

**HAREKETLİ ROBOTLARIN KİNEMATİK
DEĞİŞKENLERİNİN TAHMİN EDİLMESİ İÇİN TEST
PLATFORM TASARIMI**

**TEST PLATFORM DESIGN FOR KINEMATIC PARAMETER
ESTIMATION OF MOBILE ROBOTS**

SEZAI HACIOĞLU

**YARD. DOÇ. DR. CAN ULAŞ DOĞRUER
TEZ DANIŞMANI**

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Makine Mühendisliği Anabilim Dalı için Öngördüğü
Yüksek Lisans Tezi olarak hazırlanmıştır.

2014

SEZAI HACIOĞLU'nun hazırladığı “**Hareketli Robotların Kinematik Değişkenlerinin Tahmin Edilmesi İçin Test Platform Tasarımı**” adlı bu çalışma aşağıdaki jüri tarafından **MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI**' nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Bora YILDIRIM

Başkan

Yard. Doç. Dr. Can Ulaş DOĞRUER

Danışman

Yard. Doç. Dr. A. Buğra KOKU

Üye

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

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

beyan ederim.

21/12/2014

SEZAI HACIOĞLU

ÖZET

HAREKETLİ ROBOTLARIN KİNEMATİK DEĞİŞKENLERİNİN TAHMİN EDİLMESİ İÇİN TEST PLATFORM TASARIMI

Sezai HACIOĞLU

Yüksek Lisans, Makine Mühendisliği Bölümü

Tez Danışmanı: Yard. Doç. Dr. Can Ulaş DOĞRUER

Aralık 2014

Motor, algılayıcı, mekanik, algısal ve bilişsel konuları kapsayan mobil robotlar konusu disiplinler arası bir alana girmektedir. Robot hareket problemleri üzerinde çalışan mühendislerin, mekanizmalar, kinematik, dinamik ve kontrol teorisine hakim olmaları gerekmektedir. Güçlü algı sistemlerinin oluşturulmasında mikrodenetleyici, elektronik ve yazılım konularına hakim olunması gerekir.

Mobil robotlarda en çok araştırılan konu, robotların konum ve yönlerini doğru olarak bulunması problemidir, enkoder, jiroskop, ivmeölçer, pusula gibi algılayıcı verileri kullanılarak robotların konum ve yönleri belirlenebilir.

Mobil robotlardaki tekerlek çapları uyumsuzluğu, tekerlek açıklığının belirsizliği gibi geometrik etkenler konum tahmini problemini karmaşık hale getirir. Özellikle, Kalman süzgeci uygulamalarında süzgecin çıktısının zamanla sapmasına sebep olur. Bu tip uygulama zorluklarını gidermek için hareketli robotun kinematik modeli çeşitli testleri yapılarak hatalardan arındırılır.

Bu tez, hareketli robotların kinematik deęişkenlerini tahmin etmek için hareketli bir platform tasarımını, deęişken tahmini için test ortamı oluşturulmasını ve yazılım geliştirilmesini kapsar. Bu kapsam da laboratuvar ortamında robotun çalışma alanını görebilen bir kamera tavana yerleştirilir ve basit görüntü işleme metotları ile robotun konumu mutlak olarak tesbit edilir. Test ortamında bulunan dizüstü bilgisayar, robotun üzerine yerleştirilen mikrodenetleyici ile Xbee kablosuz bağlantı modülü ile iletişim sağlar, mikrodenetleyiciden odometre konum verisi alır ve sabit diske kayit eder. Eş zamanlı olarak motor komutlarını mikrodenetleyiciye gönderir. Mikrodenetleyici ise teker dönüş hızlarını ölçer ve motor hızlarını kontrol eder.

Test yörüngesi olarak dairesel bir yörünge seçilmiştir. Bu tip dairesel yörünge de robotun konumu ve yönü sürekli deęişir ve analiz için zengin bir veri serisi sağlar. Testler sonucunda elde edilen veriler LSE metodu ile analiz edilir ve odometreye dayalı ardışık konum verisi ile kameradan elde edilen mutlak konum verisi karşılaştırılarak geometrik deęişkenler tahmin edilir. Bu kapsamda birden çok test verisi toplandı, analiz edildi ve sonuçlar tez kapsamında sunuldu.

Anahtar Kelimeler: Parametre Tahmini, Odometre, Gömülü Sistemler, Hareketli Platformlar, Bilgisayar Görüşü.

ABSTRACT

TEST PLATFORM DESIGN FOR KINEMATIC PARAMETER ESTIMATION OF MOBILE ROBOTS

Sezai HACIOĞLU

Master of Science Degree, Department of Mechanical Engineering

Supervisor : Yard. Doç. Dr. Can Ulaş DOĞRUER

December 2014

Research on mobile requires that one has knowledge on motor, sensor, and mechanic, perceptual and cognitive topics. It is evident that robotics research is an interdisciplinary field. Researchers who works on robotic problem must have a broad knowledge about mechanisms, kinematic, dynamics and control theory. In order to devise robust perception system, researcher must have good knowledge of signal analysis, embedded electronics, and computer programming techniques.

Among many problems associated with this field of science, localization of a mobile robot has attracted the most attention from the robotics community. Sensors such as encoders, gyroscope, accelerometer, and compass has been intensively used in estimation theory frame i.e. Kalman filter, to estimate the pose and heading of mobile robots, however, estimation theory is not free of flaws and it requires that certain assumption are met before, the statistical filter is put in use.

The mechanical design of the mobile robot base is inherently inaccurate to a certain extent, for instance wheels diameters and distance between wheel centers may not be known exactly or they can vary in time. These vague parameters make the localization problem more complicated. In particular, Kalman filter which has been the primary mathematical tool for estimation problem, may diverge when plant model of the system has biased uncertainties. This divergence problem also occurs when Kalman filter is applied to the localization problem of a mobile robot. In order to alleviate these sort of problems, the kinematic model of the mobile robots are calibrated by doing certain experiments that exist in the literature.

In this thesis, in order to estimate the unknown geometric parameters of kinematic model of a wheeled mobile robot, a mobile robot and its embedded systems are designed and produced. An embedded software that runs on mobile robot electronics is developed. MATLAB software based data acquisition system that uses a web camera as the main instrument is developed. A web camera is mounted at the ceiling of an indoor environment and using simple image processing methods the absolute poses of the mobile robot along the path are computed. A laptop on which MATLAB is running, communicates with the Arduino board mounted on the mobile robot using Xbee wireless module and it send motor signals to the Arduino board, in the meantime Arduino board controls the dc motors, reads the odometer. Odometer data is recorded on the hard disk. This completes one typical cycle of a test run.

In this study, the test path is selected as a circular one. In this type of path, heading of the mobile robot changes continuously and it gives a rich data set for further analysis. After test has been completed, the data set is examined with Least Squares Estimation (LSE) method. The output of odometer is integrated with respect time to predict the future states of the mobile robot, these predicted poses are compared with the absolute poses computed with overhead camera and the uncertain parameters are estimated. In this study, comprehensive data set has been recorded and analyzed, results of these analysis are presented in the final part of thesis.

Keywords: Parameter Estimation, Odometry, Embedded Systems, Mobile Robot, Computer Vision.

TEŐEKKÜR

Tez alıőmam boyunca, bilgisiyle ve tecrübesiyle yanımda olan zorlandığım durumlarda yol gösteren ve destekleyen saygıdeęer hocam ve danıőmanın Sayın Yard. Do. Dr. Can Ulaő DOĐRUER' e tım katkılarından dolayı teőekkür ederim.

Deęerlendirmeleri ve yorumlarıyla katkıda bulunan jüri üyeleri Sayın Prof. Dr. Bora YILDIRIM ve Sayın Yard. Do. Dr. Buęra KOKU' ya teőekkür ederim.

Bu alıőma boyunca beni anlayıőla ve sabırla karőılayan eőime, aileme ve iő arkadaşlarıma teőekkür ederim

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ÇİZELGELER	ix
ŞEKİLLER	x
SİMGELER ve KISALTMALAR	xvii
1. GİRİŞ	1
1.1 Problem Tanımı	5
1.2 Tez Çalışmasının Hedefi	6
2. LİTARATÜR TARAMASI	7
2.1 Giriş	7
2.2 Literatür Özeti	7
2.3 Sonuç	10
3. PLATFORM HAREKET SİSTEMLERİ ve MOBİL ROBOT KİNEMATİĞİ	11
3.1 Giriş	11
3.2 Platform Hareket Sistemleri	11
3.2.1 Bacaklı Hareket	12
3.2.2 Tekerli Hareket	13
3.2.2.1 Tekerlek Tipleri	13
3.2.2.2 Sürüş Çeşitleri	16
3.2.2.3 Tekerlekli Hareketli Platformların Manevra Yetenekleri	22
3.3 Mobil Robot Kinematığı	23
3.3.1 İleri ve Ters Robot Kinematığı	23
3.3.2 Holonomik Olmayan Kısıtlama	25

3.3.3 Holonomik Olmayan Robotlar	25
3.3.4 Tek Tekerlek Kinematığı	25
3.3.5 Diferansiyel Sürüşlü Robot Kinematığı	26
3.4 Sonuç	28
4. PLATFORM TASARIMI	30
4.1 Giriş	30
4.2 Diferansiyel Sürüşlü Hareketli Platform Tasarımı	30
4.2.1 Platform Tasarımı	31
4.2.2 Robot Donanımı	34
4.2.2.1 Motor	35
4.2.2.2 Mikrokontrolcü	36
4.2.2.3 Motor Sürücü	37
4.2.2.4 Batarya	38
4.2.2.5 Xbee Kablosuz İletişim Modülü	39
4.2.2.6 Tekerlekler	40
4.2.2.7 Kaster Tekerlek	40
4.2.3 Algılayıcılar	41
4.2.3.1 Enkoder	41
4.2.3.2 Kamera	42
4.3 Sonuç	43
6. KONTROL MİMARİSİ	44
5.1 Giriş	44
5.2 Platform Devresi	44
5.3 Test Sistemi Mimarisi	46
5.3.1 Mikrokontrolcüde Kullanılan Komutlar	47
5.3.2 Hesaplama Bilgisayarında Kullanılan Komutlar	49
5.3.3 Bilgisayar-Mikrokontrolcü İletişimi	52

5.4 Sonuç.....	54
6. DENEYSEL ÇALIŞMALAR.....	55
6.1 Giriş.....	55
6.1 Test Düzeniği	55
6.3 Kamera Kalibrasyonu.....	58
6.1 Sonuç.....	59
7. TEORİ ve TOPLANAN VERİLERİN ANALİZ EDİLMESİ	60
7.1 Giriş.....	60
7.2 Teori	61
7.2.1 Yığın LSE Tahmini	63
7.2.2 Ardışık Değişken Tahmini.....	64
7.3 Veri Analizi	67
7.3.1 Veri Analizi Test 1	69
7.3.2 Veri Analizi Test 2	83
7.3.3 Veri Analizi Test 3	96
7.3.4 Veri Analizi Test 4.....	109
7.4 Sonuç.....	122
8. SONUÇ ve TARTIŞMA	123
KAYNAKLAR.....	125
EKLER.....	126
EK:1 MİKROKONTROLCÜ YAZILIMI	126
EK:2 TEST 1 ODOMETRE VERİSİ	139
EK:3 HAREKETLİ PLATFORM TEKNİK RESİMLERİ.....	140
ÖZGEÇMİŞ.....	155

ÇİZELGELER

Sayfa

Çizelge 3.1. Çeşitli mobil robotun hareketlilik ve yönlendirme dereceleri	23
Çizelge 4.1. Motor ve enkoder kablo renk kodları	36
Çizelge 4.2. Arduino mega 2560 kart özellikleri.....	37
Çizelge 4.3. Arduino motor sürücü modülü pin adresleri.....	38
Çizelge 4.4. Batarya özellikleri	38
Çizelge 4.5. Xbee kablosuz iletişim modülü özellikleri	39
Çizelge 5.1. İletişim safhalarında seri iletişim komutları ve içerdikleri veri paketi ve veriler.	53

ŞEKİLLER

Sayfa

Şekil 1.1: Plustech ormanlık alandan dışarıya kütükleri taşımak için tasarlanmıştır.	3
Şekil 1.2: Kutup buzul sularında araştırmada kullanılan ALTEX otonom sualtı aracı.	3
Şekil 1.3: Tur rehber robotları Rhino (A) (Bonn Üniversitesi) ve Minerva (B) (Carnegie Mellon Üniversitesi)	3
Şekil 1.4: ActivMedia Robotics' in deneysel çalışmalarda kullanılan Pioneer 3-AT robotu.	4
Şekil 1.5: K Team SA 'nin deneysel çalışmalarda kullanılan Khepera 3 robotu.	4
Şekil 1.6: I-robot'un askeri uygulamalarda kullanılan 310 SUGV robotu.	4
Şekil 3.2: Sabit tekerlek (A) , kaster tekerlek (B) , tahrikli yönlendirilmeli eş merkezli tekerlek (C) , tahrikli yönlendirilmeli kaçık merkezli tekerlek (D).....	14
Şekil 3.3: Üniversal çok yönlü tekerlek..	15
Şekil 3.4: 45° açılı mekanum tekerlek (A), -45° açılı mekanum tekerlek (B) ,genel görünüş (C).....	15
Şekil 3.5: Küresel tekerlek uygulaması.....	16
Şekil 3.6: Sol ve sağ tekerlek doğrusal hızlarına göre robotun hareketi.....	17
Şekil 3.7: Sol ve sağ teker doğrusal hızlarına göre robotun hareketi.....	17
Şekil 3.8: Üç tekerlekli mobil robot hareket şekli.	18
Şekil 3.9: Çok yönlü platformlar Üç tekerli 90° rulolu (A), Dört tekerli ve 45° rulolu (B)	19
Şekil 3.10: Dört mekanum tekerlekli mobil robotun standart kurulumu.	19

Şekil 3.12: Eş zaman sürürlü mobil robot hareketi (A), sürüş ve yönlendirme kayış sistemi (B).....	19
Şekil 3.13: Ackerman yönlendirmeli hareketli robotun dönüşü.....	21
Şekil 3.14: Kızaklamalı yönlendirmeli mobil robot.	22
Şekil 3.15: Tek tekerlek hareketi.	26
Şekil 3.16: Diferansiyel sürürlü robotun kinematığı..	27
Şekil 4.1: Üst platform CAD modeli.	32
Şekil 4.2: Motor bağlantı braketi CAD modeli.	32
Şekil 4.3: Ana şasinin patlatılmış CAD modeli.....	33
Şekil 4.4: Nesne takip platformu	34
Şekil 4.5: Hareketli platform.	34
Şekil 4.6: 67Dx54L tipi motorun ölçüsel teknik resmi.	35
Şekil 4.7: Arduino Mega 2560 Kartı.....	36
Şekil 4.8: Arduino motor sürücü modülü	37
Şekil 4.9: Li-Po batarya.....	39
Şekil 4.10: Xbee kablosuz iletişim modülü.....	39
Şekil 4.11 : Tekerlek çifti.....	40
Şekil 4.12 Kaster tekerlek.	40
Şekil 4.13: Artırımsal enkoder.....	41
Şekil 4.14 : Mutlak enkoder yapıları.....	41
Şekil 4.15: Kuadratik enkoder ve faz sinyalleri	42
Şekil 4.16: CMOS görüntü algılayıcılı kamera.	43
Şekil 5.1: Robot platformunun devre şeması	45

Şekil 5.2: Test sistemi akış diyagramı.	47
Şekil 5.3: Robotla Hesaplama /görüntü işleme bilgisayarı arasındaki seri iletişim ve program akışı.	51
Şekil 6.1. Test Düzenegi.	56
Şekil 6.2: Kamera görüş alanı (A) ve test alanı köşelerinin (B) piksel değerleri	56
Şekil 6.3: Görüş alanı, test alanı, test başlangıç yeri ve kamera orjini koordinatları.	57
Şekil 6.4: Nesne takip deseni.....	57
Şekil 6.5: Kalibrasyon tablosu.....	58
Şekil 6.5: Test düzeneginin kalibrasyonu	59
Şekil 7.1: LSE akış şeması.	66
Şekil 7.2: Mobil platformun test ortamında yörüngesini çizerken alınan kamera takip görüntüleri.	67
Şekil 7.3: Görüntü işleme takibi yöntemiyle oluşturulan robot yörüngesi.	68
Şekil 7.4: TEST1 parametre hata yaklaşımı.....	70
Şekil 7.5: TEST1 Sağ ve sol tekerlek açısız hızları.....	71
Şekil 7.6: TEST1 odometre verine göre platform yörüngesi.	72
Şekil 7.7: TEST1 kamera verine göre platform yörüngesi.....	72
Şekil 7.8: TEST1 odometre takibinde platform X/Y eksen konumu.....	73
Şekil 7.9: TEST1 kamera takibinde platform X/Y eksen konumu.....	73
Şekil 7.10: TEST1 odometre verisine göre platform yön değişikliđi.	74
Şekil 7.11: TEST1 kamera verisine göre platform yön değişikliđi.	74
Şekil 7.12: TEST1 kalibrasyondan önce verilerin X/Y konum kıyaslaması	75
Şekil 7.13: TEST1 kalibrasyondan sonra verilerin X/Y konum kıyaslaması	75

Şekil 7.14: TEST1 kalibrasyondan önce odometre ve kamera verisine göre platform yönü	76
Şekil 7.15: TEST1 kalibrasyondan sonra odometre ve kamera verisine göre platform yönü	77
Şekil 7.16: TEST1 kalibrasyondan önce odometre ve kamera platform X konumu..	77
Şekil 7.17: TEST1 kalibrasyondan sonra odometre ve kamera platform X konumu	78
Şekil 7.18: TEST1 kalibrasyondan önce odometre ve kamera verisi platform Y konumu..	79
Şekil 7.19: TEST1 kalibrasyondan sonra odometre ve kamera verisi platform Y konumu.	79
Şekil 7.20: TEST1 Kalibrasyondan önce odometre X eksen hatası	80
Şekil 7.21: TEST1 Kalibrasyondan sonra odometre X eksen hatası	80
Şekil 7.22: TEST1 kalibrasyondan önce odometre Y eksen hatası.	81
Şekil 7.23: TEST1 kalibrasyondan sonra odometre Y eksen hatası.	81
Şekil 7.24: TEST1 kalibrasyondan önce odometre yön hatası.....	82
Şekil 7.25: TEST 1 kalibrasyondan sonra odometre yön hatası.....	82
Şekil 7.26: TEST 2 parametre hata yaklaşımı	84
Şekil 7.27: TEST 2 platform sağ sol tekerlek açısal hızları.	84
Şekil 7.28: TEST 2 odometre verisine göre platform yörüngesi.	85
Şekil 7.29: TEST 2 kamera verisine göre platform yörüngesi.	86
Şekil 7.30: TEST 2 odometre takibinde platform X/Y eksen konumu.....	87
Şekil 7.31: TEST 2 kamera takibinde platform X/Y eksen konumu	87
Şekil 7.32: TEST 2 odometre verisine göre platform yön değişikliği	88
Şekil 7.33: TEST 2 Kamera verisine göre platform yön değişikliği	88
Şekil 7.34: TEST 2 Kalibrasyondan önce verilerin X/Y konum kıyaslaması.....	89
Şekil 7.35: TEST 2 Kalibrasyon sonrası verilerin X/Y konum kıyaslaması.....	89

Şekil 7.36: TEST 2 Kalibrasyon öncesi odometre ve kamera verisi platform yön açısı	90
Şekil 7.37: TEST 2 Kalibrasyon sonrası odometre ve kamera verisi platform yön açısı .	90
Şekil 7.38: TEST 2 kalibrasyon öncesi odometre ve kamera verisine göre platform X konumu.....	91
Şekil 7.39: TEST 2 kalibrasyon sonrası odometre ve kamera verisine göre platform X konumu.....	91
Şekil 7.40: TEST 2 kalibrasyon öncesi odometre ve kamera verisini platform Y konumu.	92
Şekil 7.41: TEST 2 kalibrasyon sonrası odometre ve kamera verisini platform Y konumu	92
Şekil 7.42: TEST 2 kalibrasyon öncesi odometre X eksen hatası	93
Şekil 7.43: TEST 2 kalibrasyon sonrası odometre X eksen hatası	93
Şekil 7.44: TEST 2 kalibrasyon öncesi odometre Y eksen hatası	94
Şekil 7.45: TEST 2 kalibrasyon sonrası odometre Y eksen hatası.....	94
Şekil 7.46: TEST 2 kalibrasyon öncesi odometre yön hatası.....	95
Şekil 7.47: TEST 2 kalibrasyon sonrası odometre yön hatası.....	95
Şekil 7.48: TEST 3 parametre hata yaklaşımı	97
Şekil 7.49: TEST 3 Sağ/Sol Tekerlek Açısal Hızları.....	97
Şekil 7.50: TEST 3 odometre verine göre platform yörüngesi	98
Şekil 7.51: TEST 3 kamera takibine göre platform yörüngesi.....	99
Şekil 7.52: TEST 3 odometre takibinde X/Y eksen Konumu	100
Şekil 7.53: TEST 3 kamera takibinde X/Y eksen Konumu	100
Şekil 7.54: TEST 3 odometre verisine göre platform yönü	101
Şekil 7.55: TEST 3k verisine göre platform yönü	101
Şekil 7.56: TEST 3 kalibrasyondan önce verilerin X/Y konum kıyaslaması.....	102
Şekil 7.57: TEST 3 kalibrasyondan sonra verilerin X/Y konum kıyaslaması.....	102

Şekil 7.58: TEST 3 kalibrasyondan önce odometre ve kamera platform yön açısı.....	103
Şekil 7.59: TEST 3 kalibrasyondan sonra odometre ve kamera platform yön açısı	103
Şekil 7.60: TEST 3 kalibrasyondan önce odometre ve kameraya platform X konumu ...	104
Şekil 7.61: TEST 3 kalibrasyondan sonra odometre ve kameraya platform X konumu ..	104
Şekil 7.62: TEST 3 kalibrasyondan önce odometre ve kameraya platform Y konumu ...	105
Şekil 7.63: TEST 3 kalibrasyondan sonra odometre ve kameraya platform Y konumu	105
Şekil 7.64: TEST 3 kalibrasyondan önce odometre X eksen hatası	106
Şekil 7.65: TEST 3 kalibrasyondan sonra odometre X eksen hatası.....	106
Şekil 7.66: TEST 3 kalibrasyondan önce odometre Y eksen hatası.....	107
Şekil 7.67: TEST 3 kalibrasyondan sonra odometre Y eksen hatası.....	107
Şekil 7.68: TEST 3 kalibrasyondan önce odometre yön hatası.....	108
Şekil 7.69: TEST 3 kalibrasyondan sonra odometre yön hatası.....	108
Şekil 7.70: TEST 4 parametre hata yaklaşımı	110
Şekil 7.71: TEST 4 sağ sol tekerlek açısal hızları (rad/s)	110
Şekil 7.72: TEST 4 odometre verine göre platform yörüngesi	111
Şekil 7.73: TEST 4 kamera verisine göre platform yörüngesi	112
Şekil 7.74: TEST 4 odometre takibinde platform X/Y eksen konumu	112
Şekil 7.75: TEST 4 kamera takibinde platform X/Y eksen konumu	113
Şekil 7.76: TEST 4 odometre verisine göre platform yönü	114
Şekil 7.77: TEST 4 kamera verisine göre platform yönü.....	114
Şekil 7.78: TEST 4 kalibrasyondan önce verilerin X/Y konum kıyaslaması	115
Şekil 7.79: TEST 4 kalibrasyondan sonra verilerin X/Y konum kıyaslaması	115
Şekil 7.80: TEST 4 kalibrasyondan önce odometre ve kamera platform yön açısı	116

Şekil 7.81: TEST 4 kalibrasyondan sonra odometre ve kamera platform yön açısı	116
Şekil 7.82: TEST 4 kalibrasyondan önce odometre ve kameraya platform X konumu....	117
Şekil 7.83: TEST 4 kalibrasyondan sonra odometre ve kameraya platform X konumu ..	117
Şekil 7.84: TEST 4 kalibrasyondan önce odometre ve kameraya platform Y konumu	118
Şekil 7.85: TEST 4 kalibrasyondan sonra odometre ve kameraya platform Y konumu ...	118
Şekil 7.86: TEST 4 kalibrasyondan önce odometre X eksen hatası.....	119
Şekil 7.87: TEST 4 kalibrasyondan sonra odometre X eksen hatası.....	119
Şekil 7.88: TEST 4 kalibrasyondan önce odometre Y eksen hatası.....	120
Şekil 7.89: TEST 4 kalibrasyondan sonra odometre Y eksen hatası.....	120
Şekil 7.90: TEST 4 kalibrasyondan önce odometre yön hatası.....	121
Şekil 7.91: TEST 4 kalibrasyondan sonra odometre yön hatası.....	121

SİMGELER ve KISALTMALAR

Simgeler

V_l	Sol tekerlek doğrusal hızı
V_r	Sağ tekerlek doğrusal hızı
a	Tekerlekle tekerlek açıklık merkezi arasındaki mesafe
R	Dönüş yarıçapı
M_y	Manevra yeteneği
D_h	Hareketlilik derecesi
D_y	Yönlendirilme derecesi
W	Tekerlek değişkenleri vektörü
X	Konum değişkenleri vektörü
J	Jacobian matrisi
x	X eksen konumu
y	Y eksen konumu
ϕ	Platform açısı
θ	Tekerlek dönüş açısı
r	Tekerlek yarı çapı
V_ϕ	Platform açısal dönüş hızı
$\dot{\theta}_l$	Sol tekerlek açısal hızı
$\dot{\theta}_r$	Sağ tekerlek açısal hızı
b	Tekerlek açıklık merkezi
\dot{x}_T	T noktasındaki x eksenindeki hız
\dot{y}_T	T noktasındaki y eksenindeki hız

$\dot{\phi}$	Platform açısai dönüş hızı
$f_k(\cdot)$	ayrık zaman doğrusal olmayan modeli
v_k	platformun öteleme hızı
ω_k	Platformun açısai hızı
μ_R	Sağ tekerlek ölçek faktörü
μ_L	Sol tekerlek ölçek faktörü
μ_D	Teker açıklığı ölçek faktörük
k	Ayrık zaman
z_k	Ölçüm vektörü
Z_k	Ölçüm Matrisi
ε	Kıstas değeri
P_k	Kovaryans matrisi
R	Ölçüm kovaryans matrisi

Kısaltmalar

LSE	Least Square Estimation
İHA	İnsansız Hava Aracı
OSA	Otonom Sualtı Aracı
TMR	Tekerlekli Mobil Robotlar
UMBmark	University of Michigan Benchmark
SLAM	Simultaneous Localization and Mapping
LSM	Least Square Method
EKF	Extanded Kalman Filter
AEM	Anlık Eğrilik Merkezi
ADM	Anlık Dönüş Merkezi
CPR	Counts per Revolution
PWM	Pulse Width Modulation

IEEE	Institute of Electrical and Electronics Engineers
ASME	American Society of Mechanical Engineers
AIM	Advanced Intelligent Mechatronics
MMAE	Multiple Model Adaptive Estimation

1. GİRİŞ

Robot teknolojileri, üretim alanda büyük başarılar kaydetti. Robot kolları büyük bir endüstri oluşturdu. Fabrikalarda, yüksek hızları ve hassasiyetleriyle seri üretim montaj hatlarının önemli bir parçası oldular. Kaynak, boyama gibi sık tekrarlanan işlerde büyük verimlerle çalıştılar.

Robot kollarının belirlenen alanlarda çalışması, çalışma alanlarının cıvatalandıkları yerlerde işlerini icra etmeleri, robot teknolojisini kısıtlayan en büyük sıkıntı. Halbuki sabitlendikleri zeminin hareket etmesi çok geniş alanlarda görev yapabilmelerini sağlayacak.

Mobil robotlar, otonom olarak bir yerden başka bir yere gidebilen robotlardır. Bu hareketi yaparken dışarıdan herhangi bir kullanıcı yardımı olmaksızın yapar. Endüstriyel robotların sadece özel tanımlanan çalışma alanlarında hareket edebilen endüstriyel robotların aksine mobil robotlar verilen görevleri başarmak için önceden tanımlanmış bir çalışma alanı içerisinde serbestçe hareket etme özelliği vardır.

Bu hareketlilik yeteneği, onları insan yapımı ve doğal çevrelerdeki birçok uygulamalarda çalışabilmesini sağlar. Kara mobil robotları, tekerlekli mobil robotlar ve bacaklı mobil robotlar olarak ayrılır.

Mobil robotlar insansız hava araçlarını (İHA) ve otonom sualtı araçlarını (OSA) da kapsar. Karasal uygulamalarda Tekerlekli Mobil Robotlar (TMR) mekanik yapılarının basit oluşu, az enerji tüketimi dolayısıyla uzun pil ömürleri nedeniyle çok popülerdir. Bacaklı robotlar standart olmayan merdiven, engebeli arazi gibi çevre koşullarındaki görevler için daha uygundur. Bacaklı mobil robotlar iki, üç, dört bacak gibi çeşitli bacak tertiplerine sahip olabilirler. Bazı mobil robotlar yönlendirici kollara sahiptir. Bu tip mobil yönlendirici kolları uzay araştırmalarında, askeri uygulamalarda, sağlık sektörü ve ev bakımında kullanılmaktadırlar.

İnsanların görev yapamayacağı çok tehlikeli yerlerde uzaktan görev yapabilen robotlar kullanılır. Bu uygulamalarda bu robotların hareketlerini insanlar yönlendirir. İnsan, yer tahmini ve yapılacak işlemleri algılar, robot hareketleri ise robotun kontrolcüsünden gerçekleşir.

Örneğin, Plustech en iyi yürüme dengesini sağlamak ve araziye en az hasarı vermek amacıyla tasarlanmış bacaklı mobil robottur. Robot ileri, geri yan ve köşegensel yönlerde hareket edebilir(Şekil 1.1.). Şekil 1.2.'de kullanıcı çalışırken kendi kendini dengeleyebilmek için altı pervanesini kontrol eden bir sualtı aracı gösterilmiştir. Şekil 1.3'te insanlarla aynı ortamı paylaşan otonom robotlar, insan müdahalesi gerekmez konumunu anlayabilme ve kendi kendine gezinebilme yeteneklerine sahiptirler.

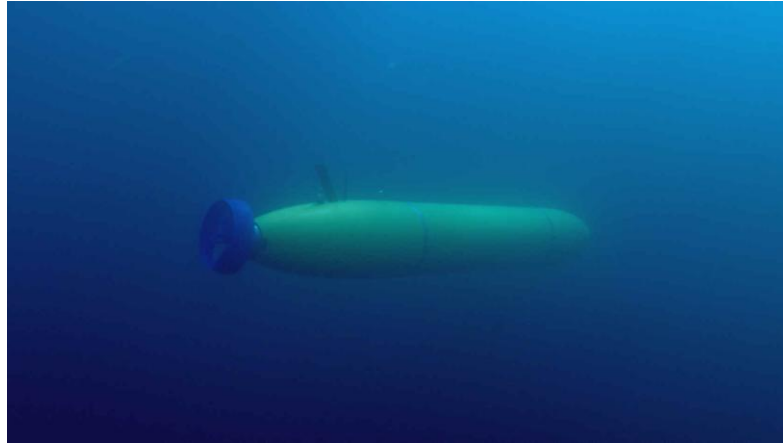
Biliş, yer sınırlandırma ve gezinmedeki yüksek seviyedeki sorunların araştırması, laboratuvar koşullarında standart araştırma robot platformlarında uygulanabilir. Çeşitli mobil robot platformları programlama arazi kabiliyetleri araştırmaları için uygundur. En popüler araştırma robotları ActivMedia Robotics, K Team SA ve I-robot olanlardır. (Şekil 1.4, 1.5, 1.6)

Birçok mobil robot çalışması mobil robotun “nerdeyim ?” sorusuna yanıt almak için yapılmaktadır. Şöyle düşünün, nerede olduğunuzu kavrayamadan gitmek istediğiniz yere nasıl gidebilirsiniz? Hareket etmeye kalktığınızda bir yerlere çarparsınız. Robotlar içinde durum aynıdır. Bu durumun çözümü ise nerede olduğunu bilmek veya tahmin edebiliyor olmaktır.

Robotların nerde olduklarını bilmelerinin en ucuz ve basit yolu tekerleklerinde bulunan enkoder verileriyle kinematik parametreleri arasındaki ilişkiden nerede olduklarını tahmin etmektir. Bu yöntem, uygulamada tam anlamıyla çalışmaz, mobil robotun geometrik hatalarından dolayı robot programlanan yere ulaşmaz. Bu hatalar genellikle robotun teker çapları arasında farklardan ve tekerlek açıklığındaki değişimlerden dolayı oluşmaktadır. Bu hataları gidermek için birçok kalibrasyon yöntemi denenmiş, bu yöntemler önceleri deneme yanılmayla yapılarak vakit kayıplarına ve kalibrasyonların uzun zaman sürmesine neden olmuş. Daha sonraları Borenstein ve arkadaşlarının geliştirdiği UMBmark yöntemiyle daha kısa zamanda ve çok daha hassas sonuçlar elde edilmiştir.



Şekil 1.1 : Plustech ormanlık alandan dışarıya kütükleri taşımak için tasarlanmıştır.[1]



Şekil 1.2. : Kutup buzul sularında arařtırmada kullanılan ALTEX otonom sualtı aracı.[2]



A

B

Şekil 1.3: Tur rehber robotları Rhino (A) (Bonn Üniversitesi) ve Minerva (B) (Carnegie Mellon Üniversitesi).[3]



Şekil 1.4 : ActivMedia Robotics' in deneysel çalışmalarda kullanılan Pioneer 3-AT robotu.

[4]



Şekil 1.5: K Team SA 'nin deneysel çalışmalarda kullanılan Khepera 3 robotu. [5]



Şekil 1.6 : I-robot'un askeri uygulamalarda kullanılan 310 SUGV robotu. [6]

1.1 Problem Tanımı

Hareketli robotların verilen görevi başarabilmeleri için en önemli şart konumlarını takip edebilmeleri ve kendilerini etraflarındaki cisimlere göre konumlandırmalarıdır. Bu yetenek daha önceden verilen bir yörüngeyi takip edilirken, ve hareketli cisimlerin olduğu bir ortamda yörüngeyi anlık olarak güncellenmesi için gerekir. Engel kaçınma, yol planlama ve yörünge takibi daha karmaşık görevlerin mesela gözetleme, devriye, alan taraması, bir cisim bir yerden başka bir yere taşınması, hareketli bir hedef cismin takip edilmesi görev ve yeteneklerinin en önemli bileşenidir. Bu temel davranışların temel bileşeni ise sağlıklı konum tahminidir. Konum tahmini probleminde temel olarak daha önceden bilinen ve haritalanmış bir ortamda robotun konumunu takip edilmesi en basit problemdir. Karmaşık ve kendi içerisinde bütünlüğü bulunan problem ise eş zamanlı haritalama ve konum takibi problemdir. Bu problem İngilizce Simultaneous Localization and Mapping problemi olarak bilinir ve bilimsel literatürde güncel kabul görmüş kısaltması SLAM ile bilinmektedir. Bahsi geçen her iki problemin çözümünde Kalman süzgeci ve türevleri sıkça kullanılmaktadır. Bu tip istatistiksel süzgeçlerin doğru çalışmasının en önemli ön şartı gürültü tiplerinin beyaz ardışık Gauss tipi gürültü olarak modellenmesidir. Bu sanal şart pratikte birçok mekanik sistem tarafından karşılanmaz dolayısıyla bu süzgeçlerde kullanılan yaklaşık modellerin kalibrasyonun yapılarak modellenemeyen sistem davranışlarının yukarıda bahsi geçen gürültü tipi ile yakınsanmasının sağlanması gerekir. Bu çalışmada tekerlekli robotun kinematik modelinin kalibrasyonu hedeflenir. Bu kapsamda sağ ve sol tekerlerin çapları ve teker merkezleri arasında mesafe istatistiksel metotlar ile bulunur. Bu çalışmanın çıktısı olan kalibrasyonu yapılmış kinematik model konum tahmini çalışmalarında kullanılabilir, ancak bu çalışmada bahsi geçen konum tahmini problemi hedeflenmemiştir, çalışma en iyi kinematik modelin hesaplandığı aşamada bitirilir. Kalibrasyonun yapılması için bir test düzeneği ve yazılım geliştirilir ve kamera ile mutlak pozisyon hesaplanır ve doğru olarak kabul edilir. Bu veri kullanılarak kalibrasyon yapılır.

1.2 Tez Çalışmasının Hedefi

Bu çalışmanın hedefi bir önceki kısımda bahsi geçen kinematik modelin değişkenleri olan sağ/sol teker çaplarının ve teker açıklığının test ortamında hassas bir şekilde bulunması olarak açıklanabilir. Tez, bunun için bir test düzeneğinin, bir hareketli robotun tasarlanması ve gerekli yazılımların geliştirilmesini kapsar. Alt bileşenleri; MATLAB tabanlı veri toplama yazılımı, usb kamera ve kamera kalibrasyon yazılımlarının geliştirilmesi, hareketli robot üzerinde ki elektronik aksamın Arduino ve kitlerinin kullanılarak yapılması, mikrodenetleyici üzerinde ki yazılımın geliştirilmesi ve en son dizüstü bilgisayarda çalışan MATLAB yazılımı ile hareketli robotun üzerinde ki Arduino tabanlı mikrodenetleyicinin Xbee kitini kullanarak haberleştirilmesini kapsar. Nihai amaç referans [7] tanımlanan ve simülasyonlar ile doğrulanan kalibrasyon metodunun gerçek test verisi ile doğrulanmasını içerir. Bu tez, elde edilen kalibrasyon değerlerinin, herhangi bir konum tahmini algoritma içerisinde kullanılmasını hedeflemez, ancak elde edilen en doğru kinematik model, kamera verileri ile doğrulanır.

2. LİTARATÜR TARAMASI

2.1 Giriş

Bu bölümde mobil robotların yörünge hatalarına neden olan sistematik hatalar (örneğin tekerlekler arası mesafe ve tekerlek yarıçapları arasındaki fark) ve sistematik olmayan hatalar (örneğin zeminin düzgün olmayışı) üzerine yapılan araştırmalar incelendi. Bu odometre hatalarının giderilmesi için yapılan çeşitli teorik çalışmalar, kalibrasyon yöntemleri hakkında çalışmalar özetlendi.

Bölüm 2.1 'de Literatür taramasının hangi konulara odaklandığından bahseden kısa bir giriş yapıldı. 2.2'de konuyla ilgili yapılan çalışmalar özetlendi. 2.3'de yapılan çalışmalardan esinlenerek bu tezde yapılacaklar detaylandı.

2.2 Literatür Özeti

Literatüre bakıldığında mobil robotların kinematik parametre tahmini hakkında birçok çalışma mevcuttur, bunlar genellikle parametre tahminindeki hataların azaltılması, parametre tahmininde yeni yöntemlerin kullanılması, odometre hata kalibrasyon metotların geliştirilmesi alanlarında yoğunlaşmıştır.

Kalibrasyon metodlarının geliştirilmesi yönünde yapılan çalışmalarda, Borenstein ve arkadaşları [8] mobil robotlardaki efektif tekerlek arası belirsizlik ve eşit olmayan tekerlek çapları nedeniyle oluşan hata kaynaklarını azaltmak için kalibrasyon yöntemi sunmuş bu yöntemle yapılan çalışmalarda odometre doğruluğunun arttırıldığı görülmüştür. Bu yöntemle karmaşık ve pahalı ekipmanlara gerek kalmadan en uygun sonuçlar elde edilmiştir. Bu çalışmadaki başka bir katkı ise tekerlekler arasındaki belirsizliğin ve eşit olmayan tekerlek çaplarının ölçümü ve doğrulanması olmuştur. Tekerlek çapı oranları arasındaki % 0.1 lik bir hatayı kolaylıkla belirleyebilmiştir.

Chiaverini ve arkadaşları [9] yeni oluşturdukları kalibrasyon tekniğiyle diferansiyel sürürlü robotların fiziksel odometre parametrelerini belirlemiş, iki çeşit parametre tahmin algoritması sunmuş ve bunları Khepera 2 mobil robotunda uygulamışlardır.

Diferansiyel sürürlü robot için algoritma üretmişler. Parametre tahmini için LSM (Least Square Method) metodu kullanmışlar, bu geliştirilen algoritmayı Khepera 2 mobil robotunda uygulamışlardır [10].

Başka çalışmalarında ise geliştirdikleri süzgeç ile mobil robotun yer tahminini yapmışlardır, bir kararlılık analizi ile de konum ve yön hatalarını sifira yaklaştırmışlardır. Bu tekniği Khepera 2 robotunda test etmişler ve Genişletilmiş Kalman Süzgeci (EKF)

yöntemiyle karşılaştırmışlar, ileri sürdükleri yöntemin Genişletilmiş Kalman Süzgecine basit bir alternatif olarak görülmüştür [11].

Woung Chun ve arkadaşları [12] , Odometre doğruluğunu arttırmak için yeni kalibrasyon şekli sunmuş; sunulan yöntem UMBmark yönteminde kalibrasyon denklemlerindeki yaklaşım hatalarını düşürmüştür. Kalibrasyon deneyinde tekerlek açıklığı ve tekerlek çap hatalarının birlikte düşünülmesini önermiştir.

Martinelli [13] diferansiyel sürürlü bir robotun sistematik olmayan hatalarını bulmak için teori sunmuş, bu stratejide robotun sadece başlangıç ve bitiş konumlarının ölçülmesi yeterli olmuş böylece sistematik olmayan hata parametreleri tahminini yapabilmişlerdir.

Kleeman ve arkadaşları [14], düşük maliyetli odometre sistemleri için hata modeli geliştirmiş bu sayede düşük maliyetli odometre sistemlerini yüksek gerçeklikte çalıştırmışlardır. Bu yöntemde oluşturulan hata kovaryansı kısa zaman aralıklarında kovaryans matrisinin aşamalı artırımına dayandırılmıştır.

Başka çalışmalarında iki gelişmiş sonar sistemi yardımıyla yeni odometre hata modeli oluşturulmuş, Kalman süzgeci SLAM yaklaşımıyla yüksek kaliteli sonar haritalar oluşturulmuştur [15].

Kuanghyun Yoo ve arkadaşları [16] araba benzeri tekerlekli robotlar için yeni kalibrasyon metodu önermiş robotun kinematik parametreleri bir çift test sonucunda başarıyla kalibre edilmiş. Kalibrasyonla belirleyici hataların azaltılmasından sonra Genişletilmiş Kalman Süzgeciyle yöntem daha da geliştirilmiştir.

Bakey ve arkadaşları [17] parametre hatalarının tahmini için adaptif tahmin yöntemi kullanmış, böylece her bir süzgeç hata parçasına bağlanmış. Her bir süzgeç dahili modellerini kullanarak algılayıcı okumasını kontrol etmiş. Gerçek değerle tahmin edilen değer arasındaki fark süzgecinin ne kadar iyi çalıştığını göstermiştir. Bu metod Pioneer 1 robotunda başarılı bir şekilde uygulanmış. Bu uygulama sonucunda düz lastik ve lastiğe cisim sıkışması hataları bu metotla tespit edilmiştir.

Didier Wolf ve arkadaşları [18] çoklu algılayıcıya sahip sistemler için otomatik kalibrasyon metodu tanımlamışlar. Bu metotta dışarıdan herhangi bir referans gereği duyulmadan, sistem üzerindeki çoklu algılayıcı verilerini kullanan fonksiyonlar tanımlamışlar. Bu yöntemin sağlıklı çalışması için her bir algılayıcının iyi kalibre edilmesi

gerektiğini vurgulamışlardır. Geliştirdikleri algoritmaları mobil robotun konum belirlemesi için uygulamışlar, uygulama sonucunda jiroskop hata katsayısını ve tekerlek açıklığı parametrelerinin nihai değerlerine, bu algoritmaları üç iterasyon yaparak ulaşımlardır.

M. Abdolmohammadi ve arkadaşları [19] çalışmalarında, diferansiyel sürüslü hareketli robotların sistematik hatalarını gidermek için yeni bir metod üzerinde durmuşlar ve bu metodu UMBmark yöntemiyle kıyaslamışlar Önerdikleri metotta, robotu düz bir çizgide doğrusal hareket ettirerek robot kalibrasyonun yapılacağını göstermişler. Bu yöntemin uygulanmasının daha basit ve daha az zaman aldığından dolayı UMBmark tekniği yerine geliştirdikleri yöntemin kullanılabileceğinden bahsetmişlerdir.

Wan Kyun Chung ve arkadaşları [20] sistematik hataların kalibrasyonu için bağlantı yinelenmeli öğrenme kontrolünü kullanarak yöntem geliştirmişler. Bu yöntemle hareketli robotun belirli bir yörünge takip etmesine gerek kalmadığını ve gelişi güzel seçilen bir yörüngeyle de kalibrasyonu yapılacağını göstermişler. Bahsettikleri bu yöntemin odometre hatalarını belirgin bir şekilde düşürdüğünü hem simülasyon yöntemiyle hem de deneysel çalışmalarda doğrulamışlar.

John James Collins ve arkadaşı [21] odometre hataların giderilmesi için yapay sinir ağları, genel yer tahmini modülü ve optimizasyon metotlarını uygulamışlar. Bu metotlardan en iyi çalışanın yapay sinir ağları metodu olduğunu göstermişlerdir. Bu yöntemi öne çıkaran kabiliyeti herhangi bir odometre benzetimini uyabilmesi ve yüksek hızda çalışması olmuştur. Yapay sinir ağları yöntemi, robot haritalandırılmasında kullanılmadan önce çalıştırılıp odometre hata modeli robota öğretilir. Bundan dolayı da bu yöntem iç ortam haritalandırmalarında oldukça kullanışlı bir yöntem olduğunu belirtmişlerdir.

Roland Siegwart ve arkadaşı [22] doğrudan bağlantılı kendi kendini kalibre edebilen otonom araç için problem analizi yapmışlar. Bu analizi yaparken sistemin kendi kendine kalibre edebilmesi için gerekli bilgi içermesi ve kalibrasyon doğruluğunu arttırmak için hangi yörünge için en iyi sonucu verdiği odaklanılmışlar. Bu kriterlere göre holonomik araçlarda yoğunlaşmışlar ve sonuçta en iyi sonucu veren ikinci dereceden diferansiyel denklem türetmişler.

Mariolino De Cecco [23] çalışmasında kendi kendini ayarlayarak kalibre edebilen tekniği geliştirmiş. Bu tekniği geliştirirken, içsel odometre pusulası olan küçük çaplı endüstri robotu kullanılmış. Geliştirdiği yöntemi kullanarak 25 metre uzunluğunda test yapmış, test

sonucunda pusuladan aldığı verilerle test ortamındaki üçlü kızılötesi vericilerden aldığı kesin konum verileri kıyasladığında çok az farkın oluştuğunu görmüştür.

Philip Chen ve arkadaşları [24] çalışmalarında özel süzgeç kullanarak uyarlanırlı hata tanımlama ve giderme sorunu üzerinde çalışmışlar. Algılayıcı sistemleri için benzetim tabanlı yaklaşımlar geliştirmişler. Bu süreçte iki özel süzgeç olan Kalman süzgeci ve olasılıklı M-algoritması üzerinde yoğunlaşmışlar. Bu iki yöntemin üzerinde karmaşıklık ve etkinlikleri yönünden kıyaslama yapılmış ve bu iki yöntem arasında olasılıklı M-algoritması daha az karmaşık olduğundan ve daha iyi sonuçlar verdiği için öne çıkmıştır.

2.3 Sonuç

Bu bölüm sonunda mobil robotların farklı kalibrasyon yöntemleri sunulmuş kalibrasyon yöntemlerinin basit olması, hata kaynaklarının azaltılması ve odometre doğruluğu artırılmalıdır. Odometre hatalarını oluşturan en belirgin parametreler: tekerlek çaplarındaki fark ve tekerlek açıklığındaki belirsizlik olduğu yapılan kalibrasyon çalışmalarında görülmüştür. Mobil robotların kalibrasyonu, robota belirli bir yörünge izlettirilerek enkoderlerden alınan odometre verileri kullanılarak yapılabildiği gibi, çoklu algılayıcıya sahip hareketli robotlar kalibrasyonlarını gelişi güzel hareketlerle kendiliğinden yapabildiği çalışmalarda görülmüştür. Parametre tahmininde en çok kullanılan yöntemin genişletilmiş Kalman süzgecidir. Yapılan çalışmalarda bu yöntem alternatif oluşturacak tahmin yöntemleri üzerinde çalışılmıştır.

3. PLATFORM HAREKET SİSTEMLERİ ve MOBİL ROBOT KİNEMATİĞİ

3.1 Giriş

Bu bölümde platform hareket sistemlerini ve mobil robot kinematığı açıklanmıştır. Platform hareket sistemleri, doğadaki canlıların hareket etmek için kullandıkları uzuvlardan esinlenerek tasarlanan bacaklı hareket sistemleri ve insan icadı olan tekerleğin farklı kombinasyonuyla oluşturulan tekerlekli hareket sistemleri çatısı altında incelenir. Seçilen hareket sistemi robot tasarımının ve robot kinematığının temelini oluşturur.

Bu hareket sistemlerini oluşturan temel unsurlar olan bacaklı hareket sistemleri bölüm 3.2.1'de açıklanmıştır. Bu hareket sistemi daha çok engebeli arazide kullanılmakta ve karmaşık mekanizmalardan oluşmaktadır. Bölüm 3.2.2'de tekerlekli hareket sisteminde bahsedilmiş bu sistemler daha düzgün alanlarda kullanılır ve yapıları basittir. Bölüm 3.2.2.1'de tekerlekli hareket sistemini oluşturan tekerlek yapılarından bahsedilmiştir bu yapılar konvansiyonel ve özel olarak sınıflandırılıp bu sınıflara dahil olan tekerlek çeşitleri açıklanmıştır. Bölüm 3.2.2.2' de sürüş çeşitlerinin nasıl hareket ettikleri tekerlek konfigürasyonlarından bahsedilmiştir. Bölüm 3.2.2.3'te sürüş çeşitlerinin manevra yetenekleri üzerinde durulmuş ve karşılaştırılmışlardır. Bölüm 3.3'te mobil robot kinematığı ile ilgili özet verildi. Bölüm 3.3.1'de mobil robotun tekerlek değişkenleriyle mobil robotun konum değişikliği arasında denklemsel ilişki kuruldu ileri ve ters kinematik modeller oluşturuldu. Bölüm 3.3.2'de holonomik olmayan kısıtlamalardan bahsedildi. Bölüm 3.3.3'te holonomik olmayan robotlara örnekler verildi. Bölüm 3.3.4'te tek tekerleğin kinematik analizi yapıldı. Bölüm 3.3.5'te diferansiyel sürürlü bir robotun kinematığı çıkarıldı. Bölüm 3.4'de platform tasarımda kullanılacak sürüş yöntemleri avantajları ve dezavantajlarıyla kıyaslanarak yapılacak olan hareketli platformda kullanılacak sürüş sistemi belirlenmiştir.

3.2 Platform Hareket Sistemleri

Hareket sistemleri, robotların karakteristiğini belirleyen en önemli konudur. Seçilen hareket sistemi çerçevesinde robot tasarımını şekillendirir.

Doğadaki canlıların hareketini sağlayan uzuvlardan esinlenerek, iki bacaklı, dört bacaklı ve altı bacaklı gibi hareket mekanizmaları tasarlanmıştır. Bunun yanı sıra tamamen insan ürünü olan tekerlekli hareket sistemleri de geliştirilmiştir.

3.2.1 Bacaklı Hareket

Bacaklı hareket, zeminle robot arasındaki noktasal temaslarla hareket etmektedir. Bu hareket sisteminin en önemli avantajı engebeli araziye uyum kabiliyeti ve manevra yeteneğidir. Engebeli arazilerde, platformun gövdesi zeminle temas etmediği ve araziyle çok noktalı temas kurduğu sürece zeminin sürtünme kalitesi çok önem arz etmez.

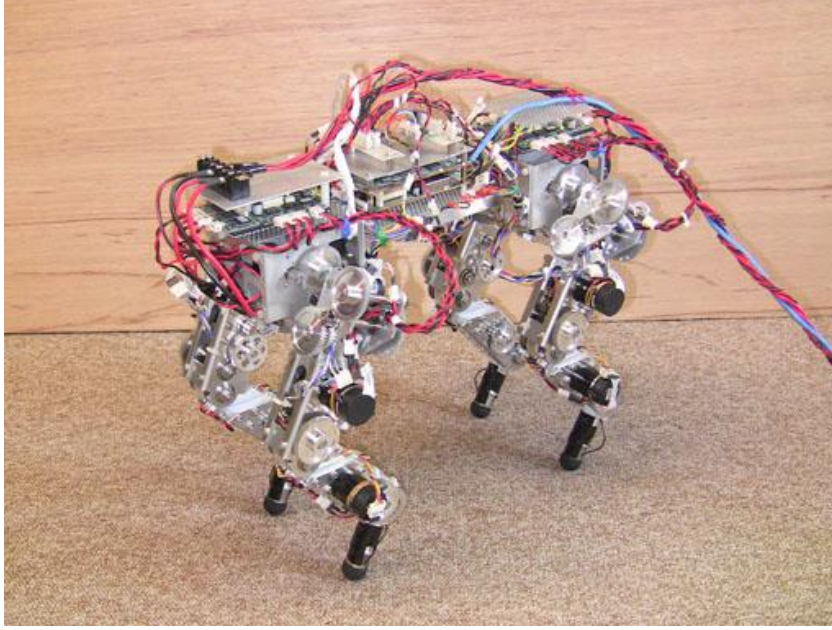
Bacaklı hareketin en büyük dezavantajı yüksek hareket mekanizmalarındaki karışıklık ve bu mekanizmaları harekete geçirmek için harcadığı yüksek enerjidir. Bacak mekanizmaları robotu kaldıracak ve hareket ettirebilecek dayanımda tasarlanır. Bacaklı hareketi, iki bacaklı hareket ve çok bacaklı hareket olarak iki guruba ayırabiliriz.

İki bacağının üzerinde durmaya dayanır. İnsansı robotlar iki bacaklı robotlardır, görünüşleri insan vücudu temel alınarak baş, gövde, bacaklar ve kollar gibi parçalardan oluşmaktadırlar. İki bacaklı hareket sistemlerinde, ayağı oluşturan mekanizmanın kontrol sistemleri oldukça karışık yapılara sahiptir. İki bacaklı harekette bacak mekanizmaları temas yüzeyinden kalkar ve sonra tekrar yerle temas eder. Platform hareket edebilmek için bacak mekanizmalarını yerle olan temas yüzeyinden kaldırır ve tekrar temas yüzeyine basar aynı zamanda platform gövdesini harekete göre dengeleyerek yürüyüşünü gerçekleştirir.

Bacaklı platformlar durağan halde olabilmeleri için en az 3 ayağının yüzeyle teması gerekir, iki bacaklı ve bir bacaklı robotların statik dengeleri yoktur bu nedenle dengelerini gövde hareketleri ya da sürekli hareket ederek sağlarlar.

Çok bacaklı platform çalışmaları önceleri düzgün zeminde hareket, yumuşak zeminlerde hareket edebilme ve basit engelleri geçme çalışmalarına için tasarlanmışlar fakat günümüzde çok bacaklı platformlar engebeli araziler, moloz yığınlarının üzerinde, depremde zarar görmüş zeminlerde ve orman yüzeyi gibi çok zorlu arazilerde hareket edebilmesi üzerine çalışmalar gerçekleştiriliyor. Bu platformların böyle yüzeylerde hareket edebilmesi için yeni tepki mekanizmaları ve denge tahmin sistemleri gibi ilave özellikler kullanılmaktadır.

Bu tür sisteme sahip olan Kotetsu robotu Şekil 3.1’de gösterilmiştir. Bacak mekanizmalarındaki yüklü ve yüksüz durumları temel alan durum modülasyonu yardımıyla engebeli arazilerde uyumlu yürüyüş yapabilme yeteneğine sahiptir.



Şekil 3.1. : Dört bacaklı koketsu robotu. [25]

3.2.2 Tekerlekli Hareket

Tekerlekli hareketli platformların (THP) hareket kabiliyetleri platformda kullanılan tekerlek tiplerine ve sürüş yöntemine dayanır. Tekerlekli hareketli platformları müze, depo, süpermarket ve hastane gibi düzgün zeminlerde çalışırlar, Engeli arazilerde çalışan tekerlekli hareketli platformlar düzgün zeminde çalışan platformlara göre fazladan süspansiyon sistemi içerir. Süspansiyon sistemi platformu hareket ettiren tekerleklerin zeminle temasını korur. Tekerlekli hareket platformları holonomik ve holonomik olmayan olarak sınıflandırılabilirler. Holonomik olmayan platformlar üç serbestlik derecesinden (ileri hareket, yatay hareket ve dönme (3)) az serbestlik derecesine sahiptirler (ileri hareket edebilme ve dönme serbestlikleri (2)). Bu platformların avantajı ise kolay imal edilmesi ve maliyetlerinin düşük olmasıdır. Holonomik platformlar ise üç serbestlik derecesiyle hareket ederler bu özellikleri sebebiyle dar alanlarda yüksek manevra kabiliyetleriyle çalışırlar. Tekerlekli platformlarda durağan denge üç tekerlek kullanımıyla doğal olarak sağlanabilir. Tekerlekli hareket platformları tasarımında kullanılan farklı tekerlek çeşitleri ve çeşitli yapılandırmalar platformun çekişini, dönme yeteneğini, kararlılığını ve kontrolünü etkiler.

3.2.2.1 Tekerlek Tipleri

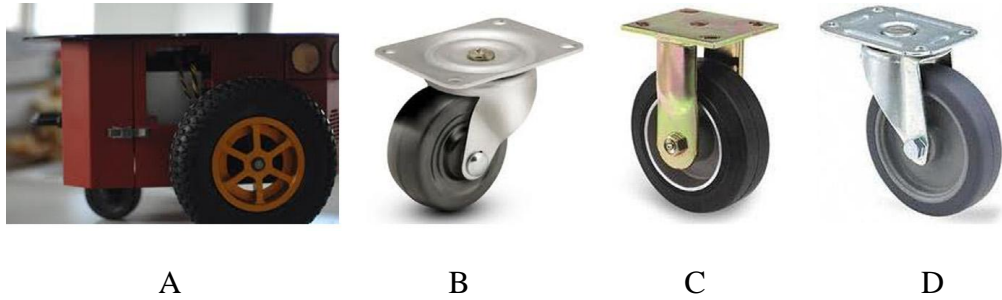
Tekerlekli hareketli platformlarda kullanılan tekerlek çeşitleri konvansiyonel tekerlekler ve özel tekerlekler olmak üzere iki başlık altında incelenebilirler.

