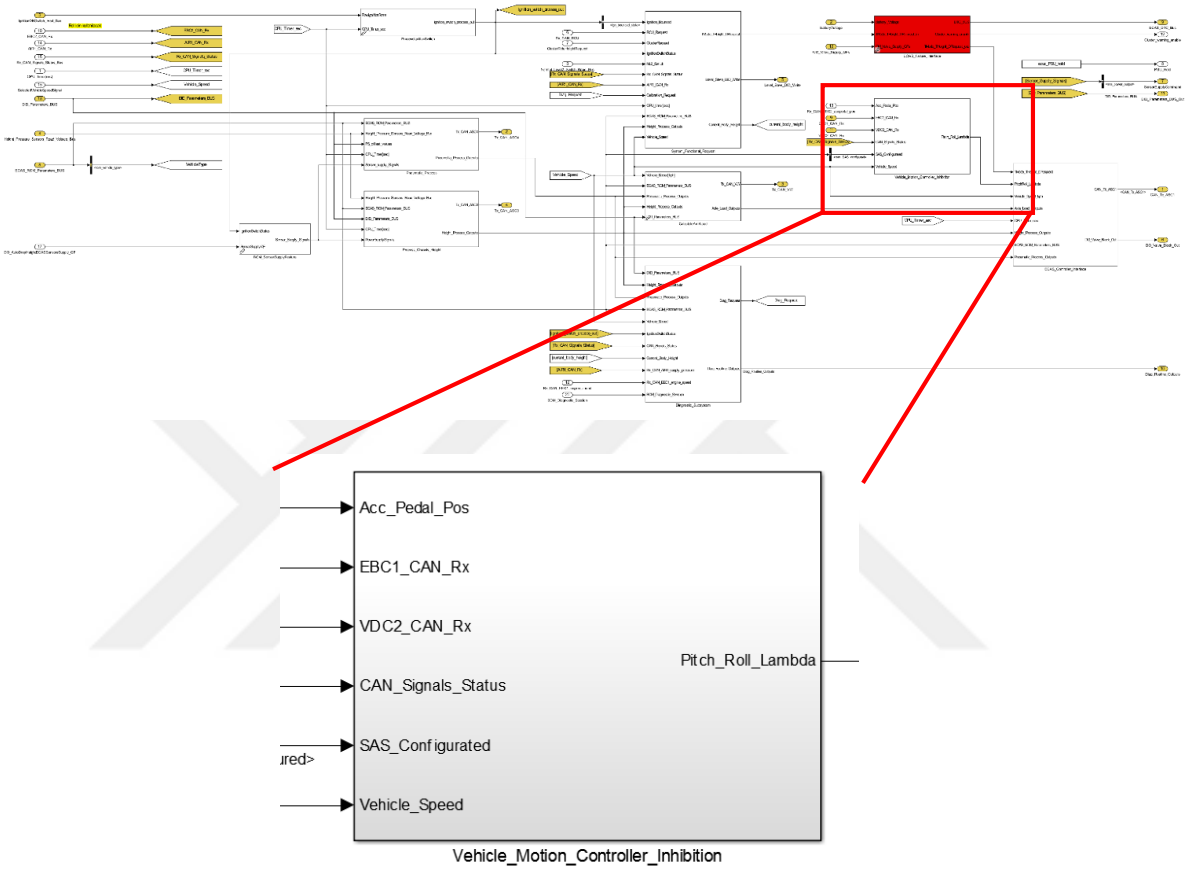
EHSKS yazılımı geliştirilirken, daha önce de bahsedilmiş olan bazı sistem gereksinimleri oluşturulur ve/veya toplanır. Toplanan gereksinimler sistem için gelen (incoming) ya da beklenen sistem gereksinimleri olarak adlandırılabilir. Buna örnek olarak devrilme (roll) hareketi sırasında EHSKS'nin çalıştırılmaması gereksinimi verilebilir. Bu gereksinim aynı zamanda bir güvenlik beklentisi ve gereksinimidir. Havalı süspansiyon sitemlerinin tepki süresinin kısıtlı olması sebebiyle ani hareketlere tepki vermemesi beklenmektedir. Bu durumu devrilme (roll) hareketi için değerlendirecek olursak; aracın sola doğru devrilme yapacak bir viraja girmesi sırasında sistem tepki verip sol körükleri doldurur ise ve bu hareketin hemen ardından sağa dönüş gerçekleşirse, sağ körüklerin soldan daha boş ve hızla doldurulamaz olduğundan aracın devrilme ihtimali oluşmaktadır.

Dolayısıyla, EHSKS yazılımının sistemin devrilmeye (roll)'a girdiğini anlaması ve bu süre boyunca herhangi bir aksiyon almaması gerekir.

Bu kısıtlama aynı şekilde frenleme (diğer bir değişle 'pitch' hareketi) için de geçerlidir. Araç sistem gereksinimlerinden bir diğeri de, frenleme anında fren hava basıncının harcanmasını engellemek adına, EHSKS'nin çalışmaması gereksinimidir.

Bu iki gereksinime göre, sistemin devrilme veya yalpalama hareketini tespit edip buna göre çalışmasını engelleme/kesme (inhibe etme) fonksiyonu oluşturulmuştur (Şekil 4.31).

Bu fonksiyon bloğu ile herhangi bir hedef gelse dahi kontrolcü çıkışları engellenerek, kontrolcünün, dolayısıyla sistemin çalışması kesilmiş olur.

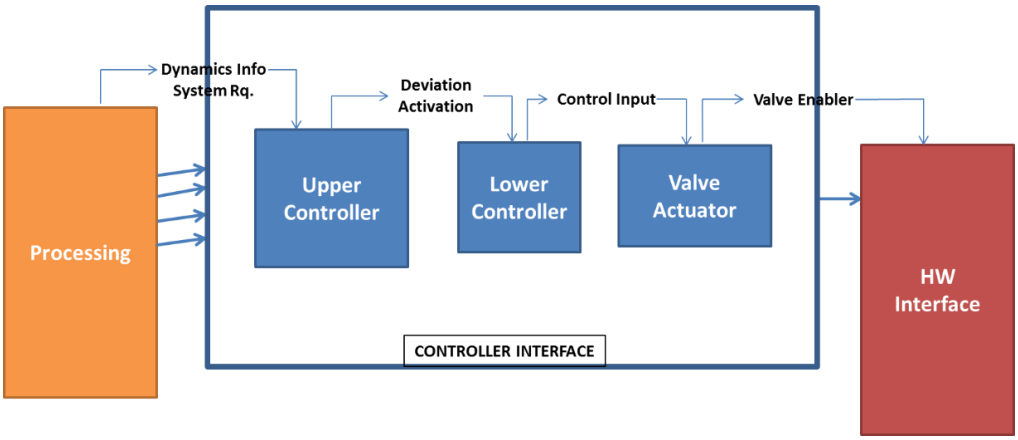


Şekil 4.31 EHSKS yazılımı, kontrolcü çalışmasını kesme fonksiyonu

4.4.1.5. EHSKS Yazılımı Kontrolcü Ara yüzü

Tez konusu olan EHSKS yazılımının kontrolcü yapısı Şekil 4.35'de de görüldüğü gibi 3 alt başlıktan oluşmaktadır.

- Üst seviye kontrolcü
- Alt seviye kontrolcü
- Valf sürücü

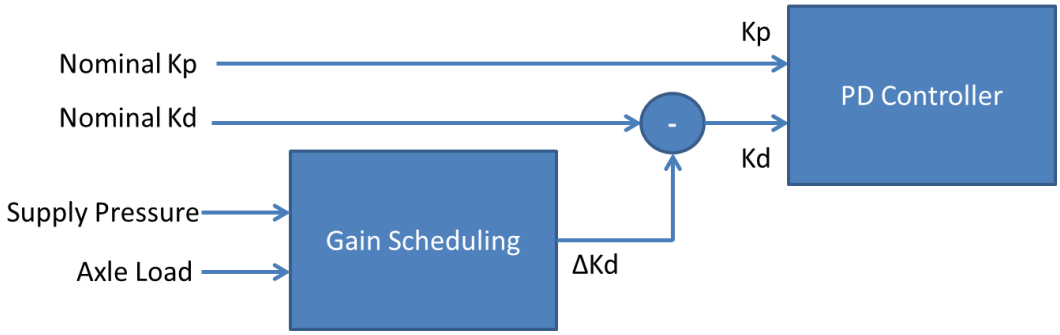


Şekil 4.32 EHSKS yazılımı, kontrolcü yapısı

Üst seviye kontrolcü, kontrolcü bloğuna tanımlanış girişleri işleyerek, yükseklik hedef modunu, hedef değerini ve hedeften sapmayı hesaplar. Alt seviye kontrolcüye çalışması gerekip gerekmediği bilgisini verir. Aynı zamanda sapma değerlerini de CAN_BUS üzerinden ihtiyacı olabilecek diğer modüllere iletir. Bu hesaplamaları yaparken değerlendirdiği bazı sinyalleri/inputları şöyle örnekleyebiliriz; kullanıcı yükseklik ister girişi, hedef yükseklik, araç hızı, kontrolcü kesme fonksiyonu çıkışları... vb.

Üst seviye kontrolcüden iletilen kontrolcü görevine istinaden alt seviye kontrolcüde kazanç ayarlamalı bir PD kontrolcü yapısı kullanılarak valf çıkışları sürülür.

Yükseklik değişim hareketleri (alçalma yükselme) integral etkili olduğundan hedef yüksekliğin sağlanması için PD kontrolcü kullanılmıştır.



Şekil 4.33 EHSKS yazılımı, kazanç ayarlamalı PD kontrolcü yapısı

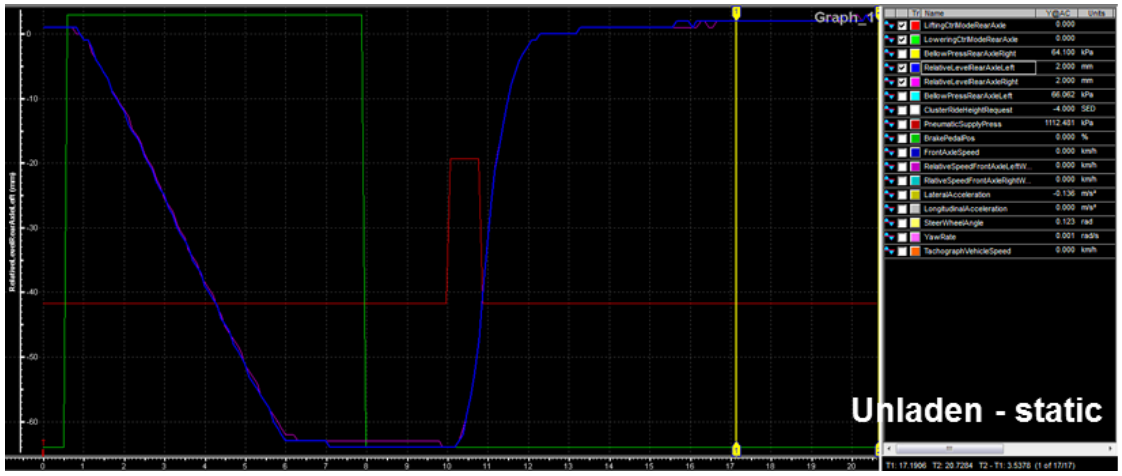
Araç ağırlığı bilgisine göre alt kontrolcü katsayıları güncellenir ve çıkışlar buna göre sürülür. Eğer sistem hatası ve/veya sensör verileri sebebiyle ağırlık bilgisi

hesaplanamıyor ise kontrolcü standart bir katsayı ile çalıştırılacak şekilde tasarımı yapılmıştır.

Şasi hareket oranı (hızı); valf tetiklemesi, besleme basıncı ve dingil yükünün bir fonksiyonu olarak modellenebilir. Bahsettiğimiz bu argümanlar kolayca değişebildiğinden, şasi yükseltme alçaltma hızı da oldukça değişken olabilmektedir. Bu nedenle kontrolcü Kp ve Kd katsayıları için kazanç katsayısı hesaplama algoritması kullanılmıştır.