Konvansiyonel tekerlekler, tahrikli sabit tekerlekler, kaster tekerlekler ve tahrikli yönlendirme tekerlek olarak sınıflandırılabilir. Tahrikli sabit tekerlekler platform üzerine sabitlenen ve bir motor tarafından sürülen tekerleklerdir, tekerlek dönüş eksenleri sabittir (Şekil 3.2A)

Kaster tekerlekler (Şekil 3.2B) bir motor tarafından tahrik edilmemiş tekerleklerdir, bu tekerlekler tekerlek dönüş eksenine ve tekerleğin zeminle temas ettiği noktadan yönelme ekseninde serbest olarak dönebilirler.

Tahrikli ve yönlendirilmeli tekerlekler hem dönme ekseninde dönebilmeleri hem de yönelme ekseninde dönebilmelerini sağlayan sürüş motorlarına sahiptirler. Yönelme eksenleri dönme eksenine göre aynı hizada ve ya farklı mesafelerde olabilir. (Şekil 3.2C ve Şekil 3.2D).

Konvansiyonel tekerlekler özel tekerleklere göre daha fazla yük taşıyabilir ve yüzey pürüzlüğünden daha az etkilenirler.



Şekil 3.2.: Sabit tekerlek (A) , kaster tekerlek (B) , tahrikli yönlendirilmeli eş merkezli tekerlek (C) , tahrikli yönlendirilmeli kaçık merkezli tekerlek (D). [26]

Özel tekerlekler, bir yönde etkin çekiş gerçekleştirecek diğer bir yönde ise etkisiz hareket edebilecek şekilde tasarlanır.

Bu tekerlekleri universal tekerlekler, mekanum tekerlekler ve küresel tekerlekler olarak üç sınıfta toplayabiliriz.

Üniversal tekerlekler, dönme esnasında etkin ve etkisiz hareket çeşitliliği sağlar. Bu tekerleklerin dış çapı etrafında küçük rulolar bulunur, bu rulolar tekerlek dönüş eksenine dikey olacak şekilde konumlandırılmıştır. Bu tekerlekler teker dönüş eksenlerinde etkin

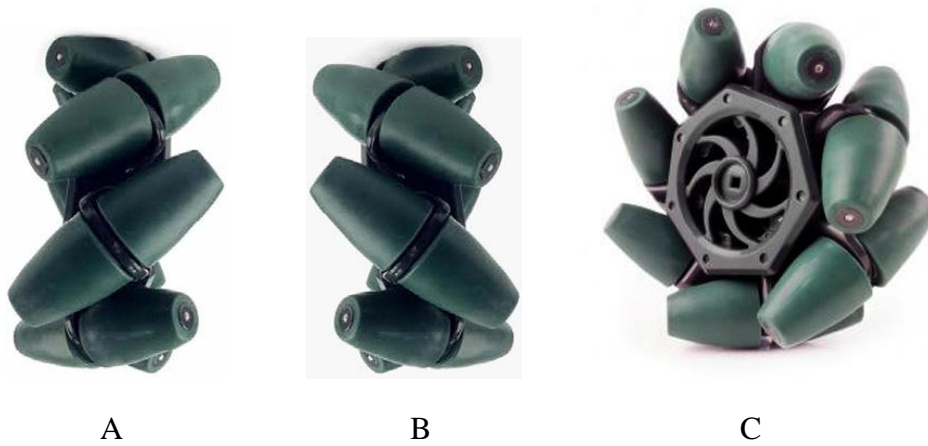
hareket sağlarken, tekerlek dönüş eksenine paralel yönde yuvarlanarak etkisiz hareket edebilirler. (Şekil 3.3.)



Şekil 3.3: Üniversal çok yönlü tekerlek [27]

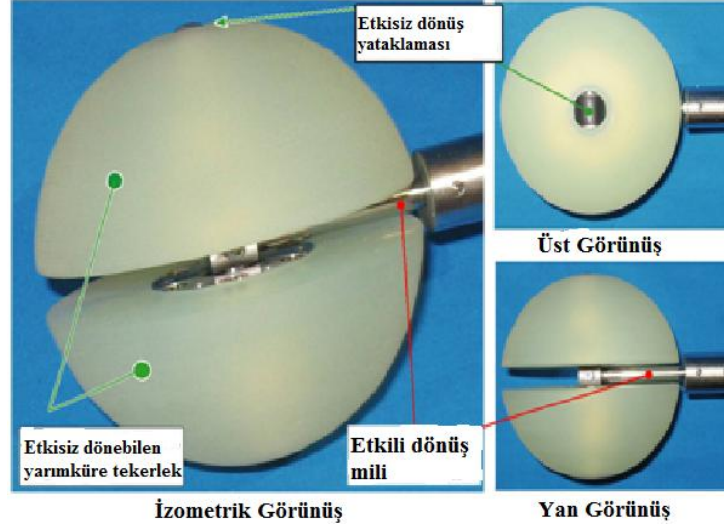
Mekanum tekerlekler, üniversal tekerleklere benzerler, fakat dış çapı etrafındaki rulolar üniversal tekerleklerdeki gibi 90° derece değil de açılı olarak konumlandırılır. (Şekil 3.4) genellikle 45° açılı olarak konulur.

Bu tip tekerleklerde, kendi dönüş ekseninde döndüğünde zemin temas noktasında dönüş eksenine dik bir kuvvet oluşur. Bu kuvvet tekerleğin dış çapında açılı olarak konumlandırılan rulolarda rulo eksenine paralel ve dik olan bileşenlerine ayrılır. Bu kuvvet bileşenleri tekerleğin dönüş eksenine açılı bir eksenle hareket etmesini sağlar. Başka bir deyişle konvansiyonel tekerlekler döndüğünde ileri ya da geri doğrusal hareket ederler. Fakat mekanum tekerlekler döndüğünde köşegensel olarak ileri ve geri giderler.



Şekil 3.4 : 45° açılı mekanum tekerlek (A), -45° açılı mekanum tekerlek (B) ,genel görünüş (C). [26]

Küresel tekerlekler, çok yönlülükleriyle diğer üniversal ve mekanum tekerleklere benzerler. Bu tekerleklere özel millerle hareket aktarımı yapılır. Mil ekseninin dönüş eksenine dik eksende etkin hareket gerçekleştirirler, paralel yönde ise etkisiz hareket gerçekleştirirler. Zor yapılarından dolayı uygulamalarda nadiren kullanılırlar (Şekil 3.5.).



Şekil 3.5: Küresel tekerlek uygulaması. [28]

3.2.2.2 Sürüş Çeşitleri

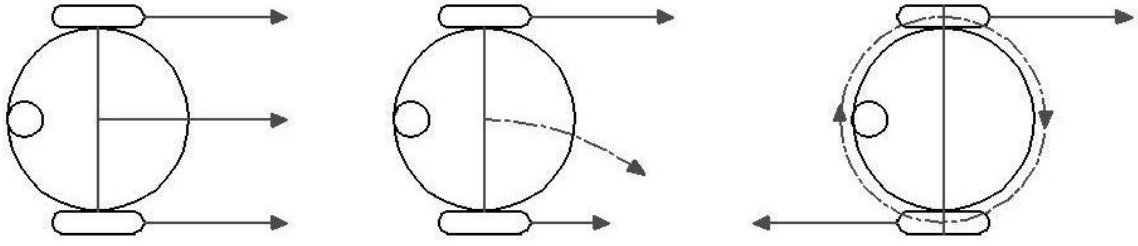
Tekerlekli mobil Robotların sürüş çeşitleri şu şekilde sınıflandırılabilir

- Diferansiyel sürüş
- Üç Tekerlekli sürüş
- Çok yönlü sürüş
- Eşzamanlı sürüş
- Ackerman yönlendirmeli sürüş
- Kızak yönlendirmeli sürüş

Diferansiyel Sürüş: Bu sürüşte platformda sağ ve sol yanlarına monte edilmiş tahrikli sabit tekerleklerden oluşmaktadır. Bu iki tekerlek birbirinden bağımsız olarak sürülür. Platformun dengesini ve kararlılığını sağlamak için bir ve ya iki adet kaster tekerlek kullanılır.

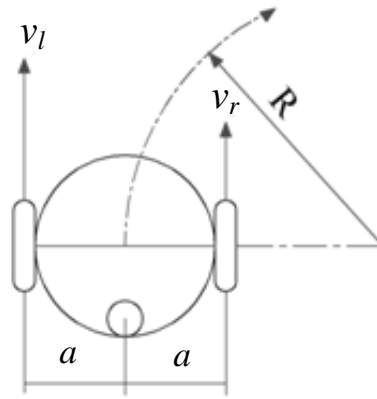
Diferansiyel sürüş ekseninin dönüşüne ihtiyaç olmadığı için basit mekanik sürüştür. Eğer tekerlekler aynı hızlarda dönerlerse robot düz ileri ve geri hareket eder. Eğer bir tekerlek diğerinden daha hızlı dönüyorsa robot kavisli bir yol izler, bu yol tekerlek hızlarından dolayı oluşan çemberin bir yayıdır. Eğer her iki tekerlek aynı hızda fakat karşı yönlere

doğru dönüyorsa robot sürüş tekerleklerinin orta noktasında döner Şekil 3.6'da bu hareket resmedilmiştir.



Şekil 3.6 : Sol ve sağ tekerlek doğrusal hızlarına göre robotun hareketi.

Platformun anlık eğrilik merkezi (AEM) tekerlek eksenlerinin geçtiği noktadan uzanır. Anlık Eğrilik Merkezinin daire merkezi iki tekerin hızlarına dayanır.(Şekil 3.7)



Şekil 3.7 :Sol ve sağ teker doğrusal hızlarına göre robotun hareketi.

$$(V_l - V_r)/2a = V_r/(R - a) \quad (3.1)$$

Böylece

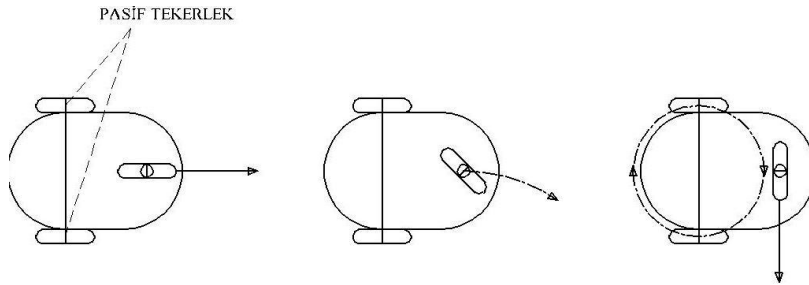
$$R = a(V_l + V_r)/(V_l - V_r) ; \quad V_l \geq V_r \quad (3.2)$$

$V_l = V_r$ olduğu zaman $R = \infty$ olur ve doğrusal hareket eder.

$V_l = -V_r$ olduğunda $R = 0$ olur ve robot tekerlek eksenlerinin orta noktasında dönüş yapmaktadır.

Üçlü Tekerlekli Sürüş: Bu sürüş hem sürülebilir hem de yönlendirilebilir ön tekerleğe ve arkada bulunan serbest hareket edebilen iki tekerlek bulunur, böylece platform, zeminle üç ayrı noktada temastadır. Platform düz hareket etmesi için tekerlek sürüş doğrultusunda orta durumda istenilen hızda sürülmelidir.

Ön tekerlek açılı ise araç kavisli bir yol izler. Eğer ön tekerlek sürüş yönüne göre doksan derece konumlanırsa, araç arka tekerleklerin orta noktasını merkez kabul edilen dairesel bir yol izler. Holonomik olmayan (diferansiyel sürüş veya üçlü sürüş) robotlar paralel olarak park edemezler, ileri ve geri bir çok manevrayla park edebilir. Şekil 3.8’de üç tekerlekli sürüşün temel hareketleri gösterilmiştir.



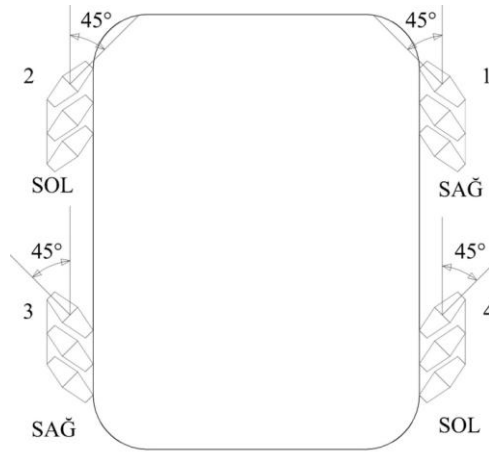
Şekil 3.8 üç tekerlekli mobil robot hareket şekli.

Çok Yönlü Sürüş: Şekil 3.9’ da görüldüğü gibi bu sürüşte üç dört veya daha fazla çok yönlü tekerlek kullanılabilir. Bu platformlarda mekanum tekerlekler kullanılır. Şekil 3.3 gösterilen universal çok yönlü tekerlektir. Tekerleğin çapı etrafında düzgün şekilde sıralanmış pasif rulolar vardır. Şekil 3.4 gösterilen mekanum tekerleklerdir. Tekerlek çapında açısız olarak konumlandırılmış pasif hareket ruloları vardır. Yapılandırma biçimleri Şekil 3.9A ve Şekil 3.9B’de gösterilmiştir.

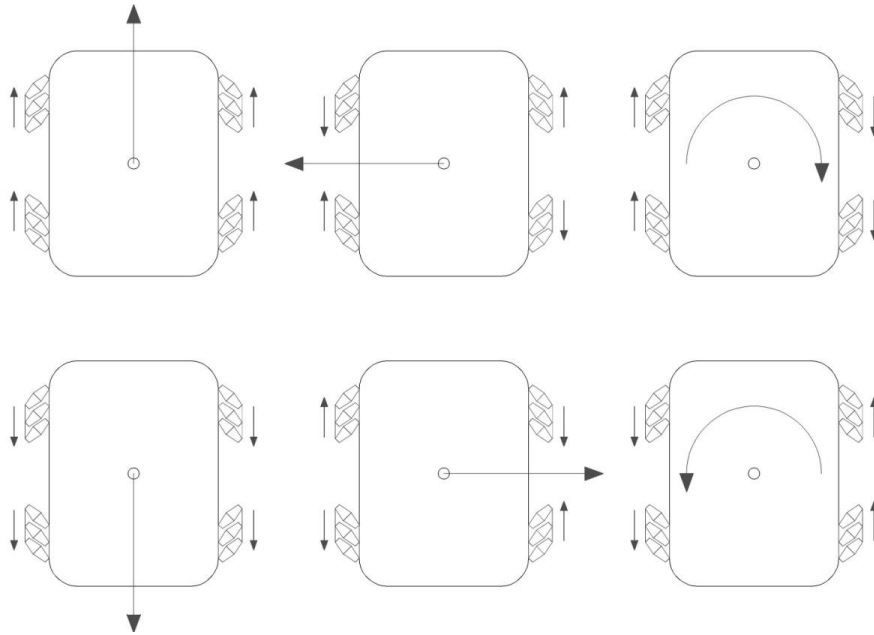
Şekil 3.10’te dört tekerlekli sistemde, iki tekerlek sol yönlü tekerlek ve diğer ikisi sağ yönlü olarak adlandırılır. Sol yönlü tekerleklerde rulolar 45° açılı, sağ yönlü tekerleklerde ise -45° derecede açılıdır. Böylece dört tekerlekli çok yönlü tekerlekli hareketli platform çeşidi Şekil 3.10’da gösterilmiştir.



(A) B
 Şekil 3.9: Çok yönlü platformlar Üç tekerli 90° rulolu (A), Dört tekerli ve 45° rulolu (B) .
 [29-30]



Şekil 3.10 : Dört mekanum tekerlekli mobil robotun standart kurulumu.



Şekil 3.11 : Dört mekanum tekerlekli mobil robotun altı temel hareketi.

Şekil 3.11’te 4 tekerlekli çok yönlü robotun temel hareketlerini gösterir. Bunlar ileri hareket, sola kayma, saat yönüne dönüş, geri hareket, sağa kayma ve saat yönüne ters hareket. Araçların sağ ve sol yanında bulunan oklar uygun tekerleklerinin hareket yönlerini gösterir.

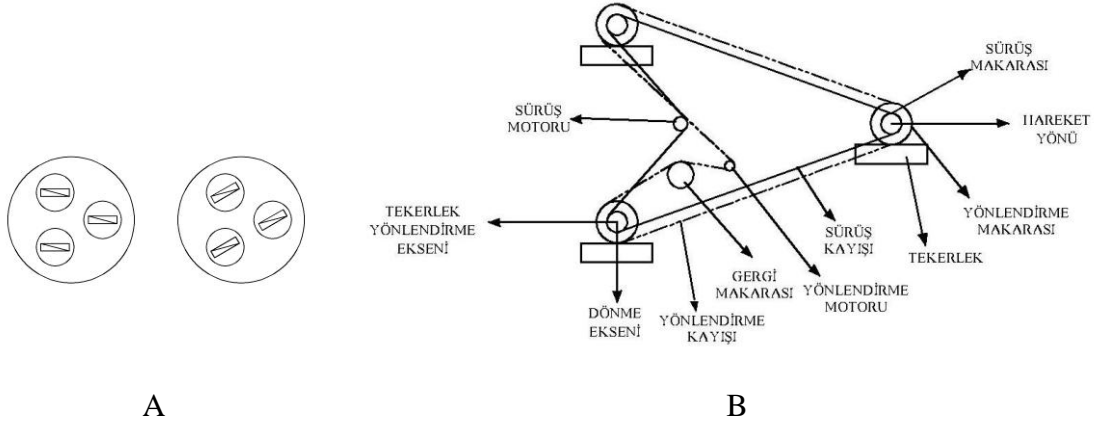
İleri hareket için bütün tekerlekler ileri hareket eder. Sola kayma hareketi için 1 ve 3 numaralı tekerlekler ileri 2 ve 4 numaralı tekerlekler geri hareket eder. Hareketler Şekil 3.11’de gösterilmiştir. Bu şekilde tekerlek aynı hızda dönmektedir. Tekerleklerinin hız büyüklüklerinin değiştirilmesiyle platformun hangi yöne hareket edebileceği anlaşılabilir.

Şekil 3.11 teki bütün hareketler tekerlek kuvvet veya hız bileşenleri kullanılarak açıklanabilir. Örneğin, sol ve sağ tekerleklerdeki simetriden dolayı, bütün tekerlekler ileri dönerse 4 adet ileri vektör noktası oluşur ve buna ek olarak 4 adet yanal vektör noktası oluşur. İki sağ ve iki sol vektörler birbirini yok eder. Böylece platform ileri hareket eder.

Eş zamanlı sürüş: Bu sürüş tipinde 3 veya daha fazla tekerler mekaniksel olarak bağlanmıştır böylece hepsi aynı hızda aynı yöne dönebilirler. Dönüş esnasında tekerleklerin dönüş eksenleri bir uyum içerisinde dönebilir. Bu mekanik direksiyon senkronizasyonu zincir, kayış veya dişli sistemleri gibi birkaç yolla sağlanabilir. Eş zamanlı sürüş sistemi tek sürüş ve direksiyon tekerlekli sistemin bir uzantısıdır. Fakat hala 2 serbestlik derecesine sahiptir.

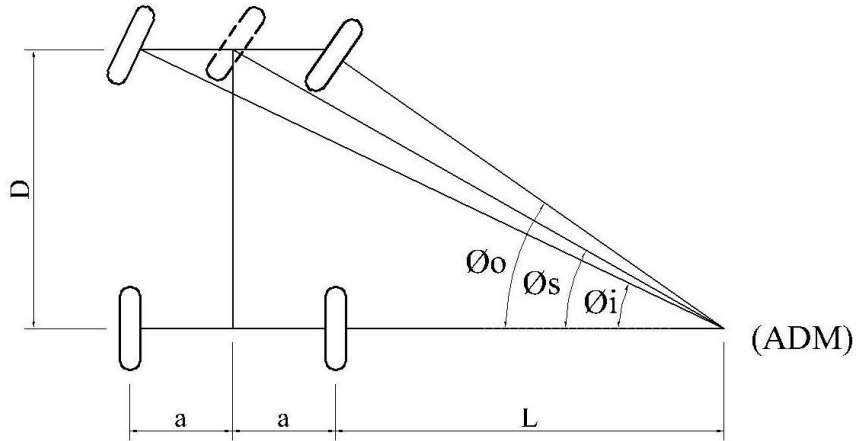
Eş zamanlı sürüşe sahip platformlar holonomiktir ve istenilen her hangi bir yöne hareket edebilir. Bununla birlikte aynı zamanda sürüş ve dönüş yapamazlar, ileri giderken yan yönde hareket yapabilmesi için platform önce durmak sonrada tekerlek yönlerini ayarlamalıdır. Şekil 3.12A üç tekerlekli eş zamanlı sürüşlü bir robotun resimsel olarak hareketini ve dönüşünü gösterir. Robot önce tekerleklerin yönü yatay olduğundan dolayı yatay hareket eder, robot farklı bir yönde hareketine devam edeceği noktada durur ve hareket etmek istediği yöne doğru tekerleklerinin açısını değiştirir. Robot tekerlek yönlendirmesini tamamladıktan sonra istenilen yönde hareketini devam ettirir.

Zincir ve kayış temelli eş zamanlı sürüş daha az doğrulukta direksiyona neden olur, bu problem dişli sistemler kullanıldığında oluşmaz. Gerçekte bu sistemlerde 2 ayrı sürüş alt sistemi kullanılmalıdır. Bu sistemlerden birisi direksiyonu sürerken diğeri de tekerlekleri sürmektedir.(Şekil 3.12 B)



Şekil 3.12 : Eş zaman sürümlü mobil robot hareketi (A), sürüş ve yönlendirme kayış sistemi (B).

Ackerman Direksiyonlu Sürüş: Otomobillerde kullanılan standart direksiyon sürüşüdür. Arka sürüş tekerleklerinden ve önde bulunan direksiyon tekerleklerinin kombinasyonundan oluşmuştur, Ackerman direksiyonlu araçlar düz gidebilir, fakat hemen orada dönüş yapamazlar, dönmeleri için minimum bir yarıçap gerekir.



Şekil 3.13: Ackerman yönlendirmeli hareketli robotun dönüşü.

Ackerman direksiyonu dönüş esnasında bütün tekerlek eksenlerinin kesiştiği genel bir nokta olacak şekilde tasarlanır, böylece tekerlek hareketlerine neden olunabilecek geometrik hatalardan kaçınılmış olunur.

Kızaklamalı yönlendirmeli sürüş : Bu diferansiyel sürüşün özel bir uygulamasıdır. Buldozerler ve zırhlı araçlar bu çekiş sistemine sahiptir. Diferansiyel sürüşlerden farklı engebeli arazilerde manevra kabiliyetini artırır ve yüksek sürtünmeye sahiptir. Paletin zeminle olan çok noktalı temasından dolayı kızaklamalı direksiyon dönüş sistemine sahip

bir robotun etkili palet temas noktaları paletin izine uyar. Şekil 3.14 tipik paletli robot platformunu göstermektedir.



Şekil 3.14: Kızaklamalı yönlendirmeli mobil robot [26]

3.2.2.3 Tekerlekli Hareketli Platformların Manevra Yetenekleri

Tekerlekli hareketli platformların manevra yeteneği M_y hareketlilik derecesi ve yönlendirilebilme derecesi toplamı olarak tanımlanır. $M_y = D_h + D_y$, burada D_h hareketlilik derecesi, D_y yönlendirilebilme derecesidir.

Hareketlilik derecesi : hareketlilik derecesini, mobil robotun hareket sistemini oluşturan tekerlek konfigürasyonu ve tekerlek çeşitleri bağımsızlık sabitlerinin sayısını belirler. Örneğin konvansiyonel tekerlekler sadece dönüş yönünde hareket edebilir dönüş yönü ekseninde hareket edemezler fakat çok yönlü tekerler özel geometrilerinden dolayı hem dönüş yönünde hem de dönüş yönü ekseninde hareket edebilirler. Konvansiyonel tekerler dönüş eksenlerine paralel hareket yapamadıkları için anlık dönüş yarıçapları hesaplanırken dönüş merkezinin bu ekseninden geçtiği kabul edilir.

Diferansiyel sürürlü robotlarda bu durum (Şekil 3.7) üst üstedir, böylece bunlar bağımsız değildir, sadece bir bağımsız kinematik sabit vardır. Genel sıfır hareket çizgisinden herhangi bir nokta anlık dönüş yarıçapı noktasını oluşturabilir. Ackerman direksiyonlu sürüş sisteminde 4 konvansiyonel tekerlek vardır, fakat 2 adet kinematik sabit bulunur.(Şekil 3.13) arka iki tekerlek diferansiyel sürüşte olduğu gibi bir kinematik sabite sahip, öndeki iki tekerde aynı dönüş yarıçapı noktasından geçtiği için, bir kinematik sabite sahiptir.

Maksimum hareketlilik derecesi 3'tür bu durum tekerlekli hareketli platformun bütün tekerleklerinin çok yönlü olmasıyla sağlanır.

Genelde hareketlilik derecesi $D_h = 3 - N_c$ 'ye eşittir. N_c , bağımsız kısıtlama sayısıdır.

Yönlendirilme Derecesi : D_y Bağımsız kontrol edilebilen yönlendirilme parametre sayısına dayanır. Bu sayı $0 \leq D_y \leq 2$ arasında değişir. Direksiyon tekeri yoksa $D_y = 0$ 'dır. $D_y = 2$ olduğu durumda robotun sabitlenmiş standart tekerleği yoktur, bu durumda iki yarı yönlendirilebilir konvansiyonel tekerlek mevcuttur. $D_y = 2$ olduğu durumda robot anlık dönüş yarıçap merkezini gezindiği alan üzerindeki herhangi bir noktaya yerleştirebileceğidir.

Genel durum ise $D_y = 1$ robotta bir veya daha fazla konvansiyonel yönlendirilmiş teker vardır. M_y manevra yeteneği, D_h hareketlilik derecesi ve D_y yönlendirilme derecesi Çizelge 3.1'de bazı sürüş çeşitleri gösterilmiştir.

Çizelge 3.1: Çeşitli mobil robotun hareketlilik ve yönlendirme dereceleri [26]

Sürüş Çeşidi	D_h	D_y	M_y
Bisiklet	1	1	2
Diferansiyel sürüş	2	0	2
Eş zamanlı sürüş	1	1	2
Üç tekerlekli sürüş	1	1	2
Ackerman yönlendirmeli sürüş	1	1	2
Çok yönlü sürüş	3	0	3

3.3 Mobil Robot Kinematığı

Robotun tekerlek yapısı ve robot geometrisi robotun yapabileceği hareketleri ve yapamayacağı hareketleri belirler. Robotun geometrik yapısından gelen bu parametreler robot kinematığının temelini oluşturur. Yapılan kinematik çalışmalar robotun kontrolü, kararlılığı ve dinamiği için öncü bir çalışmadır.

3.3.1 İleri ve Ters Robot Kinematığı

Genel koordinat sisteminde w_1, w_2, \dots, w_n tekerlek değişkenleri olsun x_1, x_2, \dots, x_m genel durum değişkenleri olduğunu düşünelim. Bu değişkenler vektorel olarak şu şekilde tanımlanır.

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

Bilinen W vektörü (tekerlek değişkenleri) ile X (konum ve yön değişkenleri) vektörünün bulunmasına direk kinematik çözüm denir. $X \in R^m$ ve $W \in R^n$ boyutlu vektörler olması bu vektörlerin doğrusal olmayan bir fonksiyonla ilişkili olduğunu gösterir.

$$X = f(W) \quad (3.3)$$

Yukarıda gösterdiğimiz denklemin tersini alırsak, W vektöründen X vektörünü bulursak bu sürece de ters kinematik denilir.

$$W = f^{-1}(X) \quad (3.4)$$

Hareket değişkenlerinin hız vektörleri $\dot{W} = [\dot{w}_1, \dot{w}_2, \dots, \dot{w}_n]^T$, $\dot{X} = [\dot{x}_1, \dot{x}_2, \dots, \dot{x}_m]^T$ şeklinde ifade edilebilir. Bu hız vektörleri arasındaki bağlantı şu şekilde kurulabilir.

$$\dot{X} = J\dot{W} \quad (3.5)$$

$$J = \begin{bmatrix} \frac{\partial x_1}{\partial w_1} & \dots & \frac{\partial x_1}{\partial w_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial x_m}{\partial w_1} & \dots & \frac{\partial x_m}{\partial w_n} \end{bmatrix} = J_{ij} \quad (3.6)$$

Denklem 3.5 robotun ileri kinematiği olarak ifade edilir.

$$J_{ij} = \frac{\partial x_i}{\partial w_j} \text{ robotun Jakobian matrisidir}$$

Jakobian matrisi robotun tekerlek değişken hızlarıyla robotun yer değiştirme hızları arasındaki ilişkisini temsil eder.

Robotun ters kinematiği ise denklem 3.7 ile ifade edilebilir. J^{-1} jakobian ters matrisi ifade eder.

$$\dot{W} = J^{-1}\dot{X} \quad (3.7)$$

3.3.2 Holonomik Olmayan Kısıtlama

Çalışma ortamındaki bir robot ortamın sağladığı hareket çeşidi kadar doğrudan hareket üretiliyorsa bu robota holonomik olmayan robot denir. Doğrudan gerçekleştirilemediği duruma ise holonomik olmayan kısıtlama denir.

Örneğin düzlemsel ortamda hareket eden bir bisiklet ileri gidebilme ve dönebilme hareketlerini yapabilir, fakat teker dönüş eksenleri doğrultusunda hareket edemez bu yüzden tekerlek dönüş eksenine yönünde holonomik olmayan kısıtlama mevcuttur.

Holonomik olmayan kısıtlamalar içeren sistemlere örnek olarak tekerlekli mobil robotlar, insansız hava araçları ve otonom sualtı araçları verilebilir. Hareket kısıtlaması olmayan bir mobil robot holonomiktir. Özel tekerlek kullanan birçok robotun hareket kısıtlaması yoktur.

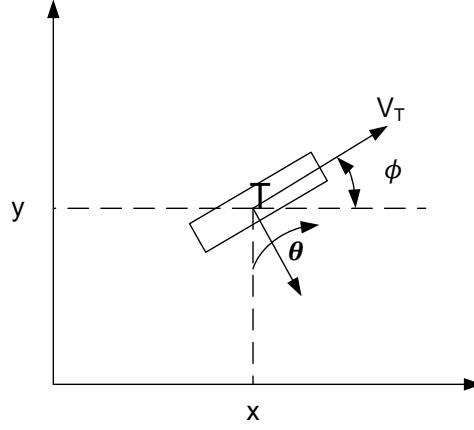
Holonomik olmayan kısıtlamalar çeşitli durumlarla oluşabilir, örneğin bir mobil robot k tane motora sahipse ve aynı zamanda n serbestlik derecesine sahipse, robot motorlarından k adet bağımsız hareket üretebilir, $n-k$ farkı bize robotta ne kadar holonomik olmayan kısıtlamaya sahip olduğunu gösterir.

3.3.3 Holonomik olmayan Robotlar

Holonomik olmayan hareketli robotlar, Tek tekerlekli sürüş, diferansiyel sürüslü hareketli robotlar, üç tekerlekli hareketli robot ve ackerman sürüslü hareketli robotlar olarak sınıflandırabiliriz. Bu çalışmada tek tekerlek ve diferansiyel sürüslü robotların kinematiği incelenmiştir.

3.3.4 Tek Tekerlek Kinematiği

Tek tekerlekli kinematik model birçok tekerlekli hareketli robotların kinematik modelinin temelini oluşturur. Tek tekerlek bir konvansiyonel tekerleğin yatay yüzeyde yuvarlanmasıdır. Tekerleğin dönmesi sonucu oluşacak hız vektörünü şu şekilde yazabiliriz $\dot{X} = [\dot{x}, \dot{y}, \dot{\phi}]^T$ Tekerleğin dönüşü ve zeminle temasından dolayı oluşan doğrusal hız $V_T = \dot{\theta}r$ ve dönüş eksenindeki açısal hızı $V_\phi = \dot{\phi}$ 'dir. Tekerleğin hız değişkenleri şu şekilde yazılabilir. $\dot{W} = [\dot{\theta}, \dot{\phi}]^T$



Şekil 3.15. : Tek tekerlek hareketi

Tekerleğin dönmesi nedeniyle oluşan genel hareket denklemleri şu şekilde yazılır.

$$\dot{x} = \dot{\theta}r \cos \phi \quad , \quad \dot{y} = \dot{\theta}r \sin \phi \quad , \quad \dot{\phi} = V_{\phi} \quad (3.8)$$

Denklem 3.5'te ki bağlantıyı uygularsak.

$$\dot{X} = J\dot{W}$$

$$[\dot{x}, \dot{y}, \dot{\phi}]^T = J[\dot{\theta}, \dot{\phi}]^T \quad (3.9)$$

Jakobian matrisini bulmak için denklem 3.6'yı kullanırız.

$$J = \begin{bmatrix} \frac{\partial \dot{x}}{\partial \dot{\theta}} & \frac{\partial \dot{x}}{\partial \dot{\phi}} \\ \frac{\partial \dot{y}}{\partial \dot{\theta}} & \frac{\partial \dot{y}}{\partial \dot{\phi}} \\ \frac{\partial \dot{\phi}}{\partial \dot{\theta}} & \frac{\partial \dot{\phi}}{\partial \dot{\phi}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{\theta}r \cos \phi}{\partial \dot{\theta}} & \frac{\partial \dot{\theta}r \cos \phi}{\partial \dot{\phi}} \\ \frac{\partial \dot{\theta}r \sin \phi}{\partial \dot{\theta}} & \frac{\partial \dot{\theta}r \sin \phi}{\partial \dot{\phi}} \\ \frac{\partial \dot{\phi}}{\partial \dot{\theta}} & \frac{\partial \dot{\phi}}{\partial \dot{\phi}} \end{bmatrix} \quad (3.10)$$

$$J = \begin{bmatrix} r \cos \phi & 0 \\ r \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \quad (3.11)$$

Kinematik model şu şekilde yazılabilir.

$$[\dot{x}, \dot{y}, \dot{\phi}]^T = \begin{bmatrix} r \cos \phi & 0 \\ r \sin \phi & 0 \\ 0 & 1 \end{bmatrix} [\dot{\theta}, \dot{\phi}]^T \quad (3.12)$$

3.3.5 Diferansiyel Sürürlü Robot Kinematığı

Bu tip robotların geometrik ve kinematik parametreleri Şekil 3.16'da gösterilmiştir. Bu robotların durum vektörleri ve hız vektörleri şöyledir.

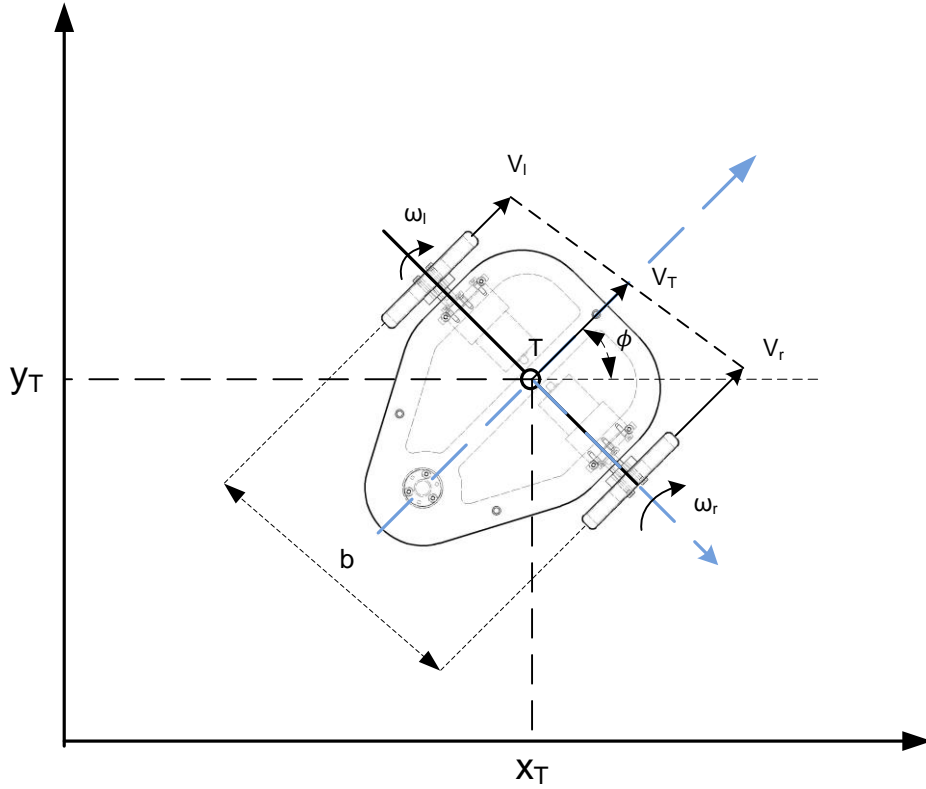
$$p = \begin{bmatrix} x_T \\ y_T \\ \phi \end{bmatrix} \quad \dot{p} = \begin{bmatrix} \dot{x}_T \\ \dot{y}_T \\ \dot{\phi} \end{bmatrix}$$

Tekerleklerin açısal konumları ve hızları $\{\theta_l \dot{\theta}_l\}$ $\{\theta_r \dot{\theta}_r\}$

V_l ve V_r sol ve sağ tekerin doğrusal hızlarıdır. V_T mobil robotun teker orta noktasının doğrusal hızıdır. Şekil 3.16'dan robotun tekerlek hızlarına göre, genel hızı ve açısal hızı şu şekilde yazılabilir.

$$V_T = \frac{1}{2}(V_r + V_l) \quad (3.13)$$

$$\dot{\phi} = \frac{(V_r - V_l)}{b} \quad (3.14)$$



Şekil 3.16: Diferansiyel sürürlü robotun kinematığı.

Kayma olmadığı farz edildiğinde $V_r = r\dot{\theta}_r$ ve $V_l = r\dot{\theta}_l$ elde edilir. Diferansiyel sürürlü robotun genel hız durumunu aşağıdaki gibi tanımlanır.

$$\dot{x}_T = \frac{r \cos \phi}{2} (\dot{\theta}_r + \dot{\theta}_l) \quad (3.15)$$

$$\dot{y}_T = \frac{r \sin \phi}{2} (\dot{\theta}_r + \dot{\theta}_l) \quad (3.16)$$

$$\dot{\phi} = \frac{r}{b} (\dot{\theta}_r - \dot{\theta}_l) \quad (3.17)$$

Denklem 3.5' teki bağlantıyı uygularsak.

$$\dot{X} = J\dot{W}$$

$$[\dot{x}_T \quad \dot{y}_T \quad \dot{\phi}]^T = J[\dot{\theta}_r \quad \dot{\theta}_l]^T \quad (3.18)$$

Jakobian matrisini bulmak için denklem 3.8'i kullanırız.

$$J = \begin{bmatrix} \frac{\partial \dot{x}_T}{\partial \dot{\theta}_r} & \frac{\partial \dot{x}_T}{\partial \dot{\theta}_l} \\ \frac{\partial \dot{y}_T}{\partial \dot{\theta}_r} & \frac{\partial \dot{y}_T}{\partial \dot{\theta}_l} \\ \frac{\partial \dot{\phi}}{\partial \dot{\theta}_r} & \frac{\partial \dot{\phi}}{\partial \dot{\theta}_l} \end{bmatrix} = \begin{bmatrix} \frac{\partial(\frac{r \cos \phi}{2}(\dot{\theta}_r + \dot{\theta}_l))}{\partial \dot{\theta}_r} & \frac{\partial(\frac{r \cos \phi}{2}(\dot{\theta}_r + \dot{\theta}_l))}{\partial \dot{\theta}_l} \\ \frac{\partial(\frac{r \sin \phi}{2}(\dot{\theta}_r + \dot{\theta}_l))}{\partial \dot{\theta}_r} & \frac{\partial(\frac{r \sin \phi}{2}(\dot{\theta}_r + \dot{\theta}_l))}{\partial \dot{\theta}_l} \\ \frac{\partial(\frac{r}{b}(\dot{\theta}_r - \dot{\theta}_l))}{\partial \dot{\theta}_r} & \frac{\partial(\frac{r}{b}(\dot{\theta}_r - \dot{\theta}_l))}{\partial \dot{\theta}_l} \end{bmatrix} \quad (3.19)$$

ve robotun Jakobianı denklem 3.19'de ki gibi bulunur.

$$J = \begin{bmatrix} \frac{r}{2} \cos \phi & \frac{r}{2} \cos \phi \\ \frac{r}{2} \sin \phi & \frac{r}{2} \sin \phi \\ \frac{r}{b} & \frac{-r}{b} \end{bmatrix} \quad (3.20)$$

İleri kinematik model şu şekilde yazılabilir.

$$[\dot{x}_T \quad \dot{y}_T \quad \dot{\phi}]^T = \begin{bmatrix} \frac{r}{2} \cos \phi & \frac{r}{2} \cos \phi \\ \frac{r}{2} \sin \phi & \frac{r}{2} \sin \phi \\ \frac{r}{b} & \frac{-r}{b} \end{bmatrix} [\dot{\theta}_r \quad \dot{\theta}_l]^T \quad (3.21)$$

3.4 Sonuç

Bu bölümde hareketli platformlarda uygulanan çeşitli tekerlekleri ve bu tekerleklerin kombinasyonundan oluşturulan sürüş çeşitlerini incelenmiştir. Bu sürüş çeşitlerinin manevra yetenekleri hakkında bilgi edinildi. Çok iyi manevra kabiliyetine sahip olan çok yönlü sürüşte özel tekerleklerin kullanılması ve bu tekerlekler pahalı ve zor bulunur. Ackerman sürüş yönteminde dört tekerlek kullanılması ve yönlendirme ve arka tekerleklerin hareketi için mekanizmalar kullanılır. Eş zamanlı sürüşte tekerleklerin hem

yönünü hem de hareketini sağlayan kayış kasnak sistemlerinin kullanılır ve bu sistemler karmaşıktır. Üçtekerli sürüşte ön tekerin hem yönlendirilebilir hem de tahrikli olması ön tekerleğe özel yataklı bir tasarım gerektirir. Yukarıda bahsedilen sürüş sistemlerinin dezavantajları, yapılacak olan hareketli platformda daha basit ve az maliyetli sistemlere yönlendirir. Bu basit sistem diferansiyel sürüş sistemidir. Bu sistemde iki tahrikli sabit teker doğrudan şasiye bağlanır, yönlendirme için fazladan mekanizma veya özel yatak tasarımı gerekmez. Bu nedenle hareketli platformun sürüş sistemin diferansiyel sürüş sistemi olacağı ortaya çıkarıldı. Mobil robot kinematiği bölümünde kinematik modeli oluşturuldu.

4.PLATFORM TASARIMI

4.1. Giriş

Platform hareket sistemlerini incelediğimizde yapımı kolay ve maliyeti düşük olduğundan dolayı diferansiyel sürürlü robot tasarlanması gerektiği kararlaştırıldı. Diferansiyel sürürlü robotlar sağ ve sol yanda ayrı motorları olan mobil araçlardır. Sağ ve sol yanda bulunan motorlar birbirinden bağımsız ve farklı hızlarda çalışabilmektedir. Diferansiyel sürüşün genel yapılandırılması genelde yanlarda iki sürüş motoru kullanılır, bu motorlarla tekerlekler bağlanır ve robotun doğrusal ve dönüş hareketi gerçekleştirilir, robot platformunun dengeli halde durması için de kaster tekerlekleri kullanılır. Eğer motora bağlı olan tekerleklerin açısal hızları aynı ise robot herhangi bir dönüş hareketi yapmadan ileriye ve ya geriye doğrusal hareket gerçekleştirir. Motora bağlı olan tekerleklerdeki açısal hızların farklı olması durumunda, açısal hızı düşük olan tekerleğe doğru robot yön değiştirerek dönme hareketi yapar. Bir tekerleğin döndüğü diğer tekerleğin sabit durduğu durumlarda ise dönüş eksenini sabit duran tekerlek olacak şekilde robot dönüş hareketini gerçekleştirir.

Bölüm 4.1’de tasarlanacak hareketli platformun seçimi ve genel özelliklerinden bahsedilen kısa bir giriş yapıldı, bölüm 4.2’de diferansiyel sürürlü hareketli platformun diğer sürüş yöntemlerine göre avantajlarından bahsedildi. Bölüm 4.2.1’de platform tasarımının nasıl yapıldığı hangi parçalardan oluştuğu parçaların nasıl imal edildiği ve parçaların birbirine nasıl monte edildiği açıklandı. Bölüm 4.2.2 de platform donanımının seçilmesine etken olan durumlardan bahsedildi, 4.2.2.1’den 4.2.2.7’ye kadar sırasıyla robot donanımını oluşturan motor, mikrokontrolcü, motor sürücüsü, batarya, Xbee kablosuz iletişim modülü, tekerlekler ve kaster tekerlek özelliklerinden bahsedildi. 4.2.3 bölümünde bu çalışmada kullanılan algılayıcılar hakkında kısa bir giriş yapıldı. 4.3.2.1 bölümünde odometre verilerinin oluşturulmasında kullanılan enkoderler hakkında genel bilgi verildi. 4.3.2.2 bölümünde mutlak konum ve yön ölçmede kullanılacak olan kamera ile ilgili bilgiler verildi.

4.2. Diferansiyel Sürürlü Hareketli Platform Tasarımı

Diferansiyel sürürlü robotların kararlılığı, manevra yeteneklerinin iyi olması ve kontrol edilebilirliği kolay olmasından dolayı tercih edilen platformlardır.

Maliyet yönünden de incelendiğinde diferansiyel sürürlü robotlar, diđer farklı sürüş sistemine sahip (Eşzamanlı sürüş, Dört teker sürüş) robotlara nazaran daha az maliyete mal olmaktadır.

Literatür çalışması sonucunda robot kinematik parametre tahmini için kullanılan bir çok yöntemin daha çok diferansiyel sürürlü robotlarda kullanılması, kapalı ortam çalışmalarında genellikle diferansiyel sürürlü robotlarla uygulamalar yapılması, bu çalışmamızda kullanacağımız hareketli platformun diferansiyel sürürlü olması konusunda bizi yönlendirmiştir.

4.2.1 Platform Tasarımı

Platform tasarımı seçilen sürüş sistemi temel alınarak gerçekleştirildi. Seçilen diferansiyel sürüş sisteminde tahrik edilen iki adet standart tekerlek ve bir adet dengeyi sağlamak için kaster tekerlek kullanılması düşünüldü. Seçilen tekerlekleri tahrik edecek olan 37Dx54L tipi motorlar seçildi. Tekerlekleri, 37D54L tipi motora bağlamak için motorun çıkış miline M4 setscrewle bağlanabilecek, tekerlek göbeğine de tekerleklerde bulunan deliklerden M3 civatayla bağlayabileceğimiz motor bağlantı flanşları tasarlandı.

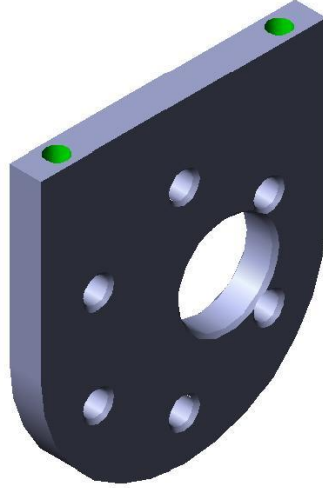
Robot şasi malzemesinin, kolay işlenebilirliği, bir çok imalat işlerinde kullanılması, kolay bulunabilirliği ve maliyetlerinin ucuz olmasından dolayı St 37 kalite imalat çeliğinden yapılmasını ön görüldü.

Robot şasisi iki ana parçadan oluşturuldu. Bunlardan birincisi üst platformdur. Üst platform üzerine mikrokontrolcü, pil, motor sürücüsü, Xbee iletişim modülü ve kaster tekerleğin bağlanacağı bir plaka olarak düşünüldü. Bu plaka St 37 malzemeden 5 mm kalınlığında CNC lazer kesim makinesinde imal edildi ve ağırlığının azaltılması maksadıyla ortası uygun şekilde boşaltıldı.

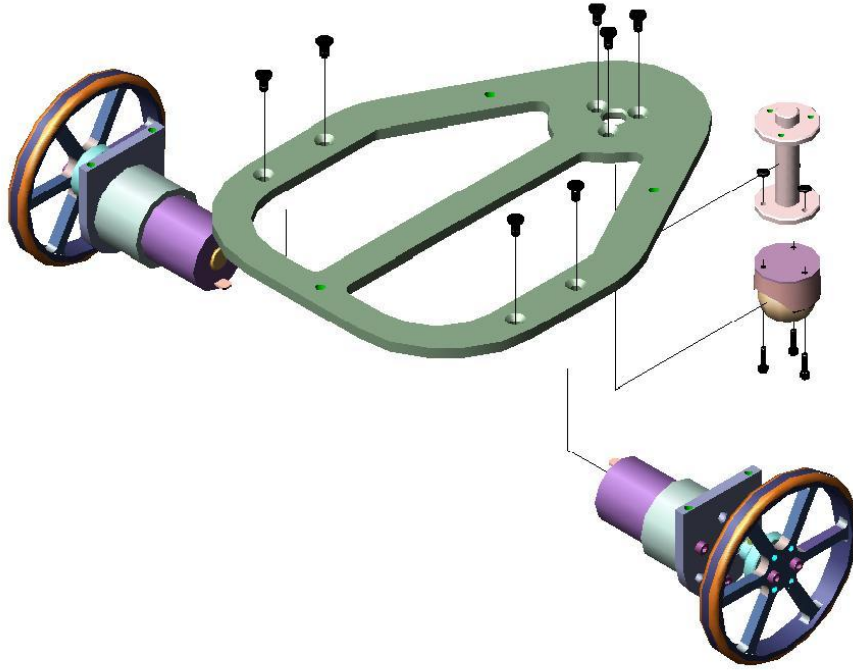
Robot şasisini oluşturan ikinci ana parça ise motorların bağlandığı ve üst plakaya M3 civatalarla bağlanan motor braketleridir. Bu braketler donanımda seçilen 37Dx54L tipi motora uyumlu şekilde tasarlandı. 37Dx54L tipi motorun bağlanabilmesi için motor şaftının olduğu bölüm braketlerde boşaltıldı ve motorun, brakete M3 civatalarla bağlanabilmesi için brakete uygun civata delikleri açıldı.



Şekil 4.1. : Üst Platform CAD modeli



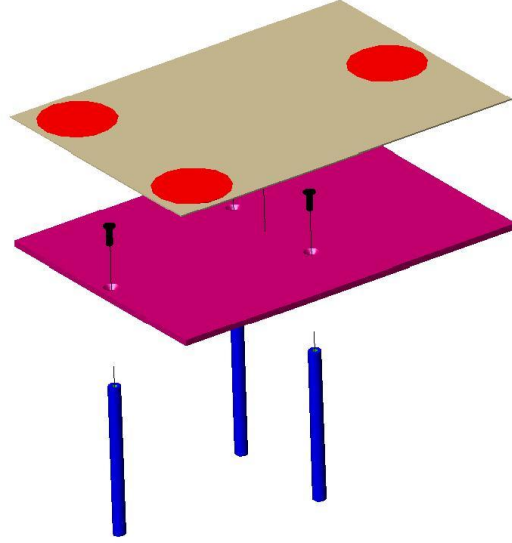
Şekil 4.2. : Motor Bağlantı Braketi CAD modeli



Şekil 4.3.: Ana şasinin patlatılmış CAD modeli

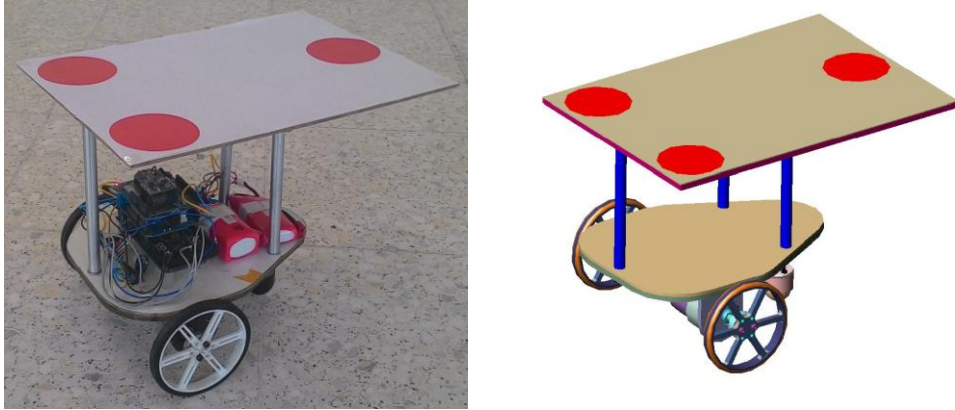
Ayrı tasarlanan bu 5 mm kalınlığındaki şasi parçaları, 37Dx54L tipi motorlarla ve kaster tekerlek, bağlantı deliklerinden M3 cıvatalarla bağlanarak diferansiyel sürürlü robotun ana şasi kompleksini oluşturmuştur. Şekil 4.3'te ana şasinin CAD modelini görebilirsiniz.

Platformun ana şasesini, motor bağlantılarını ve kaster tekerlek bağlantılarını yaptıktan sonra hareketli platformun kamerada nesne takibi yapılabilmesi için nesne takip desenini koyabileceğimiz nesne takip platformuna ihtiyaç duyuldu. Bu platformu, üst platforma bağlayabilmek için 6 mm çapında üst platformun çevresine üç delik açıldı. Nesne takip platformunu elektronik devre modüllerinden uzaklaştırmak için 10 mm çapında 150 mm boyunda alüminyum miller yapıldı ve bu millere mil ekseninde olacak şekilde karşılıklı M5 delik delindi. Yapılan alüminyum milleri üst platforma M5 cıvatalarla bağlandı, daha sonra da nesne takip platformunu bu alüminyum millerin üzerine M5 havsa başlı cıvatayla yerleştirildi. Böylece nesne takip desenini koyabileceğimiz bir platform oluşturdu. Nesne takip platformunu mobil robot üzerine bağlanmış durumda Şekil 4.4.'te görebilirsiniz.



Şekil 4.4: Nesne takip platformu

Yapılan bilgisayar destekli tasarımlar, imalat süreci ve son olarak montaj süreci sonrasında Şekil 4.5.'da gösterilen hareketli platform oluşturulmuştur.



Şekil 4.5: Hareketli platform

4.2.2 Robot Donanımı

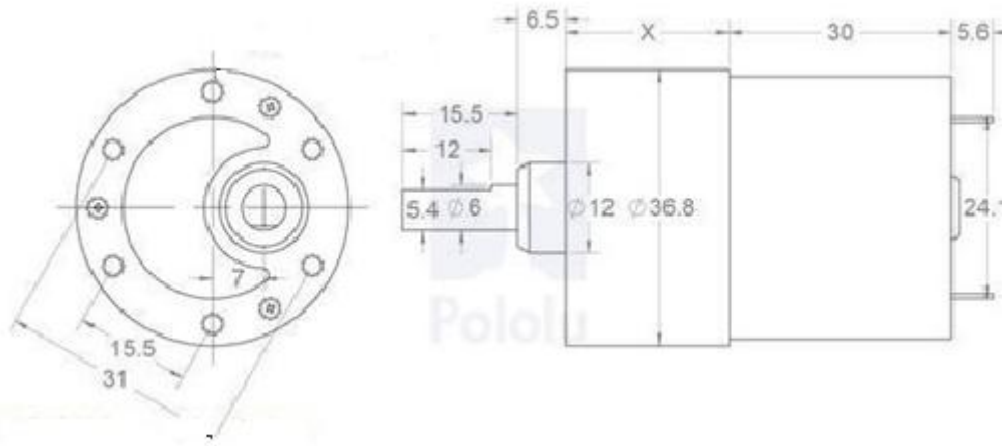
Robot donanım belirlemedeki öncelikli kriterler ise ucuza satın alınabilirlik ve donanım parçalarının stokta bulunabilmesidir. Bu kriterler göz önüne alınarak aşağıda sıralanan donanımlar temin edilmiştir.

4.2.2.1 Motor :

Bu çalışmada kullandığımız motor 67:1 oranına sahip metal redüktöre sahiptir. 12 V gerilimle çalışmakta bu gerilim değerinde redüktör mili çıkışından 150 d/d çıkış alabilirsiniz. Motor boş çalışma esnasında yaklaşık 300mA değerinde akım çeker. Zorlanma tork değeri 14 kg/cm'dir. Motorun ölçüsel teknik resmi Şekil 4.6'te verilmiştir.

Motor özellikleri şöyledir :

- Boyutları : 67Dx54L
- Ağırlığı : 128 g
- Mil Şaft Çapı : 6 mm
- Redüksiyon Oranı : 67:1
- Hız : 150 d/d @ 12V
- Boş Çektiği Akım : 300mA @ 12V
- Zorlanma Akımı : 5A @ 12V
- Zorlanma Torku : 14kg/cm @ 12V



Şekil 4.6 : 67Dx54L tipi motorun ölçüsel teknik resmi [31]

Bu motorda dahili 64 CPR kuadratik enkoder mevcuttur. Enkoder çıkış milinin tam turunda 2488 adım saymaktadır.

Motor ve Enkoder kablo renk kodları Çizelge 4.1'de verilmiştir.

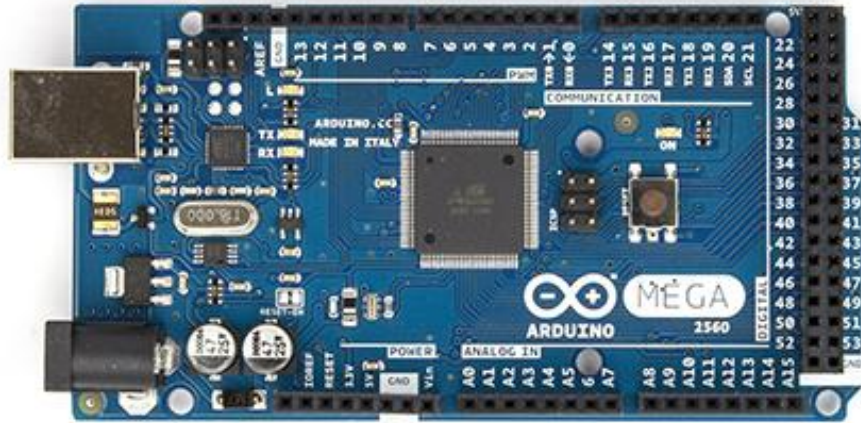
Çizelge 4.1 : Motor ve Enkoder Kablo Renk Kodları

Renk	Fonksiyon
Siyah	Motor Güç Kablosu
Kırmızı	Motor Güç Kablosu
Mavi	Enkoder Vcc Kablosu (3.5 V – 20 V)
Yeşil	Enkoder GND
Sarı	Enkoder A fazı çıkışı
Beyaz	Enkoder B fazı çıkışı

4.2.2.2 Mikrokontrolcü

Bu dönemde, pahalı olmayışları, kolay bulunabilirlikleri, özel tasarlanan kart sistemiyle diğer çevre birimlerine kolayca bağlanabilmesi gibi özelliklerden dolayı Arduino mikrokontrolcü kartları elektronik hobilerle uğraşanlar, öğrenciler ve araştırmacılar arasında en çok kullanılan mikrokontrolcü kartlar olmuştur. Bu çalışmada özellikle kullanılmasının nedeni diğer çevre birimlerine kolaylıkla bağlanabilmesidir.

Bu çalışmada Arduino Mega 2560 diğer modüllerle kolaylıkla bağlanabilmesi, çevre birimi ilavesi gerektiğinde pin sayısının yeteri kadar çok olmasından dolayı tercih edilmiştir. Şekil 4.7 ' de Arduino Mega 2560 kartı gösterilmiştir.



Şekil 4.7: Arduino Mega 2560 Kartı [32]

Arduino Mega 2560 kartında AtMega 2560 mikrokontrolcüsü kullanılmakta, 54 I/O pine sahiptirler, bu pinlerden 15 i PWM sinyal çıkışı, 16 adeti de analog girişi olarak

kullanılabilmekte; 4 adet donanım seri portu, 16 MHz kristal osilatör, bir USB bağlantısı, bir güç girişi jackı ve reset butonu bulunmaktadır. Bir mikrokontrolcüyü destekleyecek her şeyi içermektedir. USB kablosuyla kolaylıkla bilgisayarlara bağlanabilir. Arduino Mega 2560 kartının özellikleri Çizelge 4.2 ‘de verilmiştir.

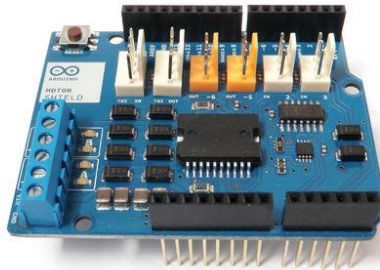
Çizelge 4.2 : Arduino Mega 2560 Kart Özellikleri

Arduino Mega 2560 Kart Özellikleri	
Mikrokontrolcü	ATmega2560
Çalışma Gerilimi	5V
Giriş Gerilimi (önerilen)	7-12V
Girdi Gerilimi (limits)	6-20V
Dijital I/O Pinleri	54 (15 PWM çıkış pini)
Analog Giriş Pinleri	16
DC akım her I/O Pinlerinde	40 mA
DC Current for 3.3V Pin	50 mA
Flash Bellek	256 KB
SRAM	8 KB
EEPROM	4 KB
Saat Hızı	16 MHz

4.2.2.3 Motor Sürücüsü

Arduino motor sürücü modülü kullandığımız Arduino Mega 2560 kartı için üretilmiş uygun bir karttır.

Arduino motor sürücü modülü motor hızlarını ve yönlerini kolaylıkla kontrol edilmesini sağlar. Adreslenen pinlerin yardımıyla arduino motor sürücü modülüne motorları dahil etmek çok kolaydır. Bu karta 12V a kadar güç desteği sağlamak mümkündür.



Şekil 4.8 : Arduino motor sürücü modülü [33]

Modül üzerinde adreslenen pinler vardır. Bu pinler motor kanalını seçmek, motor hızlarını (PWM) tayin etmek, motor dönüş yönlerini belirlemek, motorları durdurup çalıştırmak için kullanılır. Pin adresleri Çizelge 4.3'te verilmiştir.

Çizelge 4.3 : Arduino motor sürücü modülü pin adresleri

Arduino motor sürücü modülü pin adresleri		
	A Motor Kanalı	B Motor Kanalı
Yön	Dijital Pin 12	Dijital Pin 13
Hız (PWM)	Dijital Pin 3	Dijital Pin 11
Fren	Dijital Pin 9	Dijital Pin 8
Akım Ölçüm	Analog 0	Analog 1

Motorlar kanalları çalıştırılırken öncelikle yön pinlerine HIGH, LOW değerleri atanır böylece motorların yönü belirtilir. Bir sonraki aşamada motor kanallarındaki frenlemenin açılması için frenleme pinlerine LOW değerleri atanır. En son aşamada ise hız pinlerine motorun hız değerleri (0 - 255) atanarak motorlar çalıştırılır.

4.2.2.4 Batarya

Robot platformu test esnasında otonom hareket edeceğinden dolayı motorlarını hareket ettirebilmesi için dahili bir bataryaya ihtiyaç olacaktır. Bu bataryalar çok kere tekrarlanan test süreçlerinde uzun süre hizmet verebilecek tipte olmalı ve aynı zamanda testler sonucunda azalan enerjilerini yeni kullanımlar için tekrar şarj edilebilmelidir.

Piyasada satılan birçok tipte batarya bulmak mümkündür, Li-Ion bataryalar, Li-Po bataryalar, kuru akümülatörler gibi tekrar şarj edilebilir çok çeşit batarya bulunabilir. Bunların arasından Li-Po bataryalar hafif ve kısa zamanda şarj edilebilme özelliğiyle ön plana çıkmaktadır. Bu çalışmada platformda güç gereksinimlerini Li-Po bataryalardan sağlanmıştır. Şekil 4.9 'de Li-Po bataryanın teknik özellikleri gösterilmiştir. Bataryanın teknik özellikleri Çizelge 4.4 'te verilmiştir.

Çizelge 4.4: Batarya özellikleri

Batarya teknik özellikleri	
İçeriği	Lityum polimer
Voltaj	11.1 V
Kapasite	1.35 A
En	35 mm
Boy	68 mm
Yükseklik	21 mm
Ağırlık	0.1Kg



Şekil 4.9: Li-Po batarya [34]



Şekil 4.10. : Xbee kablosuz iletişim modülü [35]

4.2.2.5 Xbee Kablosuz İletişim Modülü

Bu çalışmada hareketli platformdan alınacak verilerin bilgisayar ortamında hesaplanması için Xbee kablosuz iletişim modülü kullanılmıştır. Birçok kablosuz uygulamada kullanılan bu modüller kapalı ortam çalışmalarında 100m mesafeye kadar sorunsuz iletişim kurabilmektedirler. Şekil 4.10 'da Xbee kablosuz iletişim modülü gösterilmiştir. Çizelge 4.5'te Xbee kablosuz iletişim modülü özellikleri verilmiştir.

Çizelge 4.5: Xbee kablosuz iletişim modülü özellikleri

Xbee kablosuz iletişim modülü özellikleri	
Çalışma Gerilimi	3.3V @ 50mA
Maksimum Data Aralığı	250kbps
Çıkış Gücü	1mW (+0dBm)
Kapsama Alanı	100 m
Dahili Anten	Var
10-bit ADC giriş pini	6 adet
digital I/O pini	8 adet

4.2.2.6 Tekerlekler

Robot uygulamaları için kullanılan plastik tekerlekler seçilmiştir. Bu tekerlekler 3 mm çaplı motor millerine uygun olarak tasarlanmıştır, uygulamada genellikle motora bir flanş yardımıyla montajı gerçekleşir. Tekerleğin üzerinde silikon malzemeden lastik bulunmakta, tekerlek çapı 90 mm'dir. 6 adet cıvata bağlama deliği tekerleğin flanş yardımıyla motora bağlanmasını kolaylaştırır. Şekil 4.11 'da Robotta kullanılan tekerlek çiftini görebilirsiniz.



Şekil 4.11: Tekerlek çifti [36]

4.2.2.7 Kaster Tekerlek

Kaster tekerler özellikle diferansiyel sürürlü robot platformlarında robotun kararlılığını arttırmak için kullanılırlar.

Plastik bilyalı bu küçük kaster tekerin bilya çapı 25,4 mm'dir. Platforma bir adaptör yardımıyla bağlanır. Adaptör platformun zemine göre paralel durmasını sağlar. Bu tekerlekler platformun hareketi ile herhangi bir yöne doğru dönebilirler. Böylece kaster tekerlerin robot kinematiğini etkileyen bir durumu yoktur. Şekil 4.12'de kullanılan kaster teker görülmektedir.



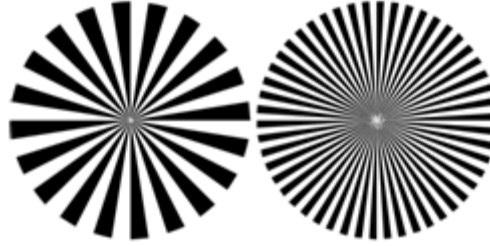
Şekil 4.12 Kaster tekerlek [37]

4.2.3 Algılayıcılar

Bu bölümde platformda kullanılan ve test esnasında odometre verisini algılayacak olan enkoderlerden 4.2.3.1 kısmında genel olarak bahsedilecek, test ortamında platformun mutlak konum ve yönünü algılayacak olan kameradan bölüm 4.2.3.2' de genel olarak bahsedilmiştir.

4.2.3.1 Enkoder

Enkoderler artırımsal enkoderler ve mutlak enkoderler olarak sınıflandırılır. Enkoderler motor kontrolünde temel geri besleme algılayıcısı olarak kullanılır. Enkoderlerin yapılarında Hall Etkisi ve ya optik algılayıcılar vardır. Optik algılayıcı olanlarda siyah-beyaz bölünmüş diskte led ışığının yansımalarını kullanarak sinyal alınır.

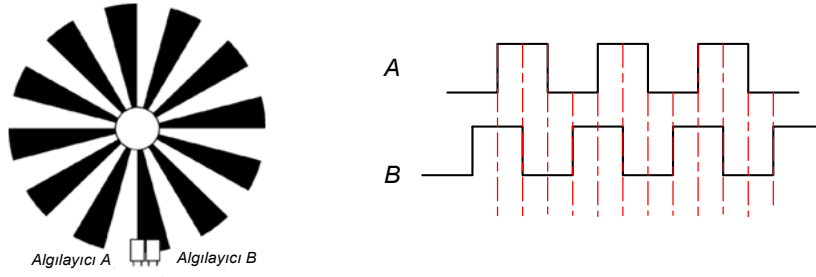


Şekil 4.13: Artırımsal enkoder [38]

Enkoderlerde kullanılan manyetik ve ya optik algılayıcılar sadece disklerde bulunan bölümleri sayarlar, fakat diskin saat yönünde ve ya saatin tersi yönünde dönüp dönmediğini anlayamaz. Bu sorunu çözmek için genelde enkoderlerde ikinci bir algılayıcı, faz farkı oluşturacak şekilde enkoderlere yerleştirilir. Bu tip enkoderlere kuadratik enkoder denilir. Eğer A fazlı sinyal önce geliyorsa, mil saat yönünün tersine döner; B fazı sinyali önce geliyorsa saat yönünde döner. Mutlak enkoderlerde desenli disk bulunur, bu diskteki desenler shaftın kesin pozisyonunu belirler.



Şekil 4.14: Mutlak enkoder yapıları [39]



Şekil 4.15: Kuadratik enkoder ve faz sinyalleri

4.2.3.2 Kamera

Kameralar görüntü algılayıcı olarak robot çalışmalarında sıklıkla kullanılırlar. Otonom robotlarda nesne takibi ve hareket özelliğini belirlemede kameralardan faydalanır.

CMOS ve CCD algılayıcılara göre kameralar sınıflandırılır. Bu algılayıcılar üzerine gelen ışığın yoğunluğunu okuyarak görüntü oluştururlar. Oluşturdukları görüntülerle çevreyi analiz ederler, nesnelere belirleme ve nesne takibi yapabilirler.

CCD algılayıcıları ışık yoğunluğunda göre yüklenebilen kapasitör yapılardan oluşmuştur. Bu kapasitör yapısının önünde lens bulunur. Algılayıcıya gelen ışık yoğunluğuna göre kapasitör yüklenir ve bu yüklenmeyi voltaj değerine dönüştürürler. Algılayıcıdan gelen voltaj değerleriyle görüntü oluşturulur.

CMOS görüntü algılayıcı transistor ve ışığa duyarlı diyotlardan oluşur. Diyotların üzerlerine gelen ışığı yoğunluğuna göre voltaj değerine dönüştürürler. Görüntü elemanları yeniden görüntü almak için bir transistor yardımıyla tekrardan kurulur. Diğer transistor dönüştürülen voltaj değerlerini güçlendirir ve son transistor de güçlendirilen bu voltaj değerlerini görüntü işlem sürecine gönderir. Bu şekilde CMOS görüntü algılayıcılardan görüntü elde edilir.

CCD görüntü algılayıcıları, düşük gürültü içeren yüksek kaliteli görüntüler sağlar, daha fazla görüntü elemanı içerirler ve ışığa karşı çok duyarlılığa sahiptirler; buna karşın CMOS görüntü algılayıcıları az maliyetli, imalatı kolay ve düşük enerji tüketimi gibi üstünlüklere sahiptirler.

Bu çalışmada test ortamında hareket eden platformun yörüngesini çizmek için platform hareketi esnasında konumunu algılayan, 640x480 görüntü elemanlı CMOS görüntü algılayıcıya sahip bir kamera kullanıldı. (Şekil 4.16)



Şekil 4.16: CMOS görüntü algılayıcı kamera [40]

4.3 Sonuç

Literatür taraması ve platform sürüş yöntemlerinin irdelenmesi sonucunda diferansiyel sürürlü platform tasarımına karar verildi. Diferansiyel sürüş sisteminin temeli olan sabit tekerleklerin ve bu tekerlekleri tahrik edecek motorun seçilmesi ile tasarım süreci başlatıldı. Motor bağlantıları, platform ana şasesi, kaster teker bağlantı adaptörü tasarlandı. Tasarlanan bu parçalar imal edilerek montajı gerçekleştirildi ve kinematik parametrelerini tahmin edebileceğimiz bir hareketli platform ortaya çıktı. Bu platformu hareket ettirebilmek için mikrokontrolcü ve motor sürücüsü donanımı seçildi, hesaplama bilgisayarı ve mikrokontrolcü arasında iletişim kurulabilmesi için Xbee haberleşme kiti alındı, bu elektronik kitleri çalıştırmak için batarya seçildi. Hareketli platformun test ortamında kameradan mutlak konum ve yönünün belirlenmesi için nesne takip platformu oluşturuldu. Platform tasarımı sonrasında test ortamında belirli bir yörüngede hareket ettireceğimiz, veri toplayabileceğimiz ve kameradan mutlak konum ve yön tespiti yapabileceğimiz bir hareketli platform meydana geldi.

5. KONTROL MİMARİSİ

5.1 Giriş

Kontrol mimarisi, tasarımını yaptığımız hareketli platformun hareket edebilmesini sağlayan elektronik modüllerin nasıl bağlandığını, platformdaki devreyle hesaplama bilgisayarı arasında iletişimin nasıl kurulduğunu, mikrokontrolcü ve bilgisayar arasındaki yazılım ve komutların nasıl çalıştığını detaylı biçimde anlatan bölümdür.

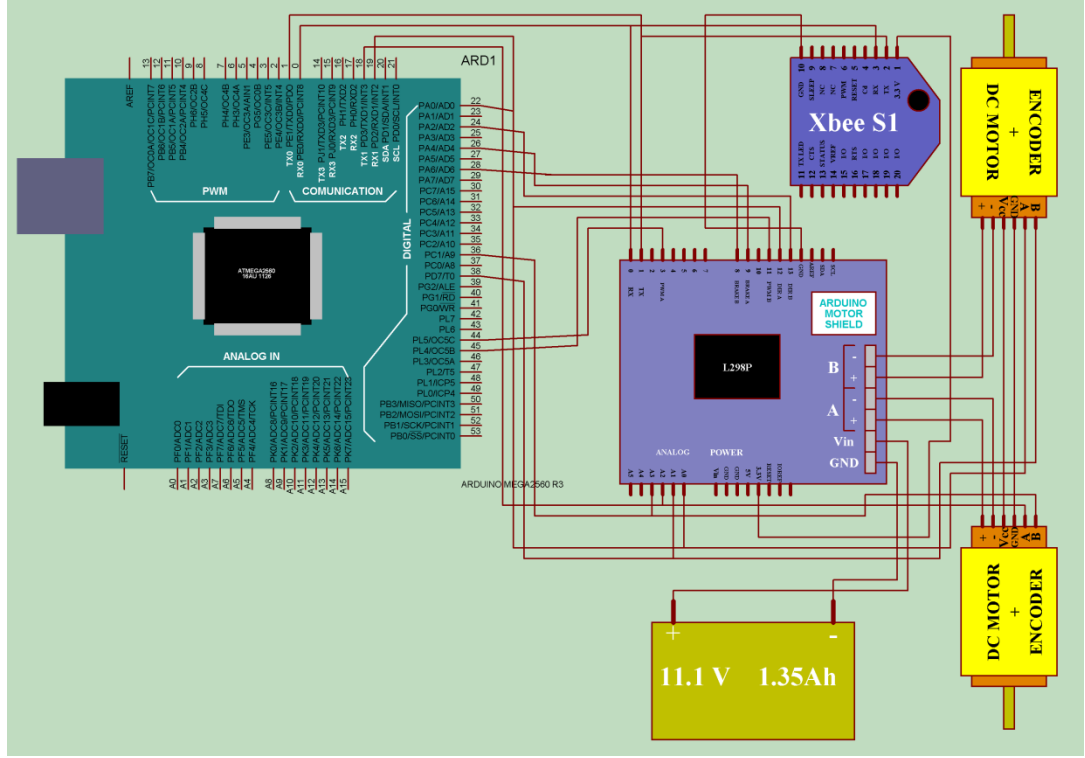
Kontrol mimarisi başlığının hangi konuları içerdiği 5.1 giriş bölümünde kısaca değinilmiştir. Platformda bağlı bulunan devrenin nasıl oluştuğu detaylı olarak 5.2 platform devresi bölümünde açıklanmıştır. Test sistemi mimarisinin nasıl işlediği 5.3 test sistemi mimarisi bölümünde genişçe anlatılmıştır. 5.3.1 bölümünde mikrokontrolcü tarafındaki komutlar tanıtılmıştır. 5.3.2 bölümünde hesaplama bilgisayarında kullanılan komutlar tanıtılmıştır. 5.3.3 bölümünde hesaplama bilgisayarı ile mikrokontrolcü arasında gerçekleşen seri iletişim detaylı olarak anlatılmıştır, mikrokontrolcünden hesaplama bilgisayarına gönderilen odometre verisi paketi ve hesaplama bilgisayarından mikrokontrolcüye gönderilen motor komut veri paketleri açıklanmıştır.

5.2 Platform Devresi

Hareketli platformun bilgisayardan gelen verileri alabilmesi, bu verilere göre hareket edebilmesi için platformun üzerine dahil edilmiş elektronik devreye ihtiyacı vardır. Platform için tasarlanan devre şemasını Şekil 5.1’de görebilirsiniz.

Devre mikrokontrolcü kartı, motor sürücü modülü ve Xbee kablosuz haberleşme modülünden oluşmuştur. Bilgisayar tarafından gönderilen veriler Xbee kablosuz haberleşme modülü tarafından alınır, mikrokontrolcü içerisindeki dahili yazılım yardımıyla bu verileri algılar ve motor sürücü modülüne yön, hız ve fren değerlerini atar, motor sürücü modülü gelen verilere göre motorları kontrol ederek, mobil platformun yön değiştirmesini, hızlanmasını, yavaşlamasını ve durmasını sağlar.

Devremizi oluşturan esas parça Arduino 2560 mikro kontrolcüdür. Motor sürücü modülü, Xbee kablosuz haberleşme modülündeki gömülü pinler ve enkoder kablo bağlantıları mikrokontrolcünün yazılımında tanımlanır.



Şekil 5.1: Robot platformunun devre şeması

Motor sürücü modülünde sağ motorun yön pini 12 , hızı pini 3 , fren pini 9 ; sol motor için yön pini 13, hız pini 11, fren pini 8 olarak modüle tanımlanmıştır. Bu modülü mikrokontrolcüye bağlarken bahsedilen pinlerle mikrokontrolcüdeki pin numaraları birbirini karşılamaı gerekir. Pinlerin uyumu sağlandıktan sonra motor sürücü modülü mikrokontrolcüye yerleştirilir. Burada dikkat edilmesi gereken husus motor sürücü modülünün 12 V gerilimli pil ile beslenecek olmasıdır. Bu gerilimin mikrokontrolcüye etkilememesi için motor sürücü modülündeki Vin pini kesilir, böylece motor sürücüyü besleyen geriliminin mikrokontrolcüye zarar vermesi önlenir.

Xbee kablosuz haberleşme modülünün pinleri TX iletim pini, RX alım pini , 3.3V ve GND pinleri modüle gömülüdür. Haberleşme modülü, motor sürüş modülünün üzerine bu belirtilen pinler motor sürüş modülündeki pinleri karşılayacak şekilde yerleştirilir. Bu şekilde yerleştirilen Xbee kablosuz haberleşme modülü mikrokontrolcüde tanımlanan standart seri porttan haberleşecektir, Eğer usb kabloyla ve aynı zamanda Xbee haberleşme modülü ile mikrokontrolcüye bağlıysa ve bilgisayardan mikrokontrolcüye yeni program yüklüyorsanız program hata verir. Bunun nedeni hem usb kablo bağlantısının hemde Xbee kablosuz haberleşme modülünün aynı seri portu kullanıyor olmasıdır. Biz böyle bir

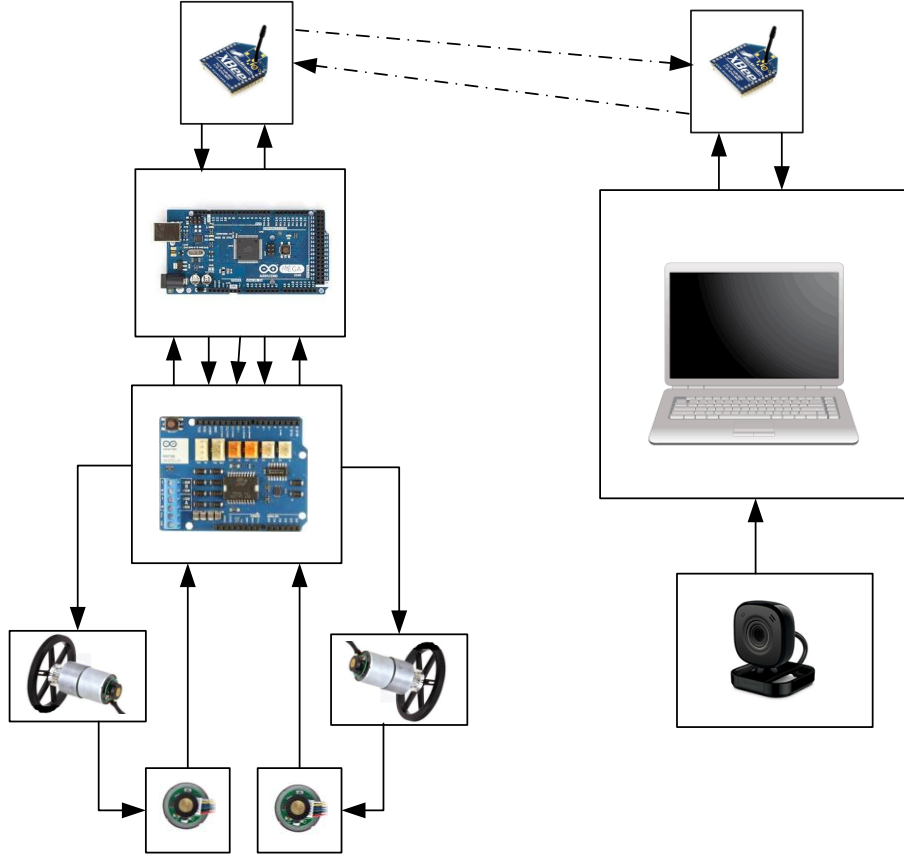
çakışma olmasını istemediğimiz için haberleşme modülündeki TX ve RX pinlerini keserek bu pinleri kablo yardımıyla mikrokontrolcünün TX3 ve RX3 pinlerine bağladık. Bu durumda Xbee iletişimi üç numaralı seri port üzerinden gerçekleşti, aynı zamanda usb kablo bağlantısı ile de standart seri port üzerinden program yüklemelerini gerçekleştirebildik.

Sağ motor güç kabloları motor sürücü modülünde bulunan A motor kanalına, sol motor güç kabloları B kanalına bağlandı. Sağ motorun enkoderinin A fazı mikro kontrolcüde 18 numaralı pinde tanımlı 5 numaralı kesme kanalına, B fazı ise 36 numaralı sayısal pine bağlandı. Sol motorun enkoderinin A fazı 19 numaralı pinde tanımlanan 4 numaralı kesme kanalına B fazı ise 38 numaralı sayısal pine bağlandı. Enkoderlerin A fazlarının kesme kanallarına bağlanması, enkoderlerin yüksek hızlarda çalışırken sinyal atımı kaçırımlarını önler ve daha doğru veriler elde etmemizi sağlar.

5.3 Test Sistemi Mimarisi

Oluşturulan test sistemi akış diyagramı Şekil 5.2.'de gösterilmiştir. Robot devresindeki mikrokontrolcü motor hareketlerinin yönetiminden ve enkoder verilerinin seri iletişim protokolüne gönderilmesinden sorumludur. Bilgisayar tarafından gönderilen motor komutları mikrokontrolcü tarafından seri iletişim portundan okunur. Okunan motor komutlarına göre motor sürücü modülü motorların hareketini gerçekleştirir. Motor hareketlerinin gerçekleşmesiyle motor millerine bağlı olan enkoderler hareket esnasında oluşan enkoder atımlarını sayarlar. Mikrokontrolcüdeki enkoder atımlarını toplar ve belli zaman aralıklarında seri iletişim protokolüne yazar. Xbee kablosuz iletişim modülü yardımıyla, hesaplama/görüntü işleme bilgisayarı ve platform üzerindeki mikrokontrolcü iletişim halindedirler. Xbee kablosuz iletişim modülü kontrol verilerinin robotun mikrokontrolcüsüne aktarılmasına, alınan enkoder verilerini de hesaplama/görüntü işleme bilgisayarına aktarmaktan sorumlu donanımdır.

Hesaplama /görüntü işleme bilgisayarına gelen enkoder verileri bilgisayardaki yazılım yardımıyla odometre verilerine dönüştürülür, aynı şekilde, kameradan alınan robot konumundaki ve açısındaki zamana bağlı değişimlerde bilgisayardaki yazılımla odometrik verilere dönüştürülür. Böylece robottaki enkoderlerden hesaplanan odometre veriler ve de kameradan ölçülen robot konum ve yönündeki anlık değişikliklerle mutlak konum ve yön verileri elde edilir. Bu durumda enkoderlerden elde edilen odometre verileriyle kameradan elde edilen mutlak konum ve yön verileri kıyaslanabilir.



Şekil 5.2: Test sistemi akış diyagramı

5.3.1 Mikrodenetleyicide Kullanılan Komutlar

__Super_Motor_Data : __Read_MotorSerial ve __Encode_Motor_Signal komutlarını içeren ve çalıştıran motor fonksiyonu.

__Read_MotorSerial : Bilgisayardan gelen motor komutunu seri iletişimde okuyan fonksiyondur. Seri iletişimde bilgisayardan robota motor verileri gönderilir, motor verileri I1,I2,I3,I4,S1,S2 şeklindedir. I1 sol motor yönü I2 sol motor fren, I3 sağ motor yönü, I4 sağ motor freni, S1 sağ motor hızı ve S2 sol motor hızıdır. Robot bu seri iletişimdeki veri paketini okuyarak hareketini gerçekleştirir.

__Super_Read_Encoder_Data: Sağ ve sol enkoderler den gelen sinyalleri okur.

__Read_Right_Encoder_Data : Sağ enkoderden gelen sinyal verisini okur

__Read_Left_Encoder_Data: Sol enkoderden gelen sinyal verisini okur

__Super_Send_Encoder_Data: Sağ enkoder verisini ve sol enkoder verisini gönderir.

__Send_Right_Encoder_Data: Sağ enkoder verilerini bilgisayarın okuyabilmesi için Header, R-Encoder Output, dTS, RightEncoderRead, RightEncoderRead, TR-END formatıyla seri iletişim kanalına yazar.

__Send_Left_Encoder_Data : Sol enkoder verilerini bilgisayarın okuyabilmesi için Header, L-Encoder Output, dTS, LeftEncoderRead, leftEncoderRead, TR-END formatıyla seri iletişim kanalına yazar.

__Super_Arduino_Computer: Program başlangıç şifresi için seri iletişimi okuyor, başlangıç verisi geldiğinde başlamaya hazır olduğunu kontrol bilgisayarına gönderiyor. Yeniden başlamak için tekrar seri iletişimi kontrol ediyor.

__WriteStartCommand : Robot 4321 olan başlamaya hazır olduğunu belirten şifreyi seri iletişim yoluna yazar.

__Read_StartCommand: Seri iletişim yolunda bilgisayardan gelen robotu başlatma şifresini okur ve robotta tanımlanan başlama şifresiyle kıyaslar, uyuşması durumunda robot çalışır.

__Super_DriveMotor: Sağ motoru ve sol motoru sürer.

__Drive_LeftMotor: Sol motora kontrol bilgisayarından atanan motor dönüş yönü, frenleme bilgisi ve motor dönüş hızı verilerini atayarak, motorun istenilen yönde istenilen hızda dönmesini gerçekleştiren fonksiyondur.

__Drive_RightMotor: Sağ motora kontrol bilgisayarından atanan motor dönüş yönü, frenleme bilgisi ve motor dönüş hızı verilerini atayarak, motorun istenilen yönde istenilen hızda dönmesini gerçekleştiren fonksiyondur.

__Setup_Serial3_Interface: Seri iletişim protokolünü başlatır, seri gecikme zamanını atar. Ve seri iletişimdeki verileri temizler

__Setup_Motor_Interface: Motorlara çalışmaya başlamadan önce ilk değerleri atar ve motorları bu atanan değerlerle durağan şekilde başlaması için sağ ve sol motorları durur vaziyette çalıştırır.

__Setup_EncoderInterface : Enkoder pinlerini atar ve sađ ve sol enkoder deęerini kesmeli okumak iin mikro denetleyici de bulunan dördüncü ve beşinci kesme kanallarına baęlar.

__doEncoderRight_ISR: Bu fonksiyon sađ enkoder a fazının yüksek olduęu ve b fazının düşük olması durumunda motorun saat yönü tersine döndüęünü farz ederek enkoder okumasını bir arttırır, b fazı yüksek durumda olduęunda saat yönü döndüęünden bir azaltır.

Eđer a fazı düşük geliyorsa ve b fazı yükseğe o zaman motor saat yönünde döner ve enkoder sayısını bir arttırır, a fazı düşük ve b fazı da düşük geliyorsa motor saat yönü tersinde dönüyordur ve enkoder sayısını 1 azaltır.

__doEncoderLeft_ISR: Bu fonksiyon sol enkoder a fazının yüksek olduęu ve b fazının düşük olması durumunda motorun saat yönü tersine döndüęünü farz ederek enkoder okumasını bir arttırır, b fazı yüksek durumda olduęunda saat yönü döndüęünden bir azaltır.

Eđer a fazı düşük geliyorsa ve b fazı yükseğe o zaman motor saat yönünde döner ve enkoder sayısını bir arttırır, a fazı düşük ve b fazı da düşük geliyorsa motor saat yönü tersinde dönüyordur ve enkoder sayısını 1 azaltır.

5.3.2 Hesaplama Bilgisayarında Kullanılan Komutlar

GUI_getARDUINODataSelfStart : Seri iletiřim portunu açarak robot bilgisayar arasındaki iletiřimi açar.

GUI_plotXBeeData : Robotan gelen enkoder verilerini örnekleme zamanına göre çizer.

GUI_ReadArdu : Arduino tarafından gönderilen verileri uygun düzende okur.

GUI_readArduinoCommunicationStart : Arduino iletiřimini kurmak için seri iletiřim kanalını okur.

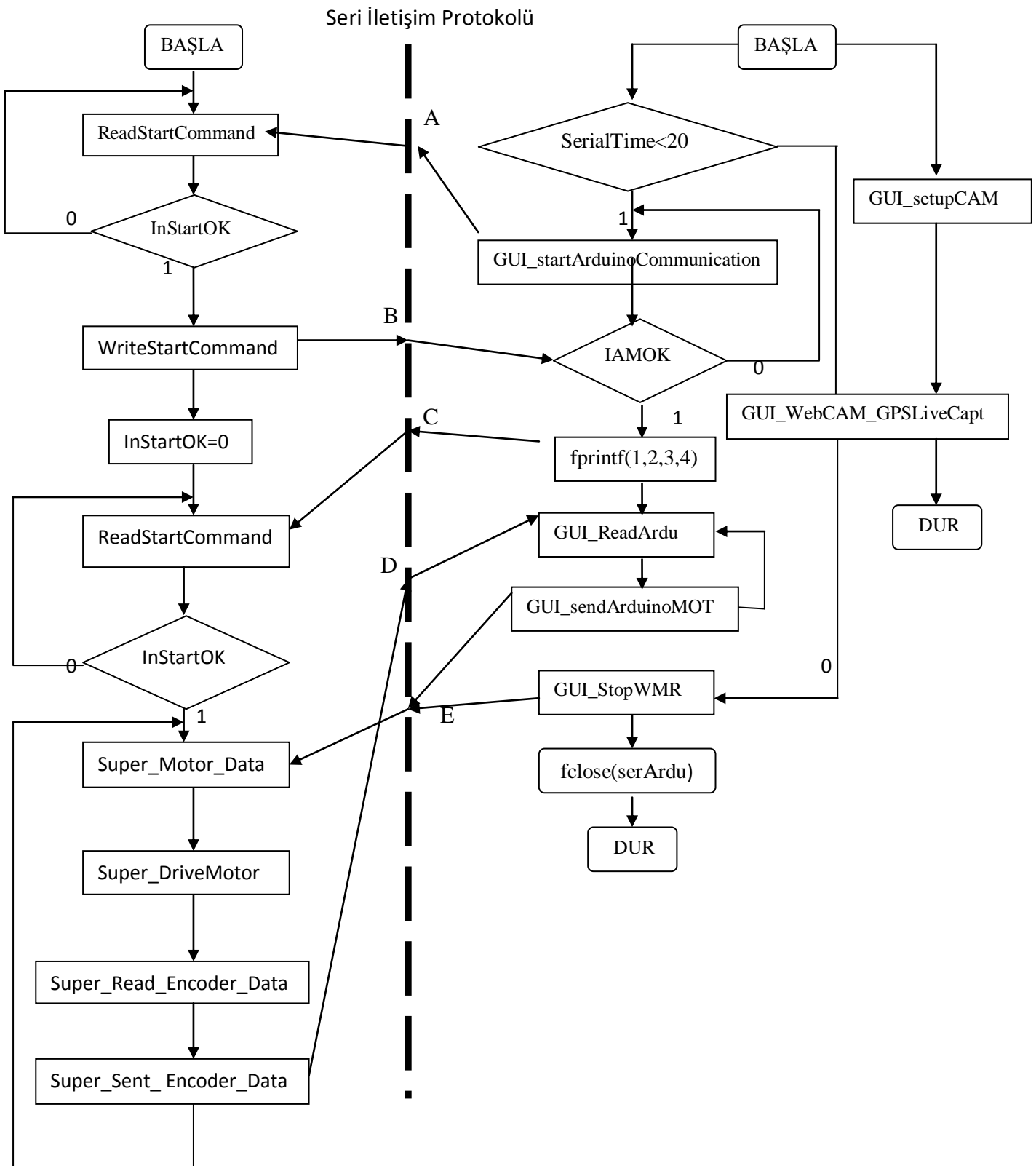
GUI_sendArduinoMOT : Motor verilerini Arduinoya gönderir.

GUI_startArduinoCommunication :Arduino ile iletiřimi başlatır.

GUI_StopWMR : Test bittięinde robotu durdurur.

- SetupSerial*** : Seri iletişim portunu kurar.
- UI_PlotNormalizedOdometerData*** : Normalize edilmiş odometre verilerini çizer.
- myCamCalibration_Script*** : Kamera kalibrasyonunu gerçekleştirir.
- GUI_setupCAM*** : Kamerayı kurar.
- GUI_WebCAM_GPSLiveCapture*** : Test esnasında video görüntülerini çalıştırır ve kaydeder.
- GUI_EndDeadReckoningData*** : Enkoder datalarının okumasını sonlandırır.
- GUI_OpenDatabaseFile*** : Kaydedilen odometre verilerini açar.
- GUI_OpenReadDatabaseFile*** : Kaydedilmiş veri tabanı dosyasını açar.
- GUI_OpenWriteDatabaseFile*** : Yazılmış veri tabanı dosyasını açar.
- GUI_ReadDeadReckoningData*** : Oluşturulan enkoder verilerini belirli formatta okur.
- GUI_SaveDeadReckoningData*** : Enkoder verilerini kaydeder.
- UI_ColorTrackinRED_Comet*** : Robot üzerindeki takip desenini izler ve robotun yörüngesini oluşturur.
- UI_Coordinate_System_Origin*** : Kameranın köşelerinin ortalamasını alarak kamera koordinat sisteminin originini bulur.
- UI_Find_Camera_StartFrame*** : Robotun ilk hareket ettiği anı bulur.
- UI_ODM_DB_Test*** : Odometre datasını basar.
- UI_ODM_Trajectory*** : Odometre verilerini kullanarak robotun gerçekleştirdiği yörüngeyi çizer.
- WebCAMW2W*** : Takip edilen nesnenin konumunu genel referans koordinatındaki konumuna dönüştürür.

+



Şekil 5.3: Robotla Hesaplama /görüntü işleme bilgisayarı arasındaki seri iletişim ve program akışı.

5.3.3 Bilgisayar ile Mikrokontrolcü arasındaki Seri İletişim

Robot devresiyle Hesaplama ve görüntü işleme bilgisayarı arasındaki seri iletişim ve program akışı Şekil 5.3 te gösterilmiştir. Robot devresine güç verildiğinde mikrokontrolcü seri iletişim protokolünü denetleyen ReadStartCommand komutu döngüsüne girer, robot devresini çalıştıran 1234 şifresini seri iletişim protokolüne düşmesini bekler

Diğer yandan Hesaplama ve görüntü işleme bilgisayarı çalıştırılır. Hesaplama ve görüntü işleme bilgisayarı bir koldan robot devresi ile haberleşme sağlayan komutları çalıştırırken diğer koldan GUI_setupCAM komutuyla kamerayı kurar ve ardından GUI_WebCAM_GPSTLiveCapture 25 saniye içindeki kamera görüntülerini kaydeder. Seri iletişim protokolü 20 saniye çalışır ve seri iletişim protokolü kapatılır.

Seri iletişim protokolu A, B, C, D ve E safhalarından oluşur. A safhasında; GUI_startArduinoCommunication komutu seri iletişim protokolüne robot programını çalıştıracak şifreyi 1234 basar. Robot programında döngüde olan ReadStartCommand komutu seri iletişim protokolündeki şifreyi okur ve döngüden çıkar.

B safhasında; WriteStartCommand komutuna geçer bu komut robot programının veri aktarımına hazır olduğunu belirten 4321 şifresini seri porta yazar. Bilgisayardaki IAMOK döngüsü robot programının hazır olduğunu belirten şifreyi okur ve döngüden çıkar.

C safhasında robot programının motorları sürme ve enkoder verilerini bilgisayara aktarmına geçmesi için son şifreyi 1234 seri iletişim protokolüne basar.

D safhasında; robot programı Super_Motor_Data komutunu çalıştırır bu komut seri iletişim protokolünden motor komutlarını okur, gönderilen veriye göre motor yönlerini, motor frenlerini ve motor dönüş hızlarını atar. Ancak ilk çalıştırmada motor verisi motorlara hareket vermeyecek şekilde robot programında önceden tanımlanmıştır. Super_DriveMotor komutu seri porttan gelen motor verisine göre istenilen hız ve yönlerde motorları çalıştırır. Super_Read_Encoder_Data komutu her bir döngüde, motorların çalışması esnasında sağ ve sol enkoderlerden okuduğu atım sayısı toplamını ve döngü zamanını okur. Super_Sent_Encoder_Data komutu okunan enkoder atım sayılarını ve döngü zamanını, veri paketi olarak seri porta yazar. Gönderilen bu veri paketi bilgisayar programındaki GUI_ReadArdu komutuyla tekrar okunur ve alınan bu veriler sırasıyla kaydedilir.

E safhasında, GUI_sendArduinoMOT komutuyla motor veri paketi seri iletişim protokolüne yazar. Böylece D ve E safhaları döngü içerisine girer. Bu döngü SerialTime , seri iletişim protokolü çalışma zamanı, değişkeninin 20 saniyeyi aşması durumunda GUI_StopWMR komutuyla motor komutlarını durduran veri paketini seri iletişim protokolüne yazarak motorları durdurur. Bilgisayar programı GUI_StopWMR komutunun ardından fclose(serArdu) komutuyla seri iletişimi kapatır.

Seri iletişim protokolüne data paketi gönderen programlar ve data paketi içeriği Çizelge 5.1’ de verilmiştir.

Çizelge 5.1 İletişim safhalarında seri iletişim komutları ve içerdikleri veri paketi ve veriler.

SAFHA	Komut	Veri paketi
A	<i>GUI_startArduinoCommunication</i>	1,2,3,4
B	<i>WriteStartCommand</i>	4,3,2,1
C	fprintf(1,2,3,4)	1,2,3,4
D	<i>Super_Sent_Encoder_Data</i>	HEADER, R-Encoder Output, dTs, RightEncoderRead, RightEncoderRead, TR-END, HEADER, L-Encoder Output, dTs, LeftEncoderRead, LeftEncoderRead, TR-END
E	GUI_sendArduinoMOT	I1,I2,I3,I4, SpeedL,SpeedR
E	GUI_StopWMR	0,0,0,0,0

Buradaki paketteki veri adları ve açıklamaları şöyledir.

HEADER : Veri paketinin başlığıdır, verilerin belirli düzende alınmasını sağlar.

R-Encoder Output : Kendisinden sonra gelecek olan sayısal verinin hangi enkodere ait olduğunu belirten karakterdir.

L-Encoder Output : Kendisinden sonra gelecek olan sayısal verinin hangi enkodere ait olduğunu belirten karakterdir.

dTs : Çevrim zamanını veren tamsayı değeridir.

<i>RightEncoderRead</i>	:	Sağ enkoder verisinin tamsayı değeridir.
<i>LeftEncoderRead</i>	:	Sol enkoder verisinin tamsayı değeridir.
<i>TR-END</i>	:	Enkoder verisinin tamamlandığını belirten karakterdir.
<i>I1</i>	:	Sağ motorun yön verisidir.
<i>I2</i>	:	Sağ motorun fren verisidir.
<i>I3</i>	:	Sol motorun yön verisidir.
<i>I4</i>	:	Sol motorun fren verisidir.
<i>SpeedL</i>	:	Sol motorun hız verisidir.
<i>SpeedR</i>	:	Sağ motorun hız verisidir.

5.4 Sonuç

Kontrol mimarisi bölümüyle platformun elektronik devresi kurulumu oluşturuldu ve platform donanımsal olarak veri toplamaya ve hesaplama bilgisayarına veri göndermeye hazır duruma getirildi. Mikrokontrolcü tarafındaki komutlar oluşturularak enkoderlerdeki verilerin okunması işlenmesi ve gönderilmesi için komutlar oluşturuldu. Hesaplama bilgisayarı tarafında mikrokontrolcünden gelecek odometre verilerini okumak, kaydetmek ve motor komutlarını mikrokontrolcüye iletmek için MATLAB programında fonksiyonlar geliştirildi. Hesaplama bilgisayarı ile mikrokontrolcü arasında gerçekleşecek veri akışı ve iletişimin nasıl gerçekleşeceği akış diyagramıyla detaylı bir biçimde anlatıldı.

Böylece kontrol mimarisi konusuyla, oluşturulan hareketli platformun hareket edebilmesi veri toplayabilmesi için gerekli devrenin yapılması ve yazılımın oluşturulması gerçekleşti.

6 DENEYSEL ÇALIŞMALAR

6.1 Giriş

Bu bölümde hareketli platformun bir yörünge etrafında hareket edebileceği, hareketi esnasında robottan odometre verilerinin alınabileceği bir test ortamı oluşturuldu. Bu ortamda kullanılacak olan kameranın görüş alanı belirlendi ve görüş alanı içerisinde test alanı oluşturuldu. Platformun hareketlerini takip etmek için nesne takip deseni tasarlandı. Kameranın daha doğru ölçümler alabilmesi için kalibrasyonu yapıldı.

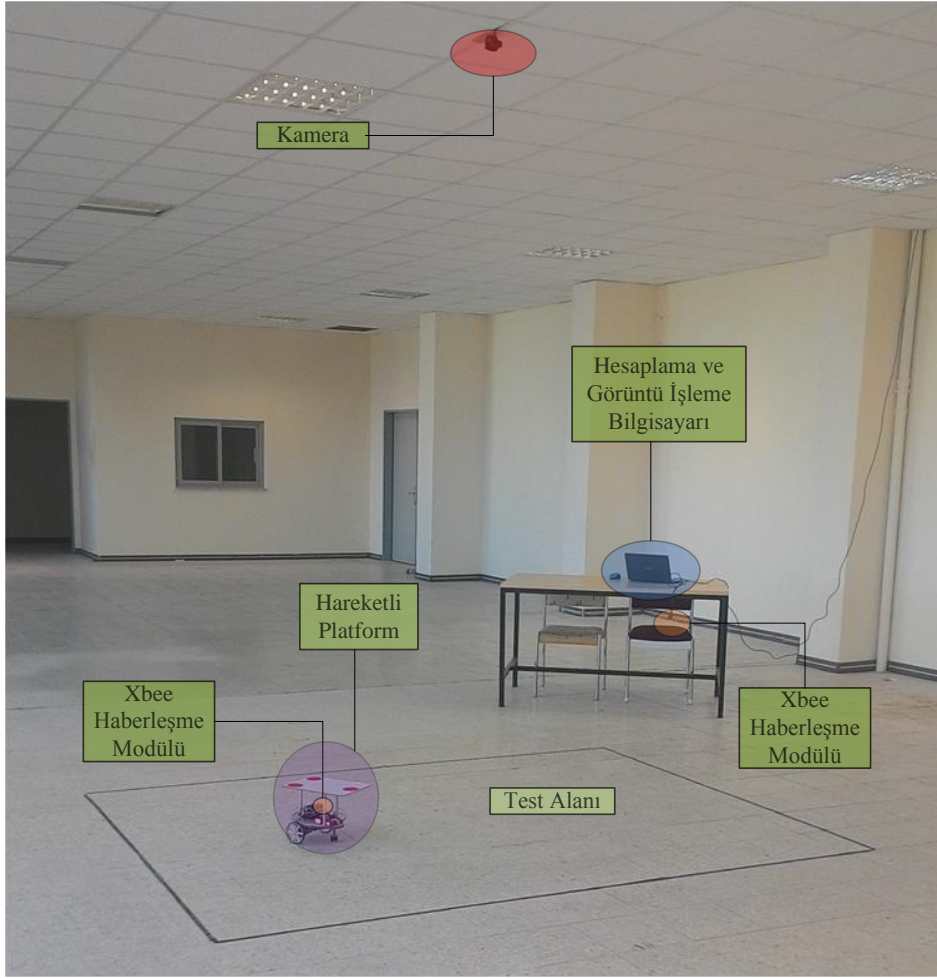
Bölüm 6.1’de yapılan deneysel çalışmalardan genel olarak bahsedildi. Bölüm 6.2’de test düzeneğinin nasıl oluşturulduğu anlatıldı. Bölüm 6.3’te test düzeneğinin önemli parçası olan kameranın daha doğru veriler alabilmesi için nasıl kalibre edildiğinden bahsedildi. 6.4 sonuçlar bölümüne deneysel çalışma sonrasında elde edilen kazanımlardan bahsedildi.

6.2 Test Düzeneği

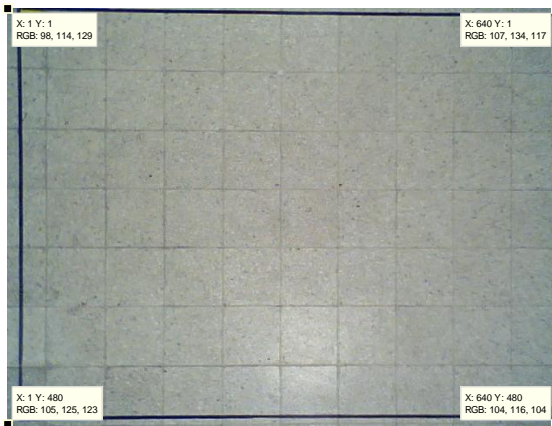
Hareketli platformun mutlak yeri ve yönünü bulmak için laboratuvarında düzenek geliştirildi. Hareketli platformun mutlak konum ve yönünü bulmak için 640x480 çözünürlükte bir kameradan faydalanıldı. Kamerayı laboratuvarın tavanına sabitlendi. Kameranın görüş alanı belirlenerek hareketli platformun test esnasında kamera görüş alanından çıkmaması için test alanı laboratuvar zeminine siyah bant ile işaretlendi. Hareketli platformun izleyeceği yörünge test alanı dışına çıkmaması için test başlangıç noktası hareketli platformun izleyeceği yörünge temel alınarak test alanında işaretlendi. Kameranın sabitlendiği yer ile hesaplama ve görüntü işleme bilgisayarı arasındaki mesafenin uzun olması kameranın kablo bağlantısına, usb uzatma kablosu ilave edilerek hesaplama ve görüntü işleme bilgisayarı olarak kullanılacak olan dizüstü bilgisayar ile kamera arasındaki bağlantı kuruldu. Şekil 6.1 ‘de oluşturulan test düzeneği görülür.

Şekil 6.1’den görüldüğü gibi test alanı siyah bantlarla işaretlenmiş olarak görülmekte. Referans alabilmemiz için siyah bant zemindeki karo çizgilerinden geçirilerek referans alabileceğimiz bir çerçeve oluşturuldu. Şekil 6.2A’da kalibrasyonu yapılan kameradan alınan ilk görüntüde görüş alanının kenar noktalarının piksel değerleri belirlendi. Şekil 6.2.B’ de siyah bantla işaretlenen test alanının köşelerinin ve daha önceden işaretlene platformun başlangıç noktasının piksel değerleri bulundu ve bu değerler kalibrasyon değerleri ile işlenerek kamera görüş alanının, test alanının ve platform başlangıç ölçüsel değerleri

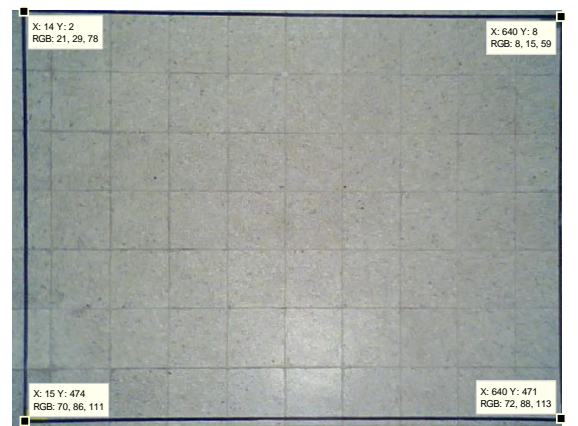
milimetre olarak bulundu. Şekil 6.3'te görüş alanı, test alanı, test başlangıç yeri ve kameranın orjin noktası şematik olarak gösterilmiştir.



Şekil 6.1 Test düzeneği

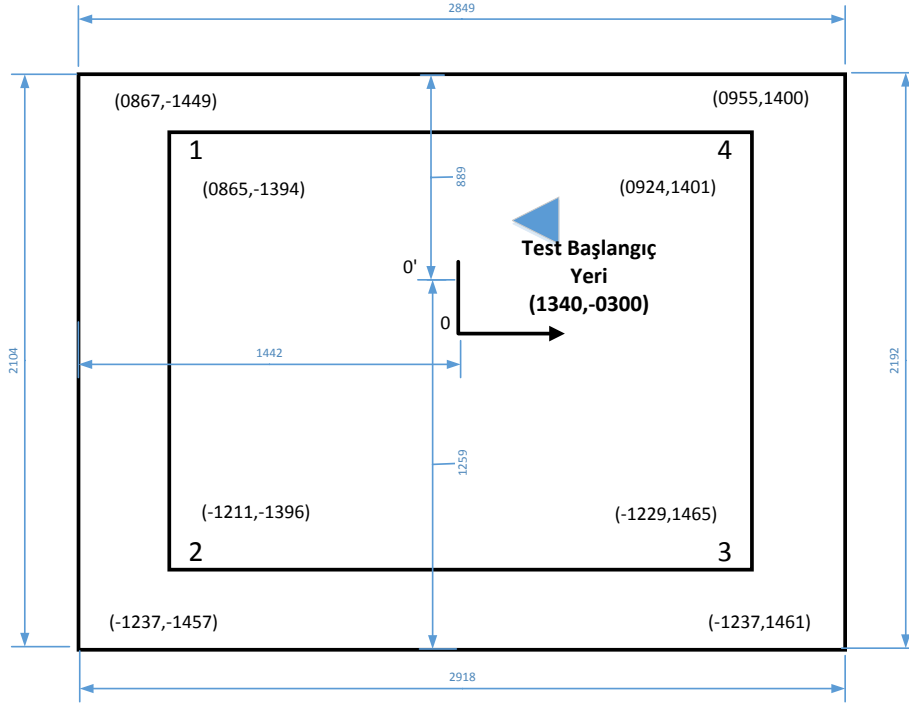


A



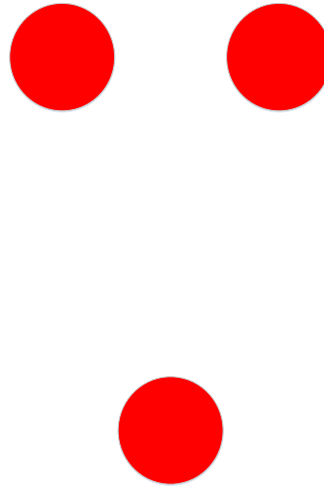
B

Şekil 6.2. : Kamera görüş alanı (A) ve test alanı köşelerinin (B) piksel değerleri



Şekil 6.3. : Görüş alanı, test alanı, test başlangıç yeri ve kamera orjini koordinatları

Şekil 6.4'te nesne takibi yapılacak desen gösterilmiştir. Desende üç tane farklı merkezde daireler mevcuttur. Daireler, merkezleri birleştirdiğimizde ikizkenar bir üçgen verecek şekilde konumlandırıldı. Daireler ortamdaki diğer nesnelere rahatlıkla ayırt edilebilmesi için kırmızı renklidir. Görüntü işleme sürecinde program kırmızı daireleri algılar ve bu renkleri görüntüden çıkararak diğer çevre renklerini de kapatır. Böylece siyah beyaz görüntü oluşur. Bu görüntüde çıkarılan yerler beyaz renkli, çıkarılmayan renkler de siyahtır. Çıkarılan dairelerin orta noktalarını bularak bu üç merkezin ağırlık ortalamasını alır ve bu noktayı takip eder.

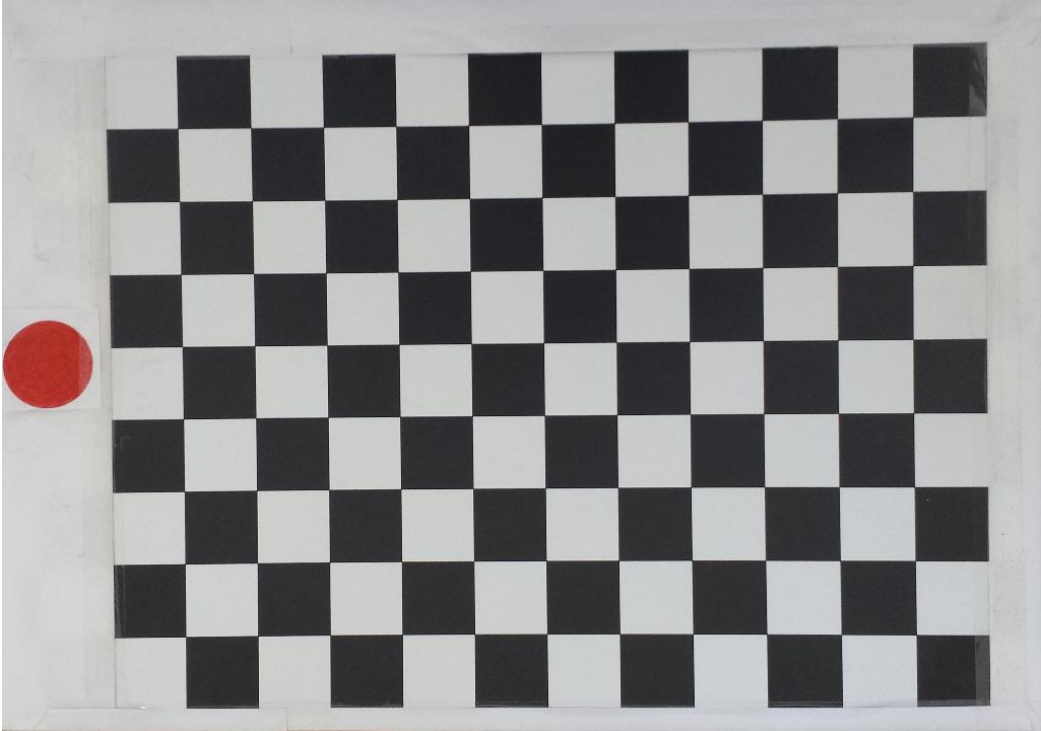


Şekil 6.4: Nesne takip deseni

6.3 Kamera Kalibrasyonu

Test düzeneğinde bulunan tavana bağlanmış kameradan sağlıklı ölçümler alabilmek için kalibre edilmesi gerekmektedir. Kalibrasyon yöntemi daha önceden belirlenmiş asimetrik desen üzerinden gerçekleştirilir. Bilgisayarda yazılan kalibrasyon programı yardımıyla desenler arasındaki renk değişimleri programa tanıtılır ve bu desenler arasındaki mesafelerde yazılıma girildikten sonra, yazılım, kullanılan kameranın parametrelerini bize verir bu parametreler bir cismin ölçüsünü kameradan belirleme, bir cismin orta noktasını bulabilme, ve hareketli bir cismin yörüngesini ölçüsel olarak bulabilmemizi sağlamaktadır.