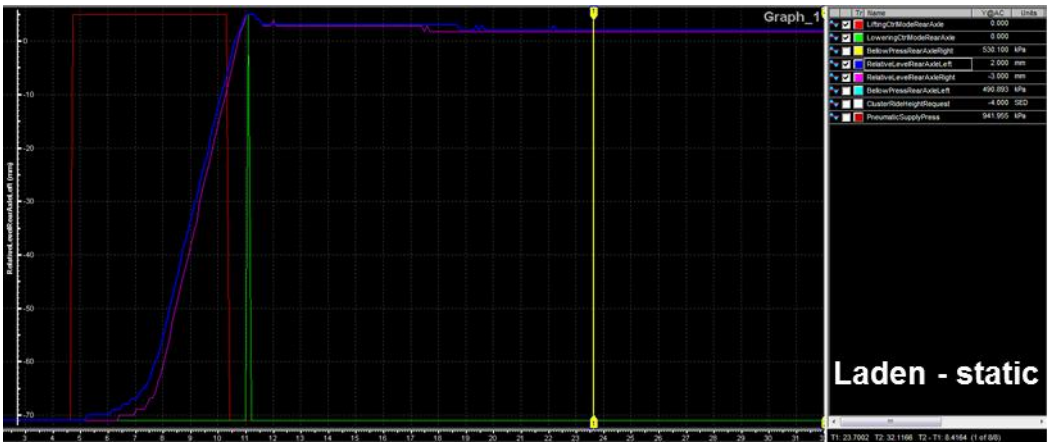
Kazanç ayarlama çalışması kural tabanlı olarak gerçekleştirilmiştir. En yüksek hareket hızında, aşımı önlemek için yüksek bir Kd değeri gerektirmiştir. Bu durum özellikle basıncın 9bar'ın üzerinde olduğu ve aracın yüksüz olduğu durumlarda gözlemlenmiştir. Bunun yanında yüksek Kd değeri hızlı aralıklarla valf aktivasyon sinyallerini uygulamakta, bu da yüklü veya alçak besleme basıncı durumlarında valfin ömrünü ciddi oranda etkilemektedir. Bu sebepten kazanç ayarlama fonksiyonu bir taramalı tablo(look-up table) tanımlanması ile çalışarak besleme basıncı ve yüke göre Kp ve Kd kazancını ayarlar.

Bu sayede kontrolcünün çalışması sırasında daha az aşım yapılması sağlanmıştır. Yüklü olan araçlarda yükseklik düşüşü çok hızlı olurken artış çok yavaş olmaktadır. Buna karşın, boş aracın yükselme hızı çok hızlı, alçalma hızı oldukça yavaştır.

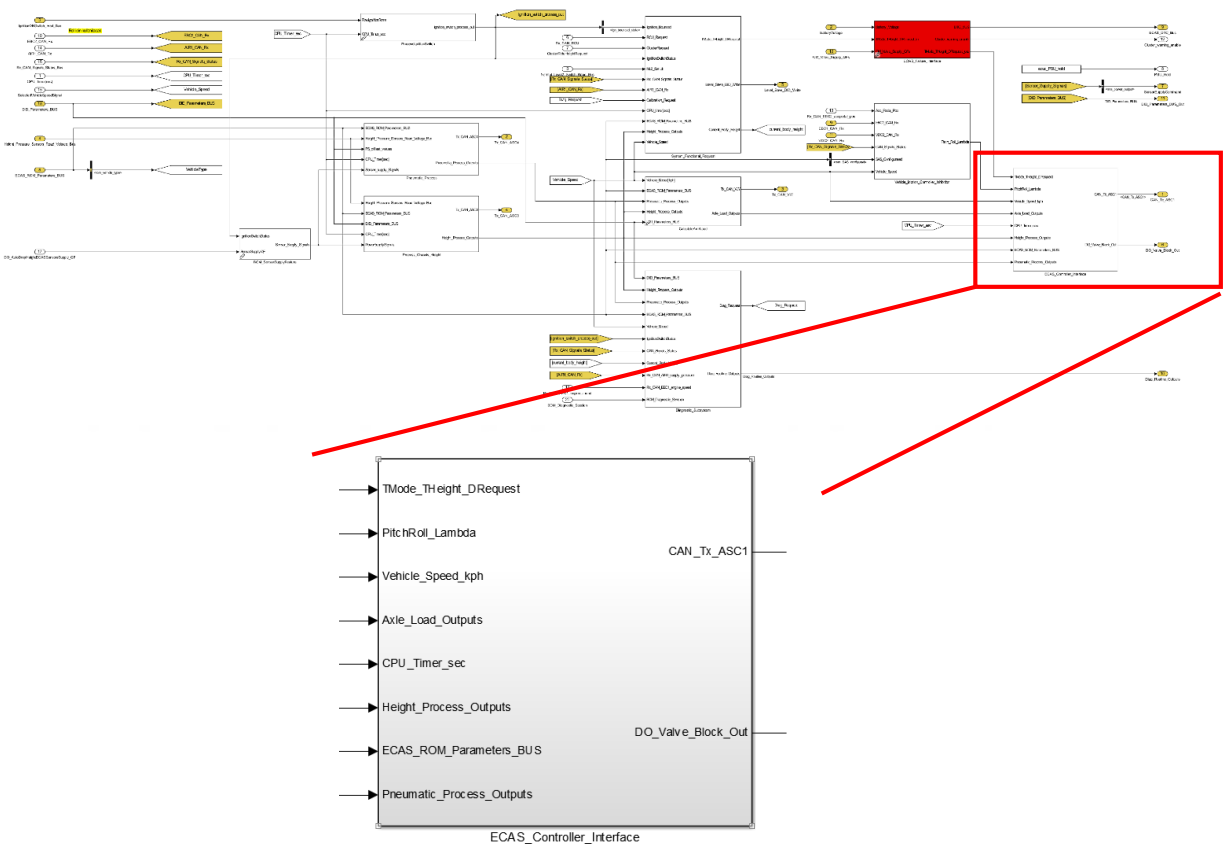


Şekil 4.34 Yüksüz araçta kontrolcü testi verisi

Bu karakteristik farkı aynı kontrolcü katsayılarının kullanılmaması açısından önemli fikir vermektedir. Mevcut sistemlerde böyle bir fark yokken. Tez konusu olan EHSKS yazılımı yüke bağlı olarak kontrolcü katsayılarını güncellemektedir.



Şekil 4.35 10ton yüklü araçta kontrolcü testi verisi



Şekil 4.36 EHSKS yazılımı, kontrolcü fonksiyonu

EHSKS'inde kullanılan valflerin aç-kapa (on-off) valfler olmasından ve kontrolcünün çalışma bant aralığı ECU'ya bağımlı ve 0.5 Hz olduğundan sistemin alt kontrolcü daha düşük olarak 300ms gibi bir bant aralığında çalışmalıdır. Bu da pasif kontrol alanında kalınması anlamına gelmektedir.

Alt kontrolcü, araçta bulunan valf konfigürasyonuna göre kontrolcü çıkışlarını düzenler.

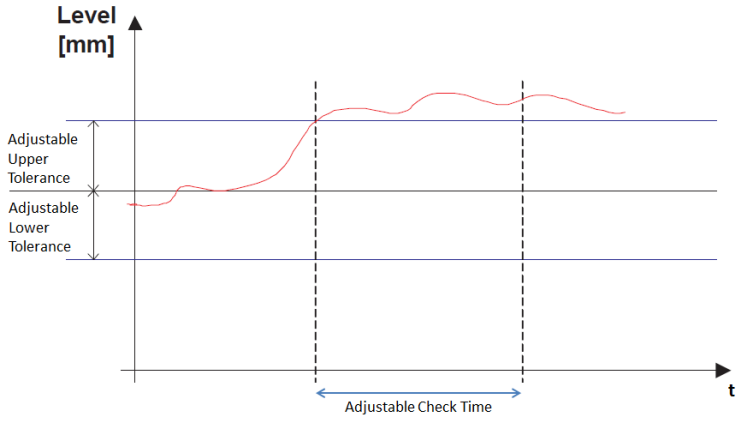
Kontrolcü içinde sapma ölçümü ve mod değişimi tespiti yapılır. Kontrolcü fonksiyonları üç temel mod arasında değerlendirilir. Bunlar; statik, dinamik ve geçiş modudur.

Hızın sıfır ve civarında olduğu durumlar statik mod olarak değerlendirilir. Araç statik modda iken vertikal değişimlerin tek sebebi yükleme olabilir.

Dinamik mod, araç hızının sıfırdan büyük olduğu durumlarda değerlendirilir. Bu durumlarda alınan yükseklik değişimlerinin sebebi yoldan gelen bozucular, yanal ve düzlemsel ivmeler olabilir.

Süspansiyonun genel davranışı ve araçta var olan ivme ve salınımlardan ötürü, hız değişimiyle doğrudan kontrolcü modunu değiştirmek doğru bir yaklaşım olmaz. Dolayısıyla bir geçiş modu koyarak belli bir süre gecikme oluşturmak ve bu gecikme süresince hızı gözlemleyerek modlar arası geçişi sağlamak yönünde bir tasarım yapılmıştır.

Tez konusu olan EHSKS kontrolcüsü araç yüksekliğini belirli bir yükseklikte sabit tutmayı hedefler. Bu yükseklik süspansiyon davranışının doğası gereği belirli bir toleransa sahiptir. Tez çalışmasında bu tolerans mevcut uygulamalar oranla daha dar tutulmuş ve ±12mm gibi tolerans aralığı seçilmiştir.

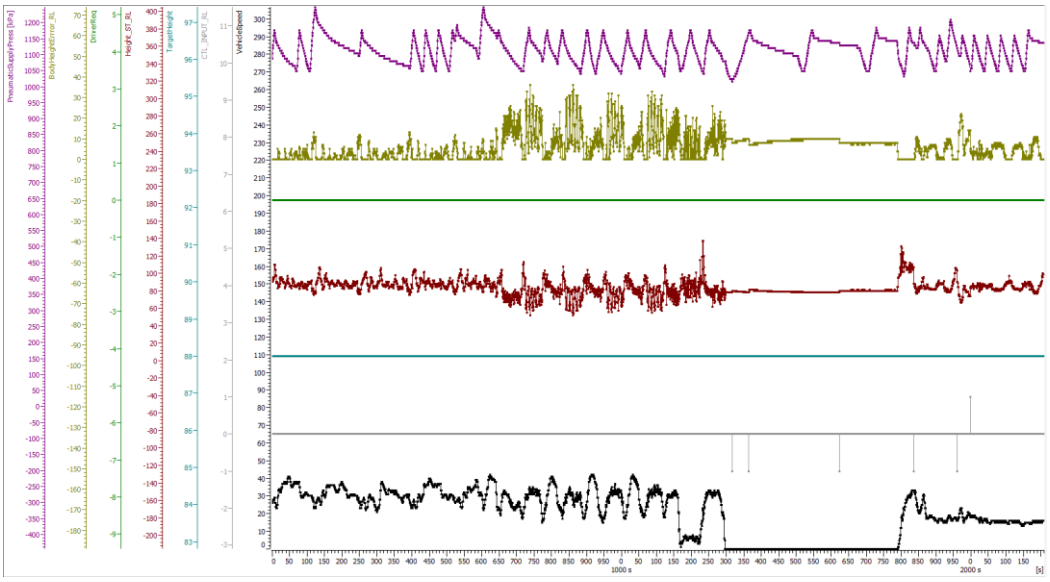


Şekil 4.37 EHSKS yazılımı, kontrolcü tolerans değeri ve tolerans dışı süre hesaplama grafiği

Sistemin hedeflenen yükseklikten tolerans dışına çıkışı sürekli olarak takip edilir. Sistem cevabı ve süspansiyon davranışı sebebiyle her tolerans dışı değerde kontrolcü çalıştırılmaz. Belirli bir süre boyunda bu değerin dışında olup olmadığı kontrol edilir. Bunu bir zamanlayıcı kullanmak gibi düşünürsek, tolerans aralığına girdiği er noktada zamanlayıcı sıfırlanıp baştan saymaya başlar. Eğer sistem belirli bir süre boyunca bu tolerans değerinin dışında kalır ise kontrolcü çalıştırılır ve hedef yükseklik toleransının sağlaması sağlanır. Bu zamanlayıcı sabitleri kontrolcü sisteminin moduna göre değişiklik gösterir, örneğin dinamik modda her bozucuyu dikkate almamak için zamanlayıcı daha uzun süre sayarken, statik modda daha kısa sayar diyebiliriz.

Kontrolcü testi için araç, dayanım testi pistince koşturulmuş, öncelikli olarak statik ardından dinamik olarak kontrolcü testleri tamamlanmıştır. Bu testlerde kontrolcü grafiği Şekil 4.39' deki gibi gözlemlenmiştir. Bu grafikte kontrolcü tepkilerini gri ile, araç yüksekliğini kırmızı ile hedef yüksekliği mavi ile körük basınçlarını mor ile takip edilebilir. Her bir sinyalin y ekseni kendi aralığını ifade etmektedir.

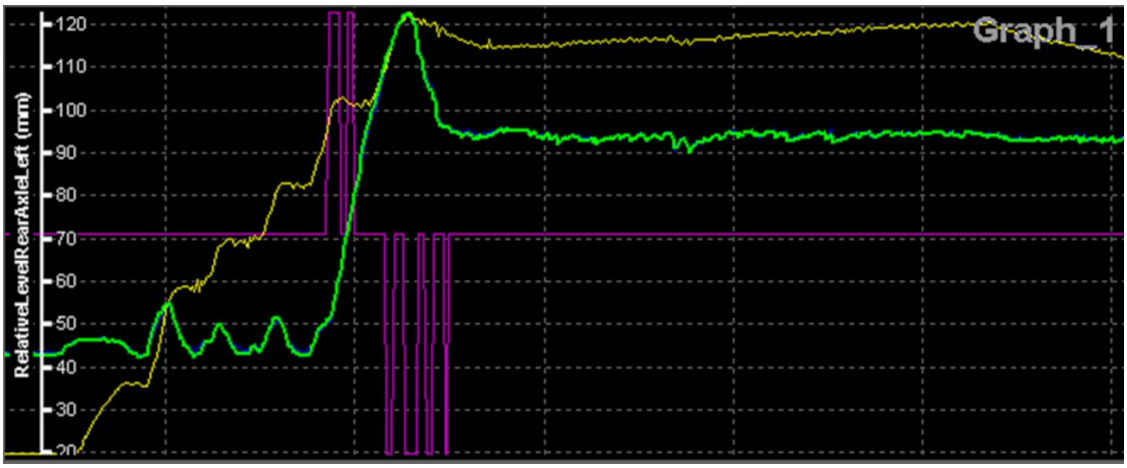
Grafikte de görülebileceği gibi, yükseklik değerleri tolerans bandında gezindiğinde, ya da kısa süreli olarak çıktığından kontrolcü çalıştırılmaz, kalıcı süre ile tolerans dışında olduğunda kontrolcü aktif olur ve hedef yüksekliğe getirmek için valf kontrolü sağlar. Aracın statik duruma geçişi ile kontrolcünün düzeltme yaptığını da grafikten kolayca okuyabiliyoruz. Bu da statik durumda sayıcı sabitinin daha kısa çalışmasından kaynaklı olarak ifade edilebilir.



Şekil 4.38 EHSKS yazılımı, araç dayanım testi sırasında kontrolcü davranışı grafiği

Dinamik testlerin bir örneği, araç belirli bir hızın üstüne çıktığında elektronik havalı süspansiyon kontrol sistemi yüksekliğinin sürüş yüksekliği seviyesine çekilme gereksinimidir. Bu gereksinim ile araç sürüş yüksekliği dışında bir yükseklikte uzun süre koşmamış olur, böylece kinematik olarak sistem koruması yapmanın yanında kullanıcı konforu ve kargo güvenliği açısından da büyük avantaj sağlanmış olunur.

Şekil 4.39'de görülen kontrolcü grafiği de hızın (sarı ile ifade edilen sinyal) belirli bir sınıra ulaştıktan sonra kontrolcünün (mor ile ifade edilen sinyal) çalışmasını göstermektedir.



Şekil 4.39 EHSKS yazılımı, dinamik sürüş yüksekliği testi örnek grafiği

5
Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılım Geliştirme Testleri ve Validasyonu

Yazılım testi ile ilgili yaygın bir anlayış, testin sadece test koşumu işleminden ibaret olduğu yönündedir. Test koşumu, test etme işleminin bir parçasıdır fakat tümü değildir.

Test sürecinde, test koşumu öncesi ve sonrasında gerçekleştirilen başka işlemler de vardır. Bu işlemler arasında planlama ve kontrol, test koşullarını seçme, test senaryolarını tasarlama ve koşturma, sonuçları kontrol etme, çıkış kriterini değerlendirme, test süreci ve test edilen sistem ile ilgili raporlama ve test fazı tamamlandıktan sonra kapanış işlemlerini sonlandırma ve tamamlama yer alır. Test ayrıca, dokümanları gözden geçirme (kaynak kod dahil) ve statik analiz işlemlerini de içerir.

Dinamik test ve statik test benzer hedeflere ulaşmak için kullanılabilir ve test edilen sistem ile yazılım geliştirme ve test süreçlerini iyileştirmek için kullanılabilecek bilgiler sağlar. Testin hedefleri aşağıdaki gibi olabilir:

- Hataları bulma
- Kalite seviyesi hakkında güven kazanma
- Karar verme için bilgi sağlama
- Hataları önleme

Yazılım testinin temel olarak yedi prensibi vardır;

- Testin amacı, yazılımda hataların olduğunu göstermektir; yazılımda hata kalmadığını ispatlamak değildir
- Yazılımı %100 test etmek imkansızdır
- Teste yazılım geliştirme sürecinin başında başlamak gerekir
- Hatalar yazılımın belli alanlarında yoğunlaşır
- Test senaryolarının belirli aralıklarla güncellenmesi ve revize edilmesi gerekir

- Test yaklaşımı ve aktiviteleri yazılım projesinin koşullarına göre değişiklik gösterir
- Testte tespit edilen hataların düzeltilmiş olması, yazılımın müşteri/kullanıcı ihtiyaçlarını tam, eksiksiz ve doğru karşılıyor olduğu anlamını taşımaz.

5.1. Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Statik Kod Analizi

Yazılımın yürütülmesini gerektiren dinamik testin aksine statik test teknikleri, kod yürütülmeden kodun veya diğer proje dokümanlarının manuel olarak incelenmesine (gözden geçirme) ve otomatik şekilde analiz edilmesine (statik analiz) dayanır.

Statik analizin hedefi, yazılımın kaynak kodundaki ve yazılım modellerindeki hataları bulmaktır. Statik analiz, incelenen yazılım yürütülmeden gerçekleştirilir; dinamik test ise yazılım kodunu yürütür. Statik analiz, dinamik testte bulunması zor olan hataları bulabilir. Statik analiz araçları, program kodunu (örn. kontrol akışı ve veri akışı) ve oluşturulan çıktıyı analiz eder [17].

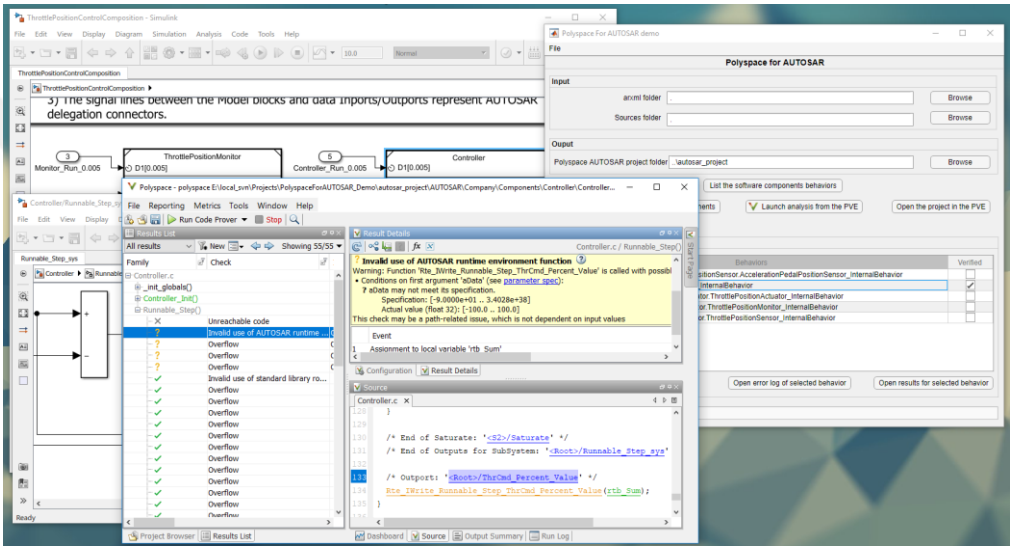
Tez kapsamında yazılan EHSKS yazılımının statik kod analizi PolySpace programı ile tamamlanmıştır. Bütün testler gibi statik kod analizi testi de bir sonuç değil geliştirme aşamasının temel araçlarından biridir.

Statik analizin avantajları şunlardır:

- Testler yürütülmeden önce hataların erken tespiti
- Yüksek karmaşıklık ölçüsü gibi metriklerin hesaplanmasıyla koddaki veya tasarımdaki şüpheli durumlarla ilgili erken uyarı
- Dinamik test ile kolayca bulunamayan hataların belirlenmesi
- Yazılım modellerindeki bağımlılıkların ve tutarsızlıkların saptanması, linkler gibi
- İyileştirilmiş kod ve tasarım sürdürülebilirliği
- Uyarıların geliştirme sırasında dikkate alınması durumunda hataların önlenmesi

Statik analiz araçları tarafından bulunan genel hatalar şöyledir:

- Değer atanmamış değişkenin yanlışlıkla referans gösterildiği durumlar
- Modüller ve bileşenler arasında tutarsız ara yüzler
- Kullanılmayan veya yanlış şekilde tanımlanmış değişkenler
- Ulaşılamayan, çağrılmayan (ölü) kod
- Eksik ve hatalı mantık (sonsuz döngüler)
- Çok karmaşık yapılar
- Programlama standardı ihlalleri
- Güvenlik açıkları
- Kod ve yazılım modellerinde söz dizimi ihlalleri



Şekil 5.1 EHSKS yazılımı; Statik kod analizi için kullanılan PolySpace programı ara yüzü

5.2 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Test Seviyeleri

Yazılım testlerinin; alfa testi, beta testi, bileşen testi, sürücü, saha testi, fonksiyonel gereksinim, entegrasyon, entegrasyon testi, fonksiyonel olmayan gereksinim, sağlamlık testi, taklit uygulama, sistem testi, test ortamı, test seviyesi, test güdümlü geliştirme, kullanıcı kabul testi gibi test edilmesi planlanan işe yönelik birçok alt dalı mevcuttur [18].

V modeli tabanlı geliştirme testlerinde temel olarak testler 4 şekilde sınıflandırılabilir.

5.2.1 Bileşen (Birim) Testi

Bileşen testi (birim, modül veya program testi olarak da bilinir), ayrı ayrı test edilebilen yazılım modüllerindeki, programlardaki, nesnelerdeki, sınıflardaki vb. hataları aramak ve bunların işleyişini doğrulamak amacıyla kullanılmıştır. Yazılım geliştirme yaşam döngüsü ve sistemin bağlamına göre sistemin geri kalanından bağımsız şekilde gerçekleştirilmiştir. Böylece her bir alt fonksiyon ilk algoritma oluşturulmasının ardında bağımsız olarak birim testine tabii tutulmuştur ve tekil olarak fonksiyonel bir hata olup olmadığı kontrol edilmiştir.

5.2.2 Entegrasyon Testi

Entegrasyon testi, bileşenler arasındaki ara yüzleri, yazılımın işletim sistemi, dosya sistemi ve donanım gibi sistemin farklı bölümleriyle etkileşimlerini ve sistemler arasındaki ara yüzleri test edilmesi için kullanılmıştır. Birbiri ile etkileşimli çalışan birkaç fonksiyon sırası ile bir araya getirilerek karşılıklı isterleri sağlayıp sağlamadıkları, birbiri ile uyumlu olarak çalışıp çalışmadıkları test edilmiştir. Entegrasyon testleri geliştirdikçe eklenen fonksiyon ve alt fonksiyonlar ile sistem testine kadar ilerlemiştir.

5.2.3 Sistem Testi

Sistem testi, tüm sistemin/yazılımın davranışı ile ilgilidir. Sistem testi, sistemin fonksiyonel ve fonksiyonel olmayan gereksinimlerini ve veri kalitesini sorgulamak amacıyla yapılmıştır. Bu seviyede sistemin tamamlanmamış gereksinimleri ile de ilgilenilmesi gerekmiştir. Fonksiyonel gereksinimlerle ilgili sistem testi, test edilecek sistem için en uygun gereksinim bazlı (kara kutu) teknikler kullanılarak başlatılmıştır. Örneğin, iş kurallarında tanımlanan etkilerin kombinasyonları için karar tablosu test tekniği kullanılabilir. Ardından, yapısal testlere (beyaz kutu) geçilerek gereksinim bazlı testlerin yakalayamadığı hatalar yakalanabilmiştir.

5.2.4 Kabul Testi

Kabul testinin amacı, sisteme, sistemin parçalarına veya sistemin fonksiyonel olmayan gereksinimlerine karşı güven oluşturmaktır. Kabul testinde ana odak hataları bulmak değildir, sistemin devreye alınmaya hazır olduğunu göstermektir.