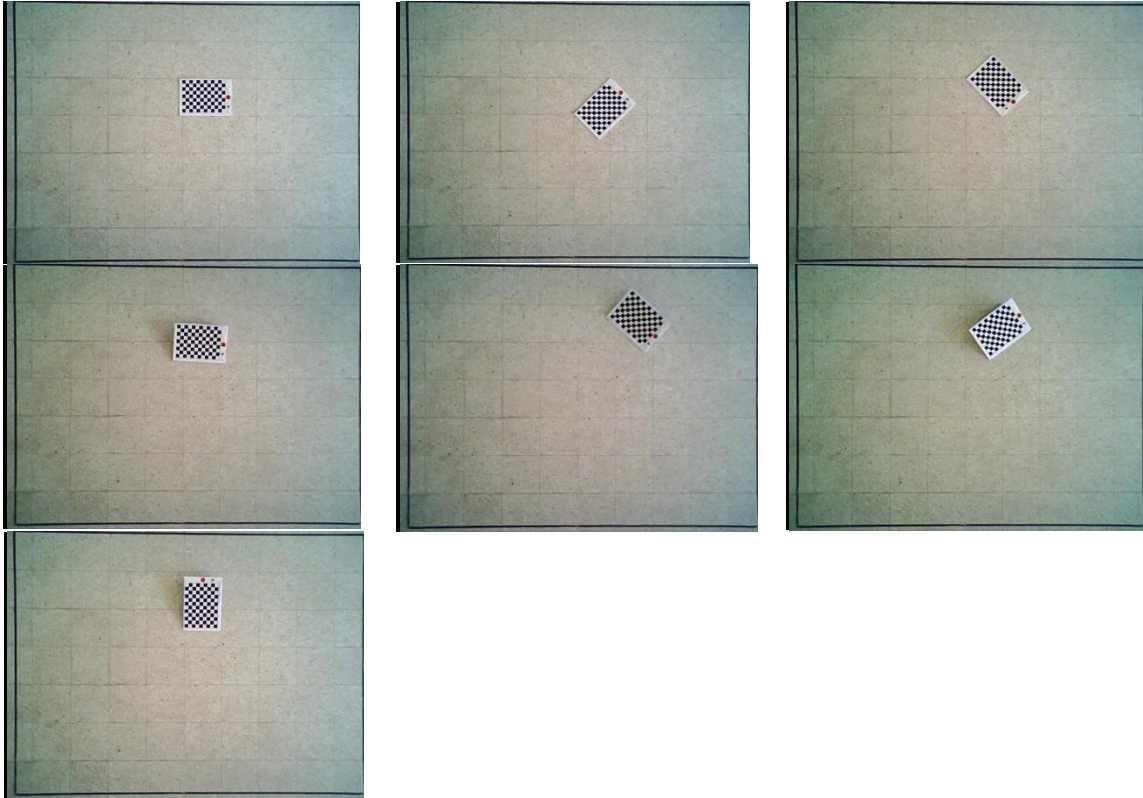
Biz bu çalışmada A3 kağıdı ebatlarında 12 satır 9 sütunlu kare desenli asimetrik kalibrasyon tablosu kullandık. Şekil 14'te kalibrasyon tablosunu görebilirsiniz. Kalibrasyon tablosundaki kare kenarları 29.5 mm ebatlarındadır, desenlerin biraz uzağında kırmızı renkli bir doğrulama şekli vardır. Bu doğrulama şekli, kalibrasyon süreci bittikten sonra kameranın ne kadar hassas ölçü alabildiğini görmek içindir.



Şekil 6.5. : Kalibrasyon tablosu

Kalibrasyon işlemini Şekil 6.6'da görebilirsiniz. Kalibrasyon tablosu ilk olarak zemindeki karolara paralel olacak şekilde yerleştirilip görüntü alınmış daha sonraki iki görüntüde kalibrasyon tablosu karoyla 45° açı yapacak şekilde yerleştirilip görüntü alınmıştır.

Dördüncü görüntüde kalibrasyon tablosu zemine açılı olarak yerleştirilmiş ve aynı zamanda karolara paralelliği korunarak görüntü alınmıştır, beşinci ve altıncı görüntüler karoyla 45° açı yapacak şekilde alınmıştır. yedinci görüntüde kalibrasyon tablosu karoyla 90° açı olacak şekilde görüntü alınmıştır. Alınan bu görüntüler sonunda yazılımla kalibre parametre bulunmuştur. Bulunan bu parametreler kullanılarak birinci görüntüdeki kırmızı renkli şeklin çapı 0.1 mm hassasiyetle ölçülmüştür.



Şekil 6.5 : Test düzeneğinin kalibrasyonu(üst bölüm soldan sağa : 1. 2. 3. görüntüler ; orta soldan sağa : 4. , 5. , 6. görüntüler ; alt: 7. görüntü)

6.4 Sonuç

Deneysel çalışmalar bölümünde test düzeneği oluşturularak hareketli platformdan odometre verilerinin ve kameradan mutlak konum ve yön verilerinin alınabileceği ortam meydana getirildi. Test düzeneğindeki kameraya yapılan kalibrasyon işlemiyle kameranın yüksek doğrulukta ölçüm alabilmesi sağlandı. Nesne takip deseninin tasarımıyla platform hareketlerinin görüş alanında belirgin biçimde ayırt edilebilmesi gerçekleştirildi.

7. TEORİ VE TOPLANAN VERİLERİN ANALİZ EDİLMESİ

7.1 Giriş

Hareketli platformun tasarlanması, donanımlarının bağlanması ve test düzeneğinin kurulmasıyla, platformdan veri toplanması ve toplanan verilerden kinematik parametrelerinin tahmin edilebilmesi için hazır duruma getirildi. Hareketli platformun kinematik parametre tahmini için kullanılacak metot bölüm 7.2 teori kısmında açıklanmıştır. Yığın LSE yöntemi bölüm 7.2.1'de açıklanmıştır. Odometre ve kamera verilerini kullanarak analiz yapılacak başka bir yöntemde bölüm 7.2.2 ardışık değişken tahmini metodu açıklanmıştır. Bölüm 7.3 Veri analizi bölümünde test düzeneğinden verilerin elde edilmesinden bahsedilmiştir. 7.3.1, 7.3.2, 7.3.3, 7.3.4 bölümlerinde yapılan dört farklı testin odometre ve kamera veri sonuçları grafiklerle gösterilmiştir. Kalibrasyon öncesi ve kalibrasyon sonrası durumlar grafiklerle gösterilip yorumlanmıştır.

7.2 Teori

Bu kısımda IEEE/ASME AIM 2014 Besacon konferansında yayınlanan teorik çalışmanın [7] bu tez kapsamında nasıl kullanılacağı açıklanacak. Hareketli robotun kinematik modelinde ki belirsiz geometrik değişkenler EKF süzgecinin kararlı tahmin yapmasını engeller. Bundan dolayı, kaynak [7] da mobil platformun belirsizliği hakkında değişik hipotezlere dayanan birden fazla EKF süzgeci paralel olarak çalıştırıldı ve bu EKF süzgeçlerin çıktısı olan konum tahmini istatistiksel yöntemlerle optimize edildi. Böylece, paralel çalışan EKF süzgeç grubu tek bir EKF süzgecinden daha kararlı bir yapıya sahip olur ve konum tahmini sağlıklı bir şekilde yapılır. Bu tip paralel çalışan birden fazla EKF süzgeci İngilizce de şu bilimsel terim ile ifade edilir; Multiple Model Adaptive Estimation ve kısaca MMAE olarak anılır. Bu çalışmada kullanılan diğer bir metot ise İngilizce literatür de Least Square Estimation olarak bilinmekte ve kısaca LSE olarak bilinmektedir.. Bu konu kapsamında İngilizce literatür de kabul görmüş olan ve yaygın bir biçimde kullanılan MMAE ve LSE kısaltmaları kullanılacaktır. Bu tip çoklu EKF süzgeçleri ilgili bilgi şu kaynak kitapta detaylı olarak bulunabilir [41]. Bahsi geçen makalede [7] çalışmanın ikinci kısmında ise, MMAE metodu ile elde edilen doğru konum verisi ve odometre verisi LSE metodu ile analiz edilerek kinematik modelin değişkenleri hesaplanır. Ancak makalede [7], metot simülasyon üzerinde sentetik veri ile test edilir. Bu çalışma kapsamında benzer bir problem tanımlanmıştır. Bu tezin esas hedeflerinden birisi olan değişken tahmini için test düzeneği ve bir örnek hareketli platformun nasıl yapılacağı konu

6.2'de ve 4.2.1'de detaylı anlatılmıştır. Bu konuda ise elde edilen ham verilerin nasıl işleneceği çalışılacaktır. Bu tez, kaynak [7] ye iki katkı sunmaktadır, bir test platformunun ve veri toplama biriminin tasarımı iki kaynak [7] de tasvir edilen metodun gerçek veri ile denenmesi. Burada anlatılan teorik bilgi kaynak [7] de anlatılan teorik bilginin ikinci kısmına benzer ancak teorik çalışmada [7], konum tahmini mobil platformun çalışma alanında bulunan noktasal cisimlere yapılan mesafe açısı ölçümleri ve odometre verisi MMAE metodu ile islenir ve mutlak konum bulunur. Bu tez de ise mutlak konum, çalışma alanının tavanına yerleştirilen bir USB kamerası ile bulunur. Dolayısı ile MMAE metodunun kullanılmasına gerek kalmaz. Aşağıda kaynak [7] da anlatılan yaklaşım tezin bütünlüğünü ve anlaşılabilirliğini artırmak için tekrar verildi.

Bu kısımda anlatılan denklemler hareketli platformun mutlak konumunun bilindiği varsayımına dayanır. Bu varsayım, çalışmada tavana yerleştirilen USB kameradan alınan resimlerin islenerek konum tahmini yapılmasına dayanır. Örnek yörüngeler, bu konunun ilerleyen kısımlarında verilir, kamera kalibrasyonu ise konu 6.3'de anlatıldı.

Hareketli platformun ileri ve ters kinematik modeli konu 3.3.1'de detaylı bir şekilde ele alındı. Bu konuda ileri kinematik f değişkeni ile gösterilir. Değişkenin ifade ettiği kinematik model konu 3.3.5 denklem 3.21 de bulunabilir. Denklem 3.21 mükemmel bir kinematik modeli ifade eder. Bu konuda ileri kinematik argümanları olan x ve u bilinmeyen geometrik değişken α vektörünün de fonksiyonu olarak tekrar ifade edilir. Böylece yeni kinematik model ve komut vektörü şu şekilde yazılır.

$$\dot{x} = f(x(\alpha), u(\alpha)) \quad (7.1)$$

Burada x modelin durum vektörünü ve $f_k(\cdot)$ ayrık zaman doğrusal olmayan modelini, k ayrık zamani ifade eder. Komut vektörü zamana ve bilinmeyen parametre vektörü α 'ya bağlı bir fonksiyondur.

$$u = [v(\alpha) \ \omega(\alpha)]^T \quad (7.2)$$

Komut vektörünün bileşenleri, doğrusal hızı , $v_k(\alpha)$ ve hareketli platformun açısal hızı $\omega_k(\alpha)$ aşağıdaki denklemlerle hesaplanır.

$$v(\alpha) = \frac{[1+\mu_R]v_R + [1+\mu_L]v_L}{2} \quad (7.3)$$

$$\omega(\alpha) = \frac{[1+\mu_R]v_R - [1+\mu_L]v_L}{[1+\mu_D]D} \quad (7.4)$$

Burada v_k ve ω_k argumani olan α soyle ifade edilir

$$\alpha = [\mu_R \mu_L \mu_D] \quad (7.5)$$

Yukarıdaki vektör tanımında , α vektörünün bileşenleri olan μ_R, μ_L, μ_D , sırasıyla sol tekerlek, sağ tekerlek ve teker açıklığının bilinmeyen ölçek faktörleridir.

İlk olarak, hareketli platforma bir yörünge komuta edilir ve platformun gerçek konumu kamera ile takip edilir.

Bir kere kamera ile belirlenen doğru yörünge kullanılarak, bilinmeyen odometre ölçek parametreleri LSE algoritması kullanarak tahmin edebilir [7] . LSE aşamasında, hesaplamaları basitleştirmek için basit bir kinematik model kullanılır.

$$\frac{dx}{dt} = f(u(\alpha, t), x(\alpha, t), t) \quad (7.6)$$

Burada , f ile ifade edilen doğrusal olmayan kinematik model şöyle yazılır

$$f(u(\alpha, t), x(\alpha, t), t) = \begin{bmatrix} v(\alpha, t) \cos(\theta(\alpha, t)) \\ v(\alpha, t) \sin(\theta(\alpha, t)) \\ \omega(\alpha) \end{bmatrix} \quad (7.7)$$

Durum vektörü x 'in bilinmeyen odometre parametrelerine hassasiyetini bulmak için, durum vektörünün α ' ya göre kısmi türevi alınır.

$$\frac{\partial^2 x}{\partial \alpha \partial t} = \frac{\partial f(\cdot)}{\partial u(\cdot)} \frac{\partial u(\cdot)}{\partial \alpha} + \frac{\partial f(\cdot)}{\partial x(\cdot)} \frac{\partial x(\cdot)}{\partial \alpha} \quad (7.8)$$

Ayrık zaman alanında, x durum vektörünün α bilinmeyen model parametrelerine göre kısmi türevinin açılımı şu şekilde yazılabilir:

$$X_{(k+1)\alpha} = [I + F_{(k)x} \Delta t] X_{(k)\alpha} + F_{(k)u} U_{(k)\alpha} \Delta t \quad (7.9)$$

Yukarıdaki denklem ardışık olarak birden fazla kere kullanılarak, sabit alfa değişkenin konum üzerindeki etkisi ileri zaman adımlarına taşınabilir. k ayrık zaman adımları için, yukarıdaki formül aşağıdaki gibi yazılabilir.

$$X_{(k+1)\alpha} = \prod_{r=0}^k [I + \Delta t F_{(r)x}] X_{(0)\alpha} + \sum_{\rho=0}^k [\prod_{r=\rho+1}^k [I + \Delta t F_{(r)x}]] F_{(\rho)u} U_{(\rho)\alpha} \Delta t \quad (7.10)$$

Yukarıdaki denklemde bahsi geçen değişkenlerin tanım kümesi ve boyutları aşağıda kısaca verildi.

$\alpha \in R^l$, $x \in R^m$, $u \in R^n$, $X_a \in R^{m \times l}$, $U_a \in R^{n \times l}$, $F_u \in R^{m \times n}$, $F_x \in R^{m \times m}$ ve $I \in R^{m \times m}$;

$l = 3$, $m = 3$, $n = 2$ ' dir.

Denklem 7.10 da kullanılan ve Jakobian matrisini ifade eden kısa yazım sekli $X_{(k)y}$ aşağıda detaylı bir şekilde açıklanmıştır.

$$X_{(k)y} = \left[X_{ij} = \left[\frac{\partial x_i}{\partial y_j} \right] \in R^{m \times n} \mid x \in R^m, y \in R^n, t = k\Delta t \right] \quad (7.11)$$

İlk kısımda LSE metodunda kullanılmak üzere gerekli olan gözlem matrisi türetilmiştir. Bu matris denklem 7.10 verilir. Bir sonraki kısımda kameradan alınan mutlak konum ve odometreden alınan verilerin LSE metodu ile nasıl işleneceği ve geometrik değişkenlerin nasıl hesaplanacağı anlatılacak. Öncelikle yığın LSE metodu daha sonra ise ardışık LSE metodu tanımlanacak ve problemin çözümünde kullanım şekilleri üzerinde durulacak.

7.2.1 Yığın LSE Tahmini

Durum vektöründeki bozuklukları δx_k , sabit odometre parametrelerindeki hatalarının $\delta \alpha$ fonksiyonu olarak aşağıdaki gibi yazılabilir.

$$\delta x_k = [X_{(k)\alpha}] \delta \alpha \quad (7.12)$$

Burada $X_{(k)\alpha}$, x durum vektörünün α değişkenine olan hassasiyetidir. Çalışmanın bu kısmında δx_k değişkenini ifade eden sembol, problemin LSE metodunun genel yazım şekline uyarlanması için değiştirilecektir. Eğer z_k , δx_k yi ifade ederse, z_k şöyle yazılabilir.

$$z_k = \delta x_k \quad (7.13)$$

k anındaki Jakobyan matrisi ise aşağıdaki sembol ile ifade edilir

$$h_{(k)} = [X_{(k)\alpha}] \quad (7.14)$$

Böylece ardışık k ölçüm için denklem setini şöyle yazabiliriz.

$$Z_k = H_k \delta \alpha \quad (7.15)$$

Burada, Z_k ve H_k aşağıda verildiği biçimdedir;

$$Z_k = [z_{(k)}^T \ z_{(k-1)}^T \ \dots \ z_{(0)}^T]^T \quad (7.16)$$

ve

$$H_k = [h_{(k)}^T \ h_{(k-1)}^T \ \dots \ h_{(0)}^T]^T \quad (7.17)$$

Bilinmeyen parametre vektörü, denklem 7.18-7-19, $\delta\alpha_{(k)}$ degiskeni kararli bir deger alana kadar tekrar tekrar kullanılarak tahmin edilebilir.

$$\delta\alpha_{(k)(i+1)} = \delta\alpha_{(k)(i)} + C_{(k)(i)}Z_{(k)(i)} \quad (7.18)$$

$$C_{(k)(i)} = [H_{(k)(i)}^T [R_{(k)(i)}]^{-1} H_{(k)(i)}]^{-1} H_{(k)(i)}^T [R_{(k)(i)}]^{-1} \quad (7.19)$$

Burada , $H_{(k)(i)}$ ve $Z_{(k)(i)}$, $\alpha_{(k)(i)}$ k anındaki i iterasyondaki deđerlerdir. $\alpha_{(k)(i)}$ k zaman göstergesine kadar kaydedilen uygun verilerin kullanılmasıyla tahmin edilir. Bu iterasyon ařađıdaki kıstas gerekleřtiđinde durdurulur.

$$\left| \frac{\delta\alpha_{(k)(i+1)} - \delta\alpha_{(k)(i)}}{\delta\alpha_{(k)(i)}} \right| < \varepsilon \quad (7.20)$$

Burada ε dođruluk iin belirlenmiř bir eřik deđeridir. Yukarıda LSE metodu iin yıđın tahmin yöntemi tarif edildi. Ancak kamera görüntülerinden elde edilen konum verisi ve odometreden elde edilen veriler ardışık olarak analiz edilebilir. Yazının devamında, kaynak [41] da tarif edilen ardışık LSE metodu kullanılmak üzere önerilir.

7.2.2 Ardışık Deđişken Tahmini

Durum vektöründeki bozuklukları δx_k , sabit odometre parametrelerindeki hatalarının $\delta\alpha$ fonksiyonu olarak ařađıdaki gibi yazılabilir.

$$\delta x_k = [X_{(k)\alpha}] \delta\alpha \quad (7.21)$$

Burada $X_{(k)\alpha}$, x durum vektörünün α deđişkenine olan hassasiyetidir. Eđer, bir önceki kısımda yapıldıđı gibi z_k $\delta\alpha_{(k)}$ 'yi ifade ederse, z_k şöyle yazılabilir

$$z_k = \delta x_k \quad (7.22)$$

k anındaki Jakobyian matrisi ise benzer bir şekilde ařađıdaki sembol ile ifade edilir

$$H_{(k)} = [X_{(k)\alpha}] \quad (7.23)$$

Böylece ardışık k anındaki ölçüm iin denklem setini şöyle yazabiliriz.

$$z_k = H_k \delta\alpha \quad (7.24)$$

Burada , z_k ve h_k k anında veri geldikçe aşağıda verildiği biçimde güncellenir;

$$z_k = [z_k] \quad (7.25)$$

ve

$$H_k = [h_k] \quad (7.26)$$

Yukarıdaki denklem seti ve aşağıdaki ardışık LSE metodu kullanıldığı zaman [41] bilinmeyen istatistiksel sabit $\delta\alpha_{k+1}$ şöyle hesaplanır.

$$S_{k+1} = H_{k+1}P_k H_{k+1}^T + R_{k+1} \quad (7.27)$$

$$W_{k+1} = P_k H_{k+1}^T S_{k+1}^{-1} \quad (7.28)$$

$$P_{k+1} = [I - W_{k+1}H_{k+1}]P_k \quad (7.29)$$

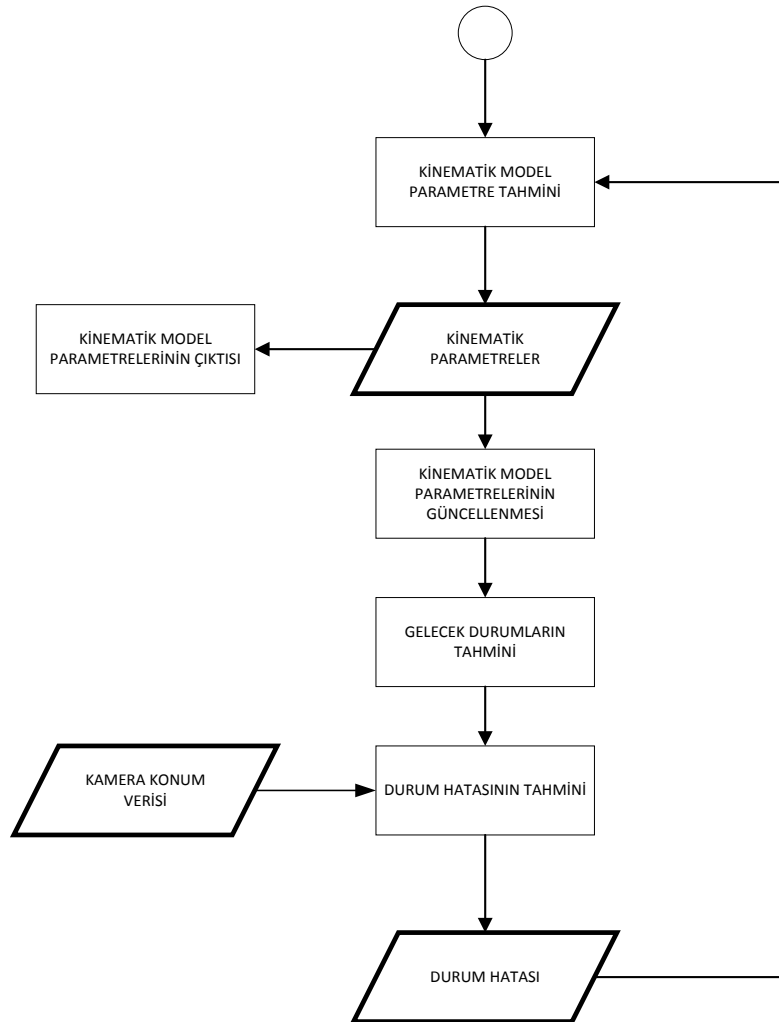
$$\delta\alpha_{k+1} = \delta\alpha_k + W_{k+1}[z_{k+1} - H_{k+1}\delta\alpha_k] \quad (7.30)$$

Yukarıda verilen denklemlerde P_k , $\delta\alpha_k$ nin k anındaki kovaryans matrisidir. R ise k anındaki ölçüm vektörünün kovaryans matrisidir.

Ardışık LSE metodu Şekil 7.1' de verilen akış şeması ile genel olarak tarif edilir. Önce kamera resimlerinden mutlak konum verisi alınır, daha sonra eldeki en iyi kinematik model ve o anda okunan odometre verisi kullanılarak konum tahmin edilir. Kameradan elde edilen mutlak konum ile kinematik model ve odometre kullanılarak elde edilen konum verisi arasında ki hata hesaplanır. Bu δx_k yı verir. Bu aşamada denklem 7.10 kullanılarak $X_{(k)\alpha}$ hesaplanır. İkinci kısımda denklem 7.15 kullanılarak ardışık LSE problemi hazırlanır. Bu problem denklem 7.25-28 kullanılarak çözülür ve böylece $\delta\alpha_k$ hesaplanır. İkinci adımda benzer olarak kameradan konum verisi alınır ve odometre ve kinematik model kullanılır konum hesaplanır ancak ilk basamakta yapılan hesaplardan farklı olarak, kinematik model ilk h adımda hesaplanan $\delta\alpha_k$ ile güncellenir. Bu güncellemeyi yapmak için denklem 7.3-4-10 kullanılır ve bütün Jakobyian matrisleri denklem 7.11 de verilen tanım ile güncellenir. Bu çalışmada Jakobyian matrisleri MATLAB symbolic toolbox kullanılarak hesaplandı. Daha sonra denklem 7.15 kullanılarak yeni bir ardışık LSE problem hazırlanır ve benzer biçimde problem denklem 7.25-28 kullanılarak çözülür.

Böylece çözümün bir döngüsü tamamlanır daha sonraki adımlar yukarıda anlatılan döngünün ikinci kısmının benzeridir.

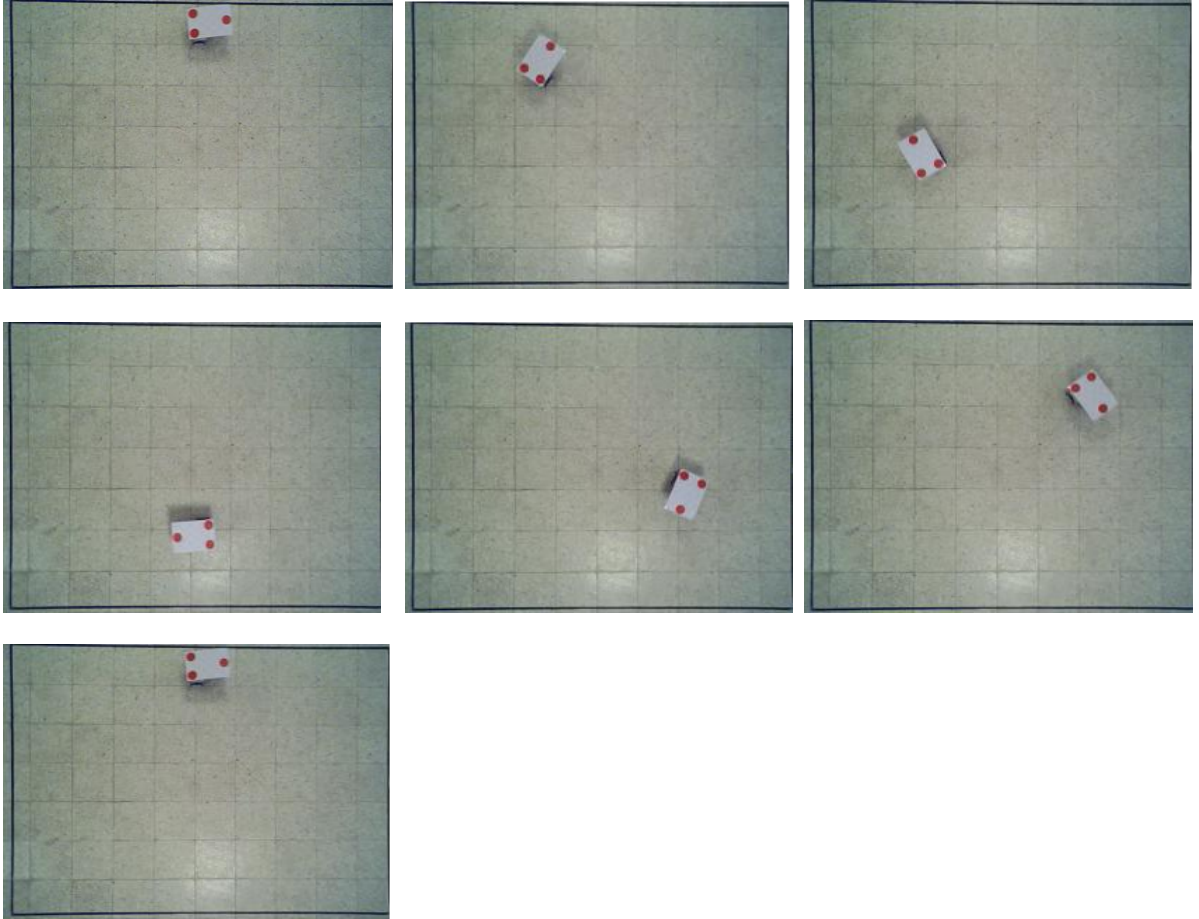
Bu konunun ilk kısmında kameradan elde edilen mutlak konum verisinin ve odometreden elde edilen verilerin nasıl analiz edileceği ve geometrik değişkenlerin nasıl hesaplanacağı soyut bir şekilde ele alındı. Bir sonraki alt konu başlığında, önce USB kamera tarafından kayıt edilen videodan alınan örnek resimler ve bu resimlerle eşleşen odometre verileri, basit bir eşik değeri testi ile konum tahmini sonuçları verilecek. Son kısımda ise test düzeneğinde elde edilen veriler sunulacak ve analiz sonuçları grafikler ile gösterilecek. Bu kısım da önerilen çözüm yöntemi doğrulanacak ve sonuçlar tartışılacak. Bu konunun son alt başlığı olan sonuç kısmında ise yöntem ve deneysel sonuçlar genel olarak tartışılacak ve konu böylece bitirilecek.



Şekil 7.1: LSE akış şeması

7.3 Veri Analizi

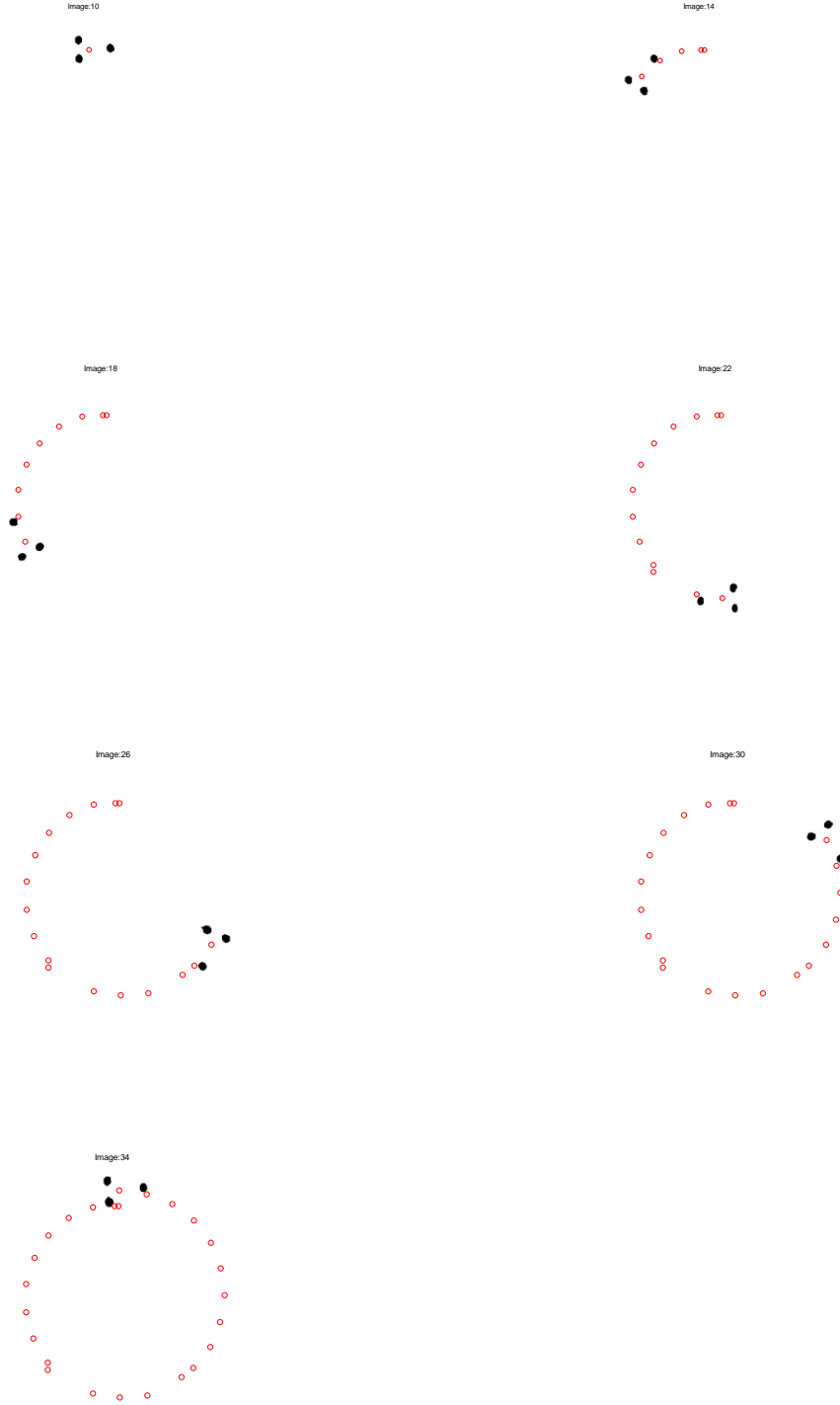
Bu bölümde hareketli platformumuzla test verilerini toplamak için yörüngesi kamera görüşünün içinde kalacak şekilde daha önceden belirlediğimiz başlangıç noktasına yerleştirilir ve devresine güç verilerek çalışır vaziyette bekler. Hesaplama ve görüntü işleme bilgisayarına kamera ve kablosuz iletişim modülünün bağlantısı kurulduktan sonra bilgisayardaki programda çalıştırılır. Bilgisayar içerisinde tanımlı olan yörünge hareketli platforma çizdirilir. Hareketli platformun çalışması esnasında platformdan gelen enkoder verileri sıralı şekilde kayıt altına alınır bu esnada test düzeneğindeki kameradan kamera takibi yapılarak robotun konumları kaydedilir. Kaydedilen bu veriler veri analizinde kullanılacaktır.



Şekil 7.2 Mobil platformun test ortamında yörüngesini çizerken alınan kamera görüntüsü.

Şekil 7.2 de test aşamasında hareketli platform yörüngesini tamamlarken kameradan alınan görüntüleri görebilirsiniz. Kamera tarafından test esnasında alınan bu görüntüler daha

sonra görüntü işleme yöntemiyle görüntülerde sadece kırmızı daire görecek şekilde işlenir ve bu kırmızı dairelerin ortalaması alınarak mobil platformun izlediği yörünge çıkarılır.



Şekil 7.3: Nesne takibi yöntemiyle oluşturulan robot yörüngesi.

Şekil 7.3' de Hesaplama ve görüntü işleme bilgisayarı tarafından işlenen görüntülerin, kırmızı dairelerle robot yörüngesini çizdiğini görebilirsiniz. Bu fotoğraflarda görünen siyah daireler platform üzerindeki nesne takip desenindeki şekillerdir. Üç siyah dairenin ortasındaki kırmızı daire ise bu noktaların ağırlık merkezidir. Bu merkeze göre robot yörüngesi çizdirilir.

7.3.1 Veri Analizi Test 1

Hazırlanan platformla yapılan ilk testte robotun üzerinde bulunan tekerlek çapları ve tekerleğin oluşturduğu açıklık aşağıda verilmiştir.

Nominal ölçüler

$$D_r = 90 \text{ (mm)},$$

$$D_l = 90 \text{ (mm)},$$

$$b = 190 \text{ (mm)},$$

Başlangıç belirsizlik matrisi $\mu_{rC} = 1 \times 10^{-6}$, $\mu_{rIC} = 1 \times 10^{-6}$, $\mu_{lC} = 1 \times 10^{-6}$ olarak atanır.

$$\begin{bmatrix} \mu_{rC} & 0 & 0 \\ 0 & \mu_{rIC} & 0 \\ 0 & 0 & \mu_{lC} \end{bmatrix} = \begin{bmatrix} 1 \times 10^{-6} & 0 & 0 \\ 0 & 1 \times 10^{-6} & 0 \\ 0 & 0 & 1 \times 10^{-6} \end{bmatrix}$$

Ölçüm Hataları : $k_s = 0.1$

Odometreye bağlı olan konum tahmin hatası zamanla artıyor. Ölçüm belirsizliği doğrusal olarak artar.

$$S_x = (1 + i \cdot k_s) \cdot 10$$

$$S_y = (1 + i \cdot k_s) \cdot 10$$

$$S_t = (1 + i \cdot k_s) \cdot 10^{-1}$$

Test verisinin ilk yüz verisi kullanılarak kalibrasyon değerleri aşağıdaki gibi çıkarılmıştır.

$$\mu_r = 0.0164 ,$$

$$\mu_l = 0.0387,$$

$$\mu_b = 0.0088$$

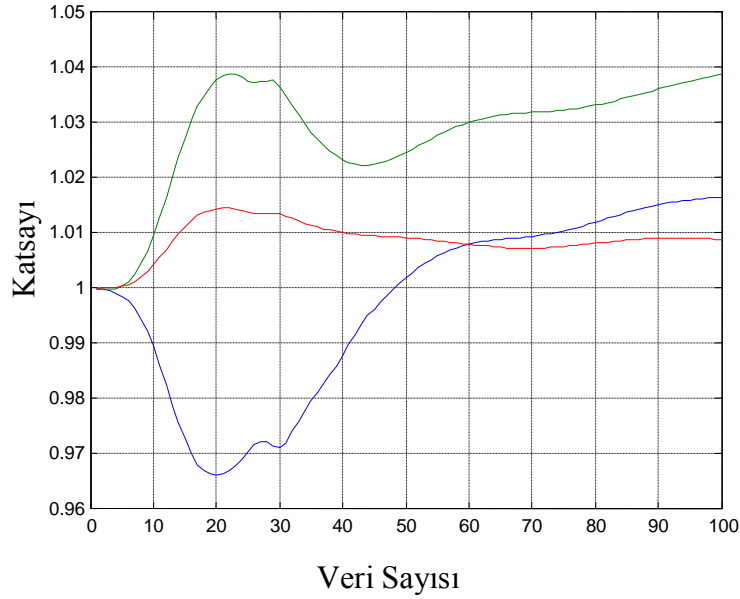
Bu değerlere göre yeniden oluşturulan sağ tekerlek, sol tekerlek çapları ve teker açıklığı aşağıdaki gibi düzenlenmiştir.

$$D_r = 91.4798$$

$$D_l = 93.4856$$

$$b = 191.6801$$

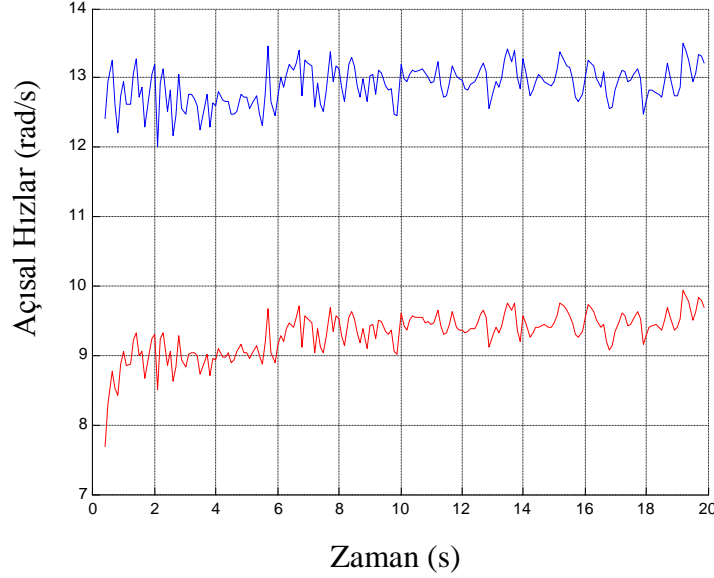
Kalibrasyon değerleri program tarafından ilk yüz noktaya göre hesaplanır, Şekil 7.4'te kalibrasyon değerlerinin hesaplanan nokta sayısına göre değişimi verilmiştir. Yeşil çizgi ile μ_l , sol teker kalibrasyon değeri ilk 60 veri hesabında salınım gösterdikten sonra doğrusal olarak son değerine yaklaşmış. Mavi çizgiyle gösterilen μ_r , sağ teker kalibrasyon değeri ilk 30 nokta hesabında gerçek değerinden çok uzaklaşıyor ve 60 veri hesabından sonra gerçek değerine doğrusal olarak yaklaşmış. Kırmızı çizgiyle gösterilen μ_b değerinde fazla bir sapma gerçekleşmiyor doğrusal olarak son değerine yaklaştığını grafikten görülebilir.



Şekil 7.4: TEST 1 parametre hata yaklaşımı

Şekil 7.5'te birinci testte elde edilen sağ ve sol tekerleğin açısal hızları görülmektedir. Mavi çizgiyle gösterilen sağ tekerlek hızı yaklaşık 13.2 rad/s dir. Kırmızıyla gösterilen sol tekerlek açısal hızı 9.6 rad/s olarak grafikte görülmektedir. Enkoder verileri eşit zaman aralıklarında alınmadığından dolayı grafik titreşimli görülmektedir. Açısal hızlardaki bu

fark yüzünden hızlı dönen tekerlek dışarıda kalacak şekilde platform daire çizecektir. $R = a(V_r + V_l)/(V_r - V_l)$ formülüyle platformun döneceği yarıçapı bulabiliriz. Burada a teker açıklığının yarısı, V_r sağ tekerin doğrusal hızı, V_l sol tekerleğin doğrusal hızıdır.

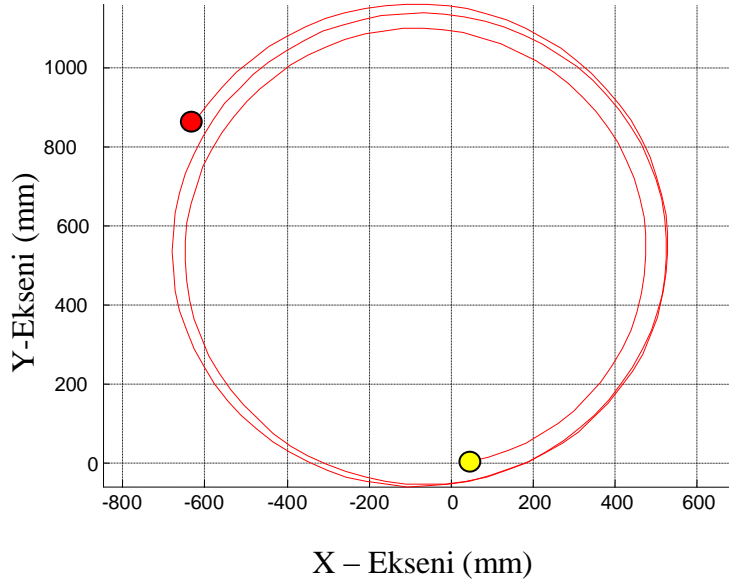


Şekil 7.5: TEST 1 Sağ ve sol tekerlek açısal hızları

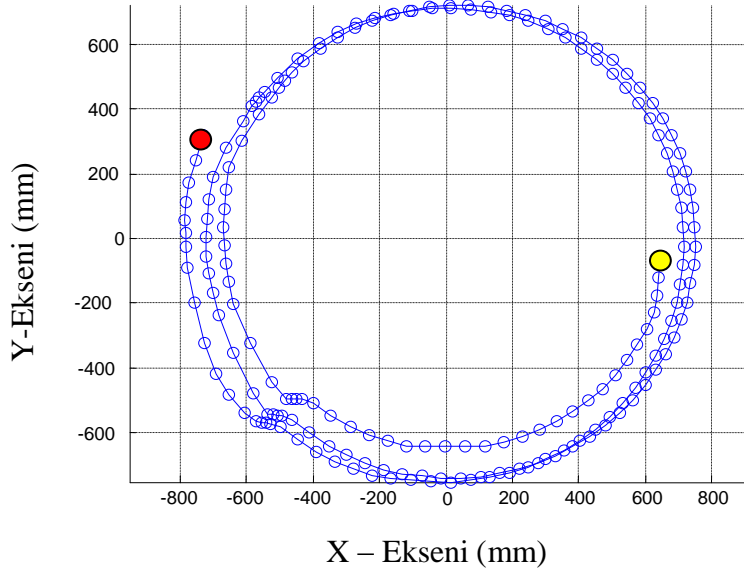
$$R = \frac{85 \left(13.2 \frac{90}{2} + 9.6 \frac{90}{2} \right)}{\left(13.2 \frac{90}{2} - 9.6 \frac{90}{2} \right)} = 538$$

Bu verilere göre platform dönüş çapı 1076 mm olacaktır. Bu değer odometre verisinin oluşturduğu çap değerine yakın olması beklenir.

Şekil 7.6 ve Şekil 7.7’de sırasıyla odometre ve kamera verilerinden elde edilen platform yörüngesi görülmektedir. Şekillerde içi dolu sarı daireyle gösterilen konum başlangıç konumu, içi dolu kırmızı daireyle gösterilen konum bitiş konumudur. Her iki şekilde de platformun dairesel yörüngesini gerçekleştirirken dönüş merkezinden uzaklaşarak daha büyük çapta dairesel bir yörünge takip ettiği görülmekte, bunun nedeni tekerleklerin dönüş esnasında ki kaymaları olabilir. Tekerleklerde oluşan bu kaymalar tekerleğin doğrusal hızının azalmasına neden olarak platformun bileşke hızının ve yönünün değişmesine sebep olmaktadır.



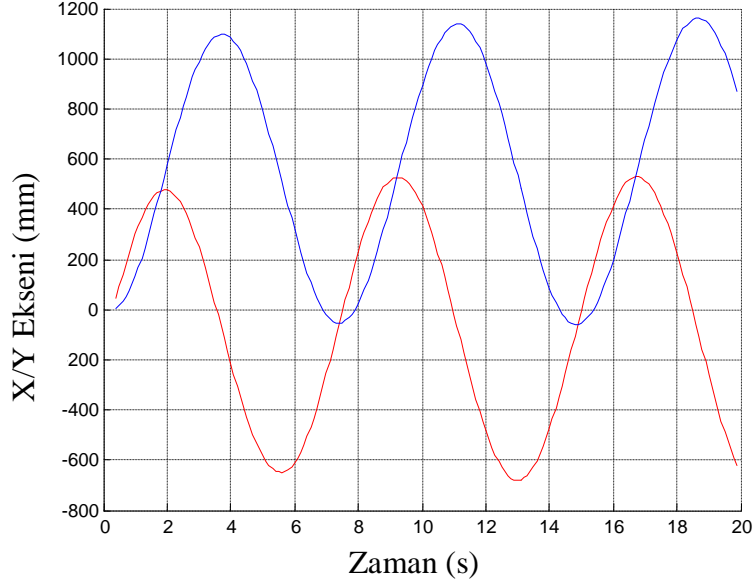
Şekil 7.6. :TEST 1 odometre verisine göre platform yörüngesi.



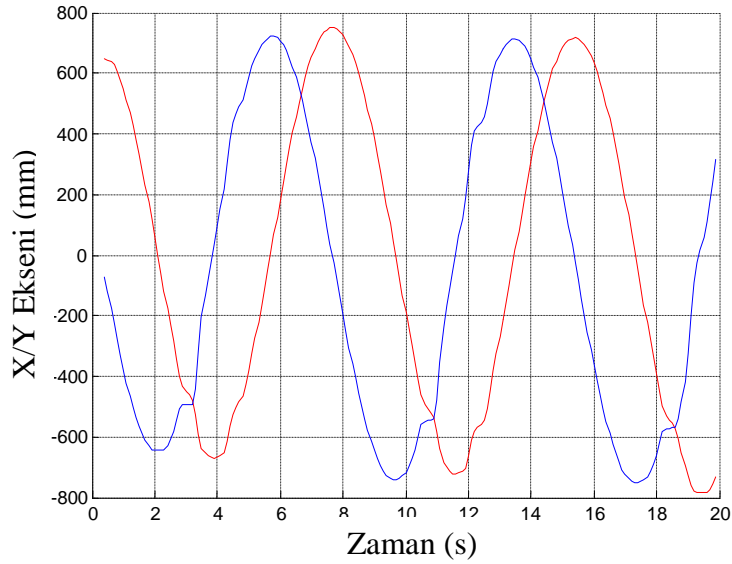
Şekil 7.7.: TEST1 kamera verisine göre platform yörüngesi.

Şekil 7.8 ve Şekil 7.9 ‘da sırasıyla odometre ve kamera verilerine göre platformun X ve Y eksenindeki konumları görülmektedir. Mavi çizgiyle gösterilen X eksenini kırmızı çizgiyle gösterilen Y eksenindeki konumlarıdır. Burada dikkat edilmesi gereken nokta robot başlangıç konumlarının odometre ve kamerada farklı olmasıdır bu yüzden konum başlangıç değerleri iki grafikte farklıdır. Bu grafiklerdeki X ve Y konum değişimlerine bakarak platformun yörünge çapını ve bu yörüngede kaç defa döndüğünü görebiliriz. Odometre verisinde X ve Y yönünde dönüş çapı yaklaşık olarak birinci turda 1100 mm,

ikinci turda 1250 mm'dir(Şekil 7.8). Kamera verisinde X ve Y yönünde dönüş çapları yaklaşık olarak birinci turda 1350 mm, ikinci turda 1425 mm'dir (Şekil 7.9). Her iki şekilden de robotun iki tam tur ve bir yarım tur gerçekleştirdiği görülür.



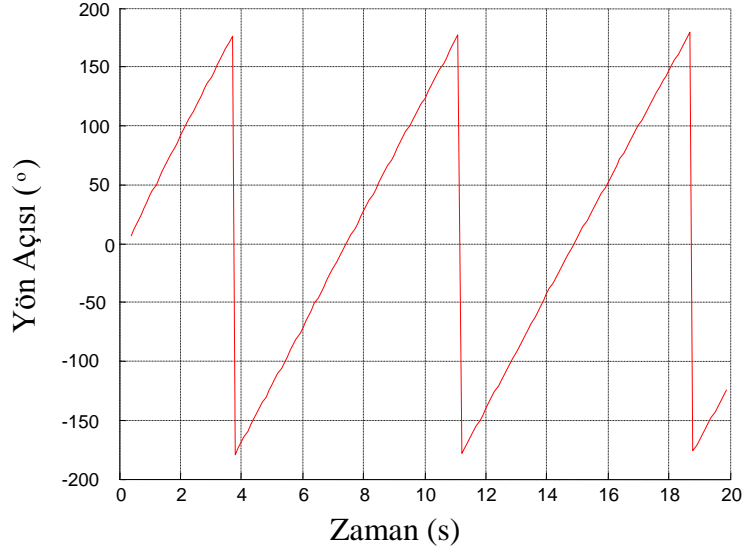
Şekil 7.8:TEST1 odometre takibinde platform X/Y eksen konumu.



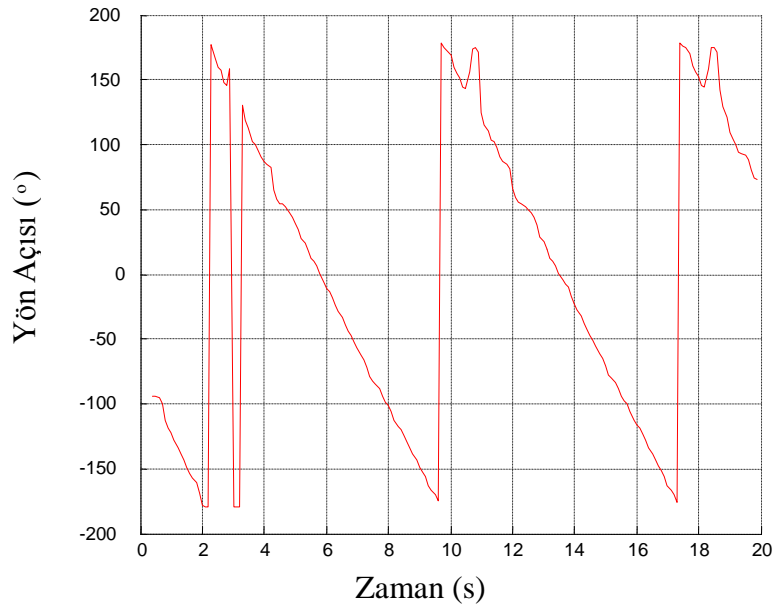
Şekil 7.9: TEST1 kamera takibinde platform X/Y eksen konumu.

Şekil 7.10 ve Şekil 7.11'de platformun yön değişimi sırasıyla odometre ve kamera verilerine göre verilmiştir. Her iki şekilde de platformun yönü $atan2(x,y)$ fonksiyonuyla

hesaplanmıştır. $\text{atan2}(x,y)$ fonksiyonunun çalışma aralığı $[-180^\circ,180^\circ]$ aralığındadır. Şekil 7.10 incelendiğinde -180° 'den 180° 'ye doğru doğrusal bir artış görülmekte, bu doğru platformun tam bir tur gerçekleştirdiğini gösterir.

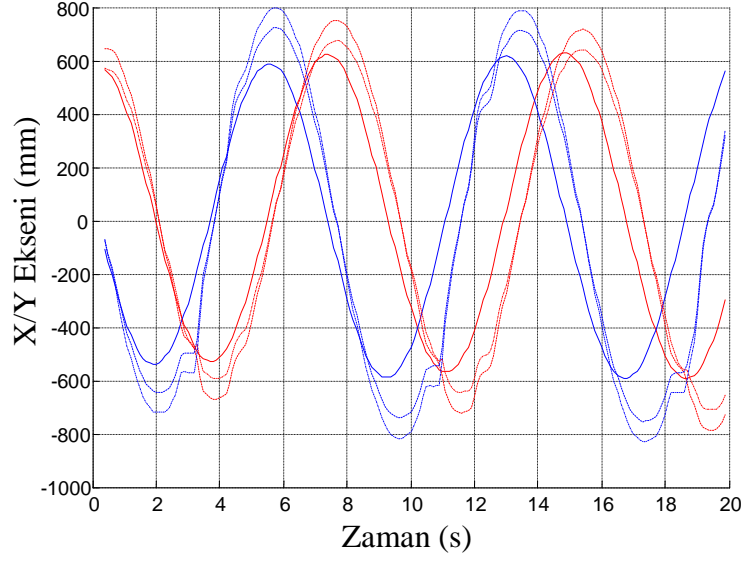


Şekil 7.10: TEST 1 odometre verisine göre platform yönü.

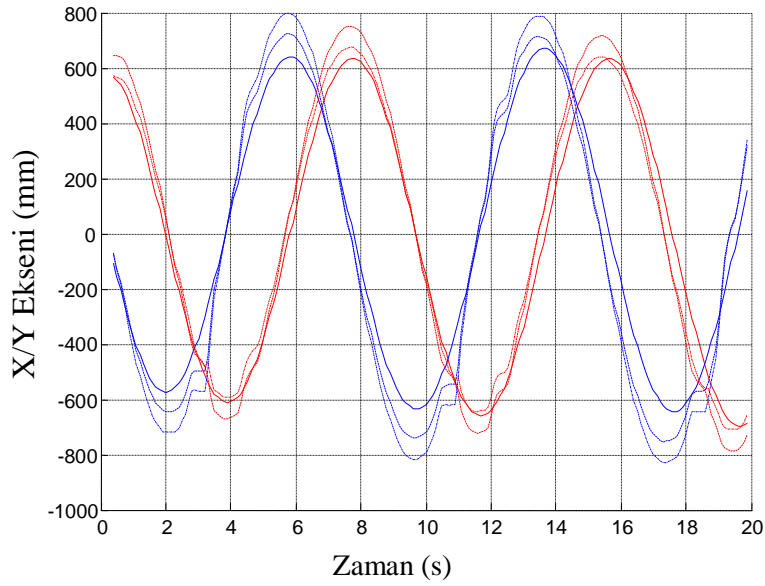


Şekil 7.11: TEST 1 kamera verisine göre platform yönü.

Şekil 7.11’da bu durum tam tersidir bunun nedeni platform başlangıç konumlarının farklı olmasından kaynaklanmaktadır. Şekil 7.11’nun üçüncü saniyesinde bir atlama görülmekte bu atlama üçüncü saniyedeki $\text{atan2}(x,y)$ hesaplamasının 180° ’yi aştığını bu yüzden negatif değerli sonuç vermesinden kaynaklanır. Kamera yörüngesi ve odometre yörüngeleri platform başlangıç yönü farklı ve yörünge dönüş yönleri birbirinin tersi olduğu için yön değişim grafikleri birbirlerine benzemektedir.



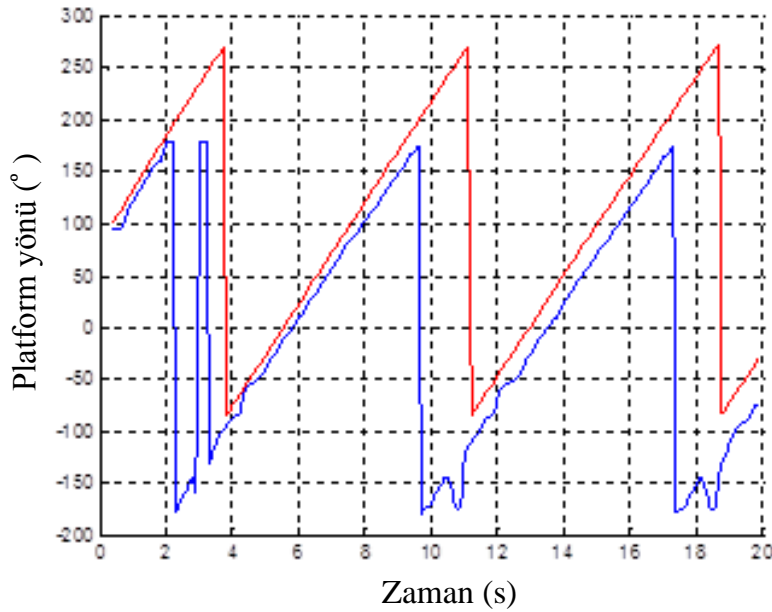
Şekil 7.12: TEST 1 kalibrasyondan önce verilerin X/Y konum kıyaslaması.



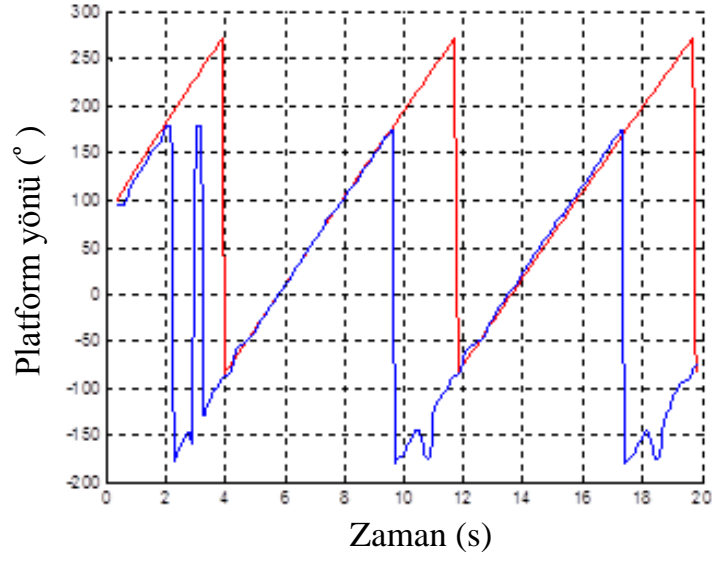
Şekil 7.13: TEST 1 kalibrasyondan sonra verilerin X/Y konum kıyaslaması.

Şekil 7.12 ve Şekil 7.13’de odometre, kamera ve düzeltilmiş kamera verileriyle oluşturulan kıyaslama grafikleri görülmektedir. Burada kırmızı düz çizgi odometre X eksenini, kırmızı kesik çizgiler kamera X eksenini ve kırmızı noktalı çizgiler düzeltilmiş kamera X eksenini göstermektedir; mavi düz çizgiler odometre Y eksenini mavi kesik çizgiler kamera y eksenini ve mavi noktalı çizgiler düzeltilmiş kamera Y eksenini gösterir. Şekil 7.12 de ki grafik parametrelerin doğrudan kullanılmasıyla elde edilmiştir. Şekil 7.13 deki grafik parametrelerin kalibrasyonu yapıldıktan sonra oluşan grafikdir. Bu iki şekil birbiriyle kıyaslandığında kalibrasyon olmadan önce odometre ve kamera verilerindeki fark görülmekte (Şekil 7.12) , kalibrasyon sonucunda oluşan grafikteki odometre ve kamera konum verilerinin birbirine yaklaştığı görülebilir.