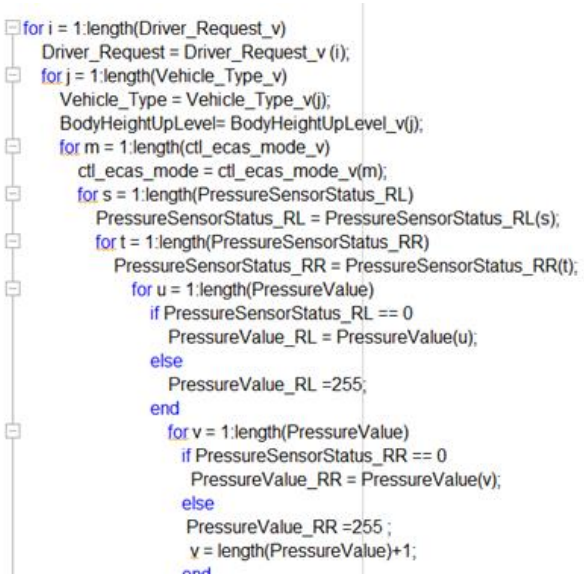
5.3 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Döngü İçinde Model Testi

Döngü içinde model testi (MIL- Model In the Loop) Matlab-Simulink ortamında yapılan, yazılmış yazılımın mevcut olabilecek bütün girişlerinin tanımlandığı ve çıkışlarının test edildiği test çeşididir.

Bu testler zamandan bağımsız lojik olarak gerçekleştirilebildiği gibi zamana ve döngüye bağlı olarak da gerçekleştirilebilir. Bu testlerin yapılabilmesi için lojik olarak olası girişler dizilere atanabilir veya zamana bağlı olarak sinyal üreteçler ile döngü içinde test edilebilir.

Yazılımın MIL testleri yapılırken herhangi detayı kaçırmamak büyük önem taşımaktadır. Oluşturulan yazılımın istenilen şekilde çalışıp çalışmadığını eksiksiz bir şekilde test edebilmek için koşulların doğru saptanması ve değerlendirilmesi gerekmiştir. Karşılaşılabilecek bütün sorunlara karşı fonksiyonların etkisi gözlemlenebilir olması için çalışılmıştır.

Sistemi eksiksiz test edip herhangi bir koşulu gözden kaçırmamak adına Matlab script kodları ile test yazılımı hazırlanmıştır. Bu yazılım oluşturulurken muhtemel her koşul ve girişin sisteme sağlanabilmesi için “for“ döngüleri kullanılmıştır. Yazılan bu kod baz alınarak test edilen fonksiyon için ayrıca bir Matlab Simulink test yazılımı oluşturulmuş ve girişler sabit değerler olarak atanmıştır. Bu şekilde sabit değerler değiştirilerek değişik kombinasyonlar ve durumları değerlendirmek mümkün hale gelmiştir. Her durum için fonksiyonun nasıl sonuç verdiği gözlemlenirken, fonksiyonel olarak karşılaşılabilecek sorunlar araçta test yapmadan önce tespit edilerek daha hızlı bir şekilde çözümlenmiş olmuştur.



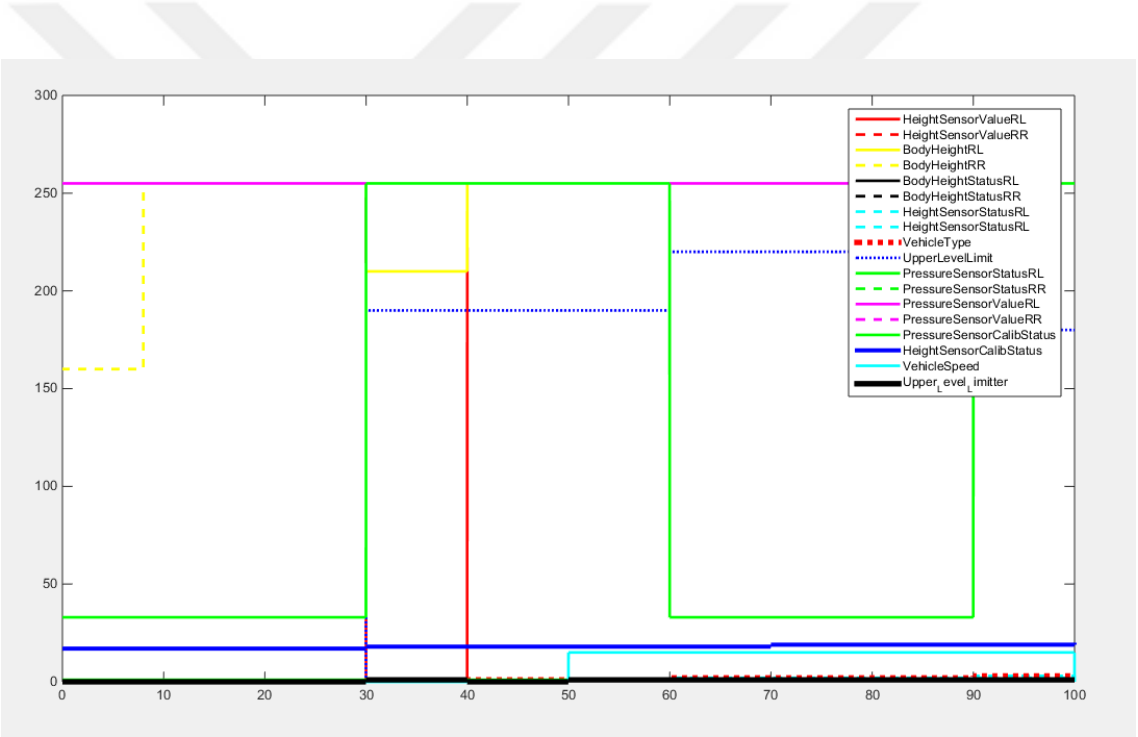
Şekil 5.2 EHSKS yazılımı; döngü içinde model testi için yazılan örnek bir test kodu

Bir örnek ile açıklamak gerekirse, Şekil 5.2'de verilen fonksiyon yazılımı test kodundan da görüldüğü üzere; test edilmesi gereken giriş değerleri birer dizi olarak tanımlanır, ardından iç içe yazılan for döngüleri ile olası bütün kombinasyonlar model üzerinde test edilmiştir. Bu kombinasyonların oluşturduğu her bir koşul değerlendirilerek, fonksiyonun vermesi gereken reaksiyonu verip vermediği kolayca gözlemlenebilmiştir. Ancak oluşturulan dizilerde önemli yerlerin tespit edilip ona göre değer ataması yapılması gerekmiştir. Bunu test edebilmek için ise "Matlab Coverage Test" isminde bir araç kullanılmıştır. Bu aracı kullanmaktaki asıl amaç, test değerlerinin doğru aralıklarla verilip verilmediğini gözlemlemek ve gerekli bütün kombinasyonların elde edilebildiğini test etmektir. Coverage test aracı, yüzde üzerinden bir sonuç çıkararak, sisteminin ne kadarının test edilebildiğini, hangi noktaların test edilemediğini, test yazılımının sistemi ne kadar kapsadığını göstermiştir. Bu test için, testin etkisi testi ifadesini kullanmak yanlış olmaz.

Sistemin test girişleri akış şemalarıyla belli zamanlarda değiştirilerek girişler arasında farklı kombinasyonlar elde edilmiştir. Bütün girişler ayarlandıktan sonra "Coverage Test" aracıyla sistem çalıştırıldığında ise Matlab üzerinden nerelerde eksikler olduğunu ve hangi ihtimalin ne kadar kullanıldığını gösteren bir rapor çıkarılmıştır. Bu rapor sayesinde girişlerin sıklıkları, saniye aralıkları değiştirilerek

olabilecek ideal test koşulları sağlanabilmiş ve testin etkisi %90'ların üzerine çıkarılarak tamamlanıştır.

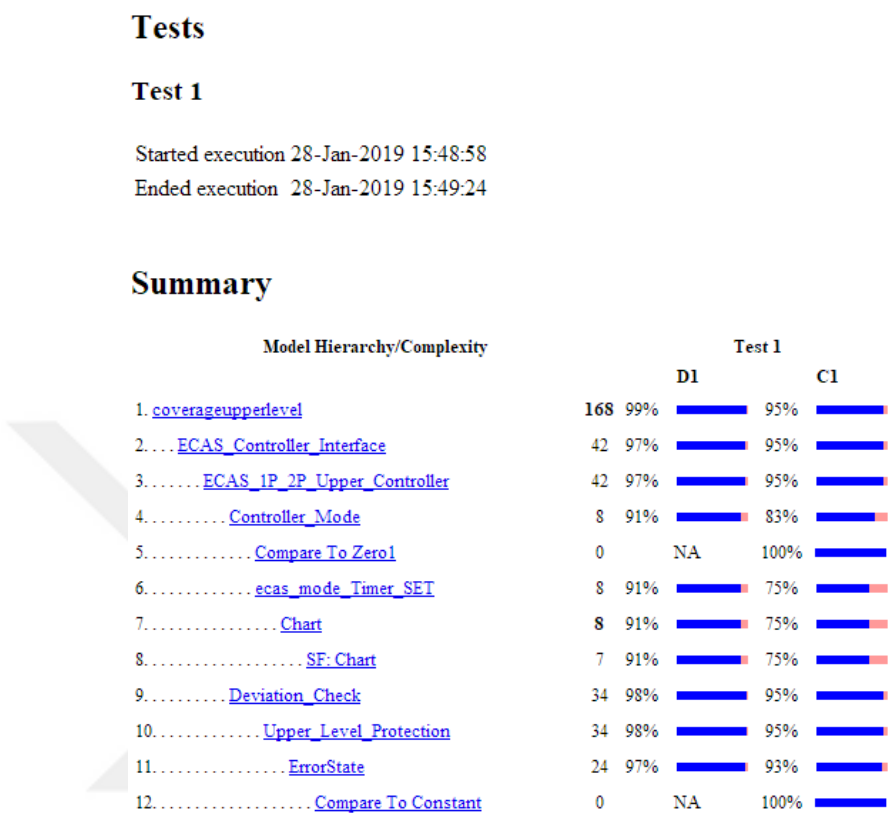
Sistem oluşturulurken verilen girişler ve alınan çıkışlar birer dizi olarak kaydedilirse bu verileri analiz etmek kolaylaşır. "Coverage test" için hazırlanan başka bir Matlab kodu ile bu dizileri grafik olarak çizdirerek alınan sonuçları zamana bağlı olarak test etmek ve gözlemlemek mümkün hale getirilmiştir. Şekil 5.3'te gösterilen grafik bunlara bir örnek olarak verilebilir. Parametrelerin değişimine göre fonksiyondan alınan cevap, buradaki gibi gözlemlenerek istenilen yanıtın alınıp alınmadığı hızlıca tespit edilebilmiştir. Testin zamana bağlı olarak verdiği döngü sonuçları gözlemlenmiş ve kontrol edilmiştir.



Şekil 5.3 EHSKS yazılımı; döngü içinde model testi için yazılan örnek bir test kodu

Şekil 5.4'de "Matlab Coverage Test" raporunun kısmı verilmiştir. Nerelerde eksik olduğunu ve verilen girişler ve sıklıkları ile sistemin ne kadarının test edilebildiğini gösteren bu rapor ile eksik test koşulları saptanarak o koşulları da sağlayacak yeni giriş kombinasyonları oluşturulabilmektedir. Bu şekilde, hem fonksiyonların çalışıp çalışılmadığı gözlemlenmiş, hem de gerekli bütün koşullar oluşturulabilir hale

gelmiştir. Sistemin fonksiyonel olarak model testleri bu şekilde tamamlanarak fiziksel testlere kalmadan fonksiyonel sorunlar erkenden çözümlenerek süreç oldukça hızlandırılmıştır.



Tests

Test 1

Started execution 28-Jan-2019 15:48:58
Ended execution 28-Jan-2019 15:49:24

Summary

Model Hierarchy/Complexity	Test 1				
		D1		C1	
1. coverageupperlevel	168	99%		95%	
2. . . . ECAS_Controller_Interface	42	97%		95%	
3. . . . . ECAS_1P_2P_Upper_Controller	42	97%		95%	
4. . . . . Controller_Mode	8	91%		83%	
5. . . . . Compare To Zero1	0		NA	100%	
6. . . . . ecas_mode_Timer_SET	8	91%		75%	
7. . . . . Chart	8	91%		75%	
8. . . . . SF: Chart	7	91%		75%	
9. . . . . Deviation_Check	34	98%		95%	
10. . . . . Upper_Level_Protection	34	98%		95%	
11. . . . . ErrorState	24	97%		93%	
12. . . . . Compare To Constant	0		NA	100%	

Şekil 5.4 EHSKS yazılımı; döngü içinde model testi için kullanılan "Matlab Coverage Test" raporu

5.4 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Döngü İçinde Donanım Testi

Döngü içinde yazılım testi (HIL- Hardware In the Loop), kompleks gerçek zamanlı gömülü sistemlerin tasarım ve test aşamasında gittikçe artan bir şekilde kullanılan bir benzetim tekniğidir [17] [18]. Bu tekniğin amacı; gömülü sistemlerinin etkili ve kolay bir şekilde tasarlanabilmesini sağlayacak bir platform oluşturmaktır. Bu tasarım ve test aşaması sırasında sistemin bütün dinamikliğini yansıtacak matematiksel bir model söz konusudur. Bu matematiksel modelin oluşturulması ise klasik sistem benzetim algoritmaları ile gerçekleştirilmektedir.

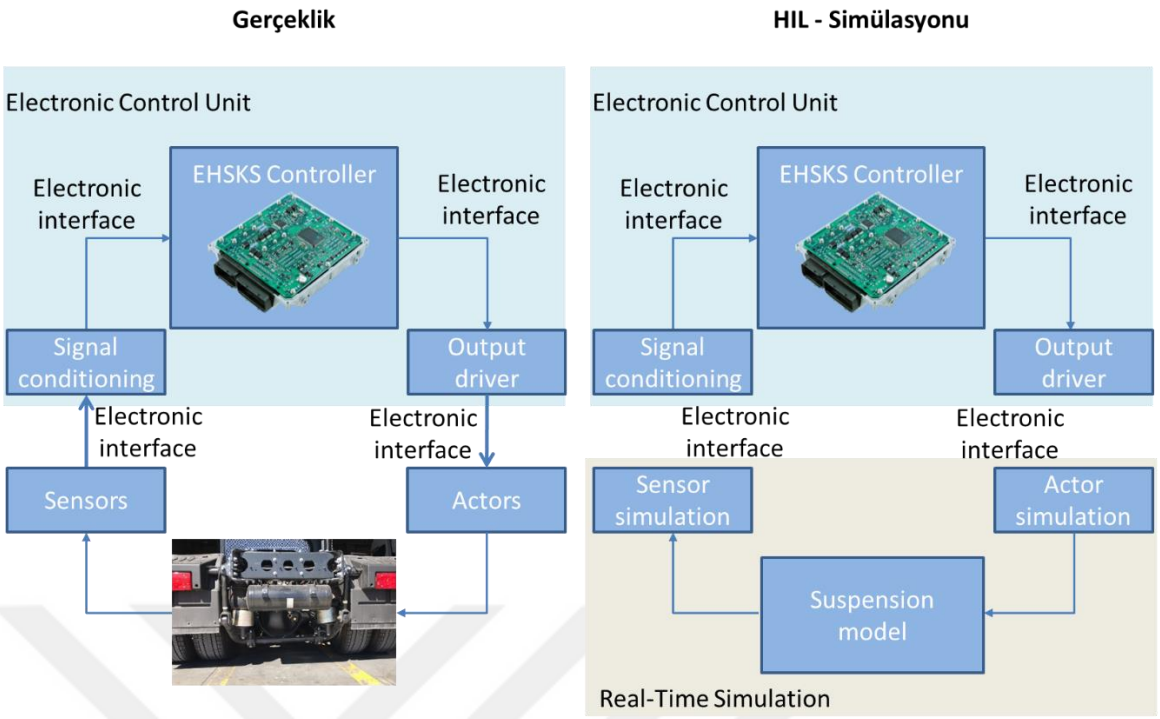
Döngü içinde Donanım (HIL) simülasyonu, günümüzde birçok ECU geliştirme projesinde uygulanan köklü bir doğrulama teknolojisi haline gelmiştir. HIL teknolojisi sayesinde, yeni yazılım ve / veya elektronik bileşenlerin vadesini artırmak için fonksiyon testleri önceki geliştirme aşamalarına kaydırılabilir. Daha önce doğrudan araçta veya geleneksel test tezgahlarında /platformlarında gerçekleştirilen maliyet ve zaman açısından pahalı test sürüş çevrimleri, simülasyon temelli işlemlerle ikame edilebilir.

Arıza durumları veya tehlikeli manevraların testleri, en azından bütün test programının bir kısmında, bilgisayara kaydırılabilir. HIL simülasyonlarının en büyük avantajı, bu test platformunun/tezgahının otomatik hale getirme yeteneğidir. Bu, tüm test döngülerinin çoğaltılmasına ve bu test tezgahlarının günde 24 saat çalışmasına izin verir.

Örneğin otomobil fren sistemlerinin test edilmesi için bir HIL platformu oluşturulursa böyle bir benzetilmiş sistemde aşağıdaki alt-sistem bileşenlerinin her biri için matematiksel bir model oluşturulabildiği gibi sistem içinde bulunan sistem modelleri de uygulamaya uyarlanarak kullanılabilmektedir.

- Süspansiyon, tekerlekler, lastikler vb. gibi araç dinamiği,
- Yol karakteristikleri,
- Fren sisteminin hidrolik bileşenlerinin dinamiği gibi.

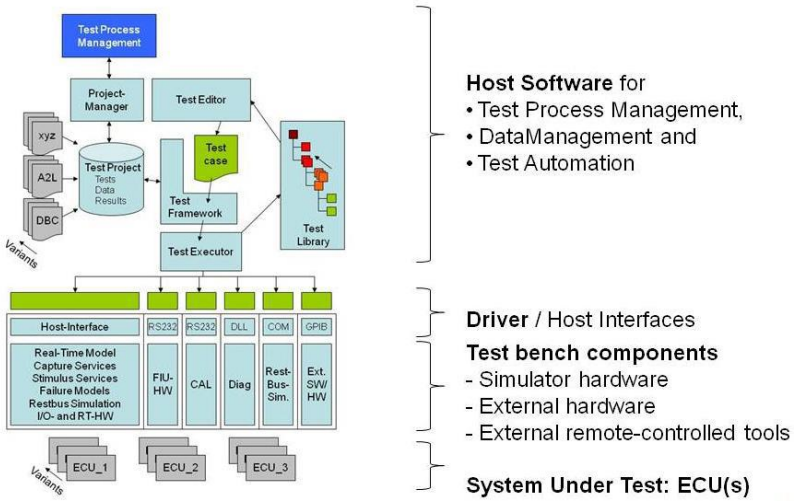
HIL sistemi fiziksel olarak araca gitmeden/bağlanmadan, testlerin bir araç modeli üzerinde koşturulması ile donanım ve yazılım testlerinin yapılması işlemidir. Bu metot ile Elektronik Havalı Süspansiyon Kontrol Sisteminin araç testlerinden önce farklı yol, araç, sürüş ve sürücü özellikleri senaryolaştırılarak sisteme özgü hazırlanmış simülator üzerinde yapılması sağlanır. Bu da geliştirme ve test aşamasında maliyet süre ve güvenlik açısından ciddi derecede avantaj sağlamaktadır.



Şekil 5.5 HIL(Hardware In the Loop) testlerinin temel prensibi

Şekil 5.5'in sol kısmında gösterildiği gibi günümüz otomotiv elektronik sisteminin kapalı bir kontrol döngüsü (Kontrolör, çıkış sürücüsü, aktörler, tesis, örneğin bir motor, sensörler ve giriş tarafı sinyal koşullandırma) parçalar halinde değiştirilmiştir. Elektriksel ara yüzler korunur. Sensörler ve aktörler, tam simüle versiyonlarla değiştirilir veya test tezgahı kurulumunda orijinal fiziksel yük bileşeni olarak bile bağlanırlar.

Kontrol döngüsünün çevresel kısım, başka bir değişle ise bu örnekte görünen motor, gerçek zamanlı olarak uygun model kesinliğinde hesaplanabilen bir simülasyon modeliyle tamamen değiştirilebilmektedir.



Şekil 5.6 HIL(Hardware In the Loop) test platformlarının mimarisi

Tez konusu olan EHSKS yazılımı geliştirme çalışmasında da kullanıldığı üzere, tipik bir HIL test platformu temel olarak şunlardan oluşur.

Ana bilgisayarda yürütülen yazılım (Host software); test platformuna bağlanabilen farklı bileşenlerin etkileşimli olarak çalıştırılmasını sağlar.

Bu anlamda operasyon, konfigürasyon, durum kontrolü, verilere erişim gibi işlemlerin yapılması anlamına gelir. Bu işlemleri katılımsız olarak da yapabilmek için, yani otomatikleştirilmiş bir şekilde uygun bir test otomasyon sistemi uygulanır. Bu test otomasyon sistemleri, test yürütücüleri, test senaryolarının parametrelerini yönetmek için test çerçeveleri, spesifik grafiksel veya tablo bazlı test editörleri, kapsamlı test kütüphaneleri gibi bileşenlerden oluşur. Böylece HIL'in ana bilgisayarda yürütülen yazılım araçlarını diğer mühendislik veri işleme araçlarına entegre edilebilmektedir. Örneğin; veri saklama, test yönetimi, gereksinim yönetimi veya hata takip sistemleri için.

Sürücü; ana bilgisayarda yürütülen yazılım, test platformuna bağlı olan bileşenlere belirli sürücüler aracılığıyla erişir. Bu sürücüler, RS232, DLL'ler, (D) COM Ara yüzleri, TCP / IP ve hatta GPIB gibi çok heterojen teknolojiler kullanır.

Test platformu bileşenleri; gerçek zamanlı simülatörün kendisi bu bileşenlerin temelidir. Burada simülasyon adımları, giriş çıkış kartlarının tepkilerine veya iletişim ara yüzüne (CAN, LIN, Flexray vb.) göre istenen zaman hassasiyetinde

hesaplanabilir. Gerçek zamanlı çalışma elektriksel ara yüz ile ECU' ya bağlanır. Eksik olan bileşenler simüle edilerek (Soft ECU) kullanılabilir.

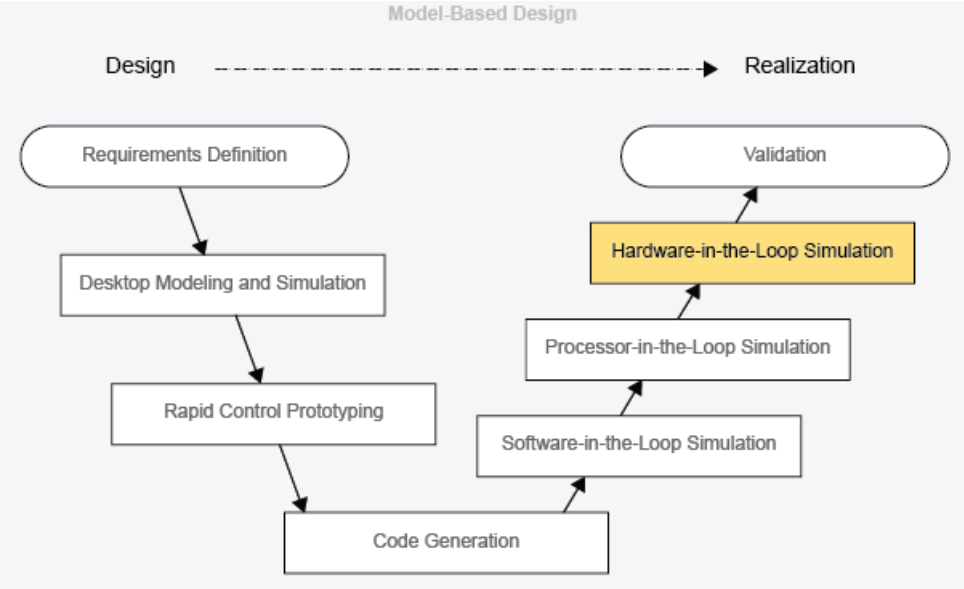
Diğer bir önemli test platformu bileşeni de Arıza Enjeksiyon Ünitesi (FIU - Failure Injection Unit) veya Elektriksel Hata Simülasyonu (EES - Electrical Error Simulation) sistemidir. Test durumları yalnızca Test Edilen Sistemin (SUT - System Under Tests) davranışının tamamen işlevsel bir ortamda kontrol edilmesini içermeyebilir. Giriş ve çıkış pinlerinde elektrik hataları olması durumunda test edilen sitemin kontrol etmek de önemlidir.

ECU iç değişkenlerine erişebilmek için dış kalibrasyon veya teşhis sistemlerinin test otomasyon ortamına entegre edilmesi gerekir.

Test Edilen Sistemler (SUT):ECU'lar; HIL test platformlarının en son ve en önemli kısmı elbette testin kendisi, yani ECU veya bir ECU ağıdır. Günümüzde bir yandan HIL teknolojisi, tek ECU'ların bileşen testi için kullanılmasının yanında, çoğu zaman tam araç ECU ağlarının entegrasyon testi için de uygulanmaktadır. Bütün ECU ağını entegre ederek komple araç testine kadar süreç genişletilebilir. ECU'lar farklı varyantlarda çok sık geliştiği için, HIL test platformlarının teçhizatı bu varyantlara uyarlanmak için düzenlenir.