Şekil 7.14 ve Şekil 7.15’te platformun kamera ve odometre verilerinden ölçülen yön grafikleri verilmiştir. Şekil 7.14 teki grafik ham verilerle oluşturulan grafikdir. Bu grafikte kırmızı çizgi odometre verisinden elde edilen yönü, mavi çizgi ise kamera verisinden elde edilen yönü göstermektedir. Odometre yönüyle kamerayla elde edilen yön uyuşmamakta ve kamera verisine göre robot erken dönüş yaptığı görülmektedir.

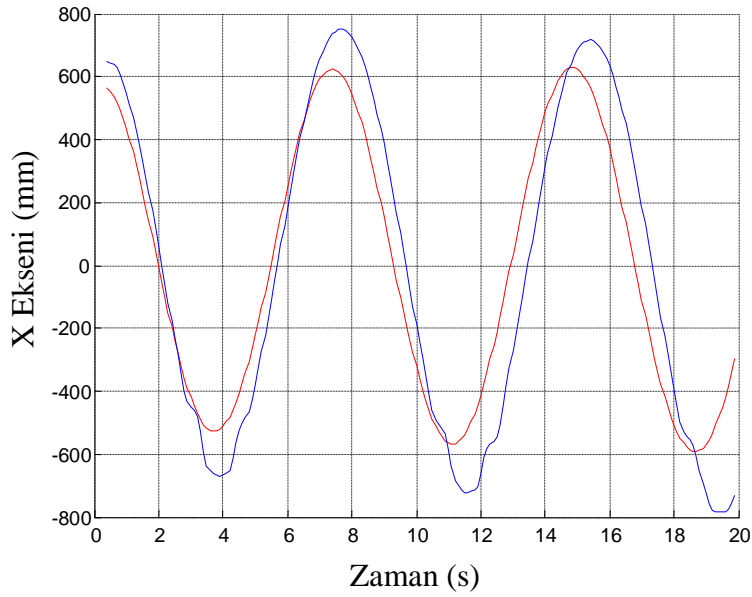


Şekil 7.14 : TEST 1 kalibrasyondan önce odometre ve kamera platform yön açısı.

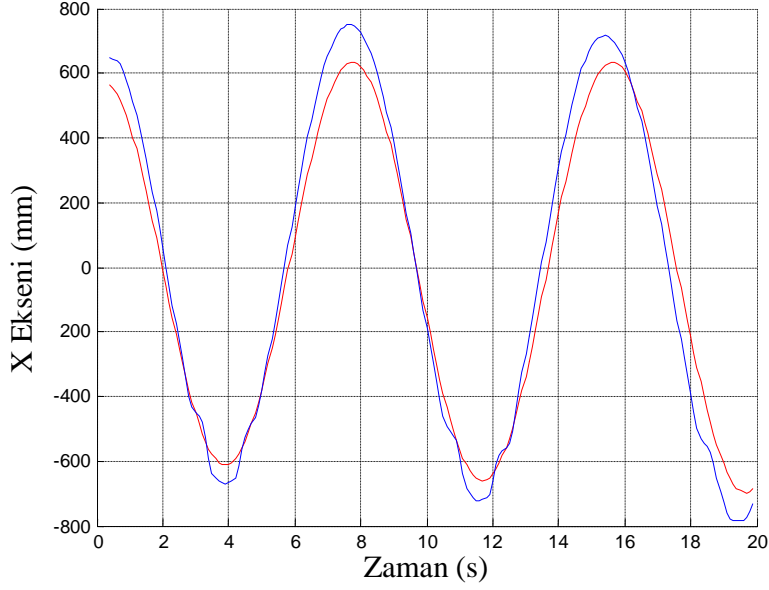


Şekil 7.15: TEST 1 kalibrasyondan sonra odometre ve kamera platform yön açısı.

Kamera verisine göre robotun erken dönüş yapmasının nedeni kamera ile yapılan nesne takibinde kameranın takip ettiği desen tekerlek orta merkezine göre 150 mm yukarıda olmasından dolayı kaynaklandığı düşünülmektedir. Şekil 7.15'te kalibrasyon sonucu oluşan grafikler birbiriyle uyumuş fakat yine kamera verisine göre platformun erken döndüğü görülmektedir. Bu grafiklerde kalibrasyon işleminin hataları çok aza indirdiği açıkça görülmüştür.



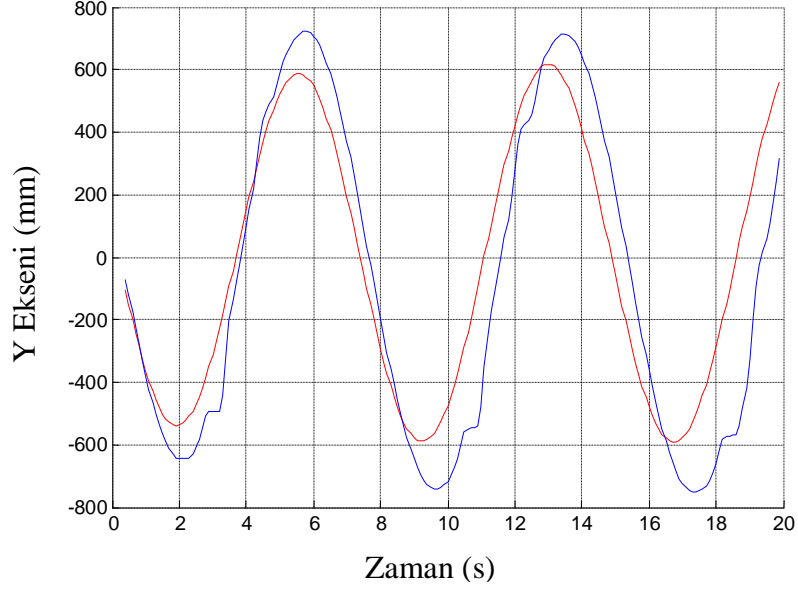
Şekil 7.16: TEST 1 kalibrasyondan önce odometre ve kamera platform X konumu.



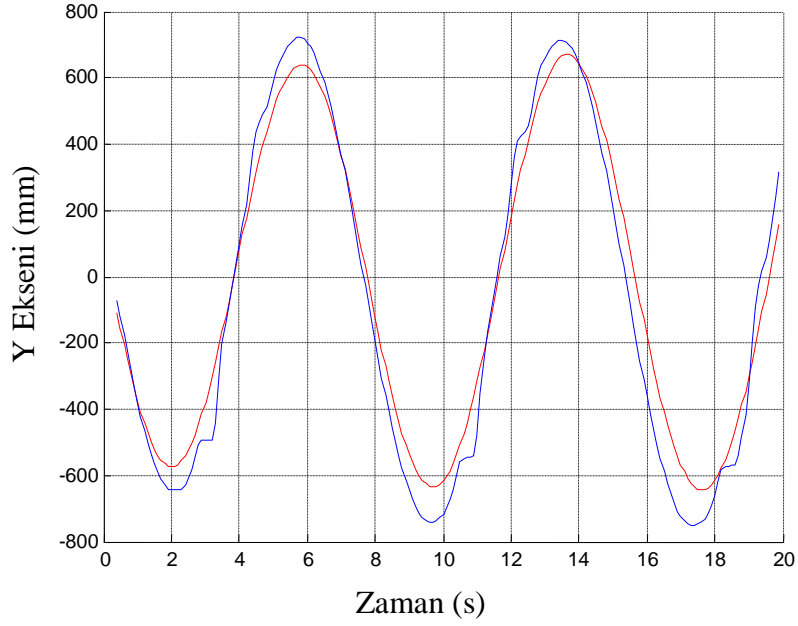
Şekil 7.17: TEST 1 kalibrasyondan sonra odometre ve kameraya platform X konumu.

Şekil 7.16 ve Şekil 7.17'de platformun odometre ve kamera verilerine göre X eksenindeki konum değişikliği görülmekte, Şekil 7.16 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasına aittir. Şekil 7.16 incelendiğinde odometre verisine göre robotun dönüş yaptığı çap kamera verisinden elde edilen dönüş çapına göre daha küçüktür. Bunun nedeni enkoder okumalarında kayıpların gerçekleştiği olabilir. Kamera verisine bakıldığında alt ve üst noktalarda düzensizlikler mevcut bunun nedeni kameranın nesne takip desenini o noktalarda net görememesinden kaynaklanabilir. Şekil 7.17 incelendiğinde kalibrasyon sonrası oluşan odometre verileri kamera verilerine yaklaşmıştır.

Şekil 7.18 ve Şekil 7.19'da platformun odometre ve kamera verilerine göre y ekseninde konum değişiklikleri görülmekte, Şekil 7.18 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasına aittir. Şekil 7.18'de odometre verileri ile kamera verileri uyuşmadığı görülüyor, özellikle eğrilerin en alt ve en üst konumlarda büyük farkların olduğu görülmektedir. kalibrasyon sonrasında Şekil 7.19'da odometre verilerinin kamera verilerine yaklaştığı görülmektedir. Enkoder okumalarındaki kayıplar odometrenin elde edilen konumlarla kameradan elde edilen konumlar arasındaki farka neden olduğu düşünülmektedir.

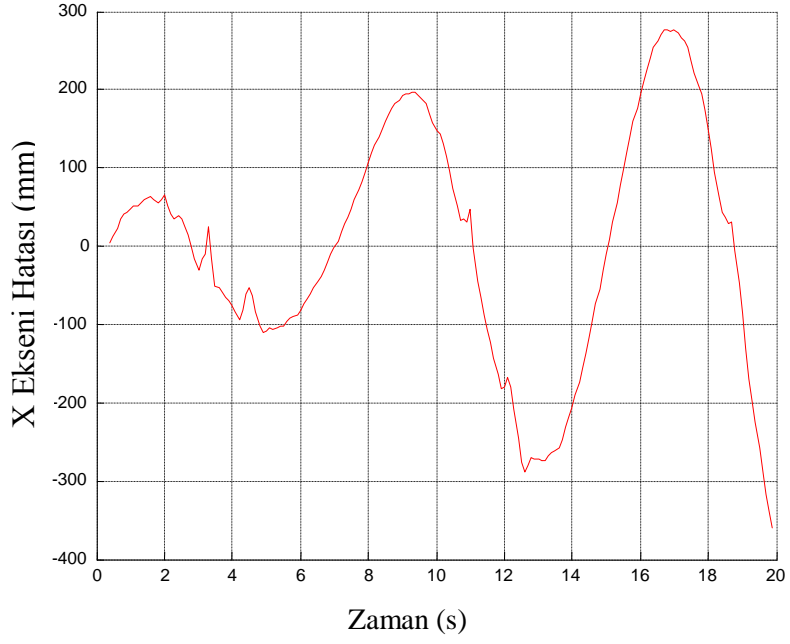


Şekil 7.18: TEST 1 kalibrasyondan önce odometre ve kamera platform Y konumu.

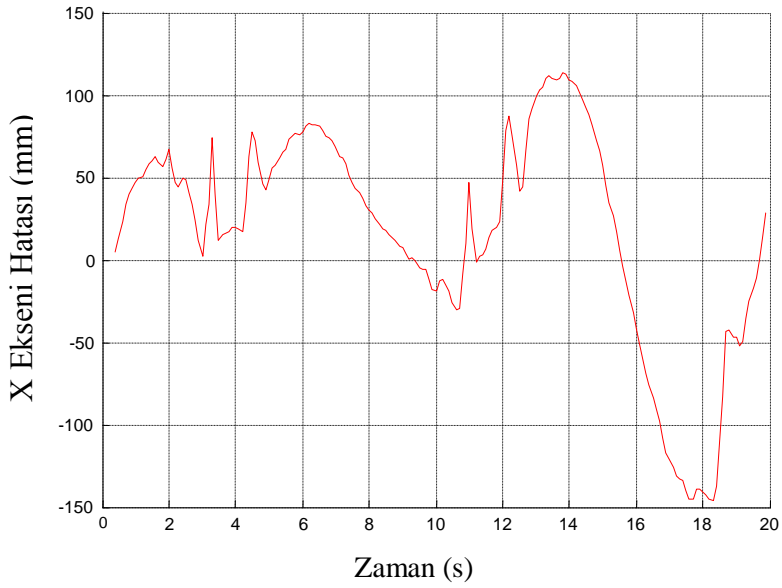


Şekil 7.19: TEST 1 kalibrasyondan sonra odometre ve kamera platform Y konumu.

Şekil 7.20 ve Şekil 7.21’de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan X konumu hataları gösterilmiştir. Şekil 7.20 incelendiğinde kalibrasyon öncesi hataların 300 mm’ye kadar çıktığı görülmektedir. Şekil 7.21 kalibrasyon sonucu X konumu odometre hataları 150 mm’yi aşmamaktadır. Burada kalibrasyon sonucu X yönündeki hataların yarı yarıya düştüğünü gösterir.

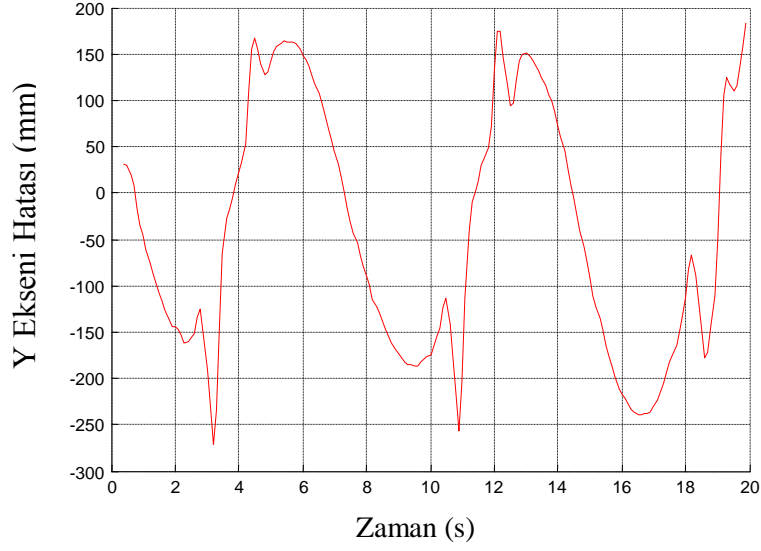


Şekil 7.20: TEST 1 kalibrasyondan önce odometre X eksenini hatası.

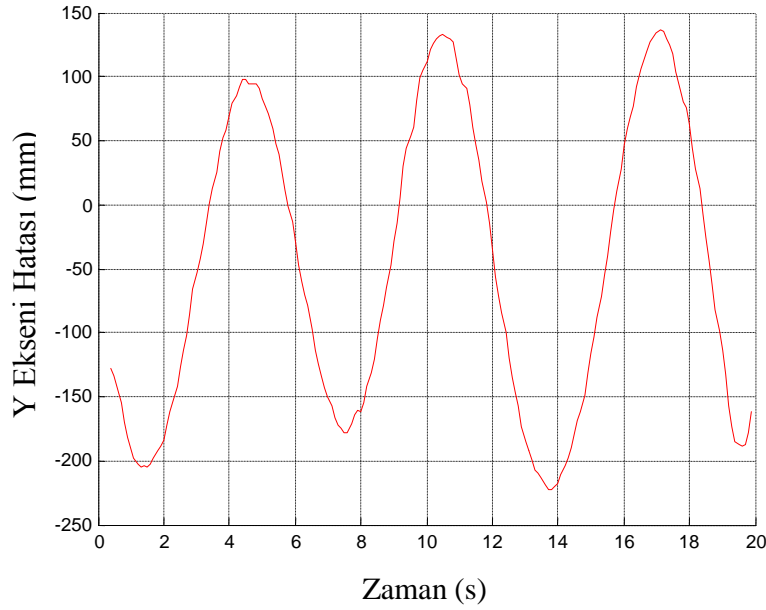


Şekil 7.21: TEST 1 kalibrasyondan sonra odometre X eksenini hatası.

Şekil 7.22 ve Şekil 7.23' de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan Y konumu hataları gösterilmiştir. Şekil 7.22 incelendiğinde kalibrasyon öncesi hataların 250 mm'ye kadar çıktığı görülmektedir. Kalibrasyon sonrası oluşan Y konum hataları Şekil 7.23' de 200 mm'ye kadar düştüğü görülüyor.

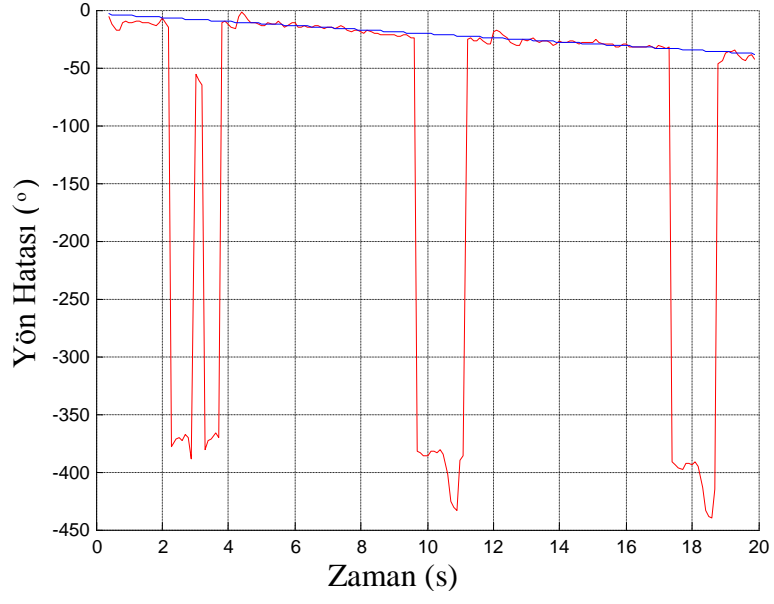


Şekil 7.22: TEST 1 kalibrasyondan önce odometre Y eksenı hatası.

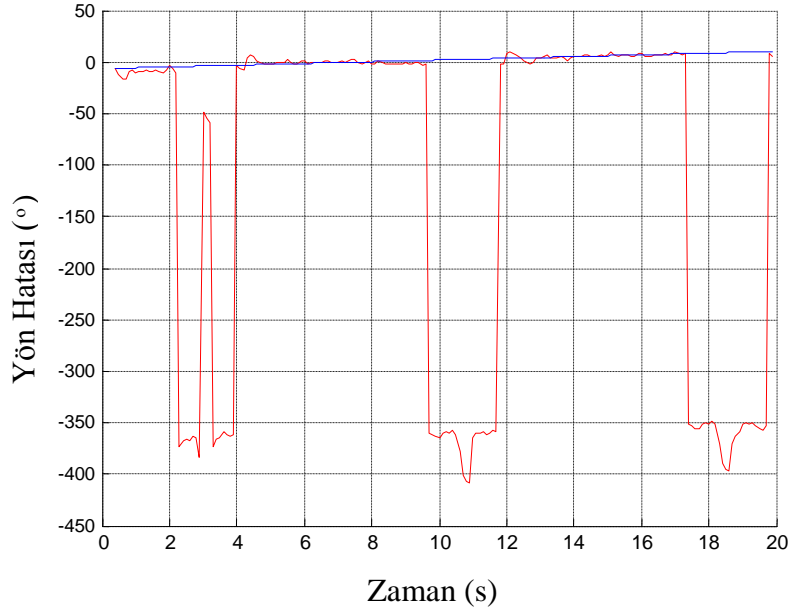


Şekil 7.23: TEST 1 kalibrasyondan sonra odometre Y eksenı hatası.

Şekil 7.24 ve Şekil 7.25'te sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan yön hataları gösterilmiştir. Kırmızı çizgiyle gösterilen düzeltilmemiş yön hataları, maviyle gösterilen eğri üzerine uydurulması yöntemiyle oluşturulan eğridir. Şekil 7.24 incelendiğinde yön hatasının 50° iken kalibrasyon sonucu bu hata 10° 'ye kadar düşmektedir.



Şekil 7.24: TEST 1 kalibrasyondan önce odometre yön hatası.



Şekil 7.25: TEST 1 kalibrasyondan sonra odometre yön hatası.

7.3.2 Veri Analizi Test 2

Bu testte birinci testteki gibi robotun standart tekerlekleri kullanılarak yapılmıştır. Sağ ve sol teker çapları eşit 90 mm ve tekerlek açıklığı 190 mm dir. Nominal ölçüler aşağıdaki verilmiştir.

$$D_r = 90 \text{ (mm)}$$

$$D_l = 90 \text{ (mm)}$$

$$b = 190 \text{ (mm)}$$

Başlangıç belirsizlik matrisi $\mu_{rC} = 1 \times 10^{-6}$, $\mu_{rIC} = 1 \times 10^{-6}$, $\mu_{lC} = 1 \times 10^{-6}$ olarak atanır.

$$\begin{bmatrix} \mu_{rC} & 0 & 0 \\ 0 & \mu_{rIC} & 0 \\ 0 & 0 & \mu_{lC} \end{bmatrix} = \begin{bmatrix} 1 \times 10^{-6} & 0 & 0 \\ 0 & 1 \times 10^{-6} & 0 \\ 0 & 0 & 1 \times 10^{-6} \end{bmatrix}$$

$$\text{Ölçüm Hataları : } k_s = 0.1$$

Odometreye bağlı olan konum tahmin hatası zamanla artıyor. Ölçüm belirsizliği doğrusal olarak artar.

$$S_x = (1 + i \cdot k_s) \cdot 10$$

$$S_y = (1 + i \cdot k_s) \cdot 10$$

$$S_t = (1 + i \cdot k_s) \cdot 10^{-1}$$

Test verisinin ilk yüz verisi kullanılarak kalibrasyon değerleri çıkarılmıştır.

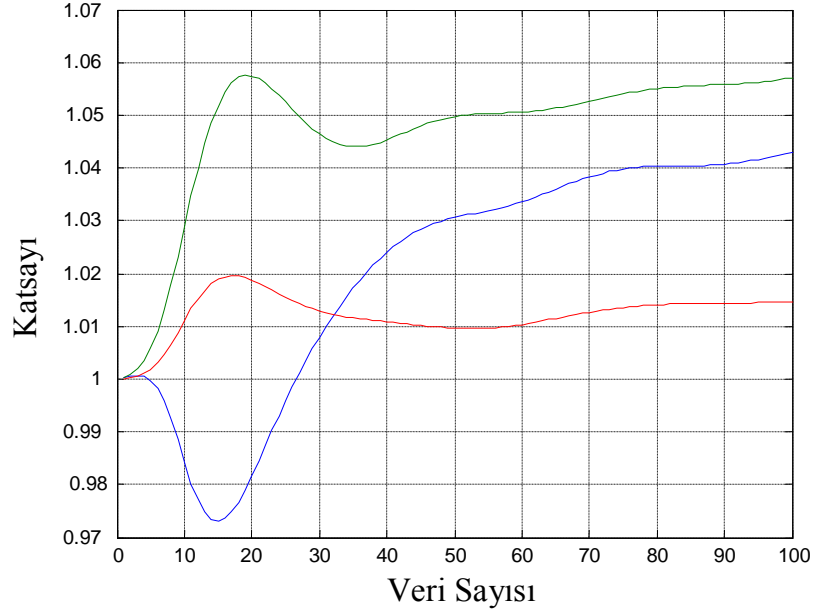
$$\mu_r = 0.0430$$

$$\mu_l = 0.0571$$

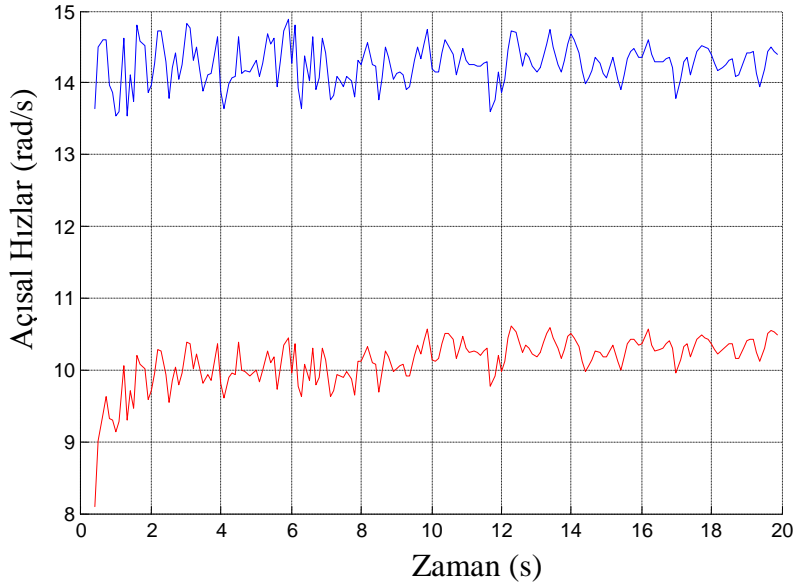
$$\mu_b = 0.0146$$

Bu değerlere göre yeniden oluşturulan sağ tekerlek, sol tekerlek çapları ve teker açıklığı aşağıdaki gibi düzenlenmiştir.

$$D_r = 93.8733 , D_l = 95.1388, b = 192.7693$$



Şekil 7.26: TEST 2 parametre hata yaklaşımı



Şekil 7.27: TEST 2 sağ ve sol tekerlek açısal hızları

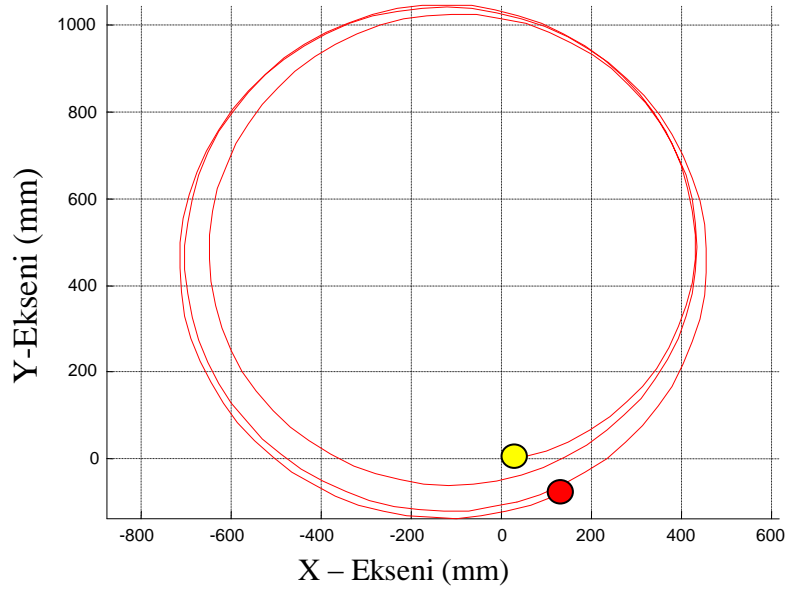
Kalibrasyon değerleri program tarafından ilk yüz noktaya göre hesaplanır, Şekil 7.26'da kalibrasyon değerlerinin hesaplanan nokta sayısına göre değişimi verilmiştir. Yeşil çizgi ile μ_l , mavi çizgiyle gösterilen μ_r ve kırmızı çizgiyle gösterilen μ_b dir. Hesaplanan ilk 15 veriye göre büyük sapmalar göstermişlerse de 50 verinin hesaplanmasıyla gerçek değerlerine yaklaşmışlardır.

Şekil 7.27’te ikinci testte elde edilen sağ ve sol tekerleğin açısal hızları görülmektedir. Mavi çizgiyle gösterilen sağ tekerlek hızı yaklaşık 14.4 rad/s dir. Kırmızıyla gösterilen sol tekerlek açısal hızı 10.5 rad/s olarak grafikte görülmektedir. Enkoder verileri eşit zaman aralıklarında alınmadığı için grafik titreşimli görülmektedir. Açısal hızlardaki bu fark yüzünden hızlı dönen tekerlek dışarıda kalacak şekilde platform daire çizecektir. $R = a(V_r + V_l)/(V_r - V_l)$ formülüyle platformun döneceği yarıçapı bulabiliriz.

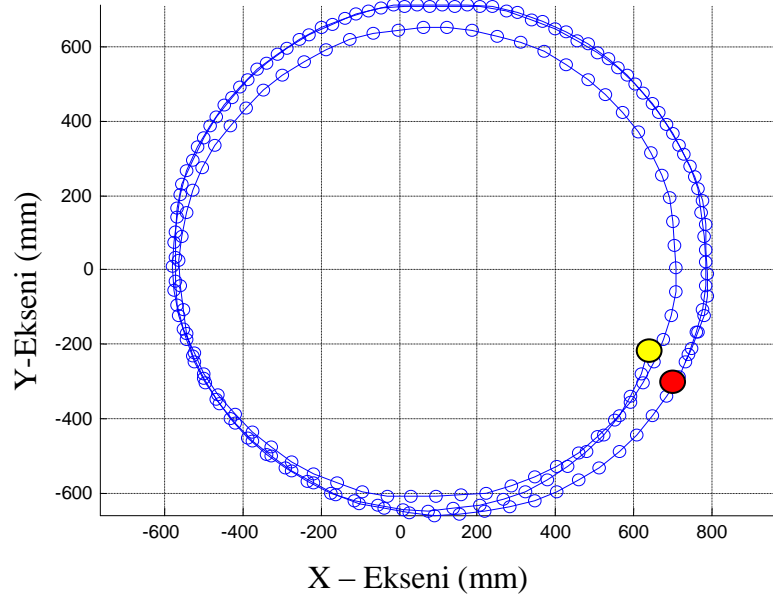
$$R = \frac{85 \left(14.4 \frac{90}{2} + 10.5 \frac{90}{2} \right)}{\left(14.4 \frac{90}{2} - 10.5 \frac{90}{2} \right)} = 542$$

Bu verilere göre platform dönüş çapı 1084 mm olacaktır. Bu değer odometre verisinin oluşturduğu çap değerine yakın olması beklenir.

Şekil 7.28 ve Şekil 7.29’da sırasıyla odometre ve kamera verilerinden elde edilen platform yörüngesi görülmektedir. Şekillerde içi dolu sarı daireyle gösterilen konum başlangıç konumu, içi dolu kırmızı daireyle gösterilen konum bitiş konumudur. Bu testte de robot ilk konumuna göre dışarı doğru açılmış, ilk çizdiği yörünge çapı daha sonraki turlarda artış göstermiştir.



Şekil 7.28 :TEST 2 odometre verisine göre platform yörüngesi.

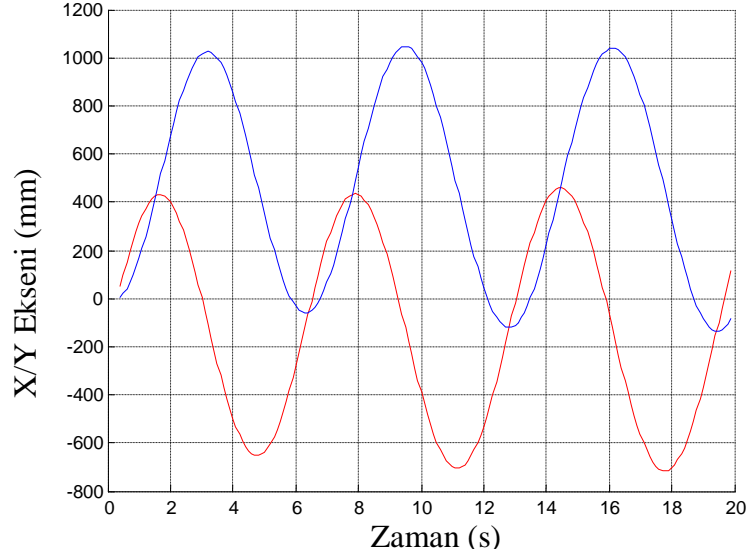


Şekil 7.29. :TEST 2 kamera verisine göre platform yörüngesi.

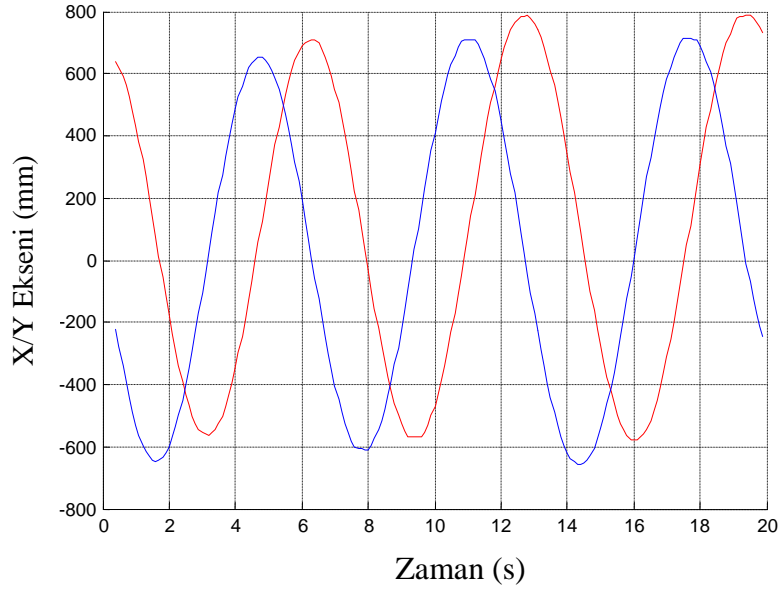
Şekil 7.30 ve Şekil 7.31 ‘de sırasıyla odometre ve kamera verilerine göre platformun X ve Y eksenindeki konumları görülmektedir. Kırmızı çizgiyle gösterilen X eksenini mavi çizgiyle gösterilen Y eksenindeki konumlarıdır. Bu testte birinci testteki gibi başlangıç farklıdır bu yüzden odometre ve kamera konum grafiklerinin başlangıç noktaları farklıdır.

Bu grafiklerdeki X ve Y konum değişimlerine bakarak platformun yörünge çapını ve bu yörüngede kaç defa döndüğünü görebiliriz. Odometre verisindeki eğrilere bakıldığında birinci periyottaki eğri yüksekliği ile üçüncü periyottaki eğri yüksekliği arasında fark olduğu görülür (Şekil 7.30). Kamera verisindeki eğrilerde de bu durum gözlenir. Bu eğrilerin yükseklik farklarının olması platformun her bir turda farklı çapta bir yörünge izlediği anlamına gelir.

Odometre verisinde X ve Y yönünde dönüş çapı yaklaşık olarak birinci turda 1100 mm, ikinci turda 1200 mm ve üçüncü turda 1300 mm’dir(Şekil 7.30). Kamera verisinde X ve Y yönünde dönüş çapları yaklaşık olarak birinci turda 1250 mm, ikinci turda 1300 mm ve üçüncü turda 1350 mm’dir (Şekil7.31). Her iki şekilden de robotun üç tam tur gerçekleştirdiği görülür.

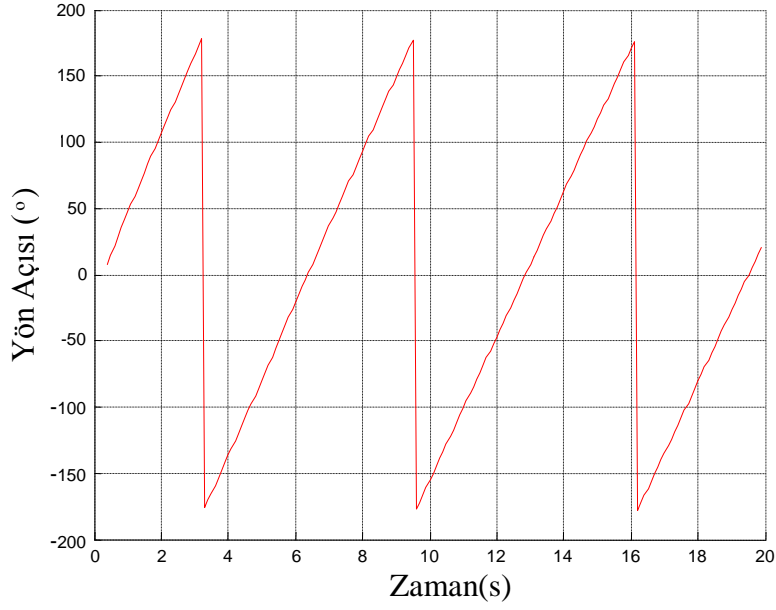


Şekil 7.30 : TEST 2 odometre takibinde platform X/Y eksen konumu.

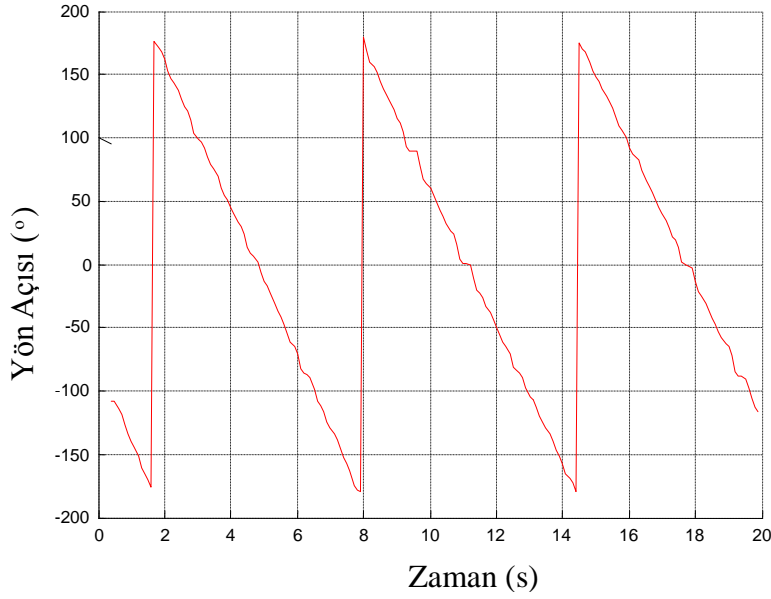


Şekil 7.31 : TEST 2 kamera takibinde platform X/Y eksen konumu.

Şekil 7.32 ve Şekil 7.33'de platformun yön değişimi sırasıyla odometre ve kamera verilerine göre verilmiştir. Her iki şekilde de platformun yönü $atan2(x,y)$ fonksiyonuyla hesaplanmıştır. $atan2(x,y)$ fonksiyonunun çalışma aralığı $[-180,180]$ aralığındadır. Birinci teste göre kamera verisinde $atan2(x,y)$ fonksiyonu hatası yoktur. Kameradan platform takibi birinci teste göre daha iyi yapıldığını söyleyebiliriz.



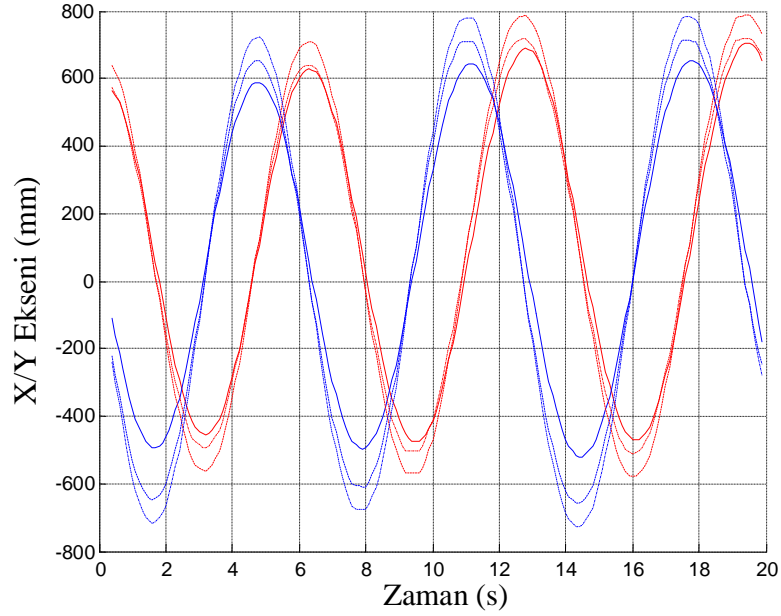
Şekil 7.32 :TEST 2 odometre verisine göre platform yönü.



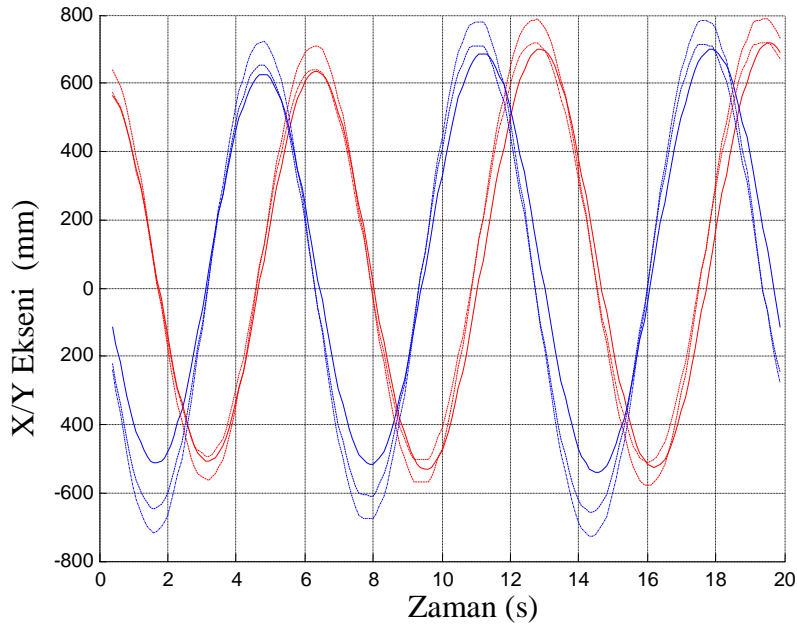
Şekil 7.33 :TEST 2 kamera verisine göre platform yönü.

Şekil 7.34 ve Şekil 7.35’de odometre, kamera ve düzeltilmiş kamera verileriyle oluşturulan kıyaslama grafikleri görülmektedir. Burada kırmızı düz çizgi odometre X eksenini, kırmızı kesik çizgiler kamera X eksenini ve kırmızı noktalı çizgiler düzeltilmiş kamera X eksenini göstermektedir; mavi düz çizgiler odometre Y eksenini mavi kesik çizgiler kamera Y eksenini ve mavi noktalı çizgiler düzeltilmiş kamera Y eksenini gösterir. Şekil 7.34 de ki grafik parametrelerin doğrudan kullanılmasıyla elde edilmiştir. Şekil 7.35 deki grafik parametrelerin kalibrasyonu yapıldıktan sonra oluşan grafikdir. Bu iki şekle bakıldığında

arasında bir büyük bir fark olmadığı görülebilir. Burada uygulanan kalibrasyonun odometre verilerini az da olsa düzelttiği söylenebilir.



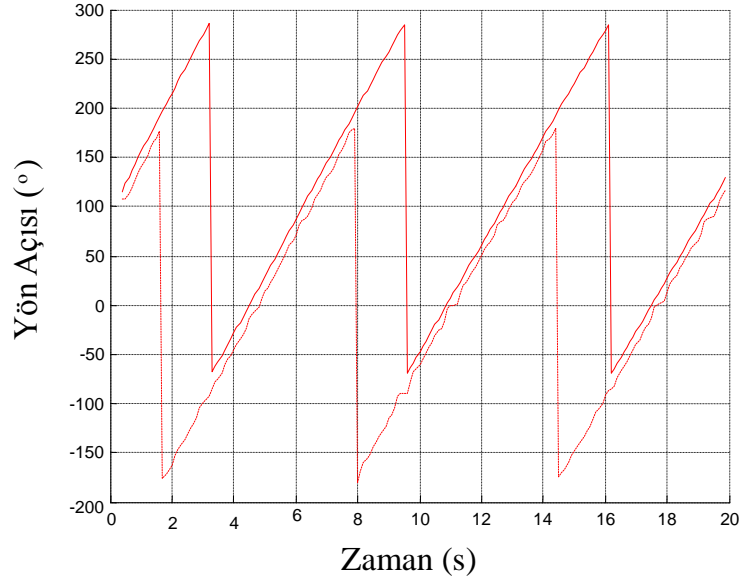
Şekil 7.34: TEST 2 kalibrasyondan önce verilerin X/Y konum kıyaslaması.



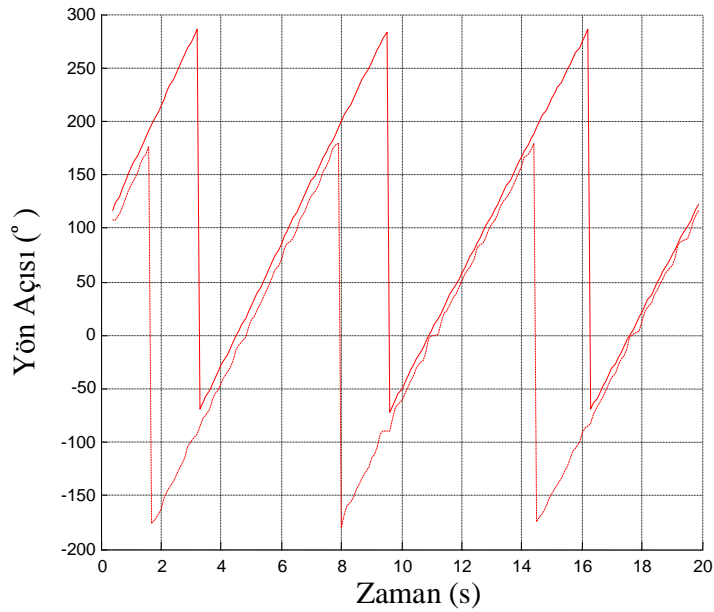
Şekil 7.35: TEST 2 kalibrasyondan sonra verilerin X/Y konum kıyaslaması.

Şekil 7.36 ve Şekil 7.37'te platformun kamera ve odometre verilerinden ölçülen konum grafikleri verilmiştir. Grafiklerde kırmızı çizgi odometre verisinden elde edilen konumu,

kırmızı kesik çizgili eğri ise kamera verisinden elde edilen konumu göstermektedir. Birinci testteki gibi kamera verisine göre robot erken dönüş yaptığı görünmekte. Bunun kameranın takip ettiği nesnenin platform teker orta noktasına göre yüksekliğinden kaynaklandığını birinci test verilerini yorumlarken bahsedilmişti. Her iki şekilde incelendiğinde her iki grafiğin birbirine benzediği görülür. Bu test sonucunda uygulanan kalibrasyon odometre verilerini çok fazla düzelttiği söylenemez.

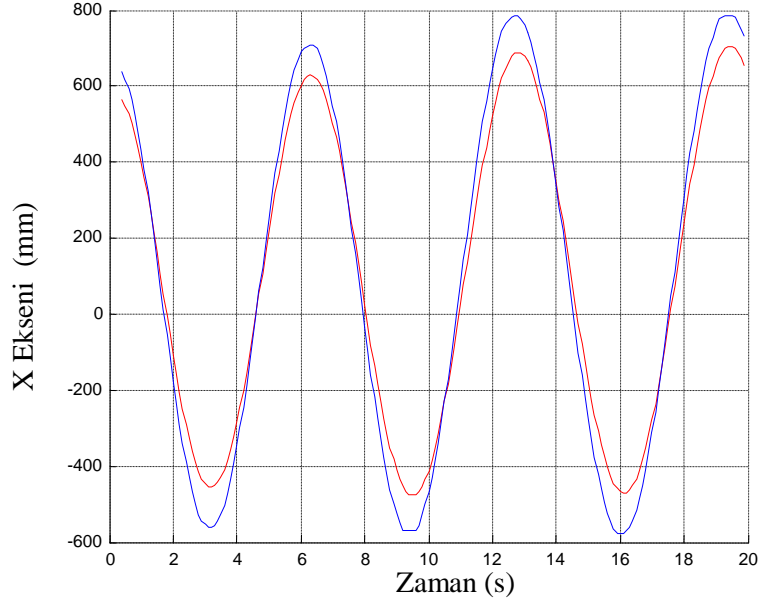


Şekil 7.36: TEST 2 kalibrasyondan önce odometre ve kamera platform yön açısı.

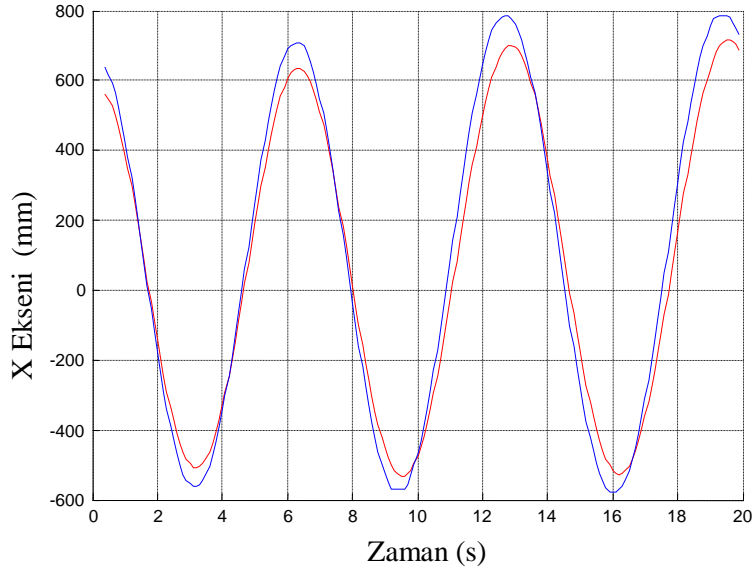


Şekil 7.37: TEST 2 kalibrasyondan sonra odometre ve kamera platform yön açısı.

Şekil 7.38 ve Şekil 7.39’da platformun odometre ve kamera verilerine göre X eksenindeki konum değişikliği görülmekte, Şekil 7.38 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasına aittir. Şekil 7.39 u incelediğimizde platform odometre verisine göre daha düşük çapta hareket ettiği görülür. Yapılan kalibrasyon sonucu Şekil 7.39’da fazla bir fark olmamasına rağmen eğrilerin uç kısımlarında az da olsa odometri eğrisi kamera eğrisine yaklaşmıştır.

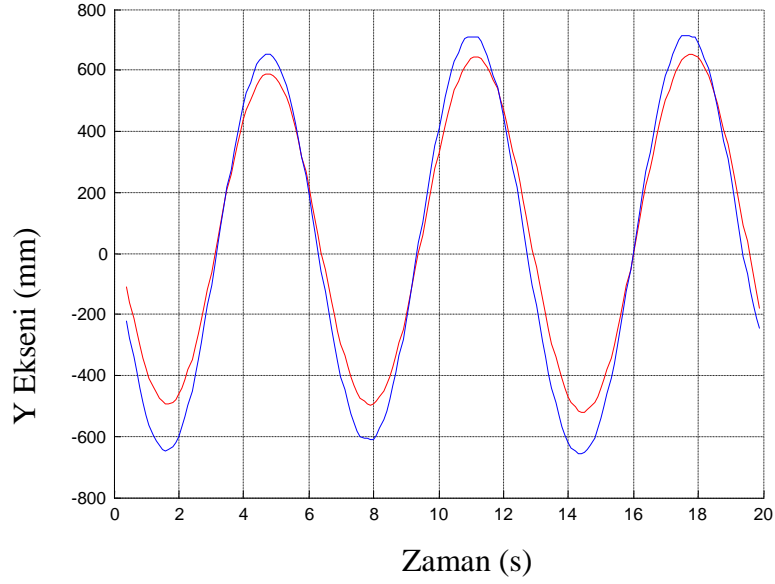


Şekil 7.38: TEST 2 kalibrasyondan önce odometre ve kamera platform X konumu.

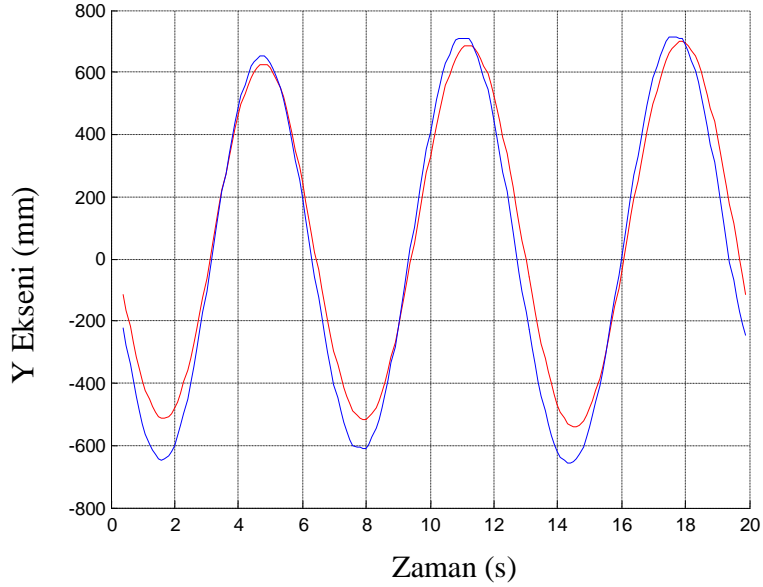


Şekil 7.39: TEST 2 kalibrasyondan sonra odometre ve kameraya platform X konumu.

Şekil 7.40 ve Şekil 7.41’de platformun odometre ve kamera verilerine göre Y ekseninde konum değişiklikleri görülmekte, Şekil 7.40 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasına aittir. Şekil 7.40 da odometre eğrisi ile kamera verisi arasında özellikle uç kısımlarda büyük farklar vardır. Kalibrasyon sonrası Şekil 7.41’de bu açıklığın azda olsa azaldığı görülür.

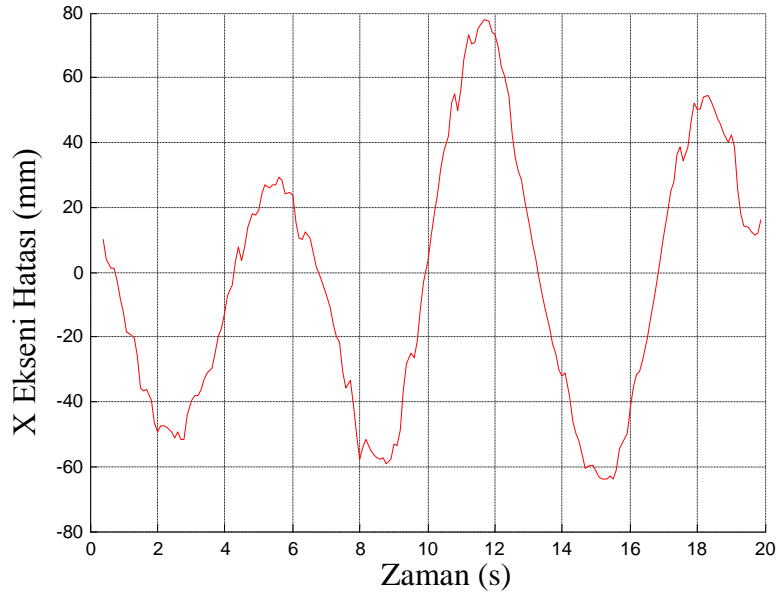


Şekil 7.40: TEST 2 kalibrasyondan önce odometre ve kamera platform Y konumu.

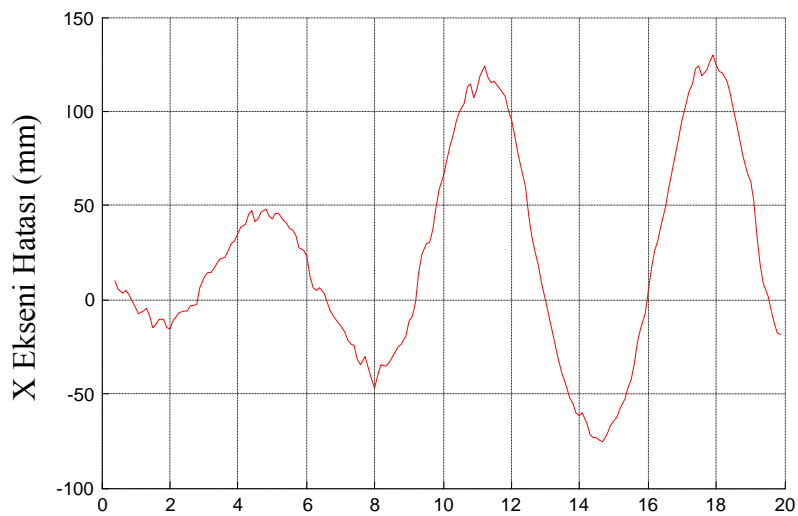


Şekil 7.41: TEST 2 kalibrasyondan sonra odometre ve kamera platform Y konumu.

Şekil 7.42 ve Şekil 7.43’da sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan X konumu hataları gösterilmiştir. Şekil 7.42 incelendiğinde kalibrasyon öncesi hataların 80 mm’ye kadar çıktığı görülmektedir. Şekil 7.43 kalibrasyon sonucu X konumu odometre hataları 120 mm civarındadır. Burada kalibrasyon sonucu X yönündeki hataların artış gösterdiği görülür. Burada kalibrasyon X konumu hatalarını düzeltmemiştir, biz kalibrasyon katsayılarını bulurken ilk yüz veriyi kullanıyoruz, eğer yüzden daha fazla veri kullansak kalibrasyon katsayıları farklı değerler alacaktır ve X konum hatasını düzeltir.

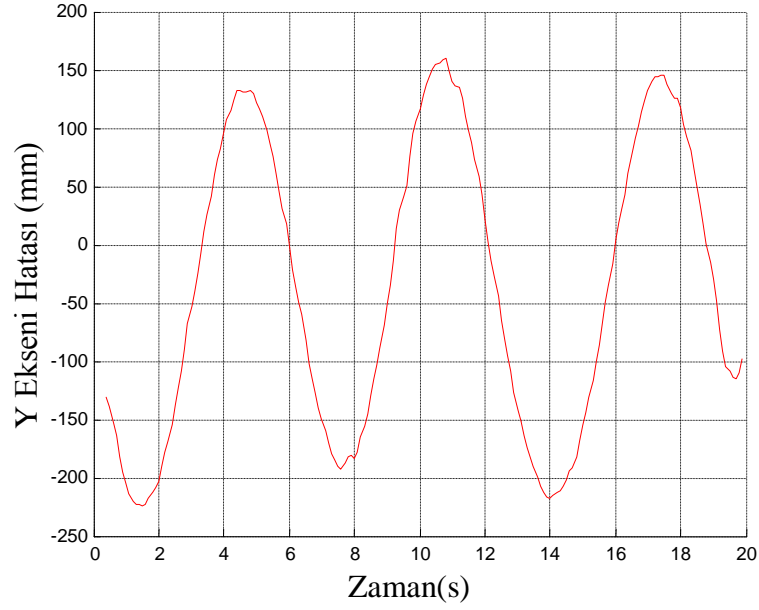


Şekil 7.42 :TEST 2 kalibrasyondan önce odometre X eksen hatası.