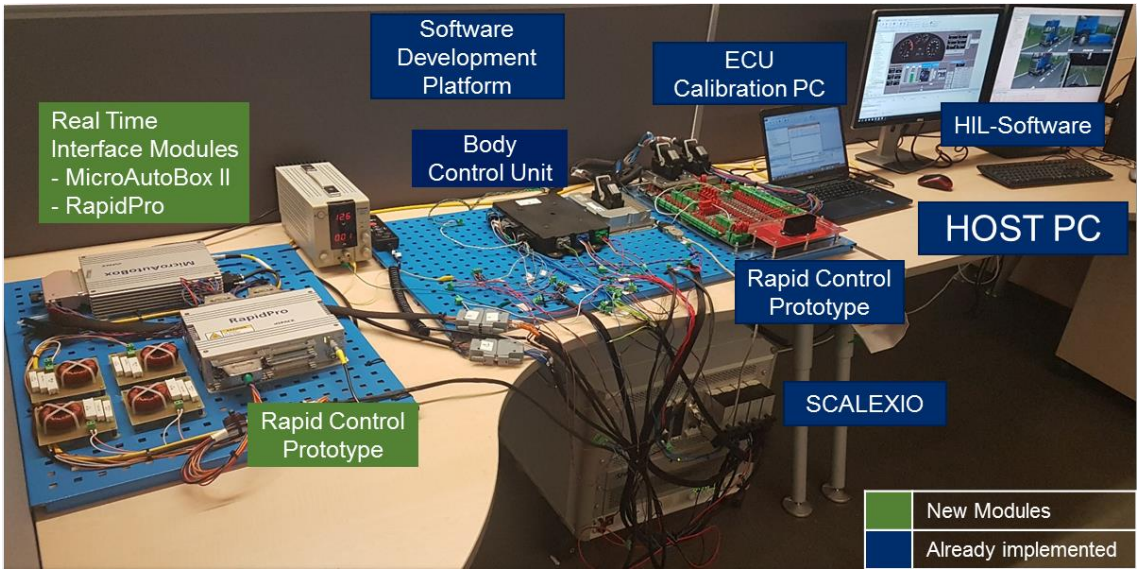
HIL simülasyonu bir tür gerçek zamanlı simülasyondur. Denetleyici ve yazılım tasarımlarını, kısaca karmaşık gerçek zamanlı gömülü sistemleri geliştirmek ve test etmek için HIL simülasyonunu kullanılır. HIL simülasyonu, kontrol cihazınızın gerçek zamanlı olarak gerçekçi sanal uyaranlara nasıl tepki verdiğini gösterir. Fiziksel sistem modelinin geçerli olup olmadığını belirlemek için HIL kullanılabilmektedir. Fiziksel modeller genişletilerek test edilecek sistemler simülasyona dahil olarak etkileşime dahil edilebilir.

HIL testleri V-döngüsünün önemli bir doğrulama ve sağlama (verification & validation) adımıdır.



Şekil 5.7 HIL testlerinin V-döngüsündeki yeri

HIL testleri, araç kullanmadan, riskli oluşturulması zor senaryoları bilgisayar ortamında modelleyerek hızlıca test edilmesini sağlar. Testleri model ortamında sağlamasının yanında, buna uygun fiziksel çıkışları da fiziksel olarak sağladığından birden fazla ECU bağlanarak da testler yapılabilir. Soft ECU adı verilen dahili ECU ile testler yapılabildiği gibi Real ECU seçeneği ile doğrudan araçta kullanılan ECU'nun test sistemine bağlanması ile ECU ile birlikte araç senaryoları fiziksel giriş ve çıkışlarla birlikte gerçeklenmiş olur. Şekil 5.8'de de EHSKS için kullanılan HIL geliştirme alanı görülmektedir. Bu alanda da kullanıldığı gibi, sistem girişleri ve çıkışları bağlanarak fiziksel testler daha anlaşılır hale getirilebilir. Tez kapsamında EHSKS yazılımı ve BCM ünitesine, HIL üzerinden EHSKS kumandası bağlanmış ve kullanıcı girişleri bu şekilde simüle edilmiştir. Aynı zamanda çıkış portlarına da HIL modülü ile birlikte EHSKS valfleri bağlanarak valf kontrol sinyalleri ile valfin açma kapama sesi gibi fiziksel çıkışlar gözlenebilmiştir.



Şekil 5.8 EHSKS yazılımı HIL testlerinin gerçekleştirilme alanı

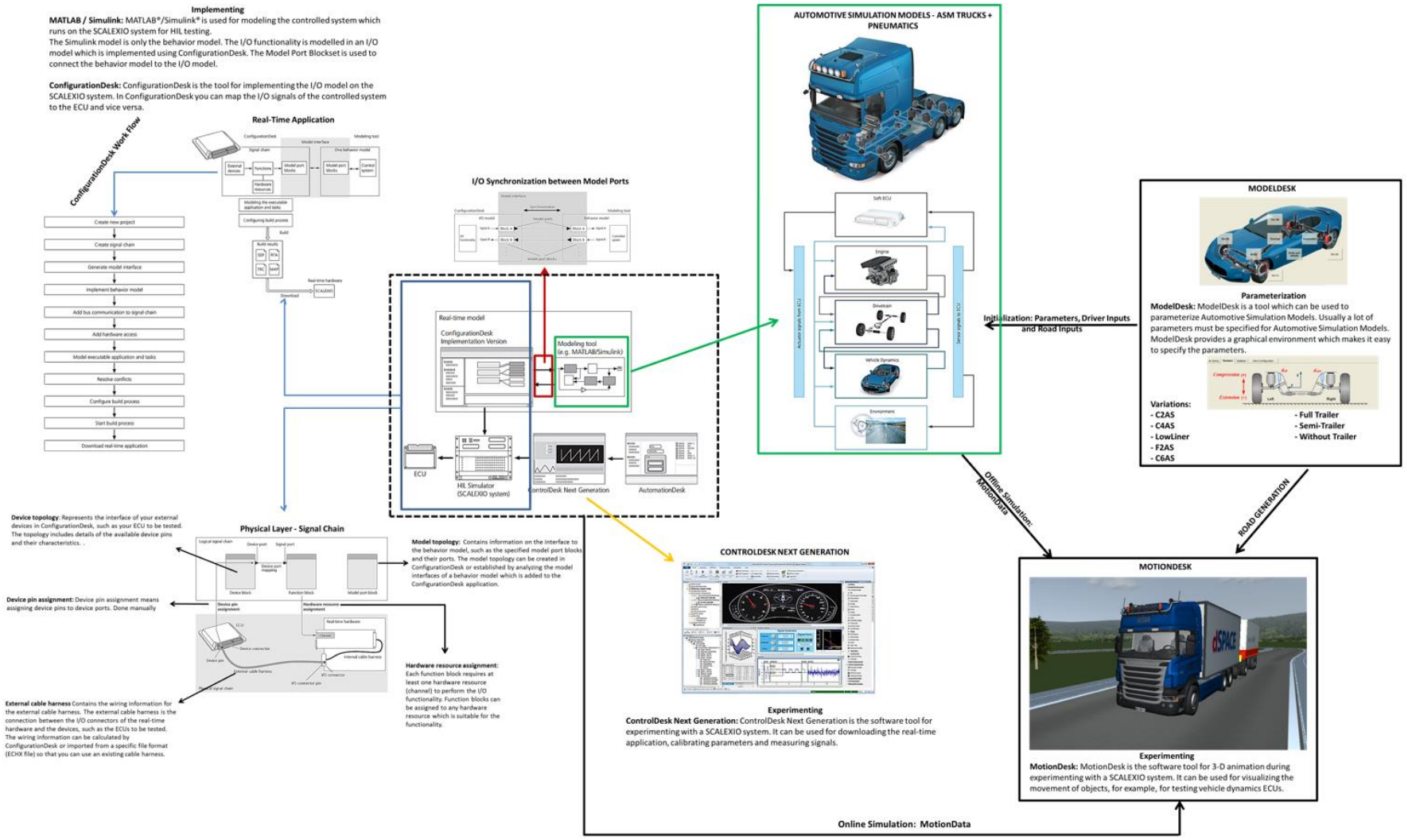
EHSKS yazılımı HIL testleri sırasında kullanılan D-Space Scalexio ürünü ve kullanılan araçları şekil 5.9'daki gibi kısaca ifade edilmiştir.

-Automotive Simulation Models (ASM) ortamı, sistem modellerinin bulunduğu platformdur. Bu platform ECU modelinden araç modeline kadar birçok çevre modelini içinde barındırır. Model değişiklikleri bu alanda yapılır.

-Model Desk ortamı, kullanılan modelin kullanıcıya açık ve değiştirilebilir parametrelerinin bulunduğu alandır. Örneğin yeni bir süspansiyon modelini oluşturup implemente etmek yerine var olan süspansiyon modelinin parametrelerini güncelleyerek araç süspansiyon sistemine en yakın süspansiyon modelini oluşturma yolu, bu platform ile uygulanabilmektedir. Ayrıca sistemde değişen her bir parametre için yeni model oluşturmak yerine bu ortamdan parametre güncelleme metodu da oldukça hızlı bir çözümdür.

- Control Desk ortamı, araç içindeki tanımlanmış bütün parametreleri değiştirme, gözlemleme ve kaydetme imkânı sağlayan platformdur.
- Motion Desk ortamı, yazılımın ve modelin simülasyonunun hareket olarak görsel açıdan sağlandığı platformdur. Deneyimlemeyi sağlayan adımdır denilebilir.

Bu platform ile koşulan test görsel olarak takip edilebilir. Giriş ve Çıkışlar gözlenebilir



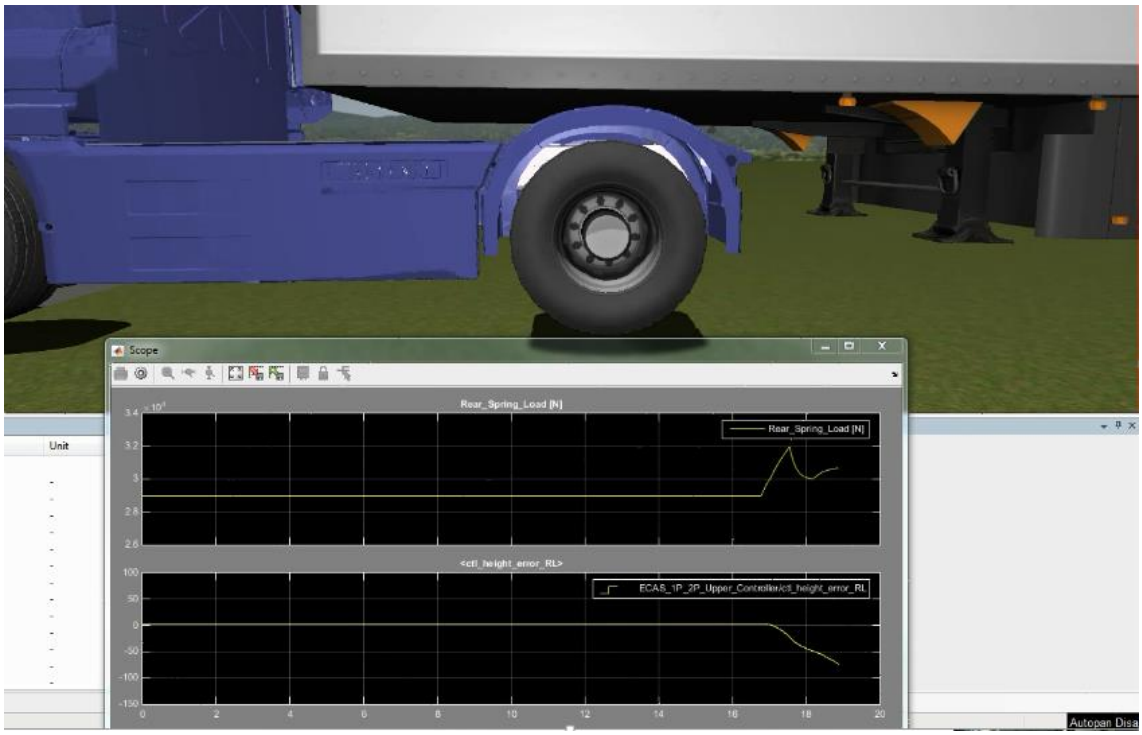
Şekil 5.9 EHSKS yazılımı HIL testleri için kullanılan D-Space Scalexio platformları ve kullanım yapısı

EHSKS yazılımın HIL platformunda testi sırasında bütün bu modüller kullanılmıştır. Kullanılan bu modüller sayesinde sistemin MotionDesk alanında sistemin tepkileri gözlemlenmiştir.

EHSKS yazılımının HIL testleri sonuçları ile araç testi sonuçlarının birbirini sağlaması temel amaç olarak alınmış ve buna göre model ve yazılım üzerinde gerekli güncellemeler sürekli olarak yapılmıştır.

HIL platformuna mevcut EHSKS yazılımı implemente edilmiş ECU bağlanıp, sistem girişleri verilmiştir.