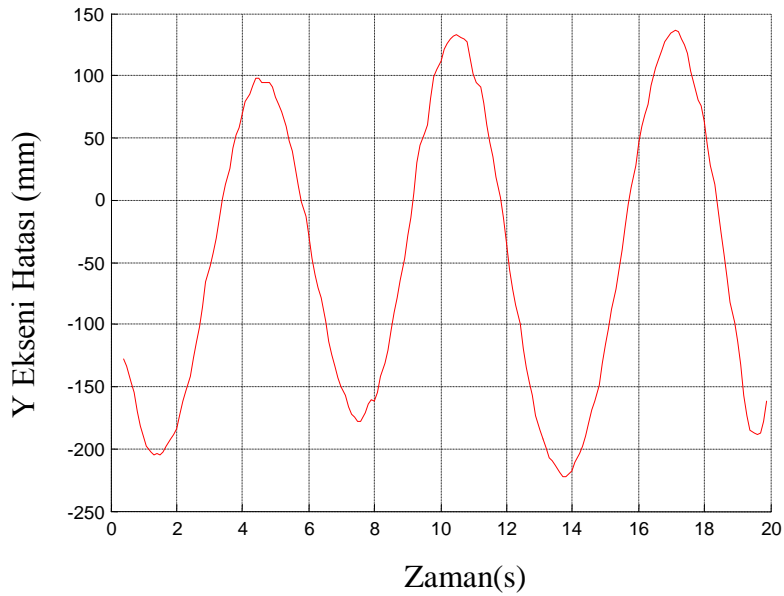


Şekil 7.43 :TEST 2 kalibrasyondan sonra odometre X eksen hatası.

Şekil 7.44 ve Şekil 7.45’ te sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan Y konumu hataları gösterilmiştir. Şekil 7.44 incelendiğinde kalibrasyon öncesi hataların 225 mm’ye kadar çıktığı görülmektedir. Kalibrasyon sonrası oluşan Y konum hataları Şekil 7.45’ de 210 mm’ye kadar düştüğü görülüyor. Burada da yapılan kalibrasyonun hataları çok fazla etkilemediği görülmekte.

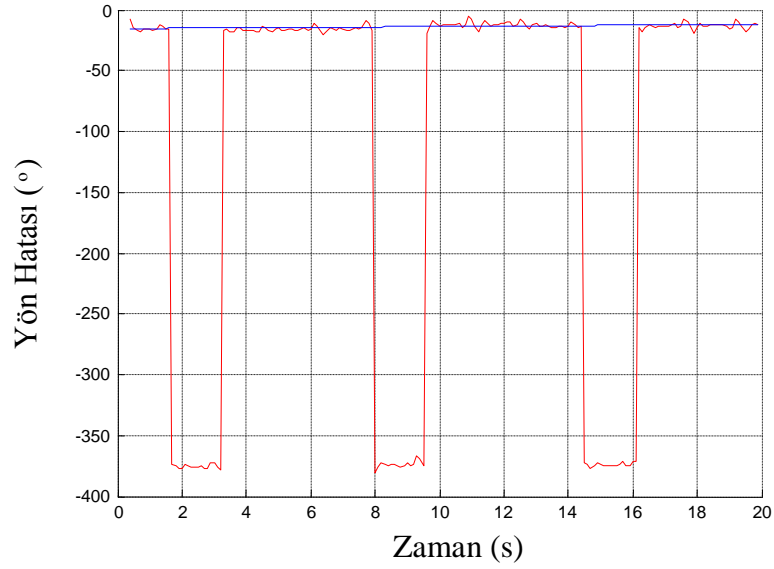


Şekil 7.44:TEST 2 kalibrasyondan önce odometre Y eksenı hatası.

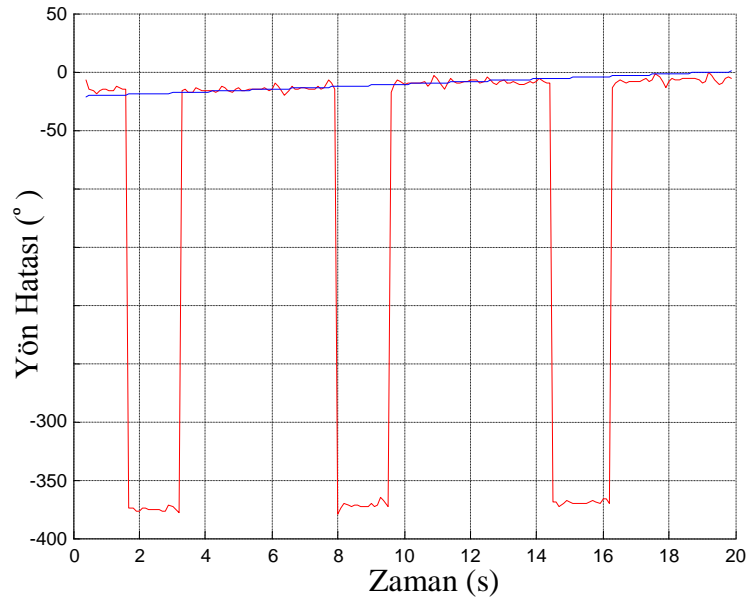


Şekil 7.45:TEST 2 kalibrasyondan sonra odometre Y eksenı hatası.

Şekil 7.46 ve Şekil 7.47’de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan yön hataları gösterilmiştir. Kırmızı çizgiyle gösterilen düzeltilmemiş yön hataları, maviyle gösterilen eğri üzerine uydurulması yöntemiyle oluşturulan eğridir. Şekil 7.46 ‘ı incelendiğinde yön hatası önce 20° başlamış ve bitiş noktasında 10° ye kadar düşmüştür. Kalibrasyon sonrasında Şekil 7.47’de yine 20° başlamış ve bitiş noktasında hiç hata gözükmemektedir. Bu sonuçlara bakarak kalibrasyonun odometre yön hatalarını düzelttiğini söyleyebiliriz.



Şekil 7.46: TEST 2 kalibrasyondan önce odometre yön hatası.



Şekil 7.47: TEST 2 kalibrasyondan sonra odometre yön hatası.

7.3.3 Veri Analizi Test 3

Üçüncü testte nominal parametreler program kısmında değiştirilerek test yapılmıştır. Bu şekilde daha yüksek hataları kalibrasyon metodunun nasıl giderceği görülecektir.

Nominal ölçüler (mm)

$$D_r = 95$$

$$D_l = 93$$

$$b = 190$$

Başlangıç belirsizlik matrisi $\mu_{rC} = 1 \times 10^{-6}$, $\mu_{rIC} = 1 \times 10^{-6}$, $\mu_{lC} = 1 \times 10^{-6}$ olarak atanır.

$$\begin{bmatrix} \mu_{rC} & 0 & 0 \\ 0 & \mu_{lC} & 0 \\ 0 & 0 & \mu_{bC} \end{bmatrix} = \begin{bmatrix} 1 \times 10^{-6} & 0 & 0 \\ 0 & 1 \times 10^{-6} & 0 \\ 0 & 0 & 1 \times 10^{-6} \end{bmatrix}$$

Ölçüm Hataları : $k_s = 0.1$

Odometreye bağlı olan konum tahmin hatası zamanla artıyor. Ölçüm belirsizliği doğrusal olarak artar.

$$S_x = (1 + i \cdot k_s) \cdot 10$$

$$S_y = (1 + i \cdot k_s) \cdot 10$$

$$S_t = (1 + i \cdot k_s) \cdot 10^{-1}$$

Test verisinin ilk yüz verisi kullanılarak kalibrasyon değerleri çıkarılmıştır.

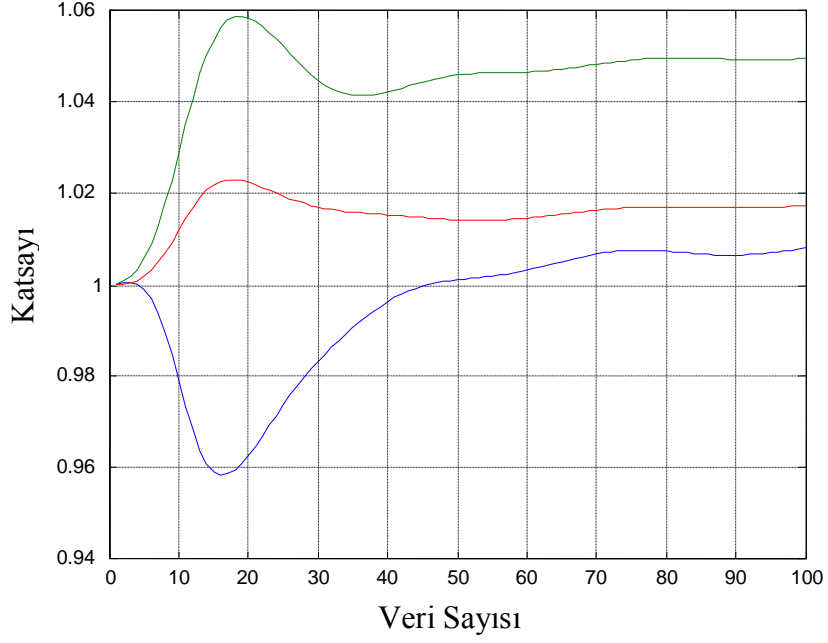
$$\mu_r = 0.0081$$

$$\mu_l = 0.0497$$

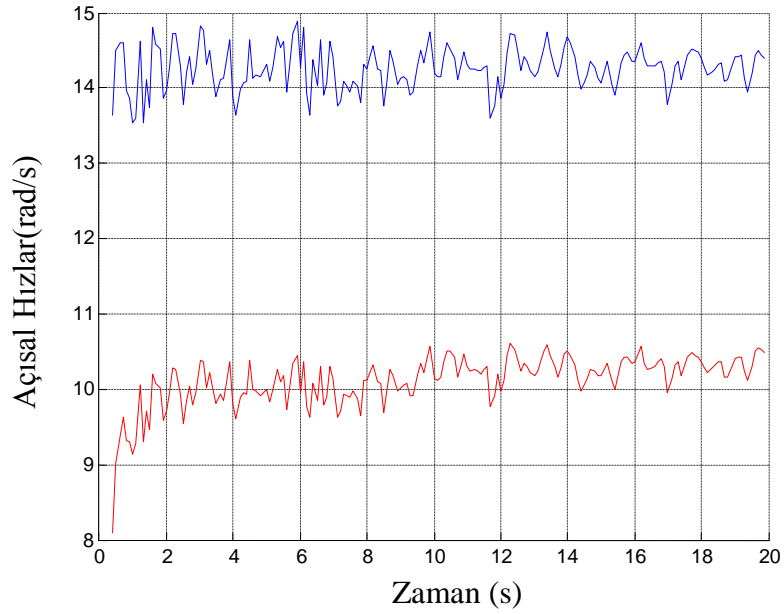
$$\mu_b = 0.0172$$

Bu değerlere göre yeniden oluşturulan sağ tekerlek, sol tekerlek çapları ve teker açıklığı aşağıdaki gibi düzenlenmiştir.

$$D_r = 95.7719, \quad D_l = 97.6209, \quad b = 193.2760$$



Şekil 7.48: TEST 3 parametre hata yaklaşımı



Şekil 7.49 : TEST 3 sağ ve sol tekerlek açısal hızları.

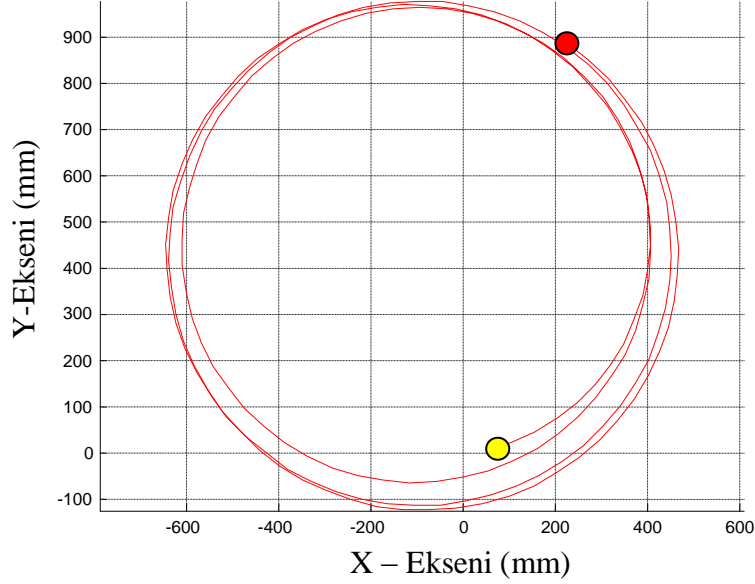
Kalibrasyon değerleri program tarafından ilk yüz noktaya göre hesaplanır, Şekil 7.48'de kalibrasyon değerlerinin hesaplanan nokta sayısına göre değişimi verilmiştir. Yeşil çizgi ile μ_l , mavi çizgiyle gösterilen μ_r ve kırmızı çizgiyle gösterilen μ_b dir. Hesaplanan

ilk 20 veriye göre büyük sapmalar göstermişlerse de 50 verinin hesaplanmasıyla gerçek değerlerine yaklaştırlardır.

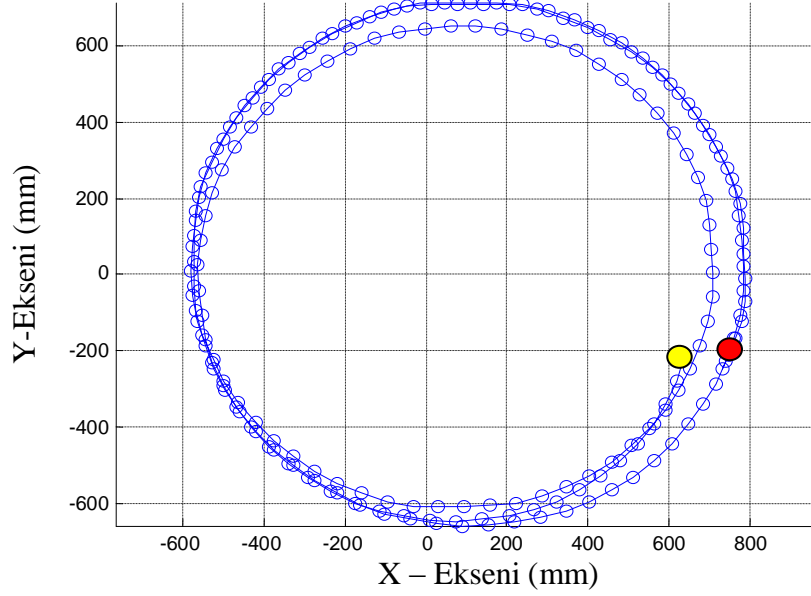
Şekil 7.49'da birinci testte elde dilen sağ ve sol tekerleğin açısal hızları görölmektedir. Mavi çizgiyle gösterilen sağ tekerlek hızı yaklaşık 14.5 rad/s dir. Kırmızıyla gösterilen sol tekerlek açısal hızı 10.5 rad/s olarak grafikte görölmektedir. Enkoder verileri eşit zaman aralıklarında alınmadığı için titreşimli görölmektedir. Açısal hızlardaki bu fark yüzünden hızlı dönen tekerlek dışarıda kalacak şekilde platform daire çizecektir. $R = a(V_r + V_l) / (V_r - V_l)$ formülüyle platformun döneceği yarıçapı bulabiliriz.

$$R = \frac{85 \left(14.5 \frac{95}{2} + 10.5 \frac{93}{2} \right)}{\left(14.5 \frac{95}{2} - 10.5 \frac{93}{2} \right)} = 499$$

Bu verilere göre platform dönüş çapı 998 mm olacaktır. Bu değer odometre verisinin oluşturduğu çap değerine yakın olması beklenir.



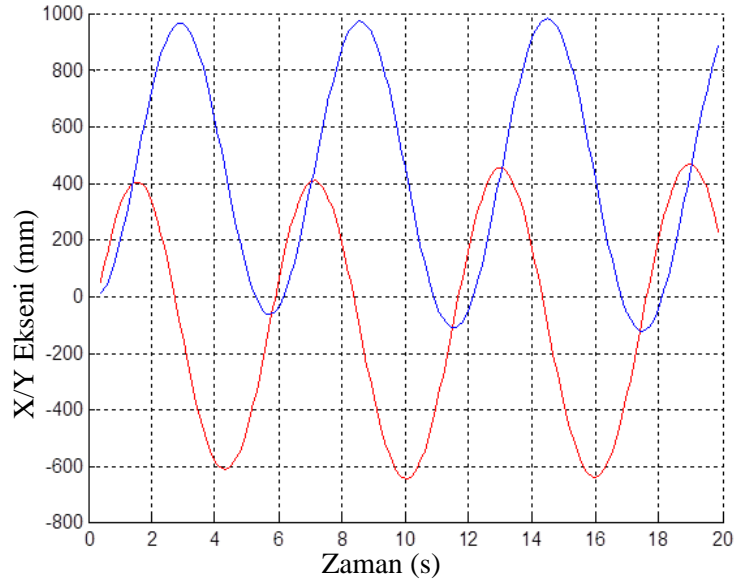
Şekil 7.50: TEST 3 odometre verisine göre platform yörüngesi.



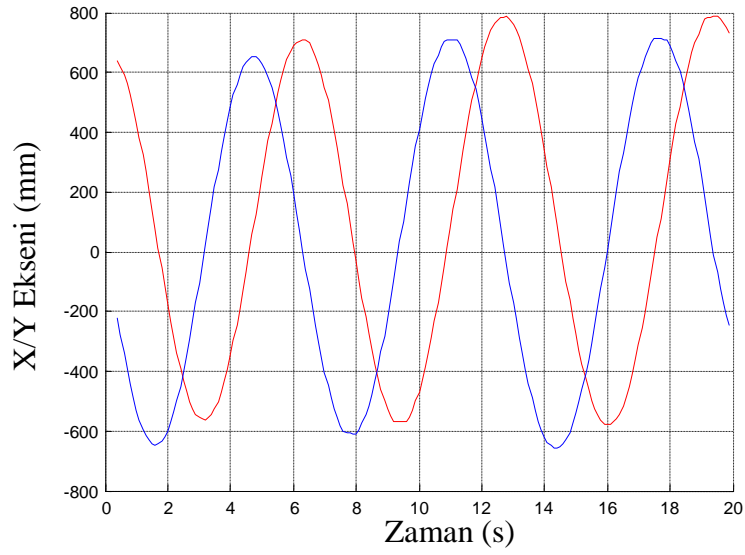
Şekil 7.51: TEST 3 kamera verisine göre platform yörüngesi.

Şekil 7.50 ve Şekil 7.51’de sırasıyla odometre ve kamera verilerinden elde edilen platform yörüngesi görülmektedir. Şekillerde içi dolu sarı daireyle gösterilen konum başlangıç konumu, içi dolu kırmızı daireyle gösterilen konum bitiş konumudur. Diğer testlerdeki gibi bu teste de platform ilk yörüngesine göre dışa doğru genişleyen bir yörünge takip etmiştir. Her iki grafikte, başlangıç yerine göre bitiş noktaları incelendiğinde bitiş noktalarının birbirinden farklı olduğu görülür. Kamera verisine göre bitiş noktası başlangıç noktasına yakinken (Şekil 7.51), odometre verisinde ise bitiş noktası başlangıç noktasından daha uzakta olduğu görülür (Şekil 7.50).

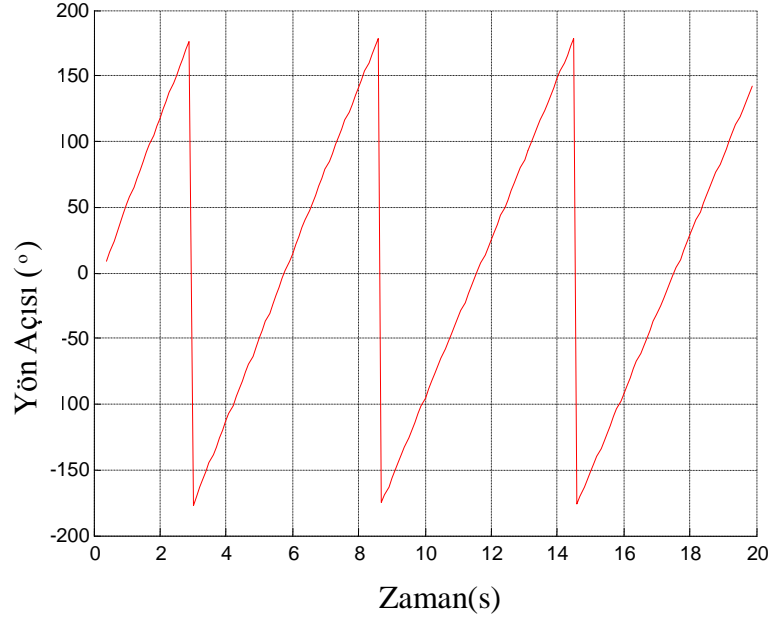
Şekil 7.52 ve Şekil 7.53 ‘de sırasıyla odometre ve kamera verilerine göre platformun X ve Y eksenindeki konumları görülmektedir. Kırmızı çizgiyle gösterilen X eksenini mavi çizgiyle gösterilen Y eksenindeki konumlardır. Odometre grafiğine baktığımızda platform üç tam bir yarım tur attığı görünmekte (Şekil 7.52) fakat kameraya göre platform üç tura yakın dönmüştür. Bunun nedeni gerçek nominal değer yerine bizim atadığımız farklı değerlerden kaynaklanmaktadır. Odometre verisine göre ilk turda yörünge yarı çapı yaklaşık 1000 mm, ikinci turda 1050 mm üçüncü turda 1150 mm civarındadır.(Şekil 7.52) Kamera verisine bakıldığında birinci turda 1300 mm, ikincisinde de 1300 mm ve üçüncü turda 1350 mm civarında olduğu görülebilir. (Şekil 7.53)



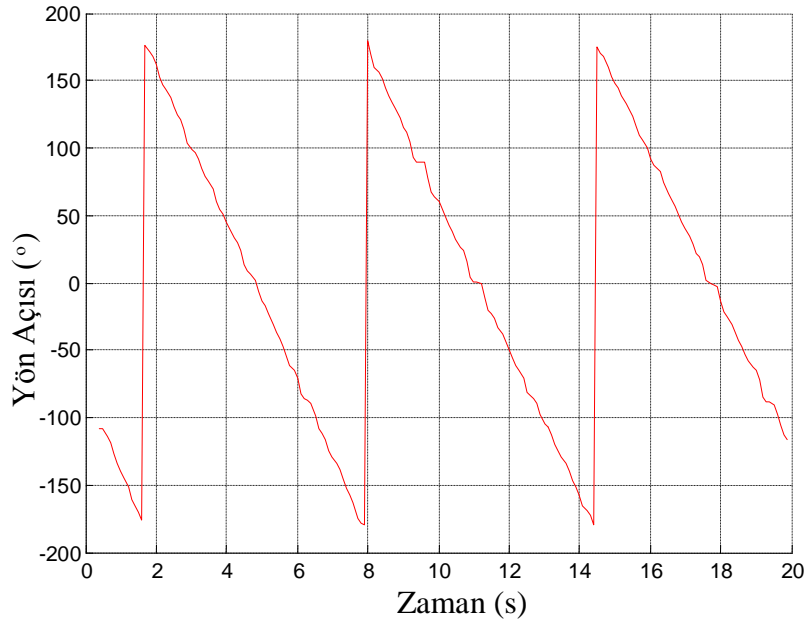
Şekil 7.52: TEST 3 odometre takibinde platform X/Y eksen konumu.



Şekil 7.53: TEST 3 kamera takibinde platform X/Y eksen konumu.



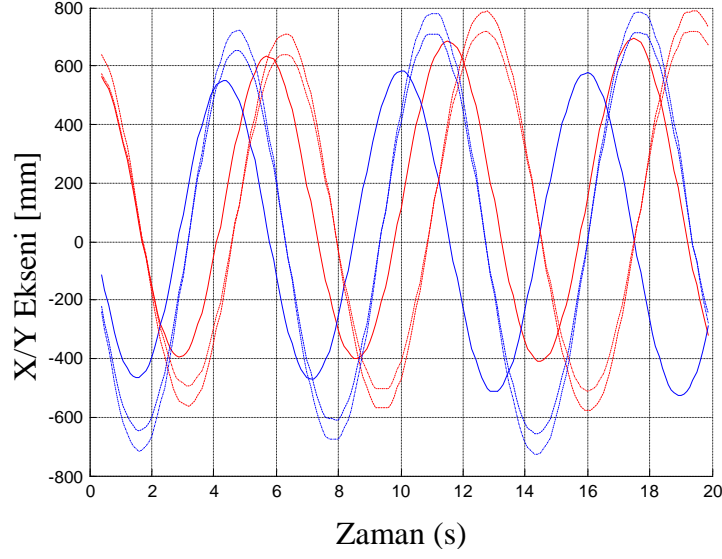
Şekil 7.54: TEST 3 odometre verisine göre platform yönü.



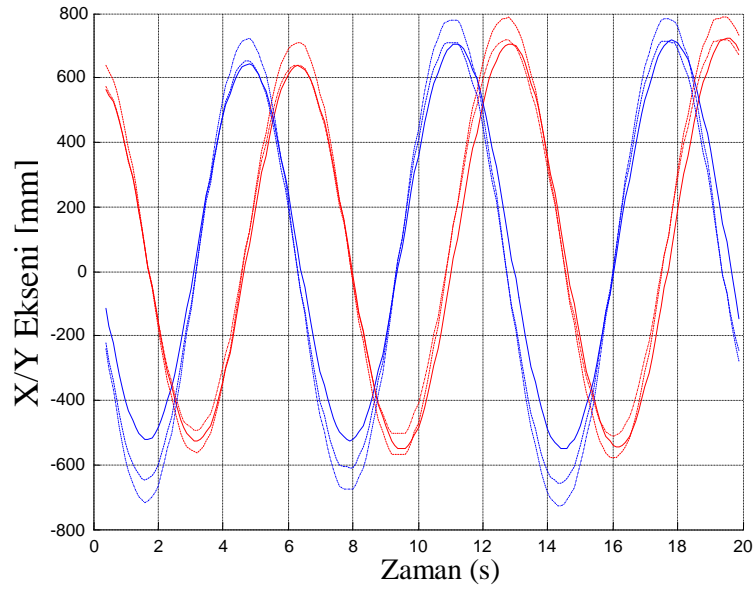
Şekil 7.55: TEST 3 kamera verisine göre platform yönü.

Şekil 7.54 ve Şekil 7.55'te platformun yön değişimi sırasıyla odometre ve kamera verilerine göre verilmiştir. Birinci testteki gibi ani sapmalar gözükmemektedir. Platformun tur sayısında bu grafiklerdeki yön değişimlerine bakılarak kolaylıkla bulunabilir.

Şekil 7.56 ve Şekil 7.57'de odometre, kamera ve düzeltilmiş kamera verileriyle oluşturulan kıyaslama grafikleri görülmektedir. Burada kırmızı düz çizgi odometre X eksenini, kırmızı kesik çizgiler kamera X eksenini ve kırmızı noktalı çizgiler düzeltilmiş kamera X eksenini



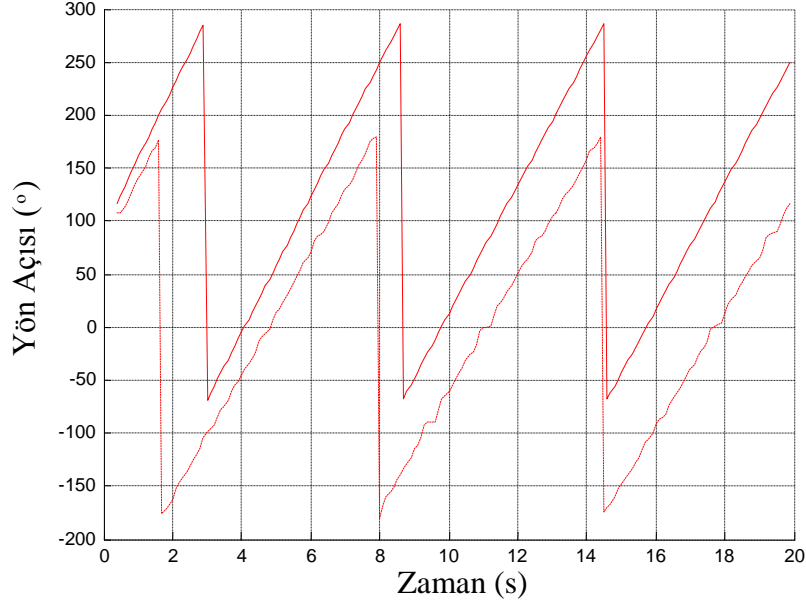
Şekil 7.56 : TEST 3 kalibrasyondan önce verilerin X/Y konum kıyaslaması.



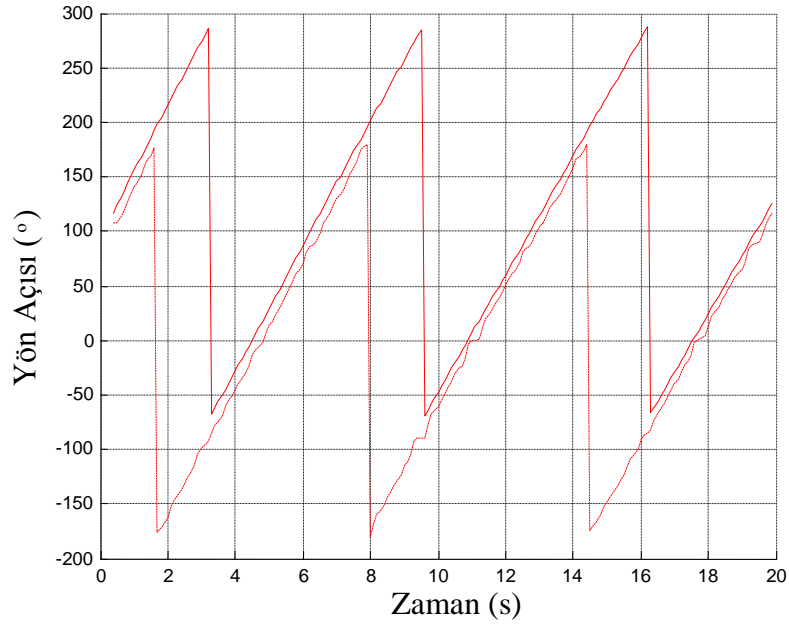
Şekil 7.57: TEST 3 kalibrasyondan sonra verilerin X/Y konum kıyaslaması.

göstermektedir; mavi düz çizgiler odometre Y eksenini mavi kesik çizgiler kamera Y eksenini ve mavi noktalı çizgiler düzeltilmiş kamera Y eksenini gösterir. Kalibrasyondan önce odometre eğrileri kamera eğrilerine göre uzak görünmekte (Şekil 7.56) kalibrasyon işlemi uygulandıktan sonra odometre eğrilerinin düzeldiği ve kamera eğrilerine yaklaştığı görülür (Şekil 7.57).

Şekil 7.58 ve Şekil 7.59'da platformun kamera ve odometre verilerinden ölçülen yön grafikleri verilmiştir. Şekil 7.58'deki grafik ham verilerle oluşturulan grafiklerdir. Bu grafikte kırmızı çizgi odometre verisinden elde edilen yönü, kırmızı kesik çizgiler ise kamera



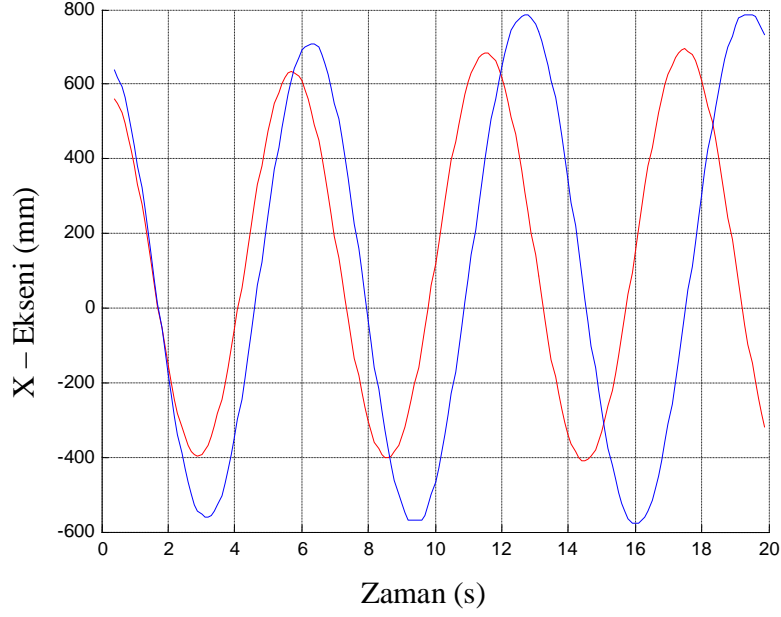
Şekil 7.58: TEST 3 kalibrasyondan önce odometre ve kamera platform yön açısı.



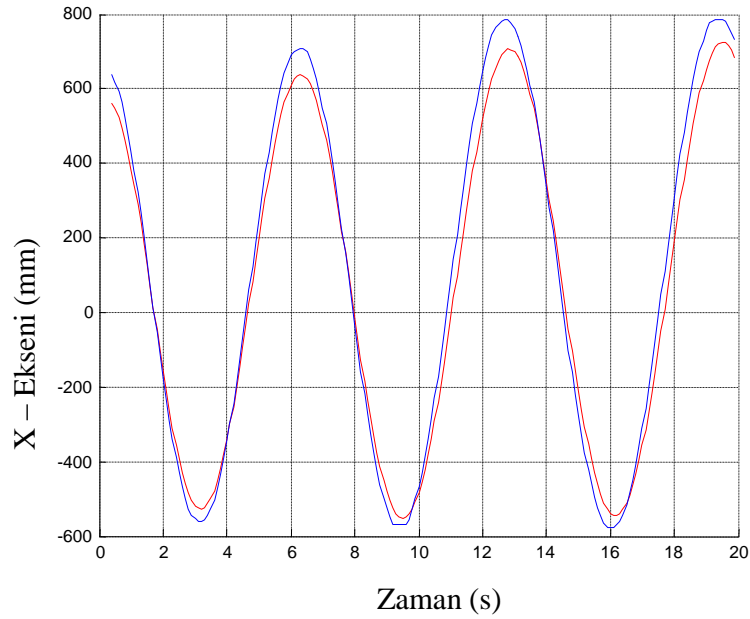
Şekil 7.59: TEST 3 kalibrasyondan sonra odometre ve kamera platform yön açısı.

verisinden elde edilen yönü göstermektedir. Kalibrasyon edilmeden önce odometre eğrisinin kamera eğrisini karşılamadığı görülüyor (Şekil 7.58). Kalibrasyon işlemi uygulandıktan sonra odometre eğrilerinin kamera eğrisine doğru yaklaştığı görülür(Şekil 7.59).

Şekil 7.60 ve Şekil 7.61'da platformun odometre ve kamera verilerine göre X eksenindeki konum değişikliği görülmekte, Şekil 7.60 kalibrasyon öncesine diğer grafikte kalibrasyon

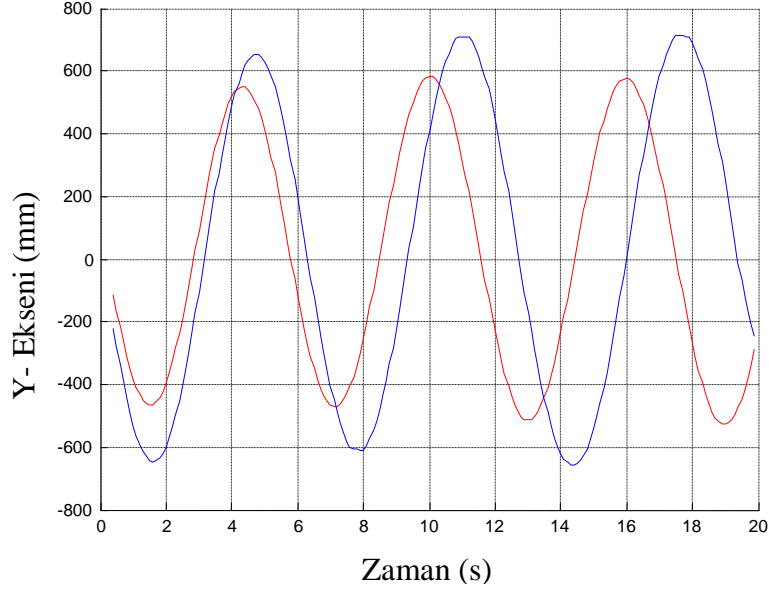


Şekil 7.60: TEST 3 kalibrasyondan önce odometre ve kamera platform X konumu.

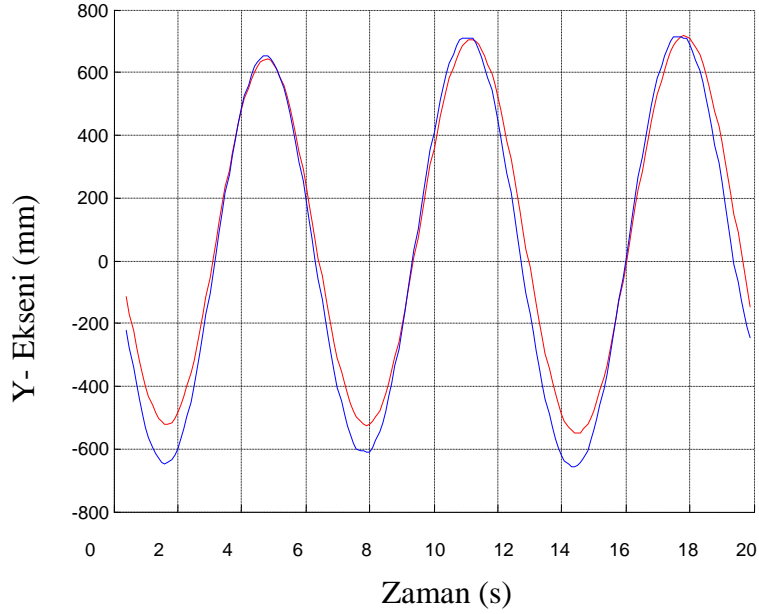


Şekil 7.61: TEST 3 kalibrasyondan sonra odometre ve kameraya platform X konumu.

sonrasına aittir. Kalibrasyon öncesinde odometre X konum eğrileriyle kamera eğrileri arasında büyük açıklıklar olduğu Şekil 7.60' da görülür. Kalibrasyon sonrasında eğrilerin birbirlerine çok yaklaştıkları Şekil 7.61'da görülebilir. Şekil 7.62 ve Şekil 7.63'de platformun odometre ve kamera verilerine göre Y eksenindeki konum değişikliği görülmekte, Şekil 7.62 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasında aittir. Kalibrasyon sonrasında odometre eğrilerinin kamera eğrilerini karşıladığı görülür (Şekil 7.63).

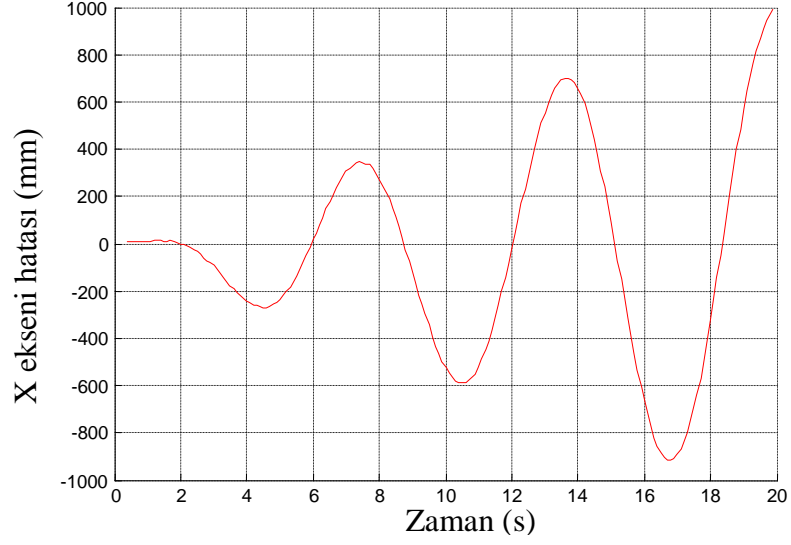


Şekil 7.62: TEST kalibrasyondan önce odometre ve kamera platform Y konumu.

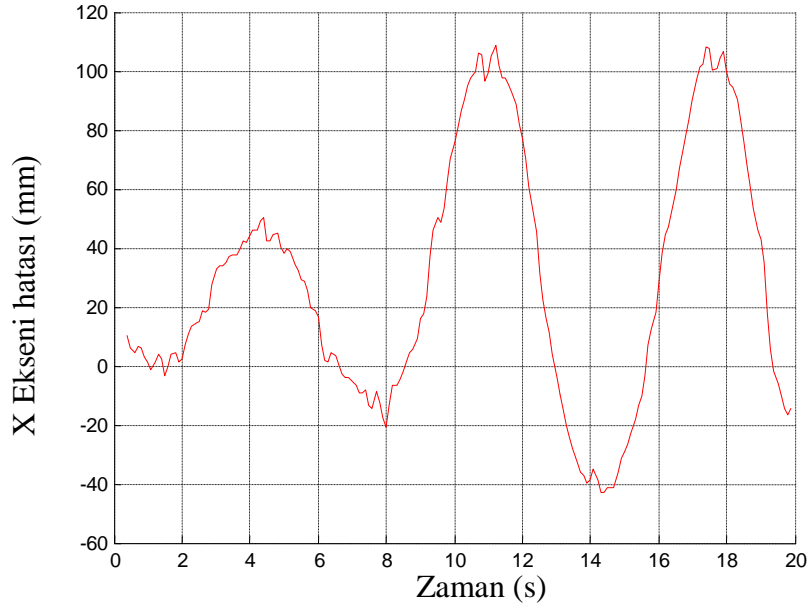


Şekil 7.63: TEST 3 kalibrasyondan sonra odometre ve kamera platform Y konumu

Şekil 7.64 ve Şekil 7.65'te sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan X konumu hataları gösterilmiştir. Şekil 7.64 incelendiğinde kalibrasyon öncesi hataların 1000 mm'ye kadar çıktığı görülmektedir. Şekil 7.65 kalibrasyon sonucu X konumu odometre hataları 120 mm'yi aşmamaktadır.

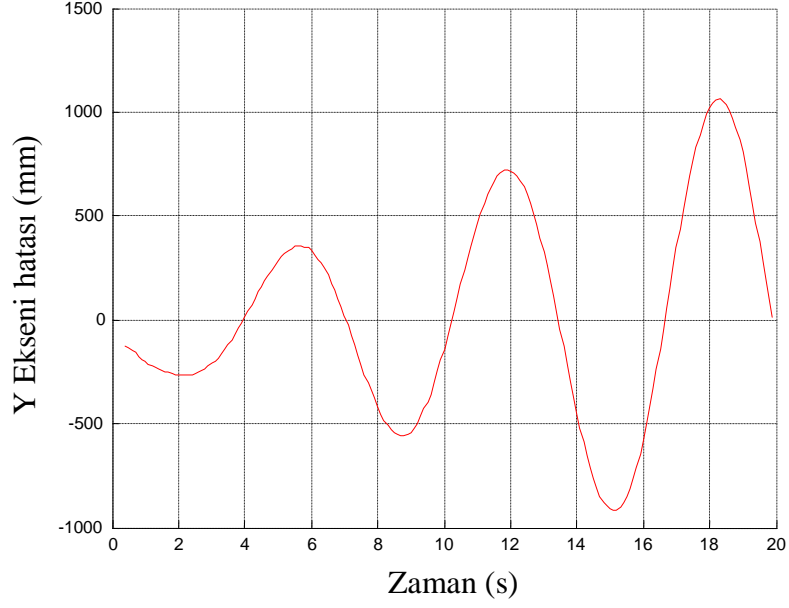


Şekil 7.64: TEST 3 kalibrasyondan önce odometre X eksen hatası.

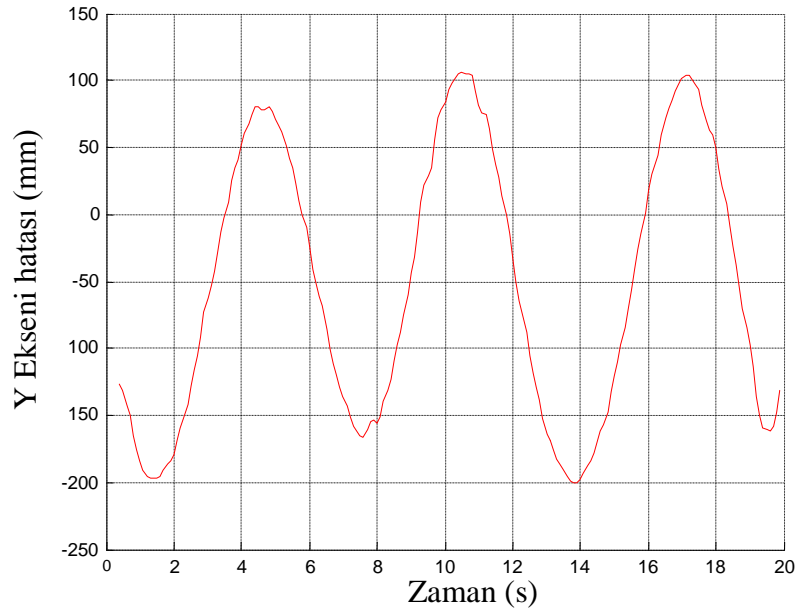


Şekil 7.65: TEST 3 kalibrasyondan sonra odometre X eksen hatası.

Şekil 7.65 ve Şekil 7.67’ de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan Y konumu hataları gösterilmiştir. Şekil 7.66 incelendiğinde kalibrasyon öncesi hataların 1100 mm’ye kadar çıktığı görülmektedir. Kalibrasyon sonrası oluşan Y konum hataları Şekil 7.67 ‘de 200 mm’ye kadar düştüğü görülüyor.

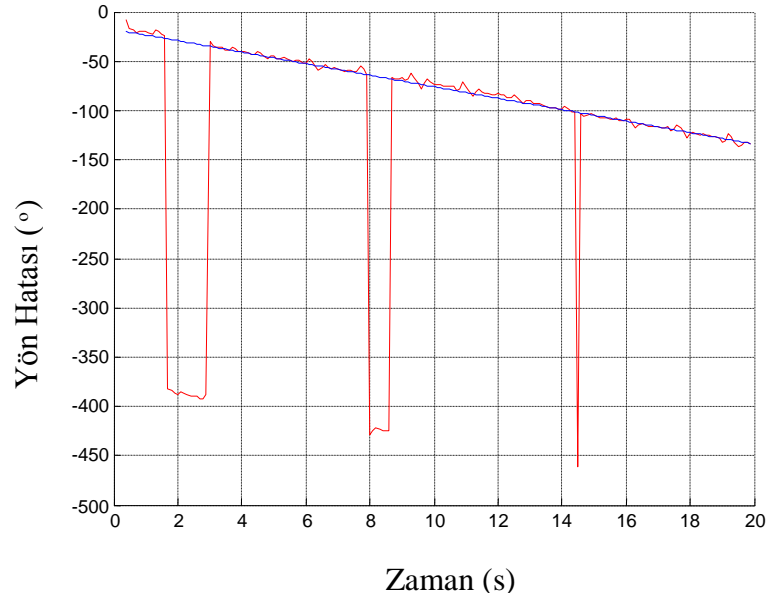


Şekil 7.66: TEST 3 kalibrasyondan önce odometre Y eksen hatası.

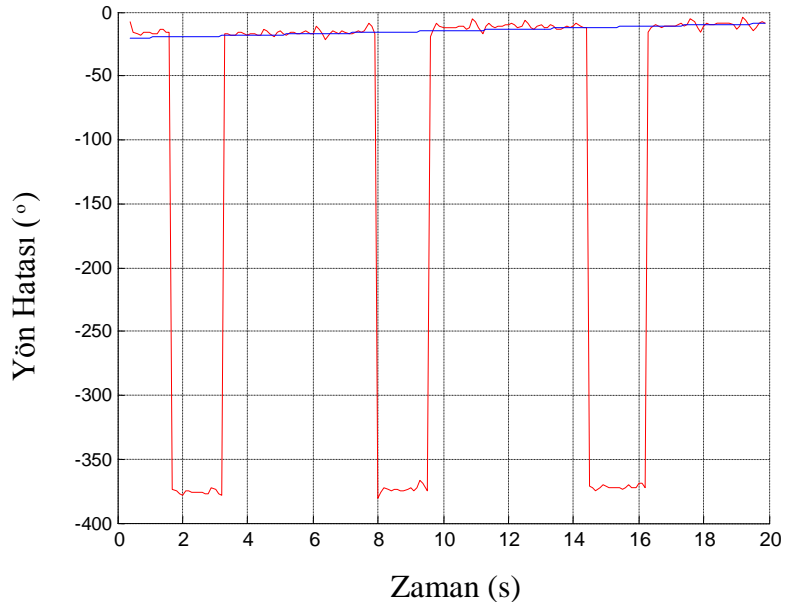


Şekil 7.67: TEST 3 kalibrasyondan sonra odometre Y eksen hatası.

Şekil 7.68 ve Şekil 7.69'de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan yön hataları gösterilmiştir. Kırmızı çizgiyle gösterilen düzeltilmemiş yön hataları, maviyle gösterilen eğri üzerine uydurulması yöntemiyle oluşturulan eğridir. Şekil 7.68'de kalibrasyon yapılmadan önceki yön hatasının 120° olduğu görülebilir. Kalibrasyon yapıldıktan sonra bu değer 10° ye kadar düşmüştür(Şekil 7.69)



Şekil 7.68: TEST 3 kalibrasyondan önce odometre yön hatası.



Şekil 7.69: TEST 3 kalibrasyondan sonra odometre yön hatası.

7.3.4 Veri Analizi Test 4

Dördüncü testte platform tekerleri aynı olmasına rağmen yazılım tarafındaki nominal ölçülerde değişikliğe gidilerek tekerlek çaplarında 6 mm fark olacak şekilde teker çap ölçüleri değiştirildi. Bu şekilde odometre eğrisinin daha da fazla sağması sağlandı.

Nominal ölçüler (mm)

$$D_r = 89 , D_l = 95 , b = 190$$

Örnekleme Zamanı: 0.1

Başlangıç belirsizlik matrisi $\mu_{rC} = 1 \times 10^{-6}$, $\mu_{rIC} = 1 \times 10^{-6}$, $\mu_{lC} = 1 \times 10^{-6}$

$$\begin{bmatrix} \mu_{rC} & 0 & 0 \\ 0 & \mu_{rIC} & 0 \\ 0 & 0 & \mu_{lC} \end{bmatrix} = \begin{bmatrix} 1 \times 10^{-6} & 0 & 0 \\ 0 & 1 \times 10^{-6} & 0 \\ 0 & 0 & 1 \times 10^{-6} \end{bmatrix}$$

Ölçüm Hataları : $k_s = 0.1$

Odometreye bağlı olan konum tahmin hatası zamanla artıyor. Ölçüm belirsizliği doğrusal olarak artar.

$$S_x = (1 + i \cdot k_s) \cdot 10$$

$$S_y = (1 + i \cdot k_s) \cdot 10$$

$$S_t = (1 + i \cdot k_s) \cdot 10^{-1}$$

Test verisinin ilk yüz verisi kullanılarak kalibrasyon değerleri çıkarılmıştır.

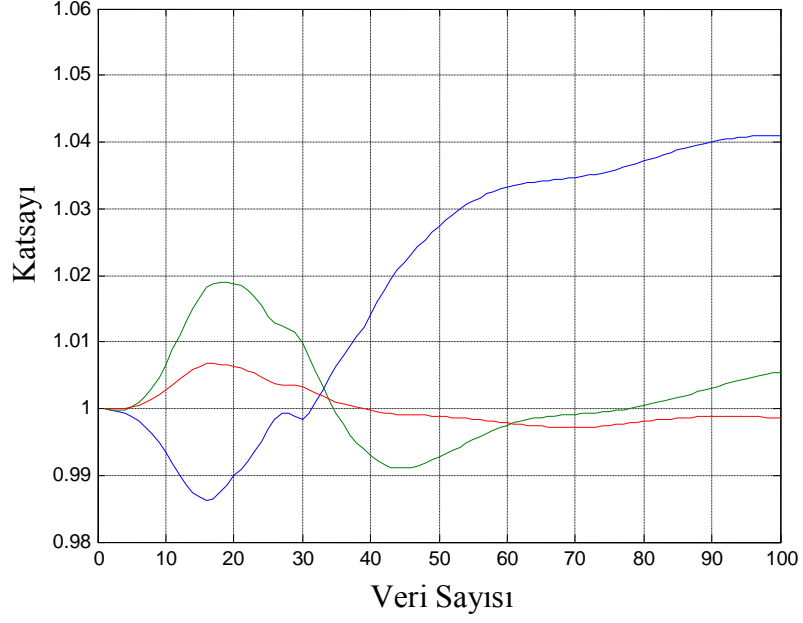
$$\mu_r = 0.0410$$

$$\mu_l = 0.0054$$

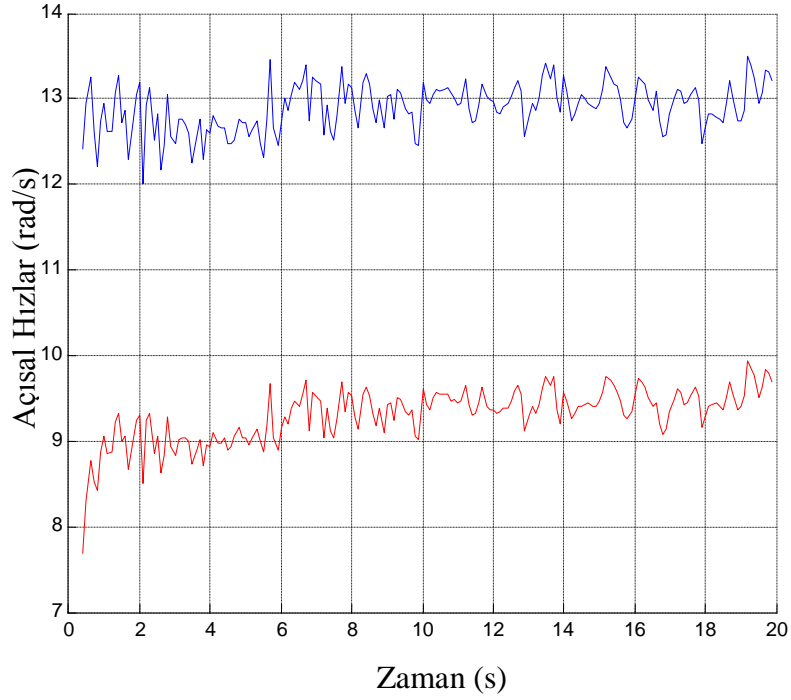
$$\mu_b = -0.0015$$

Bu değerlere göre yeniden oluşturulan sağ tekerlek, sol tekerlek çapları ve teker açıklığı aşağıdaki gibi düzenlenmiştir.

$$D_r = 92.6524 , D_l = 95.5165 , b = 189.7220$$



Şekil 7.70: TEST 4 parametre hata yaklaşımı



Şekil 7.71: TEST 4 sağ ve sol tekerlek açısal hızları.

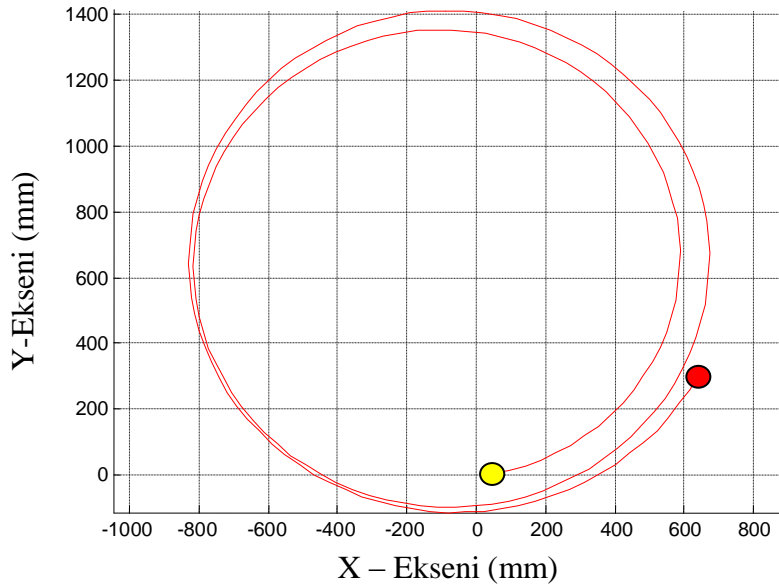
Şekil 7.70'da kalibrasyon değerlerinin hesaplanan veri sayısına göre değişimi verilmiştir. Yeşil çizgi ile μ_l , sol teker kalibrasyon değeri, mavi çizgiyle gösterilen μ_r , sağ teker

kalibrasyon değeri, kırmızı çizgiyle gösterilen μ_b değerinde tekerlek açıklığı kalibrasyon değeridir. 50 veri hesaplamasından sonra kalibrasyon değerleri son değerlerine yaklaşırlar.

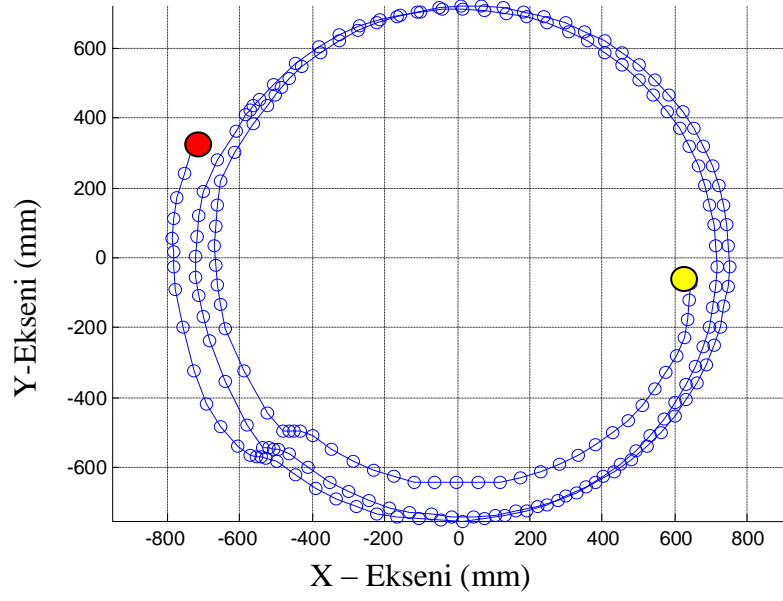
Şekil 7.71’te dördüncü testte elde edilen sağ ve sol tekerleğin açısal hızları görülmektedir. Mavi çizgiyle gösterilen sağ tekerlek hızı yaklaşık 13.2 rad/s dir. Kırmızıyla gösterilen sol tekerlek açısal hızı 9.6 rad/s olarak grafikte görülmektedir. Enkoder verileri eşit zaman aralıklarında alınmadığı için titreşimli görülmektedir. Açısal hızlardaki bu fark yüzünden hızlı dönen tekerlek dışarıda kalacak şekilde platform daire çizecektir. $R = a(V_r + V_l) / (V_r - V_l)$ formülüyle platformun döneceği yarıçapı bulabiliriz.

$$R = \frac{85 \left(13.2 \frac{89}{2} + 9.6 \frac{95}{2} \right)}{\left(13.2 \frac{89}{2} - 9.6 \frac{95}{2} \right)} = 675$$

Bu verilere göre platform dönüş çapı 1350 mm olacaktır. Bu değer odometre verisinin oluşturduğu çap değerine yakın olması beklenir.

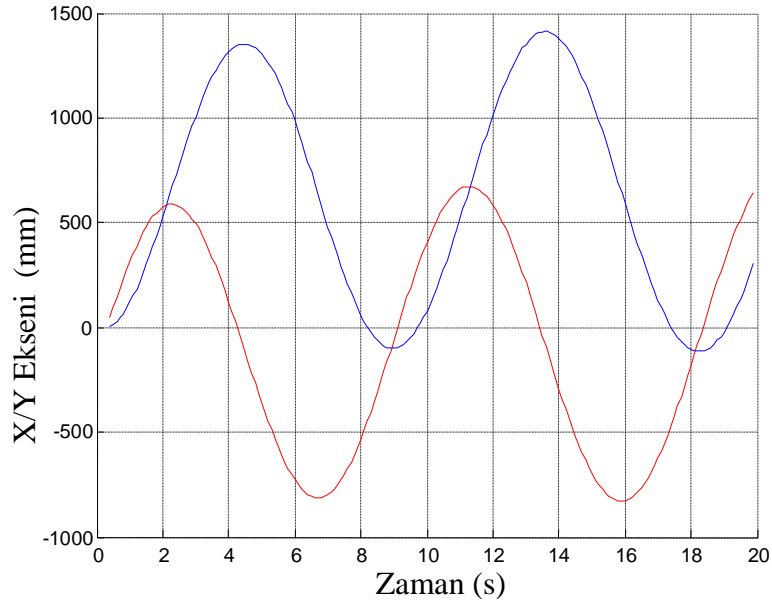


Şekil 7.72: TEST 4 odometre verisine göre platform yörüngesi.



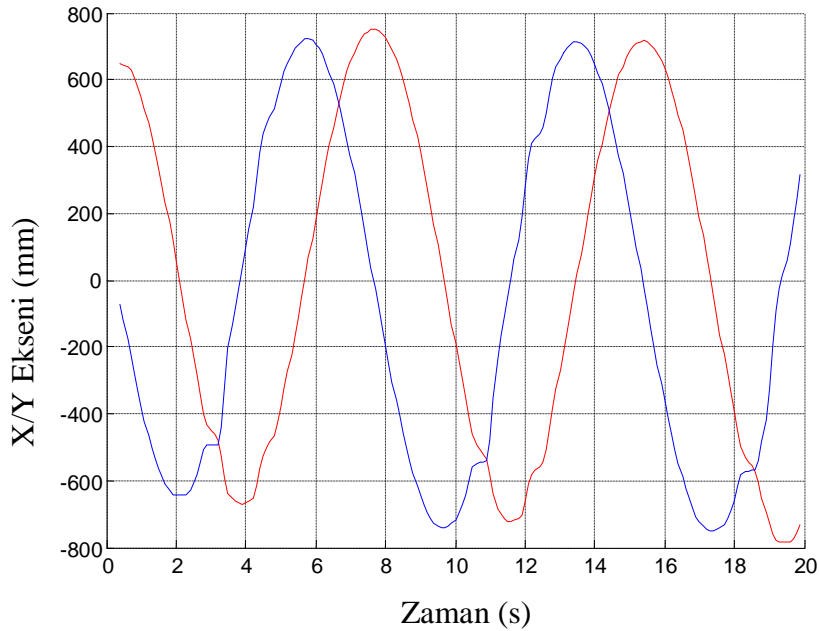
Şekil 7.73: TEST 4 kamera verisine göre platform yörüngesi.

Şekil 7.72 ve Şekil 7.73’de sırasıyla odometre ve kamera verilerinden elde edilen platform yörüngesi görülmektedir. Şekillerde içi dolu sarı daireyle gösterilen konum başlangıç konumu, içi dolu kırmızı daireyle gösterilen konum bitiş konumudur. Diğer testlerdeki gibi bu testte de platform ilk yörüngesine göre dışa doğru genişleyen bir yörünge takip etmiştir. Kamera verisinde yörüngede hafif sapmalar mevcuttur bu sapmalar nesne takibinden kaynaklanmış olabilir.



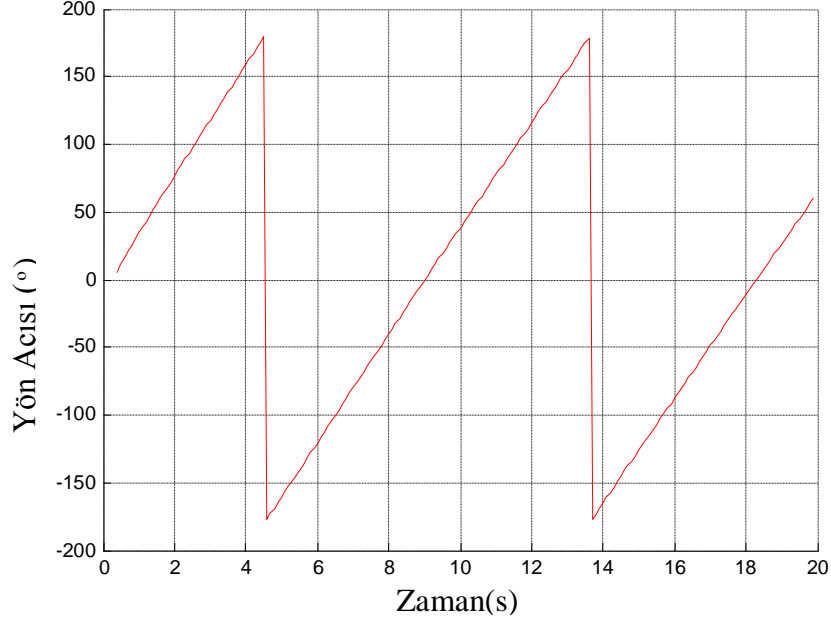
Şekil 7.74: TEST 4 odometre takibinde platform X/Y eksen konumu.

Şekil 7.74 ve Şekil 7.75 ‘de sırasıyla odometre ve kamera verilerine göre platformun X ve Y eksenindeki konumları görülmektedir. Kırmızı çizgiyle gösterilen X eksenini mavi çizgiyle gösterilen Y eksenindeki konumlardır. Odometre grafiğine baktığımızda platform iki tam tur attığı görülür (Şekil 7.74) , fakat kameraya göre platform iki tura ve bir yarım tur dönmüştür. Bunun nedeni gerçek nominal değer yerine bizim atadığımız farklı değerlerden kaynaklanmaktadır. Odometre verisine göre ilk turda yörünge yarı çapı yaklaşık 1450 mm, ikinci turda 1500 mm (Şekil 7.74). Kamera verisine bakıldığında birinci turda 1350 mm, ikincisinde de 1450 mm civarında olduğu görülebilir (Şekil 7.75).

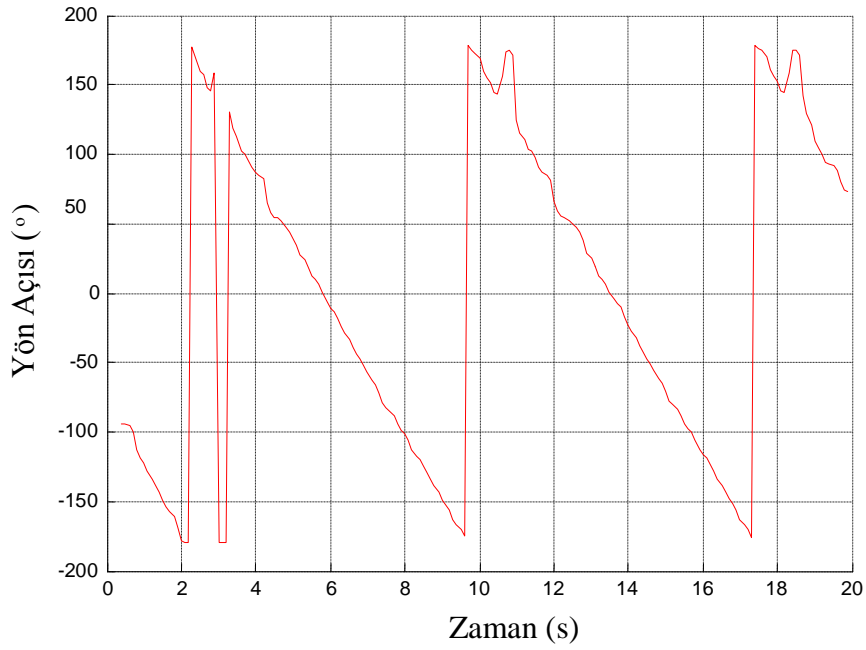


Şekil 7.75: TEST 4 kamera takibinde platform X/Y eksen konumu.

Şekil 7.76 ve Şekil 7.77’te platformun yön değişimi sırasıyla odometre ve kamera verilerine göre verilmiştir. Birinci testteki gibi ani sapmalar gözlemlenmektedir. Bu sapmaların nedenini $atan2(x,y)$ fonksiyonundan kaynaklandığı belirtildi. Şekil 7.77’de grafikte uç kısımda hafif sapmalar olduğu görülmekte bunun nedeni nesne takip esnasında nesneyi tam göremediğinden kaynaklanabilir.



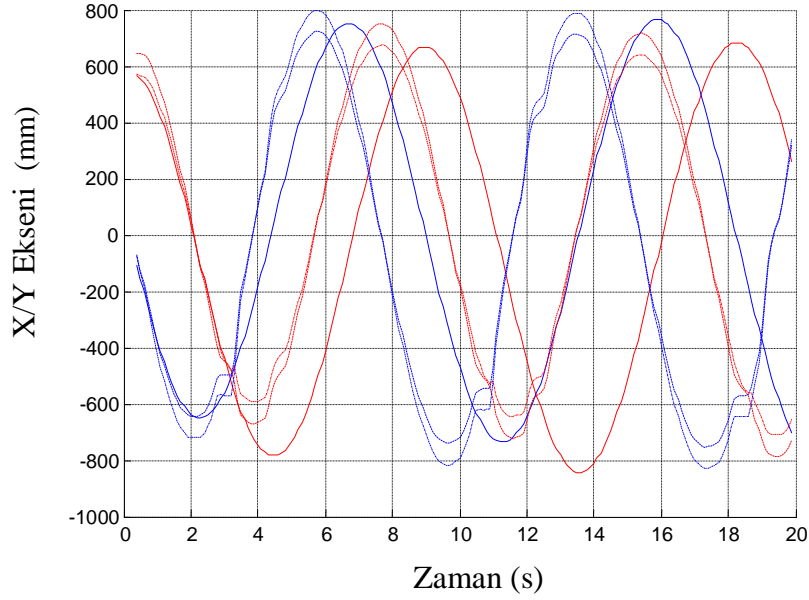
Şekil 7.76: TEST 4 odometre verisine göre platform yönü.



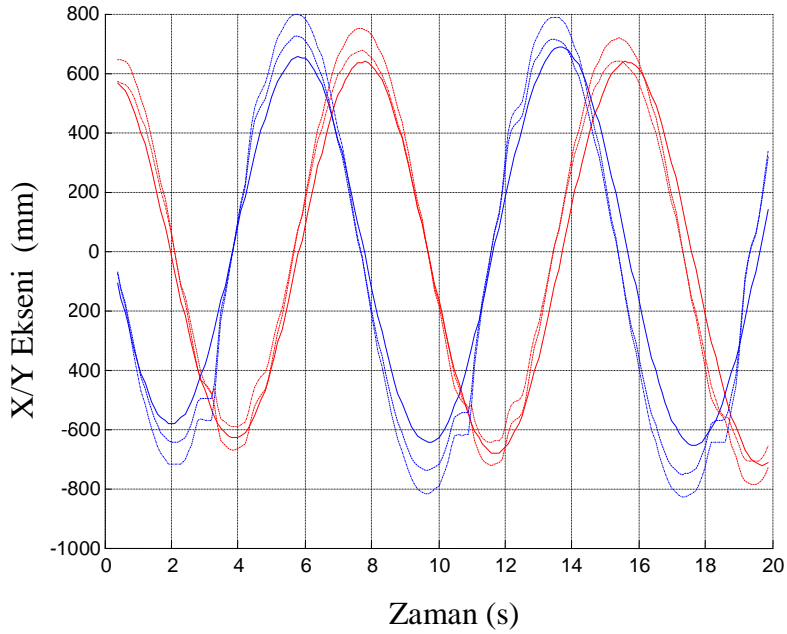
Şekil 7.77 :TEST 4 kamera verisine göre platform yönü.

Şekil 7.78 ve Şekil 7.79’da odometre, kamera ve düzeltilmiş kamera verileriyle oluşturulan kıyaslama grafikleri görülmektedir. Burada kırmızı düz çizgi odometre X eksenini, kırmızı kesik çizgiler kamera X eksenini ve kırmızı noktalı çizgiler düzeltilmiş kamera X eksenini göstermektedir; mavi düz çizgiler odometre Y eksenini mavi kesik çizgiler kamera Y

eksenini ve mavi noktali çizgiler düzeltilmiş kamera Y eksenini gösterir. Kalibrasyondan önce odometre eğrileri kamera eğrilerine göre uzak görünmekte (Şekil 7.78) kalibrasyon işlemi uygulandıktan sonra odometre eğrilerinin düzeldiği ve kamera eğrilerini karşıladığı görülür (Şekil 7.79).

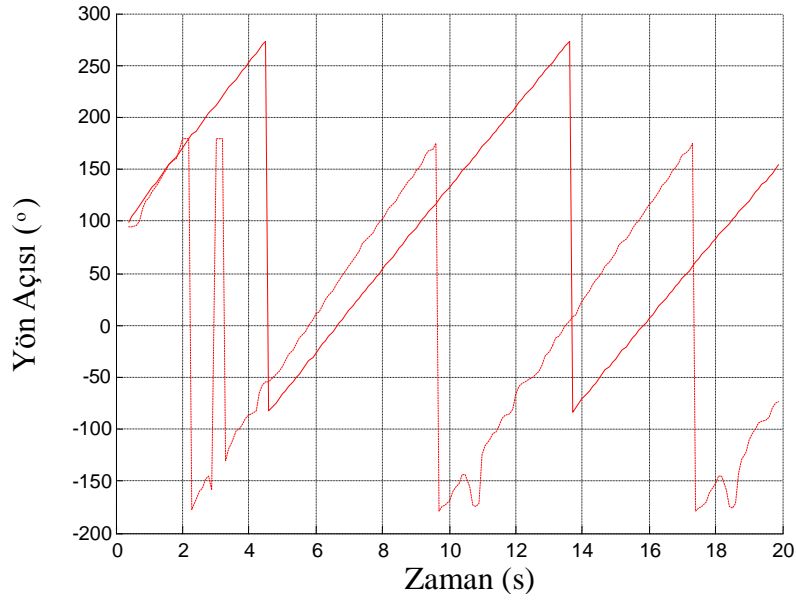


Şekil 7.78 : TEST 4 kalibrasyondan önce verilerin X/Y konum kıyaslaması.

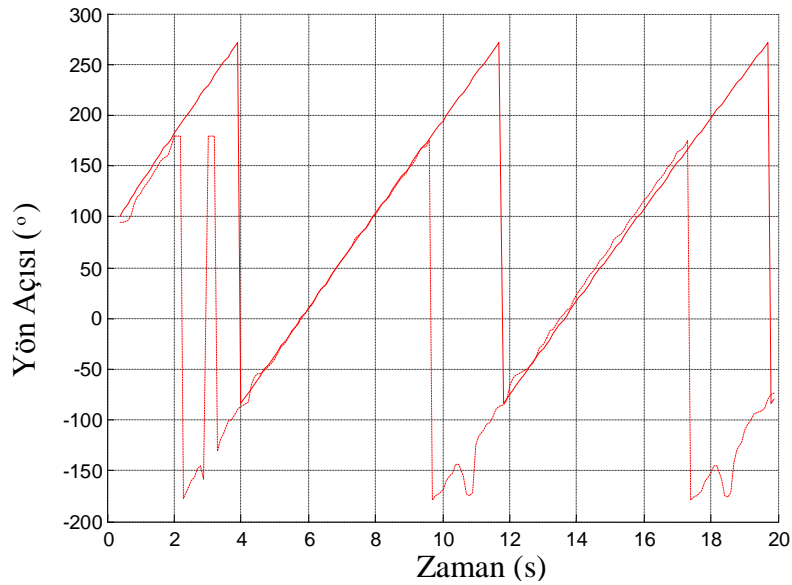


Şekil 7.79 : TEST 4 kalibrasyondan sonra verilerin X/Y konum kıyaslaması.

Şekil 7.80 ve Şekil 7.81’de platformun kamera ve odometre verilerinden ölçülen yön grafikleri verilmiştir. Şekil 7.80’deki grafik kalibrasyondan önce ham verilerle oluşturulan grafikdir. Bu grafikte kırmızı çizgi odometre verisinden elde edilen yönü, kırmızı kesik çizgiler ise kamera verisinden elde edilen yönü göstermektedir. Kalibrasyon edilmeden önce odometre eğrisinin kamera eğrisini karşılamadığı görülüyor (Şekil 7.80). Kalibrasyon işlemi uygulandıktan sonra odometre eğrilerinin kamera eğrisine doğru yaklaştığı görülür(Şekil 7.81).

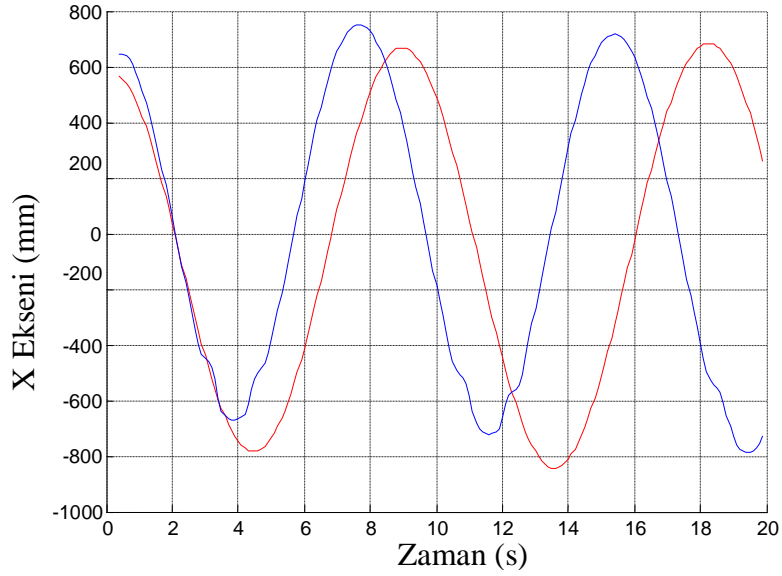


Şekil 7.80: TEST 4 kalibrasyondan önce odometre ve kamera platform yön açısı.

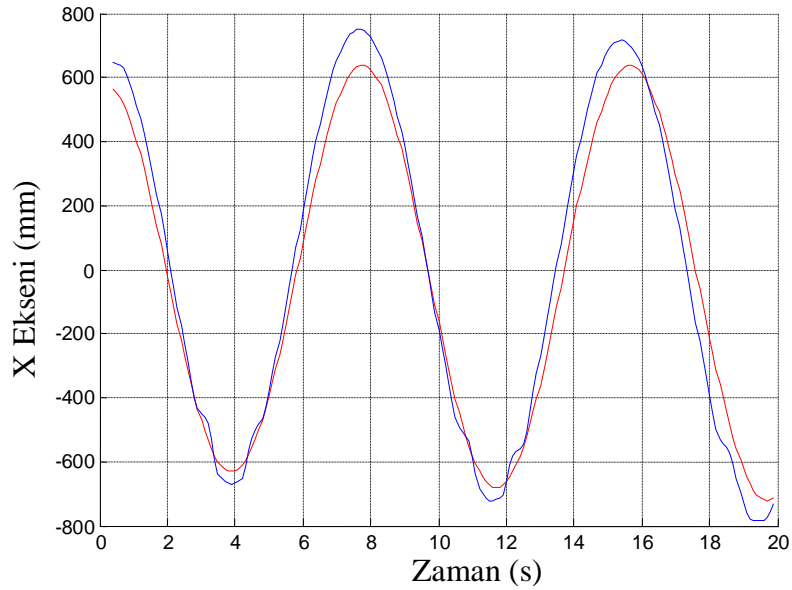


Şekil 7.81: TEST 4 kalibrasyondan sonra odometre ve kamera platform yön açısı.

Şekil 7.82 ve Şekil 7.83’da platformun odometre ve kamera verilerine göre X eksenindeki konum değişikliği görülmekte, Şekil 7.82 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasına aittir. Kalibrasyon öncesinde odometre X konumu eğrisiyle kamera X konumu eğrisi arasında büyük farklar olduğu Şekil 7.82’ de görülür. Kalibrasyon sonrasında odometre hatalarının düzeltilmesiyle, eğrilerin birbirlerine çok yaklaştıkları Şekil 7.83’de görülebilir.

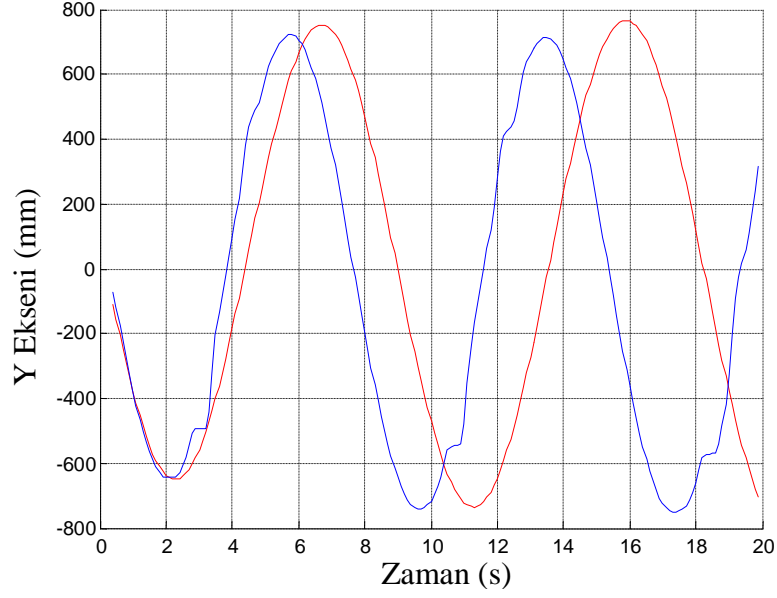


Şekil 7.82: TEST 4 kalibrasyondan önce odometre ve kamera platform X konumu.

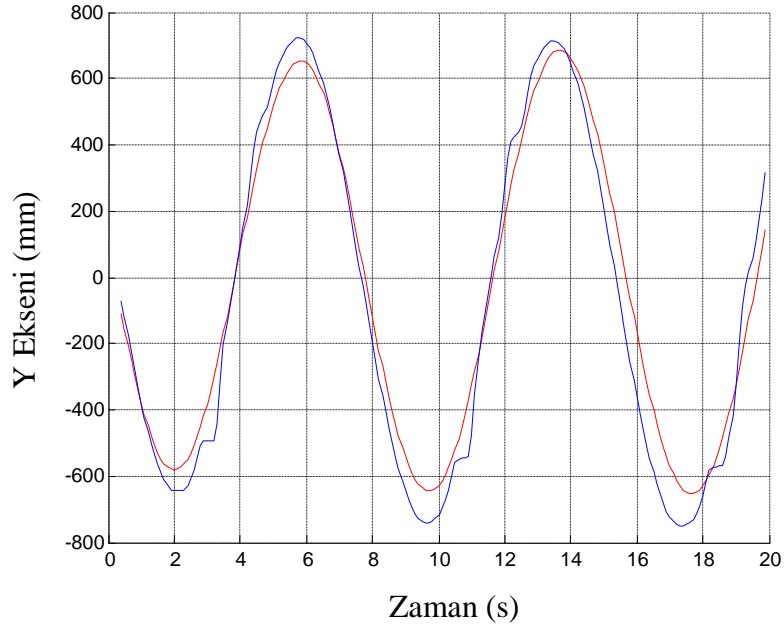


Şekil 7.83: TEST 4 kalibrasyondan sonra odometre ve kamera platform X konumu.

Şekil 7.84 ve Şekil 7.85’de platformun odometre ve kamera verilerine göre Y eksenindeki konum değişikliği görülmekte, Şekil 7.84 kalibrasyon öncesine diğer grafikte kalibrasyon sonrasına aittir. Kalibrasyon sonrasında odometre hatalarının düzeltilmesiyle odometre Y konum eğrisinin kamera Y konumu eğrisine yaklaştığı görülür (Şekil 7.85).

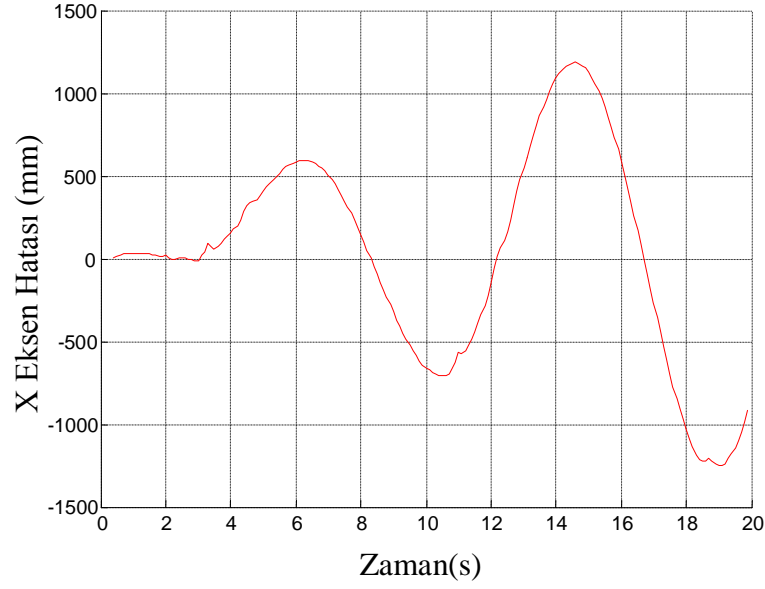


Şekil 7.84: TEST 4 kalibrasyondan önce odometre ve kamera platform Y konumu.

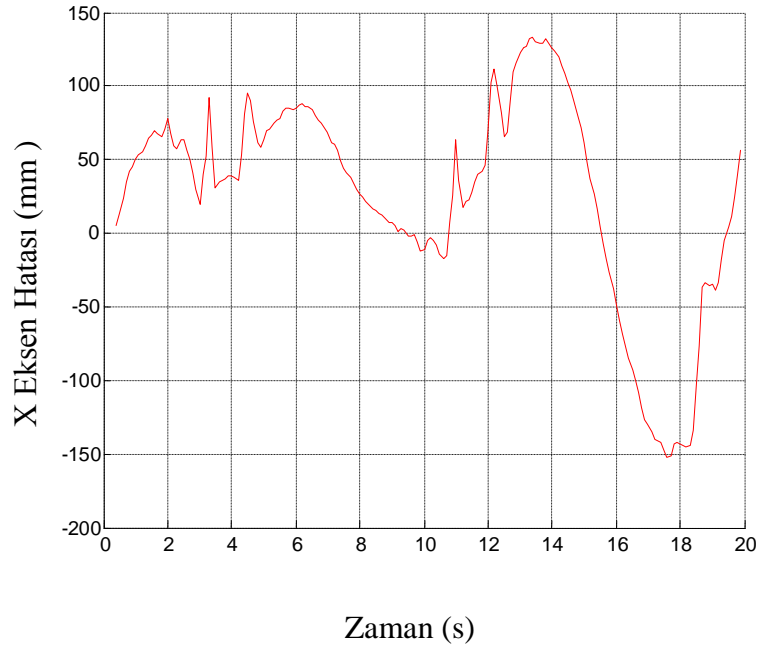


Şekil 7.85: TEST 4 kalibrasyondan sonra odometre ve kamera platform Y konumu.

Şekil 7.86 ve Şekil 7.87’da sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan X konumu hataları gösterilmiştir. Şekil 7.86 incelendiğinde kalibrasyon öncesi hataların 1300 mm’ye kadar çıktığı görülmektedir. Şekil 7.87 kalibrasyon sonucu X konumu odometre hataları 150 mm’yi aşmamaktadır.

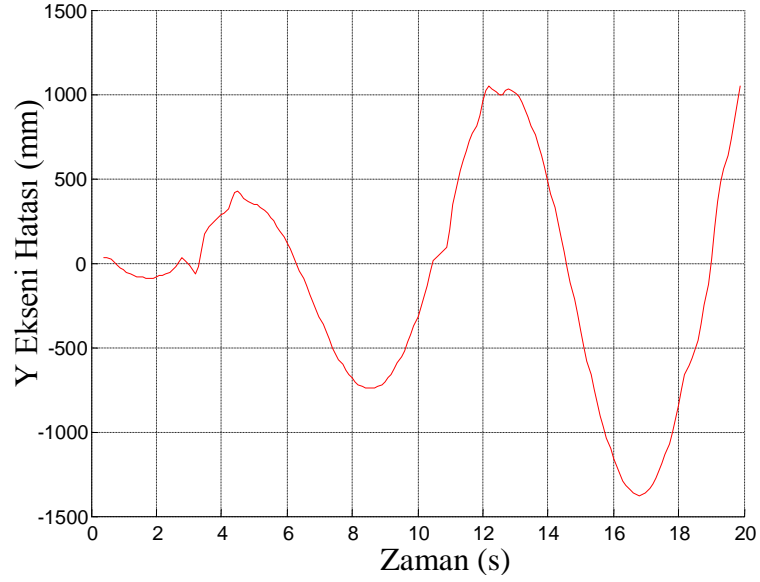


Şekil 7.86: TEST 4 kalibrasyondan önce odometre X eksen hatası.

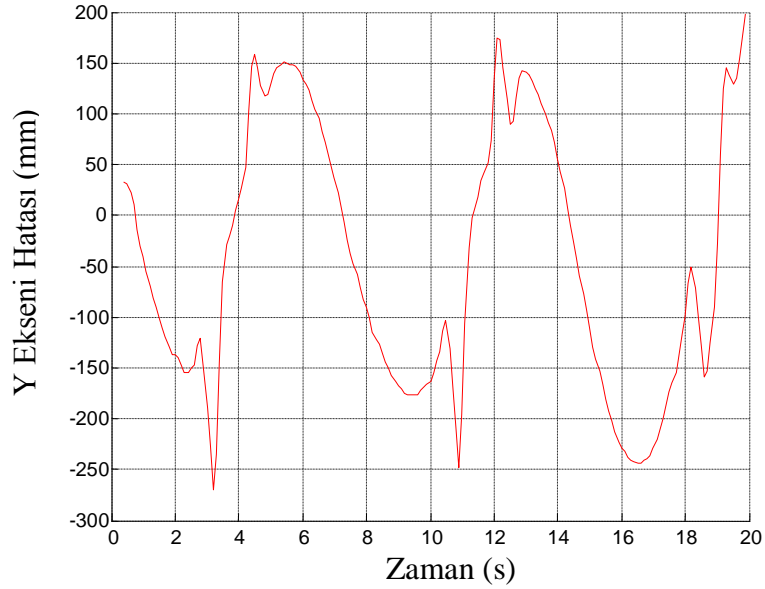


Şekil 7.87: TEST 4 kalibrasyondan sonra odometre X eksen hatası.

Şekil 7.88 ve Şekil 7.89’ de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan y konumu hataları gösterildi. Şekil 7.88 incelendiğinde kalibrasyon öncesi hataların 1400 mm’ye kadar çıktığı görülür. Kalibrasyon sonrası oluşan Y konum hataları Şekil 7.89 ‘da 275 mm’ye kadar düştüğü görülmüyor.

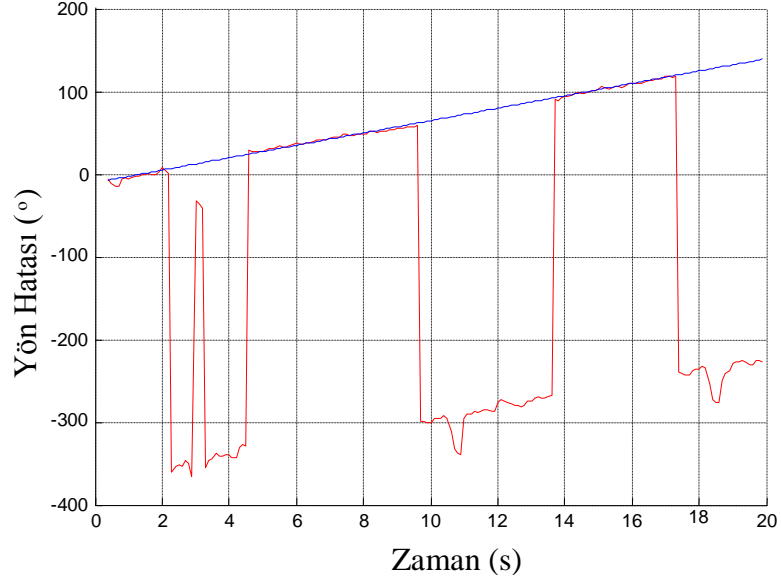


Şekil 7.88: TEST 4 kalibrasyondan önce odometre Y eksen hatası.

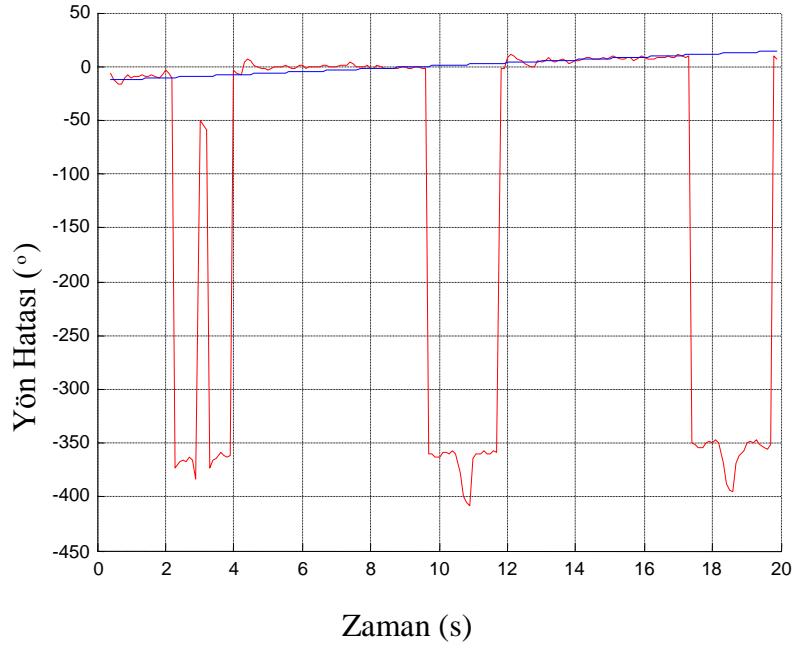


Şekil 7.89: TEST 4 kalibrasyondan sonra odometre Y eksen hatası

Şekil 7.90 ve Şekil 8.91’de sırasıyla kalibrasyondan önce ve kalibrasyondan sonra oluşan yön hataları gösterildi. Şekil 7.90’da kalibrasyon yapılmadan önceki yön hatasının 150° ye kadar çıktığı görülebilir. Kalibrasyon yapıldıktan sonra bu değer 10° ye kadar düştü.(Şekil 7.91)



Şekil 7.90: TEST 4 kalibrasyondan önce odometre yön hatası.



Şekil 7.91: TEST 4 kalibrasyondan sonra odometre yön hatası.

7.4 Sonu

Hareketli platform, oluřturulan test dzenerinde dairesel yrnge takip ettirilerek testler yapılmıřtır. Alınan bu gerek veriler, teorik alıřmada referans aldıđımız makalenin [7] kinematik parametre tahmin metodu kullanılarak iřlenmiř ve kalibrasyon deđerleri bulunmuřtur. Bu deđerlerle kinematik parametreler tekrar gncellenmiř ve odometer verileri tekrar iřlenerek kameradan alınan verilerle kıyaslanmıřtır. Bu veriler yapılan herbir test iin 7.3.1, 7.3.2, 7.3.3, 7.3.4 blmlerinde grafiklerle gsterilmiř, analiz edilmiř ve yorumlanmıřtır. Yapılan bu veri analizi sonularında referans alınan alıřmanın [7] kinematik parametre tahminiyle oluřturduđu kalibrasyon deđerlerinin, odometer verilerine uygulanmasıyla, gncellenen odometre verilerinin kameradan alınan mutlak deđerlere yaklařtıđı grlmřtr.

8 SONUÇLAR ve TARTIŞMA

Bu çalışmada, kinematik değişken parametre tahmininde kullanılmak üzere hareketli platform tasarımı ve imalatı gerçekleştirildi. Hareketli platform sürüş sistemleri incelenerek platformda kullanılacak olan sürüş sistemi seçildi ve bu yönde tasarım ve imalat gerçekleştirildi. Platformun kontrolü, veri toplaması ve haberleşmesi için gerekli donanımlar seçildi. Kontrol mimarisi kısmında platform devresi oluşturuldu ve bağlantıları detaylı şekilde anlatıldı. Bilgisayar ve mikrokontrolcü tarafındaki komutların hangi işlevleri gerçekleştirdiği açıklandı. Bilgisayarla, platform devresi arasındaki iletişimin nasıl gerçekleştiği ve hangi safhada hangi veri paketinin gönderildiği belirtildi. Deneysel çalışmalar için test düzeneğinin nasıl oluşturulduğu, deneysel çalışmalar esnasında mutlak veri ölçecek olan kameranın kalibrasyonu anlatıldı.

Parametre tahmini için teori kısmının nasıl oluşturduğu anlatıldı. Verilerin test ortamından nasıl toplandığı belirtildi. Yapılan dört test için kamera ve odometre verisine göre ayrı ayrı platform yörüngesi, platformun X eksenini, platformun Y eksenini konum grafikleri çıkartıldı. Parametre kalibrasyonu öncesi ve sonrası oluşan durumların kıyaslanabilmesi için odometre ve kamera verilerinin iç içe olduğu grafikler çıkarıldı. Odometre hatalarını daha net görebileceğimiz platform X eksenini, Y eksenini ve platform yönü hatalarını gösteren kalibrasyon öncesi ve kalibrasyon sonrası hata grafikleri oluşturuldu. Bu grafikler 7.3 veri analizi konusunda gösterildi ve yorumlandı.

Geliştirilen bu hareketli platform test düzeneğinde çalıştırılarak ham odometre ve kamera verisi alındı. Alınan bu verileri 7.2 teori kısmında anlatılan kalibrasyon metoduyla işlenerek kinematik parametre kalibrasyon değerleri çıkartıldı. Bu değerlere göre, platform kinematik parametre değerleri değiştirilerek odometre verileri tekrar işlendi. Bu işlemin sonunda kameradan alınan mutlak X eksenini, Y eksenini konumları ve platform yön değeri ile kıyaslandı. Kıyaslama sonucunda kalibrasyon metodunun odometre verilerini kalibre ederek kameradan alınan mutlak verilere yaklaştığı 7.3.1, 7.3.2, 7.3.3 ve 7.3.4 konularında test analizlerinde gösterildi.

İleride yapılacak çalışmalarda hareketli platformun sağ sol tekerlek çapları değiştirilecek şekilde farklı tekerlek tasarımları yapılarak hareketli platformun farklı tekerlek çaplarında kalibrasyon testleri yapılabilir. Tekerlek açıklığını kolaylıkla değiştirebilecek tekerlek bağlantı sistemleri geliştirilebilir, böylece daha esnek platforma sahip olunacaktır.

Platformun kinematik deęişkenleri bu esnek tasarımla her test başlangıcında kolaylıkla deęiştirilebilir. Hareketli platformlarda tekerlek çapları aynı fakat malzemeleri farklı olan tekerlekler bağlanarak testler yapılabilir. Farklı malzemelerin zeminle yaptıkları sürtünmeler farklı olacağından her iki tekerlek, kaymalardan dolayı farklı hareket edecektir.

KAYNAKLAR

- [1] <http://www.robotory.com/2002/06/plustech-walking-technology-walking.html>
- [2] http://www.mbari.org/news/news_releases/2002/Dorado/Dorado2.html
- [3] **Siciliano**, Bruno, **Khatib**, Oussama, *Springer Handbook of Robotics*, Springer, **2008**
- [4] http://en.wikipedia.org/wiki/ActivMedia_Robotics#mediaviewer/File:ActivMedia_Pioneer_3-AT_robot.jpg
- [5] <http://www.k-team.com/mobile-robotics-products/khepera-iii>
- [6] <http://www.irobot.com/For-Defense-and-Security/Robots/310-SUGV.aspx#Military>
- [7] CU Dogruer, Estimation of odometer parameters with MMAE and LSE, *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference*, 1728-1733 , Besacon ,**2014**
- [8] Borenstein J. , Feng L. ,Measurement and Correction of Systematic Odometry Error in Mobile Robots, *IEEE Transactions on Robotics and Automation*, Vol. 12, No.6, **1996**
- [9] Antonelli G. , Chiaverini S. , Linear estimation of the physical odometric parameters for differential-drive mobile robots, *Autonomous Robot*, 23, 59-68, **2007**
- [10] Antonelli G. , Chiaverini S. ,Experimental Odometry Calibration of the Mobile Robot Khepera II Based on the Least-Squares Technique, *Robotics and Automation*, 1465 – 1470, **2005**
- [11] Antonelli G. , Chiaverini S. , A Deterministic Filter for Simultaneous Localization and Odometry Calibration of Differential-Drive Mobile Robots, *Proceedings of the 3rd European Conference on Mobile Robots*, EMCR **2007**
- [12] Jung C. , Chung W. , Accurate Calibration of Two Wheel Diffirential Mobile Robots by Using Experimental Heading Errors, *IEEE International Conference on Robotics and Automation*, Saint Paul, Minesota, USA, **2012**
- [13] Martinelli A. , The Accurancy on the Parameter Estimation of an Odometry System of a Mobie Robot, *IEEE International Conference on Robotics & Automation*, **2002**
- [14] Chong K.S. , .Kleeman L. Acuurate Odometry and Error Modeling for a Mobile Robot, *IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, **1997**

- [15] Advanced Sonar and Odometry Error Modeling for Simultaneous Localisation and Map Building, *Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, **2003**
- [16] Lee K. Chung W. , Yoo K. Kinematic parameter calibration of a car-like mobile robot to improve odometry accuracy, *Mechatronics*, 20, 582-595, **2010**
- [17] Roumelitos S.I. , Sukhatme G.S. , Bekey G.A. ,Fault Detection and Identification in a Mobile Robot using Multiple-Model Estimation , *IEEE International Conference of Robotics and Automation*, Belgium, May **1998**
- [18] Hans-Joachim, von der Hardt, Rend Husson Didier Wolf, An Automatic Calibration Method for a Multisensor System: Application to a Mobile Robot Localization System, 1998 *IEEE International Conference on Robotics & Automation*, Leuven, Belgium May **1998**
- [19] Y. Maddahi, N. Sepehri., A. Maddahi and M. Abdolmohammadi, Calibration of wheeled mobile robots with differential drive mechanisms: an experimental approach, *Robotica*, **2012**
- [20] Sharifuddin Mondal, Youngmok Yun and Wan Kyun Chung, Calibration Technique for Odometry of Mobile Robots using Terminal Iterative Learning Control, *The 5th International Conference on Ubiquitous Robots and Ambient Intelligence*, **2008**
- [21] Haoming Xu and John James Collins, Estimating the Odometry Error of a Mobile Robot by Neural Networks, *Machine Learning and Applications*, MLA '09. International Conference, **2009**
- [22] Agostino Martinelli and Roland Siegwart , Observability Properties and Optimal Trajectories for On-line Odometry Self-Calibration , *Proceedings of the 45th IEEE Conference on Decision & Control*, Manchester Grand Hyatt Hotel, San Diego, CA, USA, December 13-15, **2006**
- [23] Mariolino De Cecco, Self-Calibration of AGV Inertial-Odometric Navigation Using Absolute-Reference Measurements, *IEEE Instrumentation and Measurement Technology Conference*, Anchorage, Ay USA, 21-23 May **2002**
- [24] Tao Wei, Yufei Huang, and Philip Chen, Particle Filtering for Adaptive Sensor Fault Detection and Identification Proceedings of the 2006 *IEEE International Conference on Robotics and Automation*, Orlando, Florida - May **2006**
- [25] <http://robotics.mech.kit.ac.jp/kimura/research/Quadruped/photo-movie-kotetsu-e.html>
- [26] S.G.Tzafestas, *Introduction to Mobile Robot Control*, Elsevier, **2014**
- [27] <http://www.rotacaster.com.au/shop/product/item11512/rotacaster-125mm-single--85a-pu-grey-grey.html?process=shop%2FproductView.php>

- [28] http://www-hh.mech.eng.osaka-u.ac.jp/robotics/Omni-Ball_e.html
- [29] http://nodna.de/3WD-100mm-Omni-Wheel-Mini-Mobile-Robot_1
- [30] http://en.wikipedia.org/wiki/Mecanum_wheel#mediaviewer/File:UranusOmniDirectionalRobotPodnar.png
- [31] <http://www.robotistan.com/12V-37mm-150-Rpm-671-Reduktorlu-DC-Motor-671-Metal-Gearmotor-37Dx54L-mm,PR-1219.html>
- [32] <http://arduino.cc/en/Main/arduinoBoardMega2560>
- [33] <http://arduino.cc/en/Main/ArduinoMotorShieldR3>
- [34] <http://www.roboweb.net/rw-ml-1445.html>
- [35] http://www.robotistan.com/XBee-Moduller,LA_209-2.html
- [36] <http://www.robishop.com/90x10mm-Teker-Cifti-Beyaz-Wheel-90x10mm-Pair-White,PR-1974.html>
- [37] <http://www.robishop.com/Ball-Caster-with-1-Plastic-Ball-Sarhos-Teker-254mm,PR-1732.html>
- [38] <http://thedenneys.org/pub/robot/encoders/>
- [39] http://www.inverter-plc.net/enkoder/absolut_enkoder.html
- [40] <http://www.microsoft.com/hardware/en-us/d/lifecam-vx-800>
- [41] Yaakov Bar-Shalom, X. Rong Li, Thiagalingam Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*, John Wiley & Sons, **2001**

EKLER

EK 1 : MİKROKONTROLCÜ KODU

```
#if ARDUINO >= 100
#include "Arduino.h"
#else
#include "WConstants.h"
#endif

void __Super_Motor_Data() ;
void __Read_MotorSerial() ;

void __Super_Read_Encoder_Data() ;
void __Read_Right_Encoder_Data() ;
void __Read_Left_Encoder_Data() ;

void __Super_Send_Encoder_Data() ;
void __Send_Right_Encoder_Data() ;
void __Send_Left_Encoder_Data() ;

void __Super_Arduino_Computer() ;
void __WriteStartCommand() ;
void __Read_StartCommand() ;

void __Super_DriveMotor() ;
void __Drive_LeftMotor() ;
void __Drive_RightMotor() ;

void __Setup_Serial3_Interface() ;
void __Setup_Motor_Interface() ;
void __Setup_EncoderInterface() ;

void __doEncoderRight_ISR() ;
void __doEncoderLeft_ISR() ;

unsigned long SerialMillis =0 ;
float dTs =0 ; // örnekleme zamanı ilk değeri tanımlı
```

```

int    inchar                ; // karakterin tanımı
int    inData[4]             ; // vektörün boyut tanımı
int    inSTART[4]={1,2,3,4} ; // Başlama vektörü tanımı
boolean inSTARTOK =0        ; //Başlangıç hazır ilk komut değeri
ataması

const int NUMBER_OF_FIELDS  = 6 ; // Virgülle ayrılmış kaç sayı gelecek
int    fieldIndex          = 0 ; // İlk atanan alan endeksi
int    MotorCommand[NUMBER_OF_FIELDS] ; // Motor komutunun sıra vektörü
int    speedpinA  = 3       ; //A motoru hız pini tanımı
int    speedpinB  = 11      ; //B motoru hız pini tanımı
int    pinI1     = 12       ; //A motoru yön pini tanımı
int    pinI2     = 9        ; //A motoru fren pini tanımı
int    pinI3     = 13       ; // B motoru yön pini tanımı
int    pinI4     = 8        ; // B motoru fren pini tanımı

boolean I1                ; // I1 vektör elemanının sayı
karakteristiği
boolean I2                ; // I2 vektör elemanının sayı
karakteristiği
boolean I3                ; // I3 vektör elemanının sayı
karakteristiği
boolean I4                ; // I4 vektör elemanının sayı
karakteristiği
int    SpeedLeft          ; //SpeedLeft vektör elemanının sayı
karakteristiği
int    SpeedRight         ; //SpeedRight vektör elemanının sayı
karakteristiği

long    SerTimeOut=5      ; // Seri protokolün zaman aşımı 5 milisaniye

// Enkoder pin tanımları Kesme 5 - pin 18, Kesme 4 - pin 19
const int LeftEncoderA  =18 ; // Sol enkoder a fazı pin 18 de tanımlı
const int LeftEncoderB  =36 ; // Sol enkoder B fazı pin 36 da tanımlı
const int RightEncoderA =19 ; // Sağ enkoder A fazı pin 19 da tanımlı
const int RightEncoderB =38 ; // Sol enkoder B fazı pin 38 de tanımlı

signed int RightEncoderRead=0 ; // Sağ enkoder ilk okuma değeri

```

```

signed int LeftEncoderRead =0 ; // Sol enkoder ilk okuma değeri

int right_pwm =0 ; // Sağ pwm ilk değeri tanımı
int left_pwm =0 ; // Sol pwm ilk değeri tanımı
char Static_MotorPulse =50 ; // Durağan motor pwm değeri

float left_wheel_Diameter =90.00 ;
float right_wheel_Diameter =90.00 ;
int K_esigDrev =2350 ;
float SecK1 =1000.0 ; // Örnekleme zamanı bölüm katsayısı
tanımı
//*****
*****
//MAIN LOOP
void setup()
{
__Setup_Serial3_Interface() ;// 3. Seri port iletişim kurulumu
__Super_Arduino_Computer() ;//
__Setup_Motor_Interface() ;// Motor arayüzü kurulumu
__Setup_EncoderInterface() ;// Enkoder arayüzü kurulumu
__Super_Send_Encoder_Data() ;// Enkoder veri paket gönderimi kurulumu
SerialMillis=millis() ;// Seri iletişim zamanının çalıştırılması
}

void loop()
{
int i ; //i değişkeninin tanımı
__Super_Motor_Data() ;// Seri porttan motor verisini okunması
__Super_DriveMotor() ;// Platform motorlarının sürülmesi
__Super_Read_Encoder_Data() ;// Motordaki enkoder verilerinin okunması
dT=millis()-SerialMillis ;// Örnekleme zamanı çıkarılması
dT=dT/SecK1 ;// Örnekleme zamanı saniyeye çevirimi
SerialMillis=millis() ;// Seri iletişim zamanına son değer atanması
__Super_Send_Encoder_Data() ;// Enkoder verileri seri porta gönderilmesi
RightEncoderRead = 0 ;// Sağ enkoder verisinin sıfırlanması
LeftEncoderRead = 0 ;// Sol enkoder verisinin sıfırlanması
}

```



```

//*****
*****

//MAIN LOOP

//*****
*****

//SETUP FUNCTIONS // Kurulum Fonksiyonları

void __Setup_Serial3_Interface()
{
  Serial3.begin(57600)           ;// iletişim hızı ayarlanması
  Serial3.setTimeout(SerTimeOut) ;// Gecikme zamanının atanması
  Serial3.flush()               ;// Seri portun temizlenmesi
}

void __Setup_Motor_Interface()
{
  SpeedLeft =0                  ;// Sol motor hızı pwm değeri ataması
  SpeedRight =0                 ;// Sağ motor hızı pwm değeri ataması
  I1=0                          ;// Sol motor yön değeri ataması
  I2=0                          ;// Sol motor fren değeri ataması
  I3=0                          ;// Sağ motor yön değeri ataması
  I4=0                          ;// Sağ motor fren değeri ataması
  __Drive_RightMotor()          ;// Sağ motor sürülmesi
  __Drive_Left_Motor()         ;// Sol motor sürülmesi
}

void __Setup_EncoderInterface()
{
  pinMode(LeftEncoderA ,INPUT)   ;// Sol enkoder A fazının girdi olarak tanımı
  pinMode(LeftEncoderB ,INPUT)   ;// Sol enkoder B fazının girdi olarak tanımı
  pinMode(RightEncoderA,INPUT)   ;//Sağ enkoder A fazının girdi olarak tanımı
  pinMode(RightEncoderB,INPUT)   ;//Sağ enkoder B fazının girdi olarak tanımı

  attachInterrupt(5, __doEncoderLeft__ISR, CHANGE) ;// encoder pin on interrupt 5 (pin 18)
  attachInterrupt(4, __doEncoderRight_ISR, CHANGE) ;// encoder pin on interrupt 4 (pin 19)
}

```

```

//*****
*****

//SETUP FUNCTIONS // Kurulum fonksiyonları bu bölümde toplanmıştır

//*****
*****

//STARTUP COMMUNICATION FUNCTIONS // İletişim başlama fonksiyonu
void __Super_Arduino_Computer()
{
  while(!inSTARTOK) {
    __ReadStartCommand()           // Başlama komutu gelinceye kadar seri portu
okuma
  }

  __WriteStartCommand()           // Arduino başlama komutu yazımı

  inSTARTOK=0                     // Başlama komutunun tekrar okunması için
sıfırlanması

  while(!inSTARTOK) {
    __ReadStartCommand()           // Başlama komutu gelinceye kadar seri portu
okuma
  }
}

void __ReadStartCommand()
{
  while(Serial3.available()<4)
  {
    // Seri portta dörtten az veri varsa boşa çalışır
  }

  for(fieldIndex = 0; fieldIndex < 4; fieldIndex ++)
  {
    inchar=Serial3.parseInt()      // Seri porta yazılı veriyi karakter değişkene
yazılması
  }
}

```

```

    inData[fieldIndex]=inchar                ;// Seri porta gelen verilerle vektör oluşturur
}

inSTARTOK=1                                ;// Veri akışını başlatır
for(fieldIndex = 0; fieldIndex < 4; fieldIndex ++)
{
    if(inData[fieldIndex]!=inSTART[fieldIndex]) // Eğer gelen veriyle başlangıç vektörü aynı
değilse
    {

        inSTARTOK=0                          ;// Veri akışı başlamaz
        Serial3.print(inSTARTOK)              ;// Seri porta başlama değeri yazılır
        Serial3.flush()                       ;// Seri port temizlenir
        break                                  ;// Şart bozulur.
    }
}

void __WriteStartCommand()
{
    Serial3.print(4)                           ;// Seri porta 4 yazılır
    Serial3.print(3)                           ;// Seri porta 3 yazılır
    Serial3.print(2)                           ;//Seri porta 2 yazılır
    Serial3.println(1)                         ;//Seri porta 1 yazılır ve satır sonlandırılır
}

//*****
//STARTUP COMMUNICATION FUNCTIONS           // Başlangıç Komut Fonksiyonları

//*****

//MOTOR FUNCTIONS                          // Motor fonksiyonları
void __Super_Motor_Data()
{
    __Read_MotorSerial()                       ;// Seri porttaki motor verisini okur
    __Encode_Motor_Signal()                   ;// Motor sinyallerini çözümler
}

```

```

}

void __Read_MotorSerial()
{
  Serial3.flush()                ;// Seri portu temizler
  while(Serial3.available()<6)
  {
                                // Seri portta altı veriden az veri varsa boşta çalışır
  }

  for(fieldIndex = 0; fieldIndex < 6; fieldIndex ++)
  {
    MotorCommand[fieldIndex] = Serial3.parseInt(); // Seri porttaki verileri parseller
  }
  I1=MotorCommand[0]            ;// Sol motor yön değerini I1'e atar
  I2=MotorCommand[1]            ;// Sol motor fren değerini I2'e atar
  I3=MotorCommand[2]            ;// Sağ motor yön değerini I3'e atar
  I4=MotorCommand[3]            ;// Sağ motor fren değerini I4'e atar
  SpeedLeft =MotorCommand[4]     ;// Sol motor hız değerini atar
  SpeedRight=MotorCommand[5]     ;// Sağ motor hız değerini atar
}

void __Encode_Motor_Signal()
{
  left_pwm=SpeedLeft;           ;//Sol motor hızı değerini motora atar
  right_pwm=SpeedRight          ;//Sağ motor hızı değerini motora atar
}

//*****
//*****
//MOTOR FUNCTIONS                // Motor Fonksiyonları

//*****
//*****
//MOTOR DRIVE FUNCTIONS
void __Super_DriveMotor()
{
  __Drive_RightMotor()          ;// Sağ motor sürülür
}

```

```

__Drive_Left_Motor()           ;//Sol motor sürülür
}

void __Drive_Left_Motor()
{
    analogWrite(speedpinB,left_pwm)           ;//Sol motor hız değeri hız pinine yazılır
    digitalWrite(pinI1,I1)                   ;//Sol motor yön değerini yön pinine atar
    digitalWrite(pinI2,I2)                   ;// Sol motor fren değerini fren pinine atar
}

void __Drive_RightMotor()
{
    analogWrite(speedpinA,right_pwm)          ;// Sağ motor hız değeri hız pinine yazılır

    digitalWrite(pinI3,I3)                   ;// Sağ motor yön değerini yön pinine atar
    digitalWrite(pinI4,I4)                   ; //Sağ motor fren değerini fren pinine atar

}

//*****
//MOTOR DRIVE FUNCTIONS                // Motor Sürüş Fonksiyonları

//*****

//ENCODER FUNCTIONS
void __Super_Send_Encoder_Data()
{
    __Send_Right_EncoderData()              ;// Sağ enkoder datasının gönderilmesi
    __Send_Left__EncoderData()              ;// Sol enkoder datasının gönderilmesi
}

void __Send_Right_EncoderData()
{
    Serial3.flush()                          ;//Seri portun temizlenmesi
    Serial3.print("HEADER")                  ;// Veri paketi başlığı yazımı
    Serial3.print(",")                       ;// Virgül yazımı
}

```

```

Serial3.print("R-Encoder Output")           ;//Sağ enkoder çıktısı karakter yazımı
Serial3.print(",")                           ;// Virgül yazımı
Serial3.print(dTs)                           ;// Örnekleme zamanı
Serial3.print(",")                           ;// Virgül yazımı
Serial3.print(RightEncoderRead)              ;// Sağ enkoder değerinin yazımı
Serial3.print(",")                           ;// Virgül yazımı
Serial3.print(RightEncoderRead)              ;// Sağ enkoder değerinin yazımı
Serial3.print(",")                           ;// Virgül yazımı
Serial3.print("TR-END")                      ;// İletinin bitme karakteri yazımı
}

void __Send_Left__EncoderData()
{

Serial3.print(",")                           ; // Virgül yazımı
Serial3.print("HEADER")                      ;// Veri paketi başlığı yazımı
Serial3.print(",")                           ; // Virgül yazımı
Serial3.print("L-Encoder Output")           ;// Sol enkoder çıktısı karakter yazımı
Serial3.print(",")                           ; // Virgül yazımı
Serial3.print(dTs)                           ;// Örnekleme zamanı
Serial3.print(",")                           ; // Virgül yazımı
Serial3.print(LeftEncoderRead)               ;// Sol enkoder değerinin yazımı
Serial3.print(",")                           ; // Virgül yazımı
Serial3.print(LeftEncoderRead)               ;// Sol enkoder değerinin yazımı
Serial3.print(",")                           ; // Virgül yazımı
Serial3.println("TR-END")                    ;// İletinin bitme karakteri yazımı ve satır sonu
}

void __Super_Read_Encoder_Data()
{
__Read_Right_Encoder_Data()                  ;// Sağ enkoder verisinin okunması
__Read_Left__Encoder_Data()                  ;//Sol enkoder verisinin okunması
}

void __Read_Right_Encoder_Data()
{
RightEncoderRead=RightEncoderRead           ;// Sağ enkoder verisinin atanması
}