İlk olarak süspansiyon yükselme testini inceleyecek olursak;

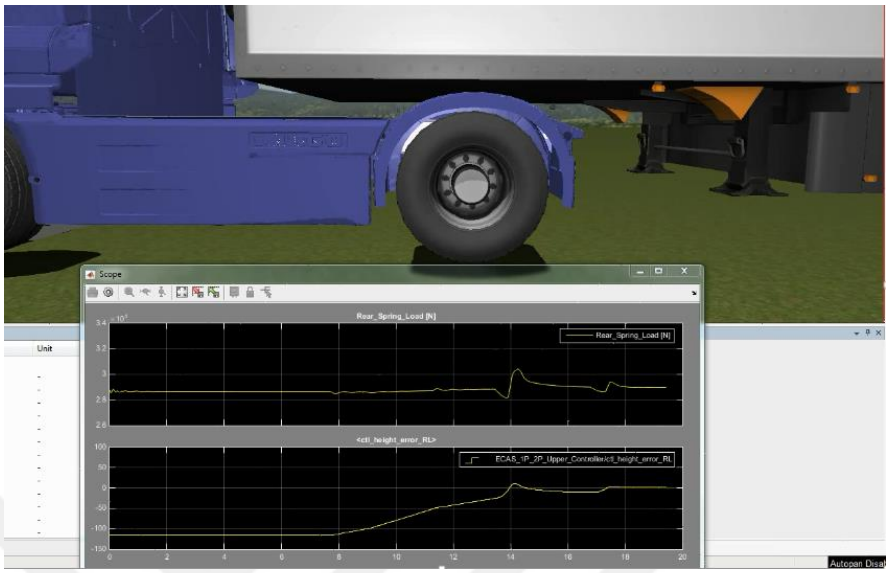


Şekil 5.10 HIL testi sırasında yükselme girişi verilen sistemin ilk durum değişimi

Şekil 5.10'da da görüldüğü gibi sisteme 17sn itibariyle sisteme yükseltme talebi girişi yapılmıştır. Üst grafikte görülen körük yükleri artmaktadır. Alt grafikte görülen hedeften sapma (error) değerinin sıfır noktasından uzaklaşarak gösterdiği değişimi de gözlenebilmektedir.

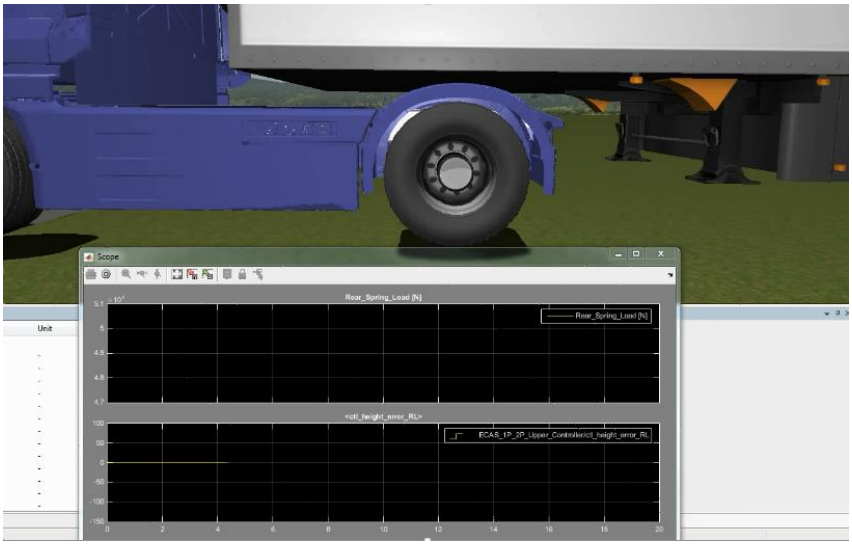
Bu yükseltme testinin ardından aracın tanımlı sürüş yüksekliğine getirilmesi testi de aynı şekilde tamamlanmıştır.

Şekil 5.11'de de gözlendiği gibi 8. Saniye itibariyle sisteme sürüş yüksekliği talebi girişi yapılmıştır. Buna bağlı olarak yükseklikteki hata değerinin azalarak sıfır noktasına oturduğunu gözlemlenebilmektedir.



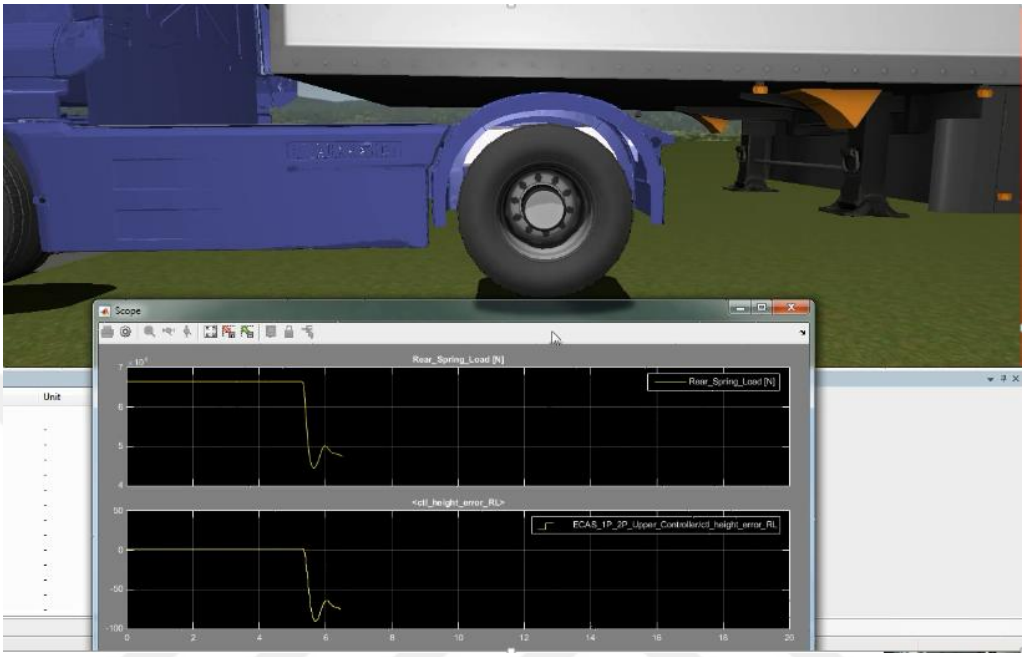
Şekil 5.11 HIL testi sırasında sürüş yüksekliği girişi verilen sistemin durum değişimi

HIL platform araç üstünde birçok değeri test esnasında değiştirmeye imkan sağlamaktadır. Örnek olarak Şekil 5.12'de de gösterildiği gibi sistem üzerinde tanımlanmış olan 10ton yük ani şekilde sıfır değerine çekilmiştir.



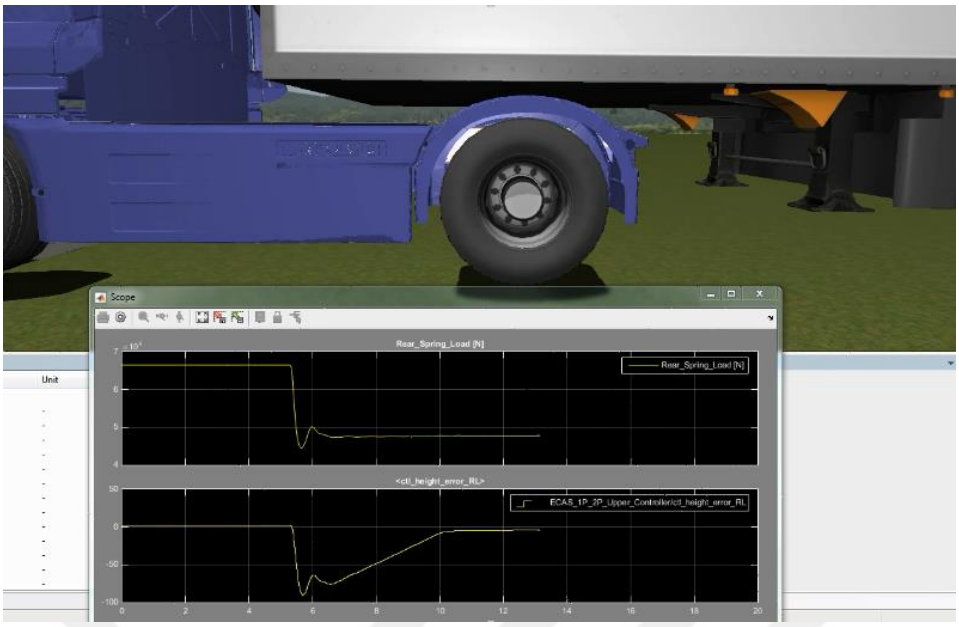
Şekil 5.12 HIL testi sırasında araç statik konumda iken ani yük değişimi başlangıç durumu

5,5. Saniye de yapılan bu işlem körüklerin içindeki havanın bu kadar ani boşaltılamaması sebebiyle aracın yüksekliğini ani olarak arttırmıştır. Bu artım, şekil 5.13'te görülen hata değeri değişimden ya da çamurluğun konumundan gözlemlenebilmektedir.



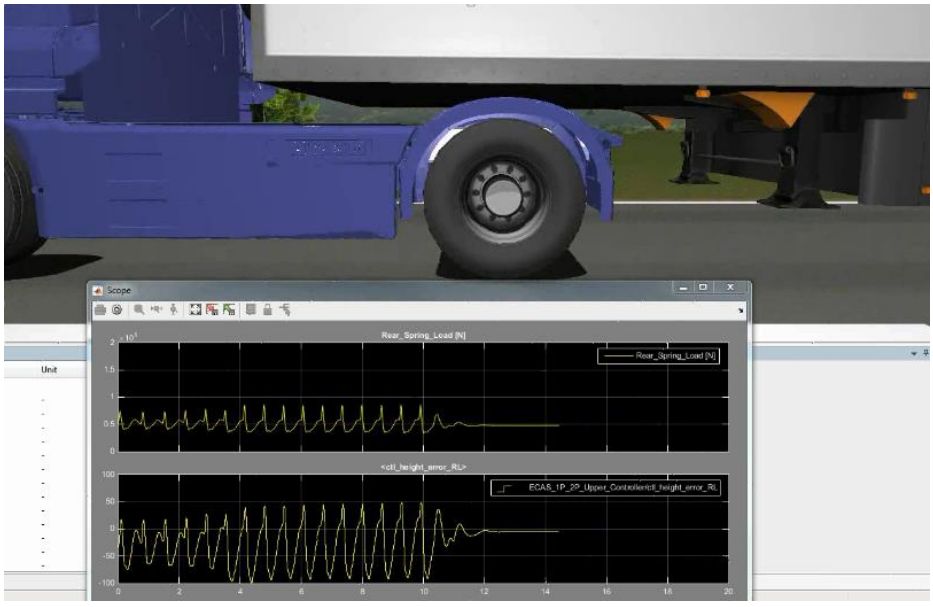
Şekil 5.13 HIL testi sırasında araç statik konumda iken ani yük değişimi ile aracın yükselmesi

Ancak sistem yazılımı ve kontrolü sürekli olarak hedef konuma gelmeye eğilimli olduğundan Şekil 5.14'te de görüldüğü gibi, kontrolcü tarafından hedef sıfıra oturulmuştur.



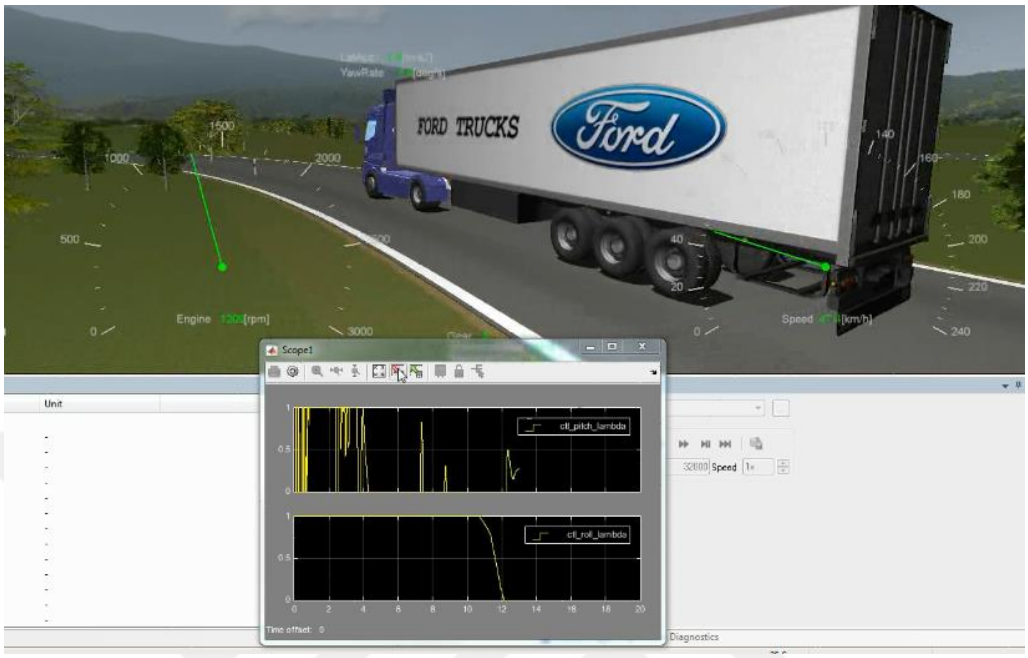
Şekil 5.14 HIL testi sırasında araç statik konumda iken ani yük değişimi sonrasında sürüş yüksekliğine oturması

HIL testleri farklı sürüş alanları ve yollar oluşturmaya da olanak sağlamaktadır. Şekil 5.16'da da gösterildiği gibi, HIL platform üzerinden tanımlana düzensiz engebeli bir yolda dinamik test yapılmıştır. Bu yol sırasında sistemin tepki vermediği, tanımlandığı gibi, düzensizlik biter bitmez kendini tekrar doğru sürüş konumuna getirdiği gözlemlenmiştir.



Şekil 5.15 HIL testi sırasında düzensiz engebeli yolda dinamik araç tepkisi

Aracın sürüş sırasında devrilme katsayısının kontrol edilmesi için de HIL platformunda testler yapılmıştır. Bu test, araç testlerinden önce araç dinamiği limitlerinin doğruluk ve kalibrasyonu için değerler sağlamıştır.



Şekil 5.16 Devrilme (Roll) hareketi HIL testi

5.5 Elektronik Havalı Süspansiyon Kontrol Sistemi Yazılımı Araç Testleri

Araç testleri, geliştirme ve doğrulama işlemleri tamamlandıktan sonra derlenmiş yazılımın araca yüklenmesi ve fiziksel olarak test edilmesini içeren testlerdir. Bu şekilde gerçek zamanlı olarak fonksiyonun, diğer modüller ve fonksiyonlar ile etkileşimi, modelden bağımsız olarak doğrudan araçtan alınan tepkilerin gözlemlendiği test aşamasıdır.



Şekil 5.17 Bozuk yol testi koşan bir araç

Araç testleri sırasında, her bir yol durumunu simüle edebilmek için büyük bir alan ve farklı özelliklerde bozukluklardan oluşan uzun bir parkur kullanılmıştır. Bu testler Şekil 5.18'de de görülen Ford Otosan İnönü fabrikası araç test alanında yapılmıştır.



Şekil 5.18 Araç test parkurunun görünüşü

5.5.1 Araç testi için yazılım implementasyonu

Araç testine gitmeden veya hedef ECU için yazılım yüklenmeden önce yazılımın implemente edilmiş olması gerekmektedir. Burada en önemli kısımlardan biri doğru implementasyonun yapılabilmesi için giriş çıkış dönüştürmelerinin doğru yapılmasıdır. Bu adımda hata yapılması durumunda data tipleri uyumsuzluğuna bağlı veri aşım hataları ya da yazılımda hatalar yaşanabilir.

Tüm implementasyon aşamaları tamamlandıktan sonra elde edilen fonksiyon modelinden C kodu üretilmesi işlemi başlar. C kodu üretilmesi için yüklenecek ECU'ya uygun konfigürasyon yapılmalıdır. Böylece fonksiyonun ECU içerisinde beklenildiği gibi çalışmamasının önüne geçilmiş olur.

C kodu üretilmesinin ardından Matlab tarafından otomatik olarak oluşturulan C ve H dosyaları, yine Da Vinci ile üretilen bir yazılım üreticisine yüklenir. Böylece tüm katmanlar ortak bir çatı altında bir araya getirilmiş olur. Bu üretici ile tüm katmanlar birbirine bağlanır, A2L ve vbf dosyalarının üretimi tamamlanır. Bu işlem ile ECU üzerine yüklenip çalıştırılabilecek dosyalar elde edilmiş olur.

A2L dosyası, içerisinde ECU'nun parametrelerini, değişkenlerini ve ara yüzlerini barındırır. ECU üzerinde çalışan yazılıma ancak bu dosya kullanılarak erişilebilir, takibi yapılabilir ya da müdahale edilebilir.

VBF (Versatile Binary Format) ise temel olarak ECU içerisinde çalışan makine kodunu temsil etmektedir.

Üretilen bu dosyalar araç testlerine başlamak mümkündür. Bu dosyalar ancak uygun bir yükleme programı kullanılarak araca yüklenebilir. Üretilen bu dosyalara ek olarak üretici bir adet anahtar dosyası da üretir. Bu anahtar dosyası da ECU'ya bağlanabilmek ve işlem yapabilmek için gereklidir. Anahtar dosyası ile ECU kişinin kullanımına açıldıktan sonra araca yazılım yükleme işlemi yapılır.

Araç yazılımı yükleme işlemi yapıldıktan sonra, belirlenen araç test gereksinimlerine göre araç testleri tamamlanır ve raporlanır. Araç testleri fonksiyonel testlerin bir çoğu önceden tamamlanmış olduğundan, daha çok araca özgü parametreleri belirleme, performans ve sürüş dinamiği değerlendirme gibi odaklarla yapılır.

Araç testleri sırasında, CAN_BUS hattına bağlı bir data toplama aracı ile araçta var olan bütün CAN sinyalleri depolanır ve buna göre objektif değerlendirmeler yapılır.

6
Sonuç ve Öneriler

Sonuç olarak, elektronik havalı süspansiyon kontrol sisteminin tasarımı tamamlanmış ve 2018 yılında geliştirme çalışmaları tamamlanmış 4x2 çekici araçlara sistem implementasyonu gerçekleştirilmiştir. Bu çalışmanın önemi, giriş çıkışları olan ve bu parametrelere göre uygun çıkışları üreten gömülü bir mekatronik sistemin ağır ticari araçlara integrasyonunda izlenen adımlardır. Bu adımlar, sistem geliştirme aşamasında sıkça kullanılan v-döngüsü prosedüründe tanımlanmış ve takip edilmiştir. Buna göre belirlenen sistemin fonksiyonel gereksinimlerini karşılayacak tasarım getirilmiş ve taşıta uyarlaması yapılmıştır. Her adım bir önceki adımı doğrulamış ve son olarak araç testlerinde sistemin fonksiyonel gereksinimleri karşılanmıştır.

Öncelikli olarak, sistem gereksinimleri tanımlanmıştır. Regülasyon ve araç seviyesi gereksinimlerle birlikte, sistem için yaklaşık 24 adet üst seviye sistem gereksinimi, yazılım seviyesinde ise yaklaşık 138 adet alt seviye sistem gereksinimi belirlenmiştir. Bu gereksinimlere göre, ilk olarak sistemin üst seviye tasarımı yapılmıştır. Üst seviye sistem tasarımının ardından, sistem alt fonksiyonlara bölünmüş ve her bir fonksiyon için alt seviye sistem gereksinimleri oluşturulmuştur. Sistem gereksinimlerinin ardından fonksiyon akış diyagramları oluşturulmuştur. Bu diyagramlar sistemin model tabanlı yazılımını oluşturmada en büyük referanslar olmuştur. Oluşturulan her bir fonksiyon modeli sırasıyla MIL ve HIL testlerine tabi tutulmuştur. Testler genişletilerek araç üzerinde sağlama (validation) aşamasına kadar getirilmiş ve sonuçlandırılmıştır. Fonksiyonel testler ve araç testleri karşılaştırılarak elektronik havalı süspansiyon kontrol sistemi valide edilmiştir.

Bu çalışmalar sonucunda, değişimi ve müdahale edilebilirliği olmayan bir yazılım yerine; %100 oranda algoritma değişiminin entegre edilebileceği bir yazılım geliştirilmiştir. Araç dinamiği uzmanları ile yapılan sürüş testlerinden alınan

sübjektif verilere göre büyük oranda konfor arttırılmıştır. Bu artış, sistemin hedefe oturma sırasında oluşan aşım değerlerinin %30 oranında azalması ile sağlanmış olmuştur. Aşırı yükleme önlemleri bulunmayan bir sisteme, ek olarak, sırasıyla çalışan dört adet güvenlik fonksiyonu eklenerek geliştirilmesi ile %45 oranda güvenlik anlamında daha gürbüz bir sistem tasarımı yapılmıştır.

Sistem için ayrıca kullanılan ECU silinerek; yazılım, araçta hazır olarak bulunan bir ECU içine gömülmüştür. Bu ECU maliyetinin ortadan kalkması ve farklı sensör yapılarının kullanılabilirliğinin artması ile araç başına yaklaşık 70 Avro maliyet azaltılması sağlanmıştır.

Bunun yanında bu kontrol sisteminin, otomotiv seri üretim hattında kullanılabilmesi için EOL, diyagnostik ve kullanıcı etkileşim ara yüzü konfigürasyonları da tamamlanmış ve çevre birimlerle uyum içinde çalışacak bir sistem geliştirmesi tamamlanmıştır.

Kaynakça

[1] B. Bhardwaj, The Complete Book on Production of Automobile Components & Allied Products: Auto parts business start up, Auto parts manufacturing Business, Automobile Based Small Scale Industries, Automobile industry, Automobile manufacturing Industry in India, Automobi, Niir Project Consultancy Services, 2014.

[2] R. Rajamani, Vehicle Dynamics and Control, minneapolis: Springer Science & Business Media, 2011.

[3] S. K. Gupta, A Textbook of Automobile Engineering, India: S. Chand Publishing, 2014.

[4] H. Peng, M. Çakmakçı. A. G. Ulsoy, Automotive Control Systems, New York: Cambridge University Press, 2012.

[5] G. Rizzo, L. Glielmo, C Pianese, F Vasca, Advances in Automotive Control 2004 (2-volume Set), Elsevier, 2005.

[6] J. Gieck, Riding on Air: A History of Air Suspension, Warrendale: Society of Automotive Engineers, Inc., 1999.

[7] Sheldon Axle Company Spring Engineering Department, Leaf Springs: Their Characteristics and Methods of Specification, New York: The Company, 1912.

[8] «airspring-tr,» wassTech, 2013. Çevrimiçi: http://airspring-tr.com/en/air-springs.php. Erişildi: 13 March 2019.

[9] S. E. Seker, "Yazılım Geliştirme Modelleri ve Sistem/Yazılım Yaşam Döngüsü" YBS Ansiklopedi, Istanbul, 2015.

[10] B. Caillaud, D. Nickovic, R. Passerone, B. Raclet, P. Reinkemeir, A. Sangiovanni, W. Damm, T. Henzinger, L. Albert Benveniste, "Contracts for Systems Design: Theory", INRIA, France, 2015.

[11] Thomas Stahl, Model-Driven Software Development, England: Jhon Wiley & Sons, 2006.

[12] L. Motus, Bran Selic, "Using models in real-time software design," *IEEE control systems magazine,* cilt 23, no. 3, pp. 31-42, 2003.

[13] J. Warmer, W. Bast, Anneke Kleppe, MDA Explained: The Model Driven Architecture : Practice and, Addison Wesley, 2003.

[14] M. Kleidermacher, D. Kleidermacher, Embedded Systems Security, Newnes, 2012.

[15] J. S. Guido Sandmann, "Development of AUTOSAR Software Components with Model-Based Design," 2008.

[16] «www.autosar.org,» Autosar, 2019. Çevrimiçi: https://www.autosar.org/standards/classic-platform/. Erişildi: 13 March 2019.

[17] "www.sae.org," Society of Automotive Engineers, 2019. Çevrimiçi: https://www.sae.org/standardsdev/groundvehicle/j1939a.htm. Erişildi: 13 March 2019].

[18] E. V. Veenendaal, "The Testing Practitioner," UTN Publishers, 2004.

[19] L. Copeland, "A Practitioner's Guide to Software Test Design," Artech House, 2004.

[20] R. Isermann, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Engineering Practice,* cilt 7, no. 5, pp. 643-653, 1999.

[21] W. Lee, "A cost- and time-effective hardware-in-the-loop simulation platform for automotive engine control systems," *SAGE* , cilt 217, no. 1, pp. 41-52, 2003.

Tezden Üretilmiş Yayınlar

İletişim Bilgisi: cansuusahbaz@gmail.com

Konferans Bildirileri

1. C. Şahbaz, T.Keleş, E. E. Sezişli, M. Görürgöz, H. Üvet, "Havalı Süspansiyonlu Ağır Ticari Araçlar İçin Yükseklik Kontrol Sistemi Geliştirme Çalışması, OTEKON, The International Automotive Technologies Congress, pp.720-730, 2018.