```

```

}

void __Read_Left_Encoder_Data()
{
    LeftEncoderRead=LeftEncoderRead           ;// Sol enkoder verisinin atanması
}

//*****
*****

//ENCODER FUNCTIONS                               // Enkoder Fonksiyonları

//*****
*****

//INTERRUPT FUNCTIONS
void __doEncoderLeft__ISR()
{
    if (digitalRead(LeftEncoderA) == HIGH){           //Eğer A fazında yüksek atımlar varsa
        if (digitalRead(LeftEncoderB) == HIGH) {     // ve B fazında da yüksek atımlar varsa
                                                    // Enkoder saat yönünün tersine döner

            LeftEncoderRead=LeftEncoderRead+1       ; // Enkoder değeri bir arttırılır
        }
        else{                                         // Eğer B fazı atımları düşükse
            LeftEncoderRead=LeftEncoderRead-1       ; // Enkoder değeri bir azaltılır
        }
    }
    else                                             // Eğer A fazında düşük atımlar varsa
    {
        if (digitalRead(LeftEncoderB) == LOW) {     // B fazında da düşük atımlar varsa
                                                    // Enkoder saat yönünün tersine döner

            LeftEncoderRead=LeftEncoderRead+1       ; // Enkoder değeri bir arttırılır
        }
        else                                         // Eğer B fazı atımları yüksekse
        {
            LeftEncoderRead=LeftEncoderRead-1       ; // Enkoder değeri bir azaltılır
        }
    }
}

```

```

}

void __doEncoderRight_ISR()
{
    if (digitalRead(RightEncoderA) == HIGH){           //Eğer A fazında yüksek atımlar varsa
        if (digitalRead(RightEncoderB)== LOW) {       // ve B fazında da düşük atımlar varsa
                                                    // Enkoder saat yönünün tersine döner

            RightEncoderRead=RightEncoderRead+1      ;// Enkoder değeri bir arttırılır
        }
        else{                                         // Eğer B fazı atımları yüksekse
            RightEncoderRead=RightEncoderRead-1      ;// Enkoder değeri bir azaltılır
        }
    }
    else                                             // Eğer A fazında düşük atımlar varsa
    {
        if (digitalRead(RightEncoderB) == HIGH) {    // B fazında da yüksek atımlar varsa
                                                    // Enkoder saat yönünün tersine döner

            RightEncoderRead=RightEncoderRead+1      ;// Enkoder değeri bir arttırılır
        }
        Else                                         // Eğer B fazı atımları düşükse
        {
            RightEncoderRead=RightEncoderRead-1      ;// Enkoder değeri bir azaltılır
        }
    }
}

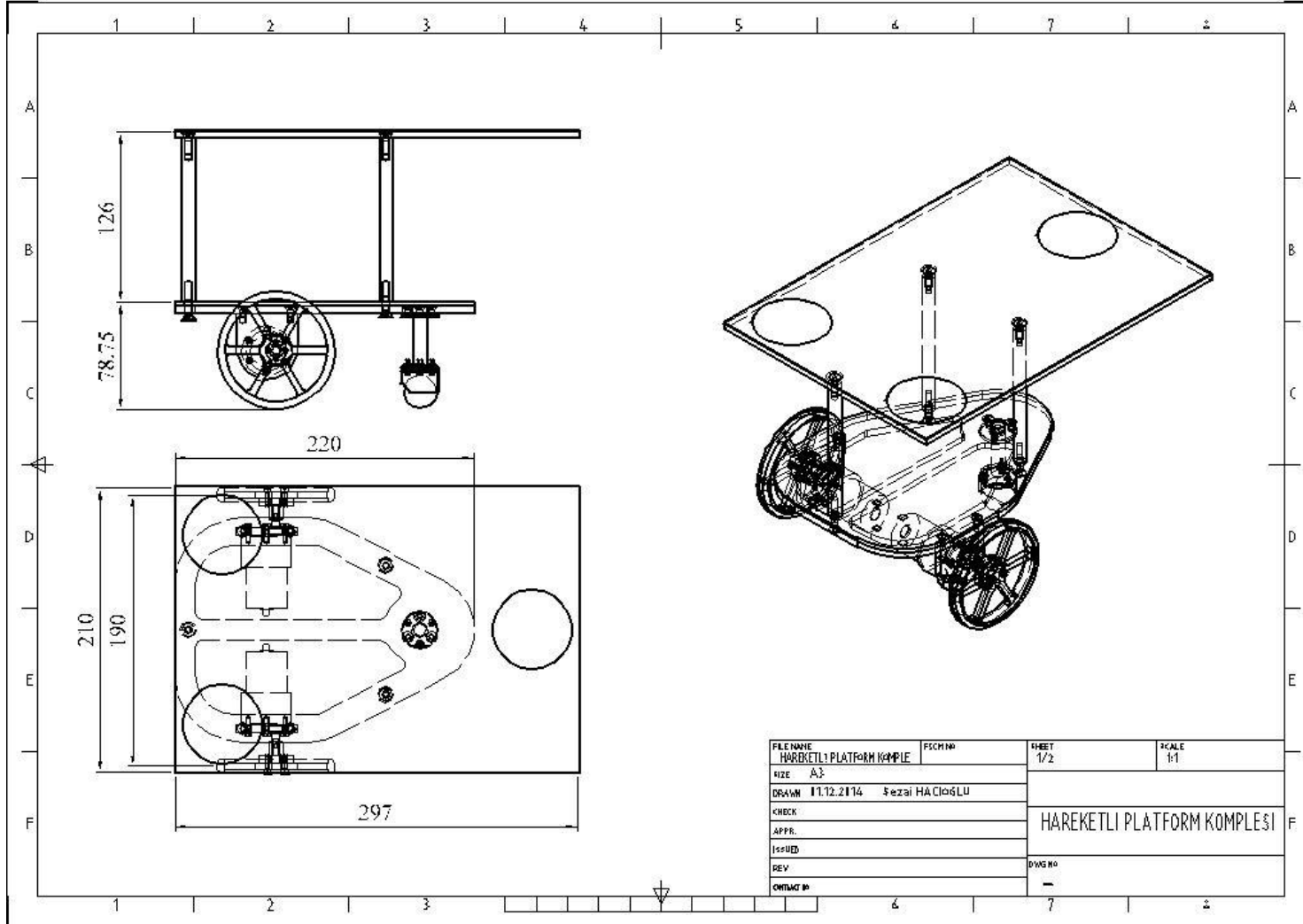
//*****

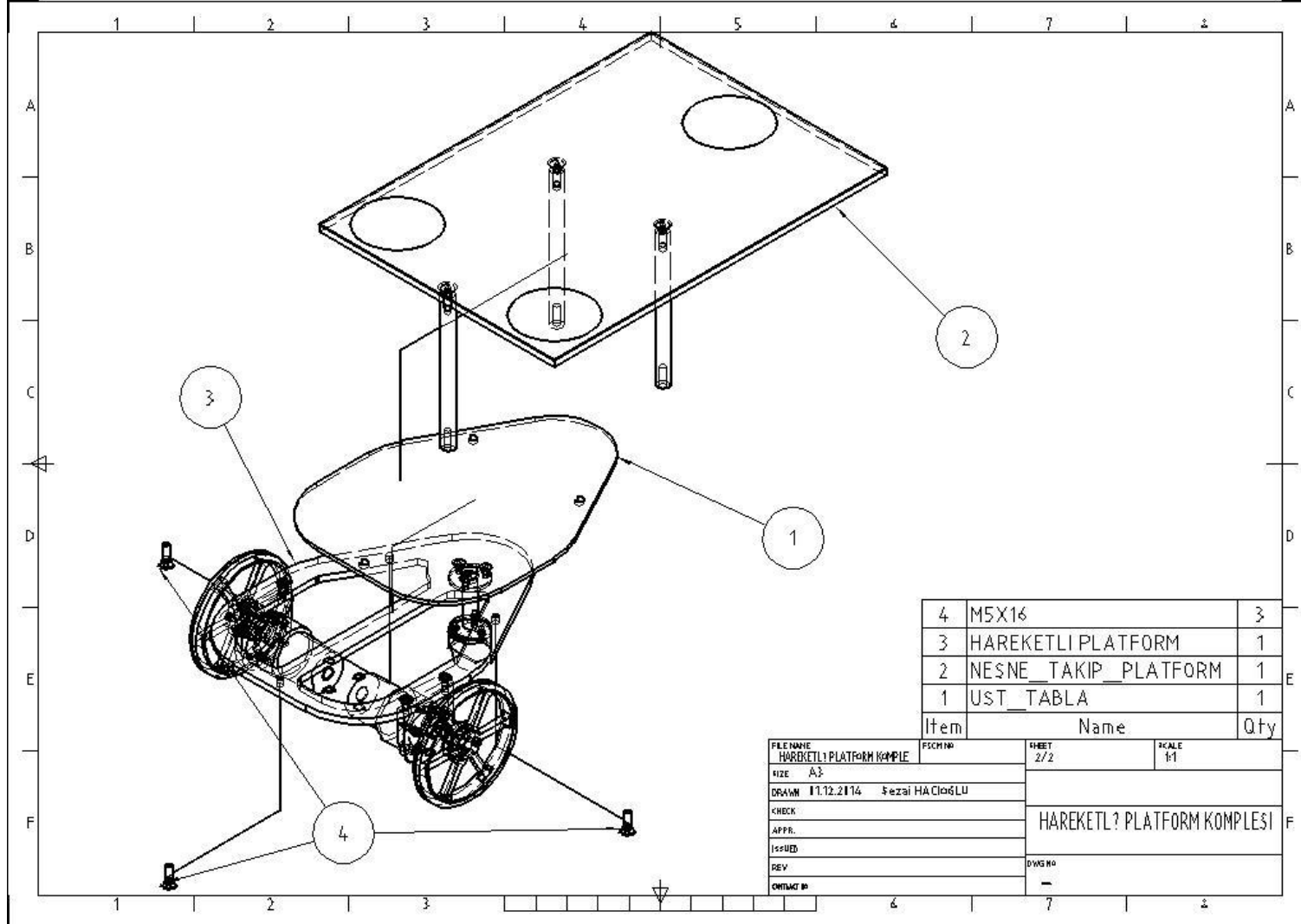
```


EK: 2 TEST 1 ODOMETRE VERİSİ

Veri sayısı	Zaman aralığı (s)	Sol enkoder değeri (atım)	Sağ enkoder değeri (atım)
1	0.0000	0	0
2	0.2100	0	0
3	0.0900	66	195
4	0.1300	233	565
5	0.1000	257	495
6	0.0900	263	454
7	0.1100	341	560
8	0.1000	331	524
9	0.0900	318	486
10	0.1100	391	584
11	0.1000	342	501
12	0.0900	327	475
13	0.0900	341	492
14	0.1000	369	531
15	0.0800	287	407
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.
121	0.1300	502	709
122	0.2000	762	1071
123	0.1400	528	741
124	0.2100	797	1122
125	0.2400	933	1313
126	0.1400	545	762
127	0.2500	953	1346
128	0.1400	531	745
129	0.2100	824	1162
130	0.2800	1090	1545
131	0.1400	544	768
132	0.2500	978	1387
133	0.2600	995	1408
134	0.1500	577	816
135	0.2600	1011	1438
-1	-1.0000	-1	-1

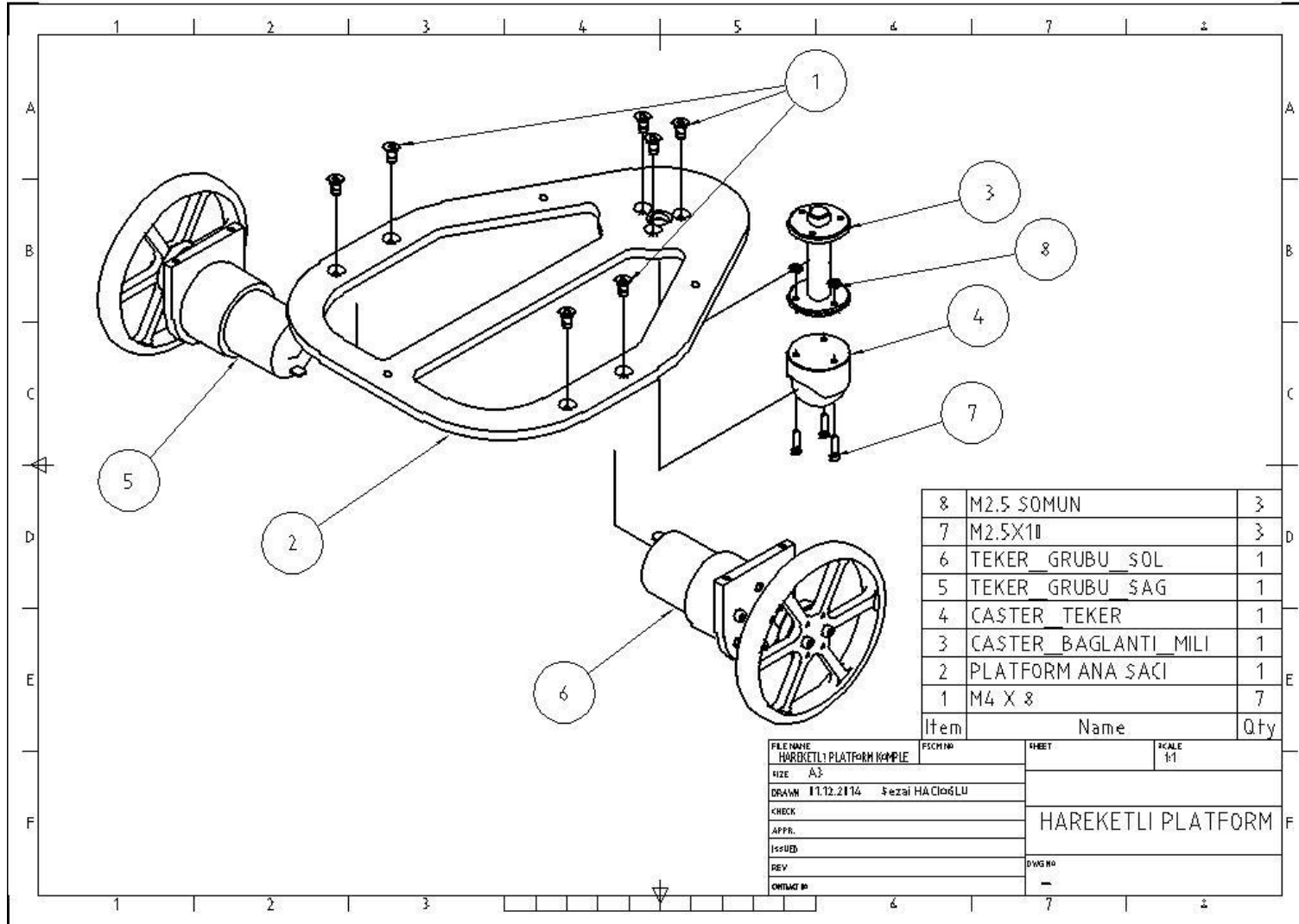
EK: 3 HAREKETLİ PLATFORM TEKNİK RESİMLERİ

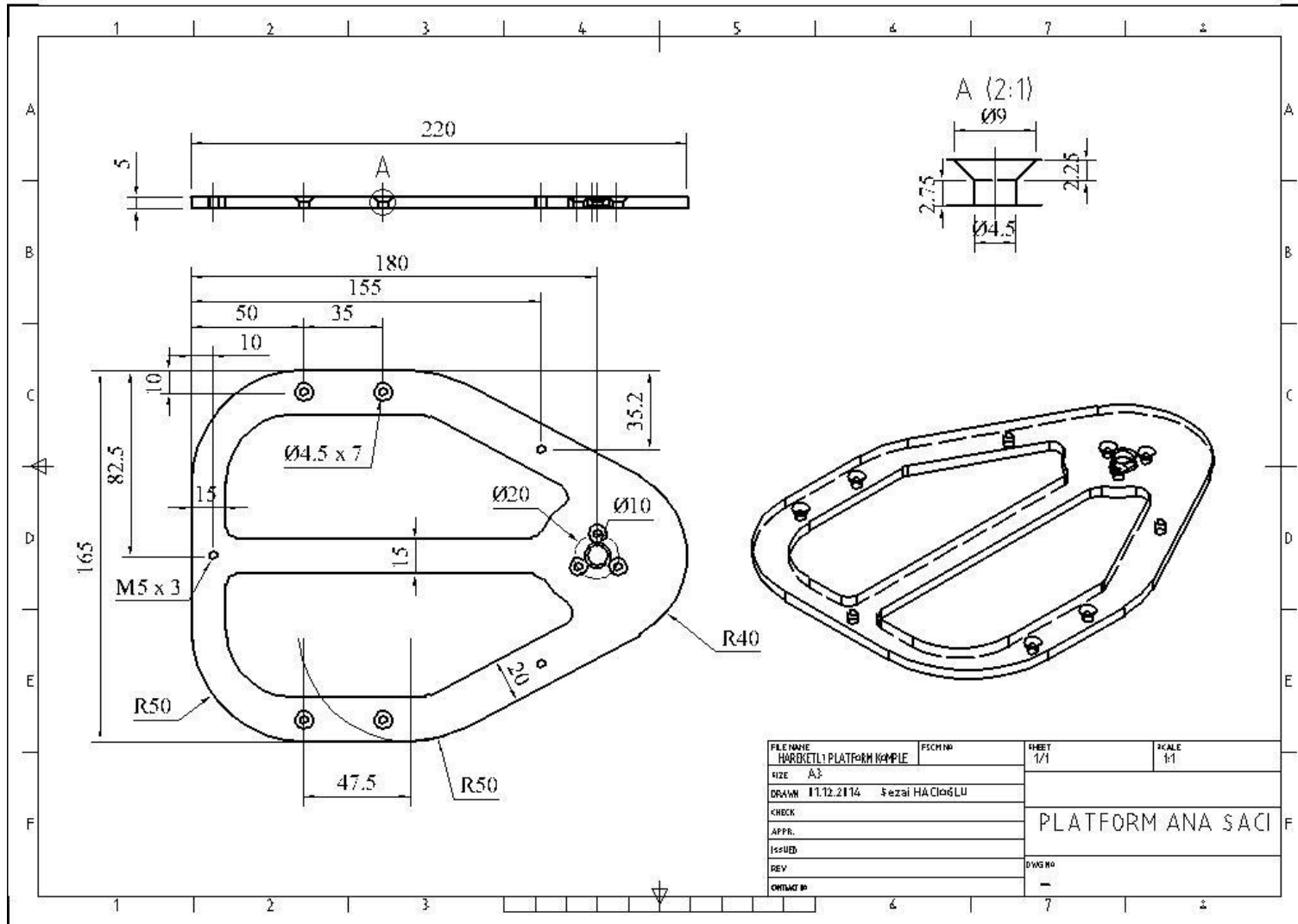


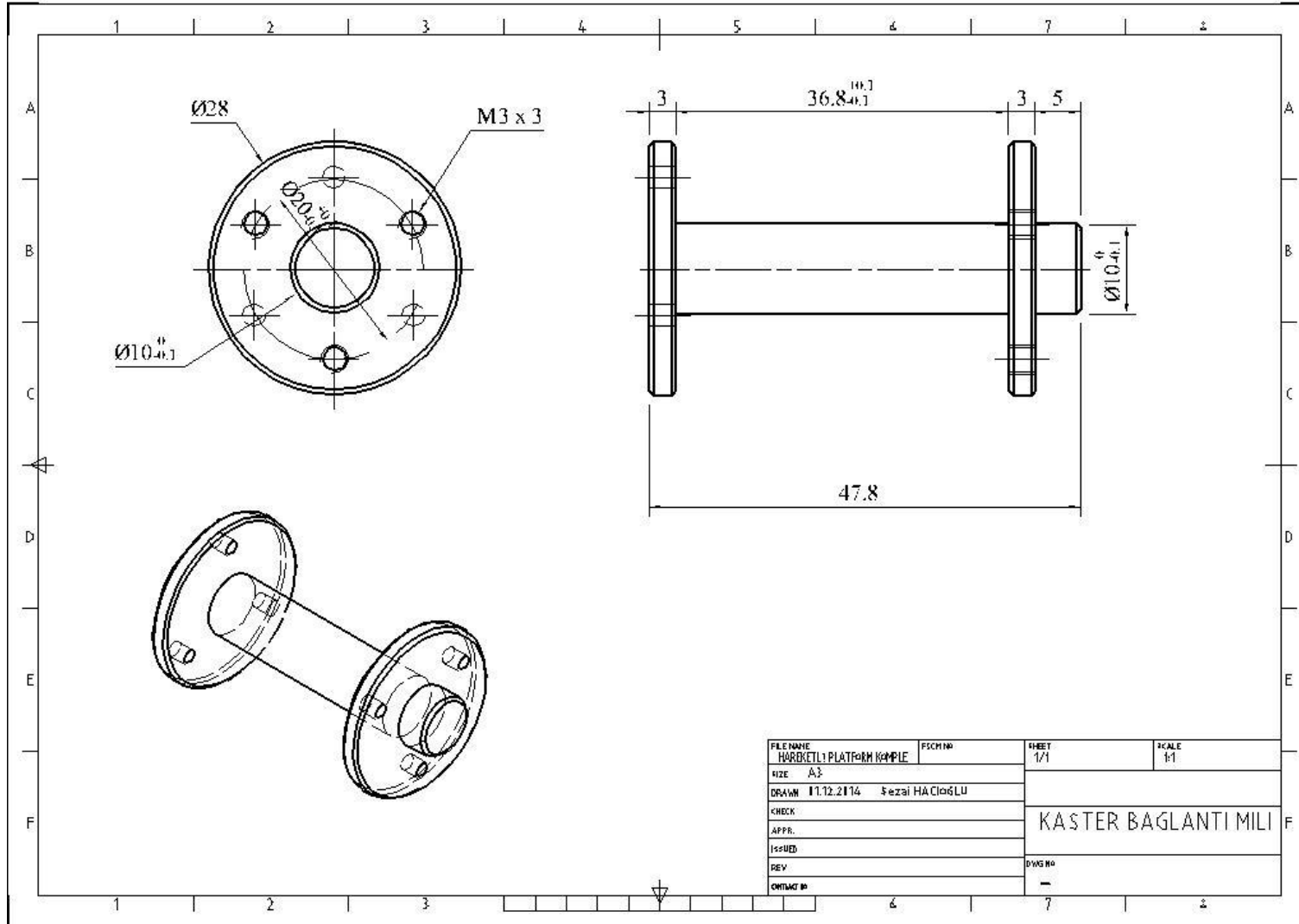


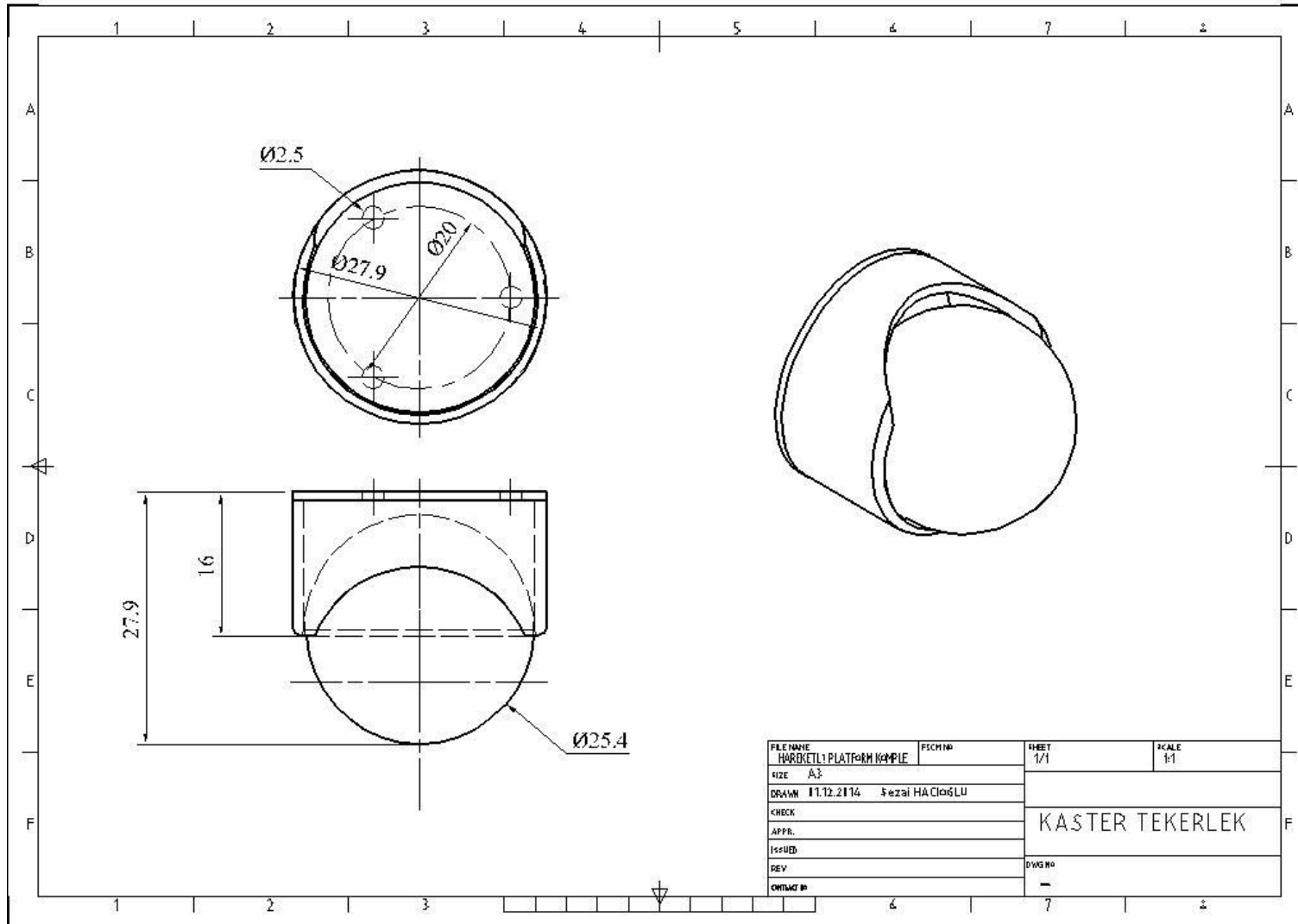
4	M5X16	3
3	HAREKETLI PLATFORM	1
2	NESNE TAKIP PLATFORM	1
1	UST TABLA	1
Item	Name	Qty

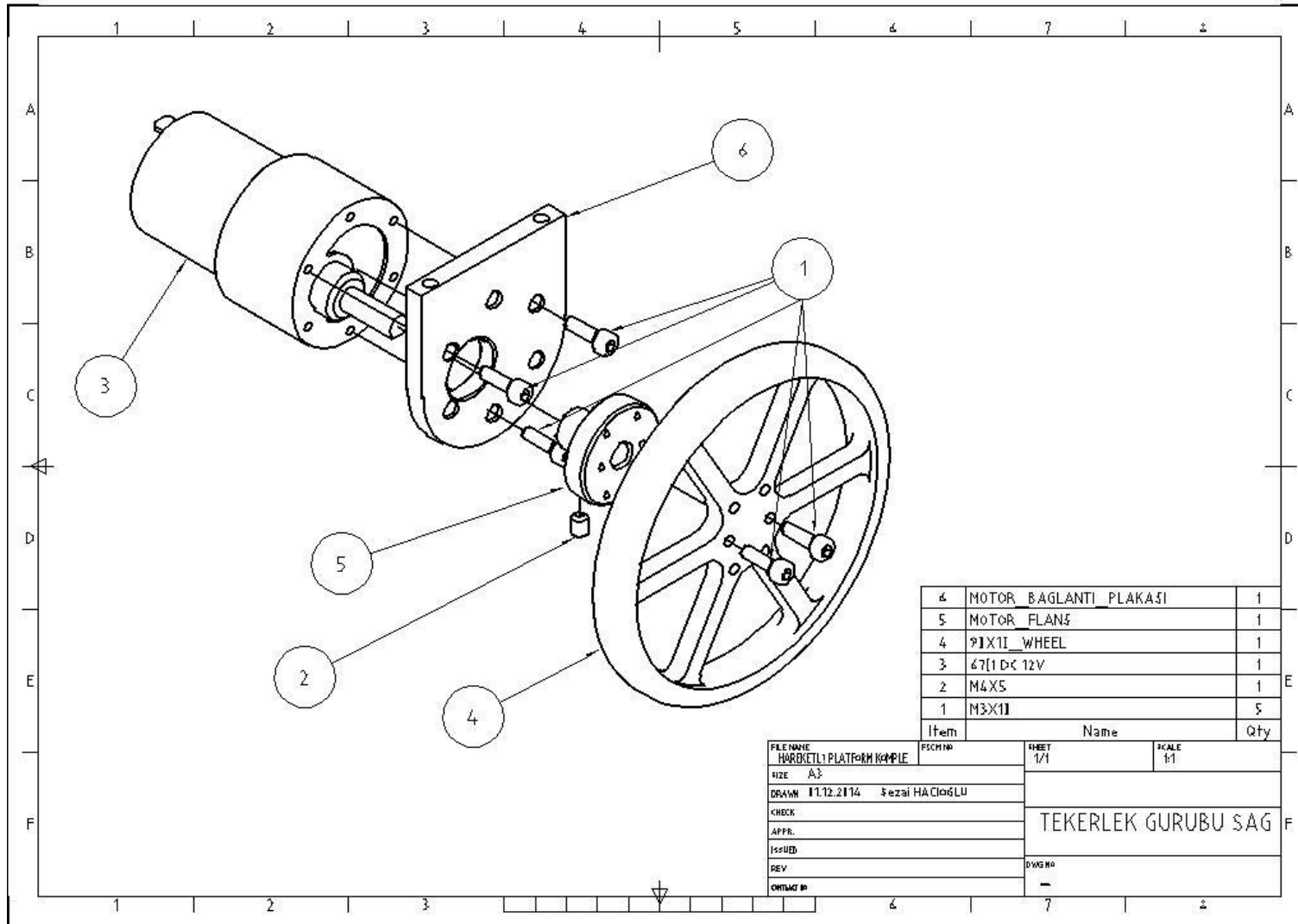
FILE NAME	HAREKETLI PLATFORM KOMPLE	SHEET	2/2	SCALE	1:1
SIZE	A3				
DRAWN	11.12.2014 İzzet HACIOĞLU				
CHECK		HAREKETLİ PLATFORM KOMPLESİ			
APPR.					
ISSUED					
REV					
CONTING #		DWG NO	-		





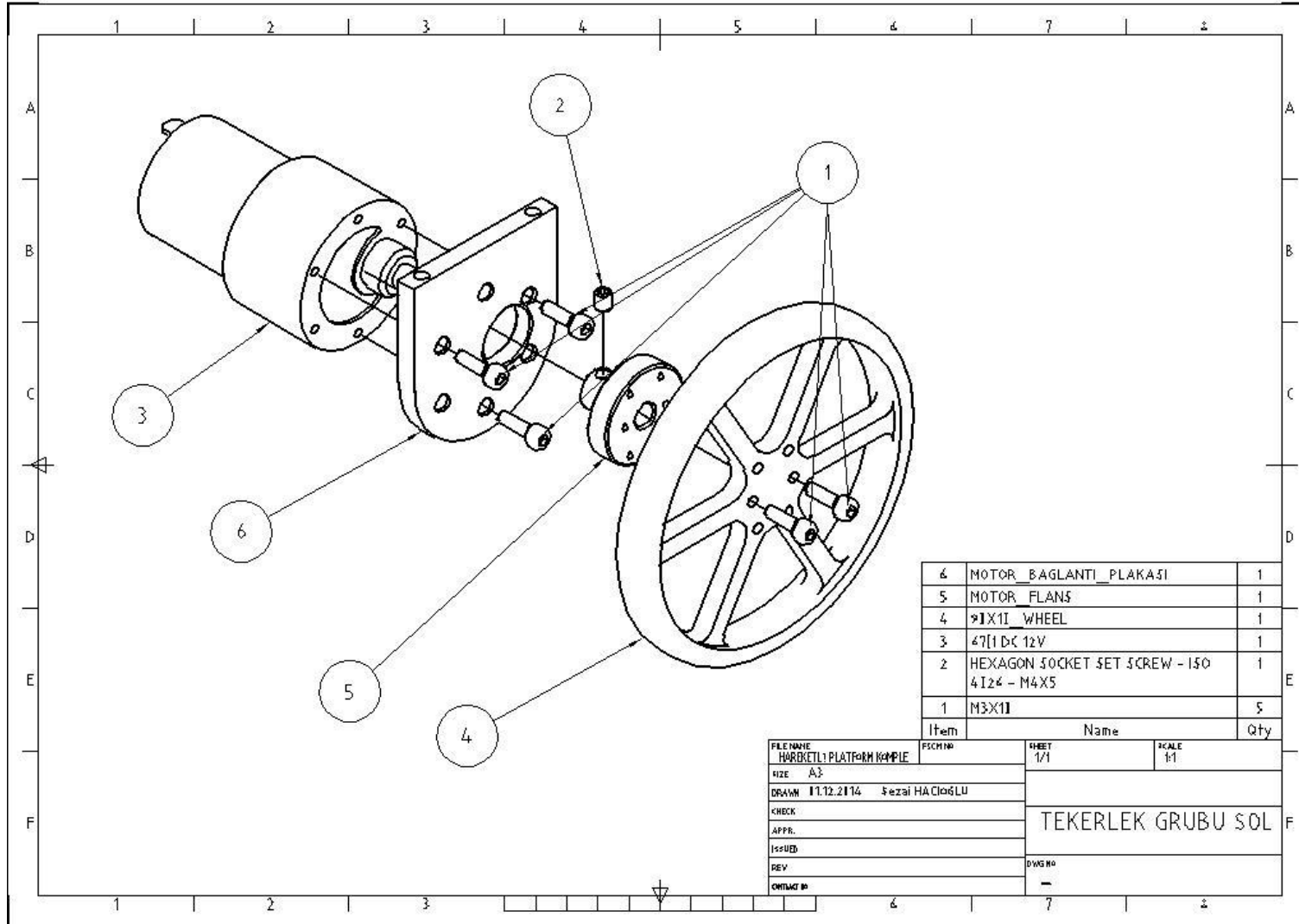






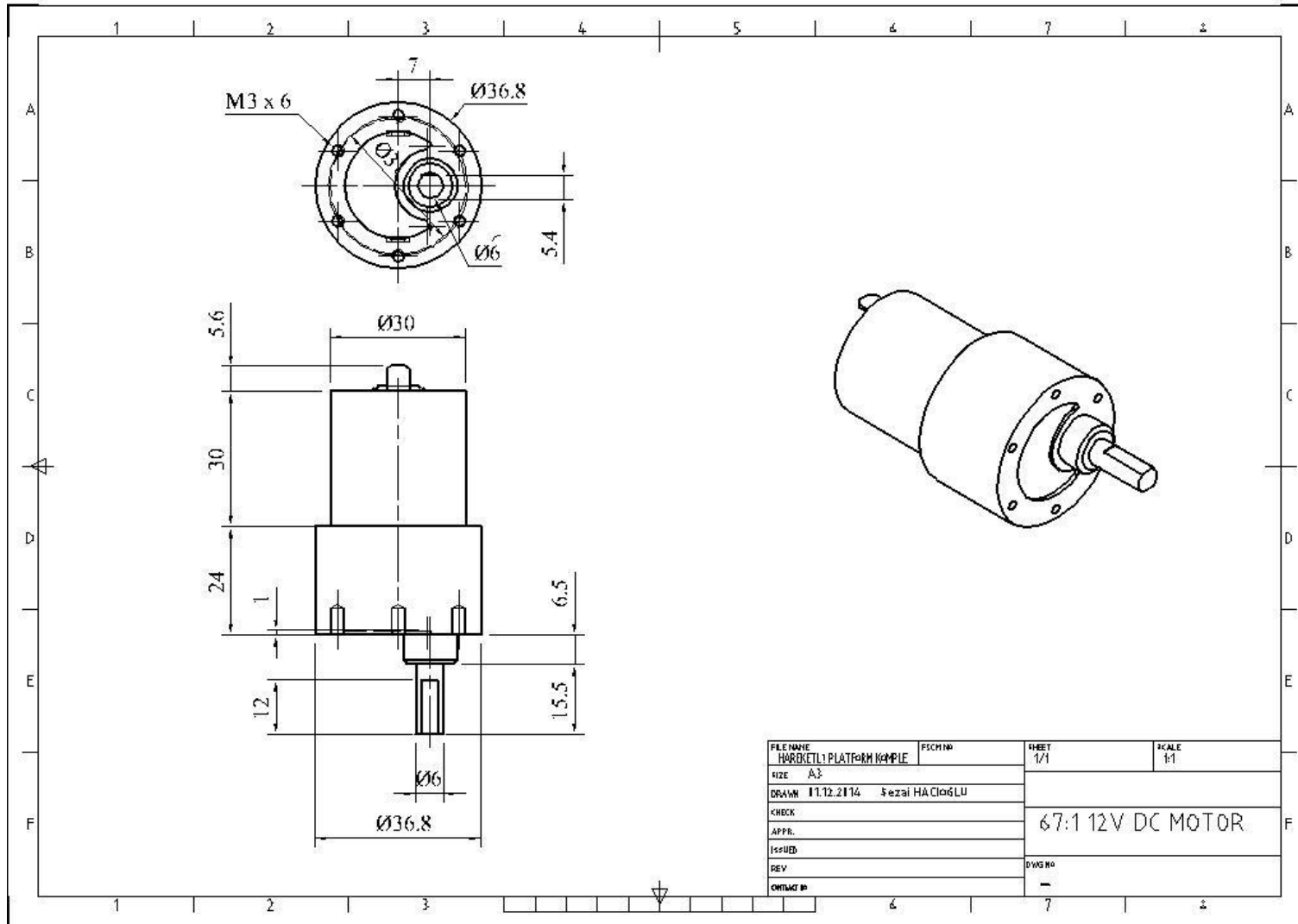
4	MOTOR_BAGLANTI_PLAKASI	1
5	MOTOR_FLANS	1
4	91X11_WHEEL	1
3	67(1.D< 12V	1
2	M4X5	1
1	M3X11	5
Item	Name	Qty

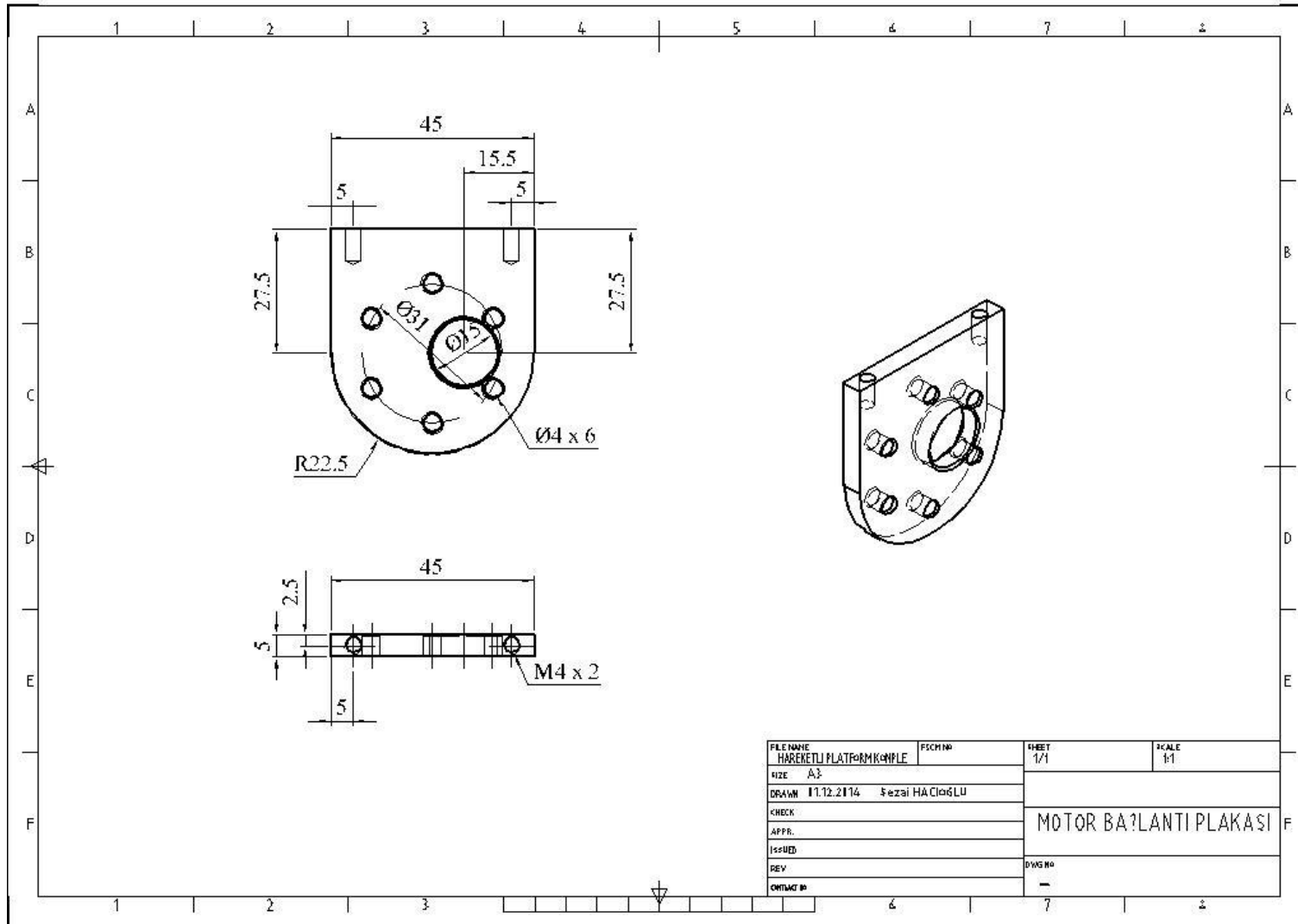
FILENAME	HARBETLİ PLATFORM KOMPLE	FIG. NO	SHEET	SCALE
SIZE	A3		1/1	1:1
DRAWN	11.12.2014 Fezal HACIÖĞLU	TEKERLEK GURUBU SAĞ		
CHECK				
APPR.				
ISSUED				
REV		DWG NO		
CONTAG #				

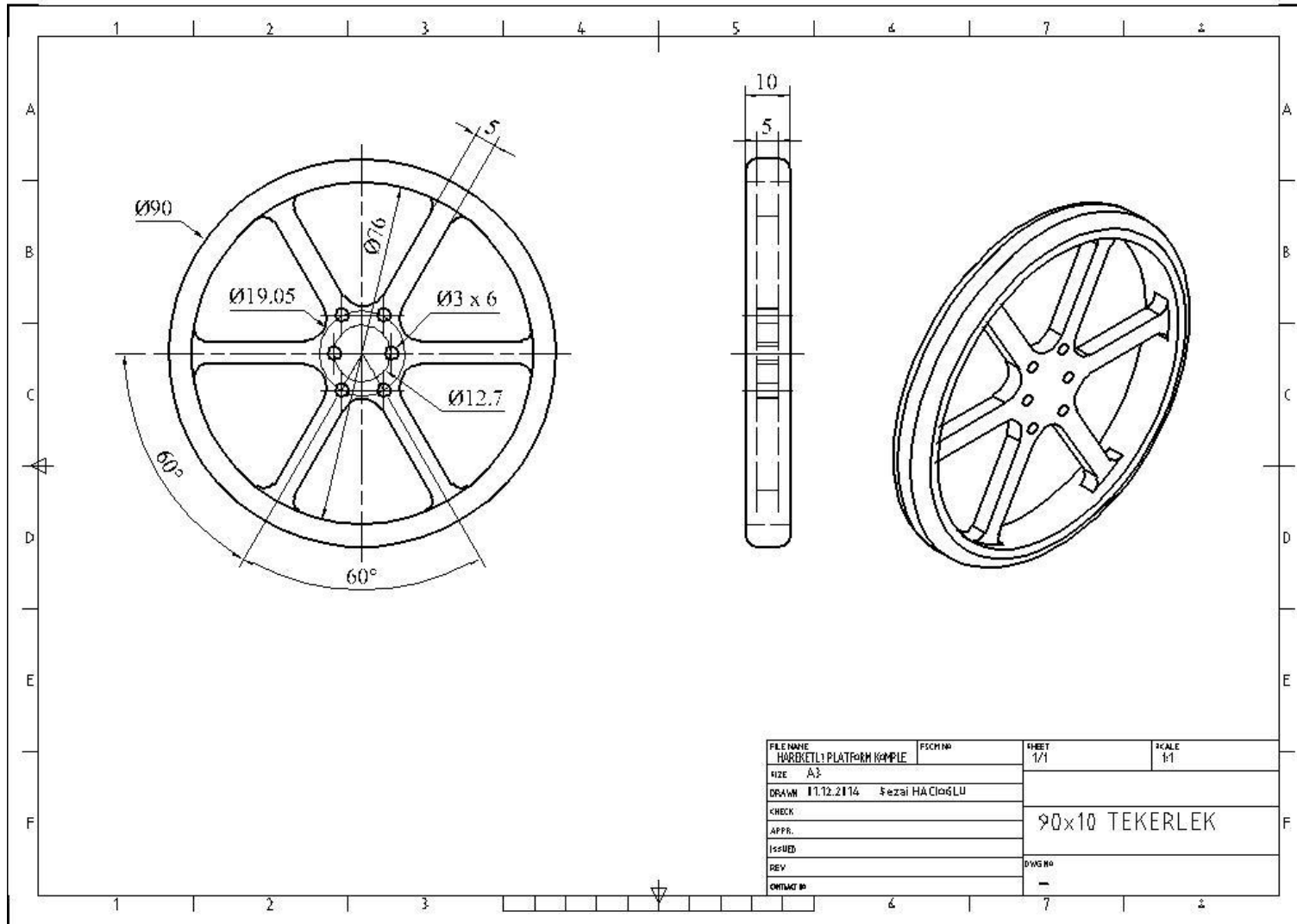


Item	Name	Qty
6	MOTOR_BAGLANTI_PLAKASI	1
5	MOTOR_FLANS	1
4	91X11 WHEEL	1
3	4711 DC 12V	1
2	HEXAGON SOCKET SET SCREW - ISO 4124 - M4X5	1
1	M3X11	5

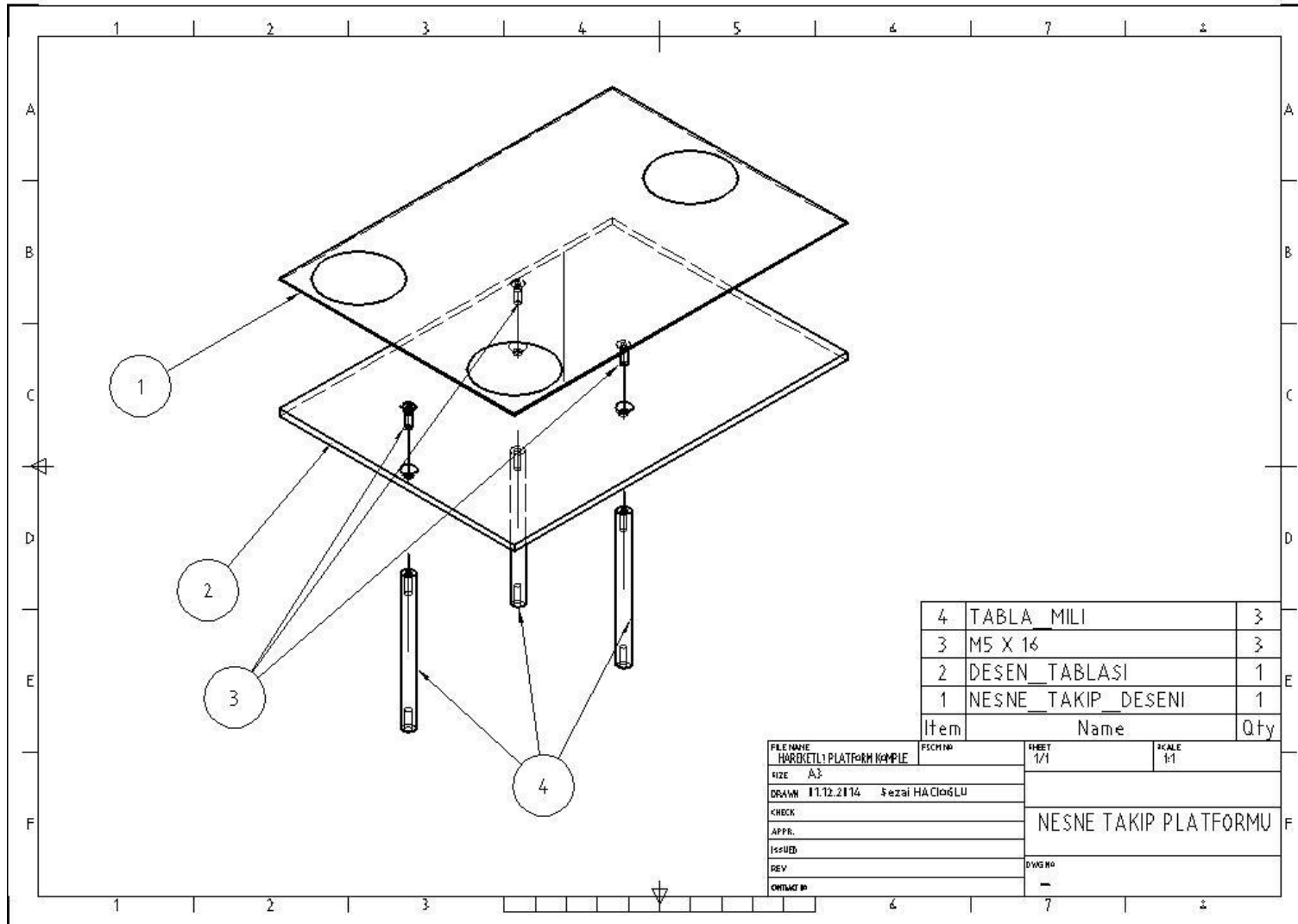
FILENAME HARBETLİ PLATFORM KOMPLE	PSCM NO	SHEET 1/1	SCALE 1:1
SIZE A3			
DRAWN 11.12.2114 Fezal HACIÖĞLU			
CHECK			
APPR.			
ISSUED			
REV			
CONTAG NO			
	TEKERLEK GRUBU SOL		
	DWG NO		
	-		





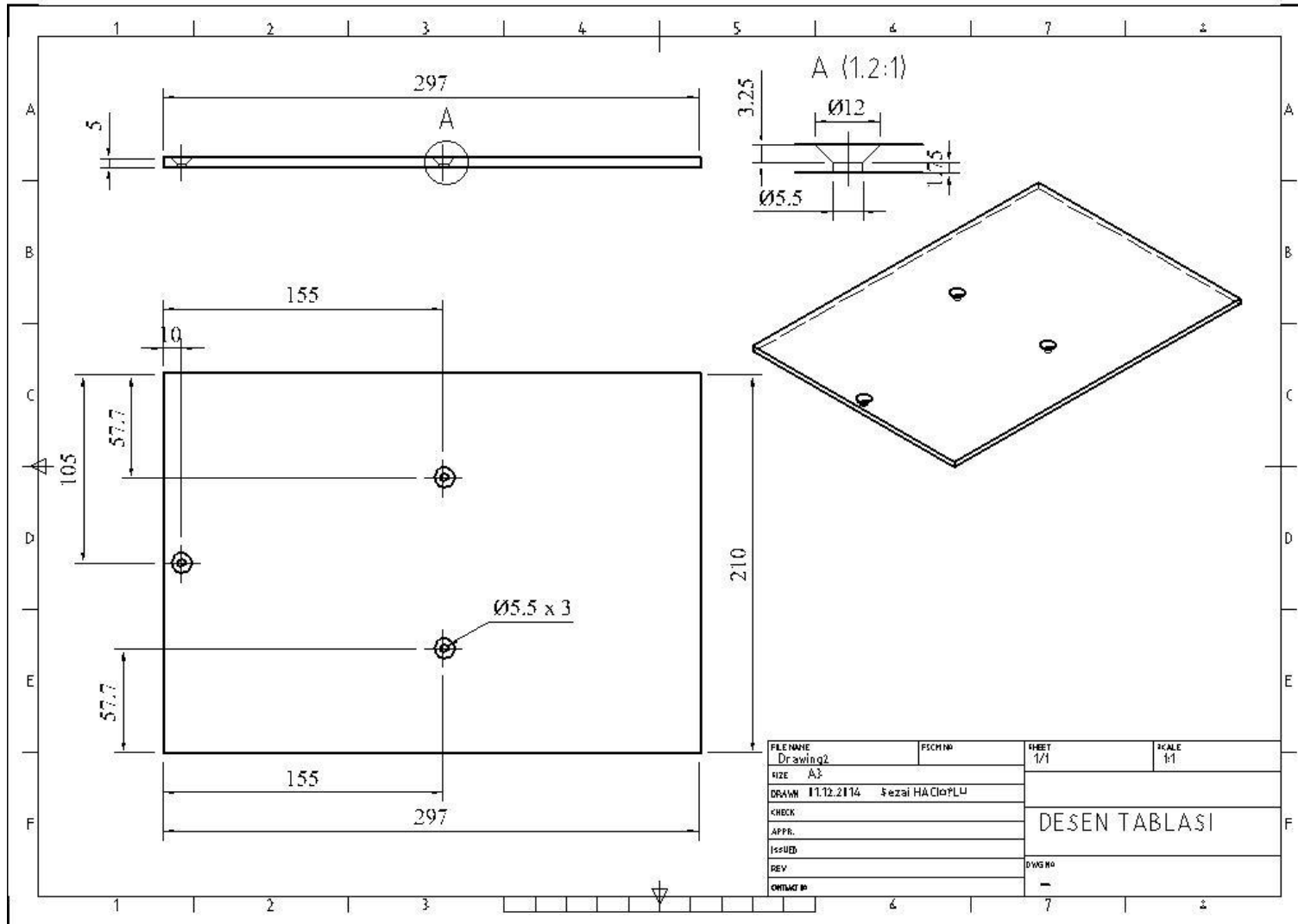


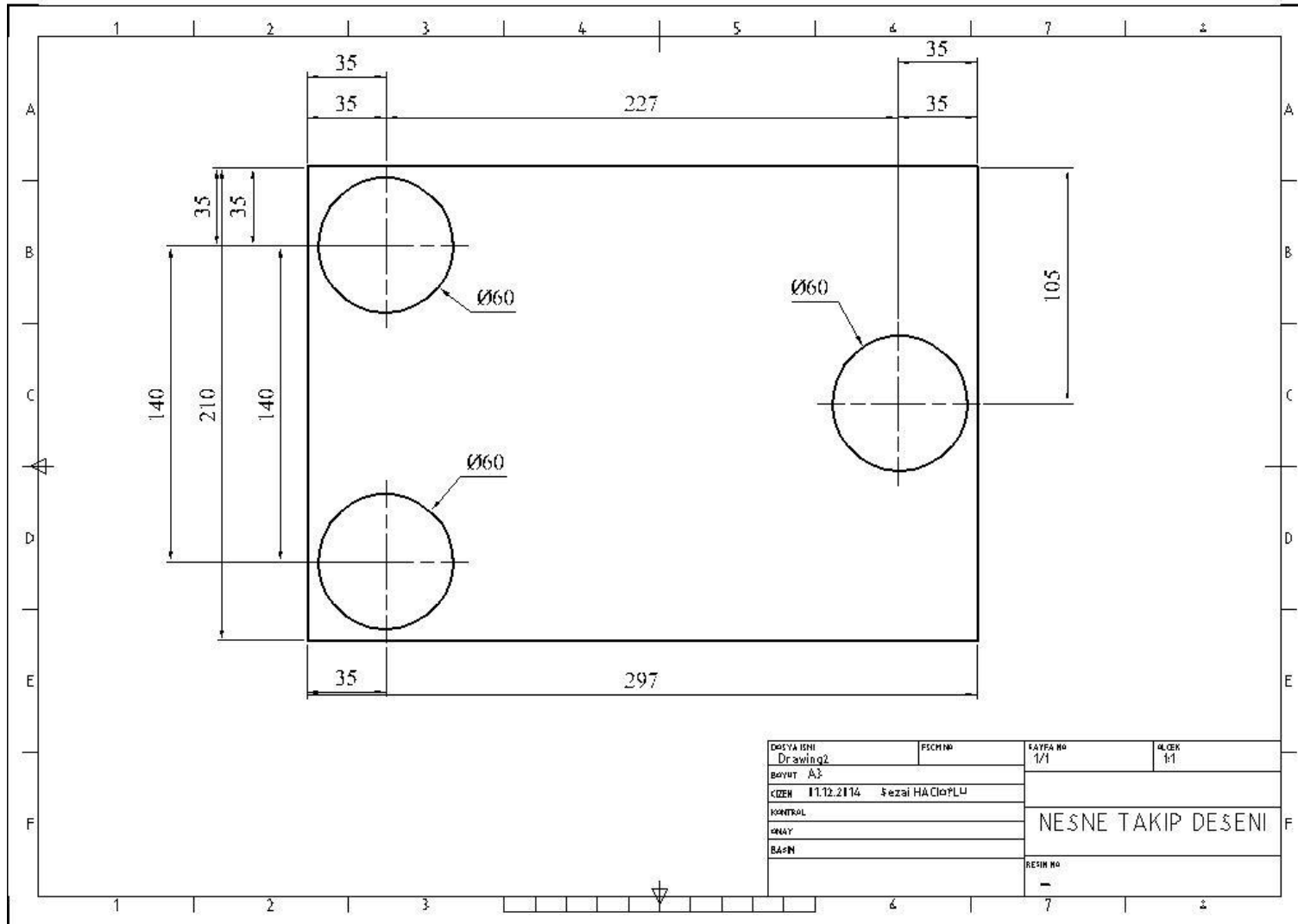
FILENAME	HARBETLİ PLATFORM KAMPE	PSCM NO	SHEET	SCALE
SIZE	A3		1/1	1/1
DRAWN	11.12.2114	FEZAL HAÇIĞLU	90x10 TEKERLEK	
CHECK				
APPR.				
ISSUED				
REV			DWG NO	
CONTIACI NO				

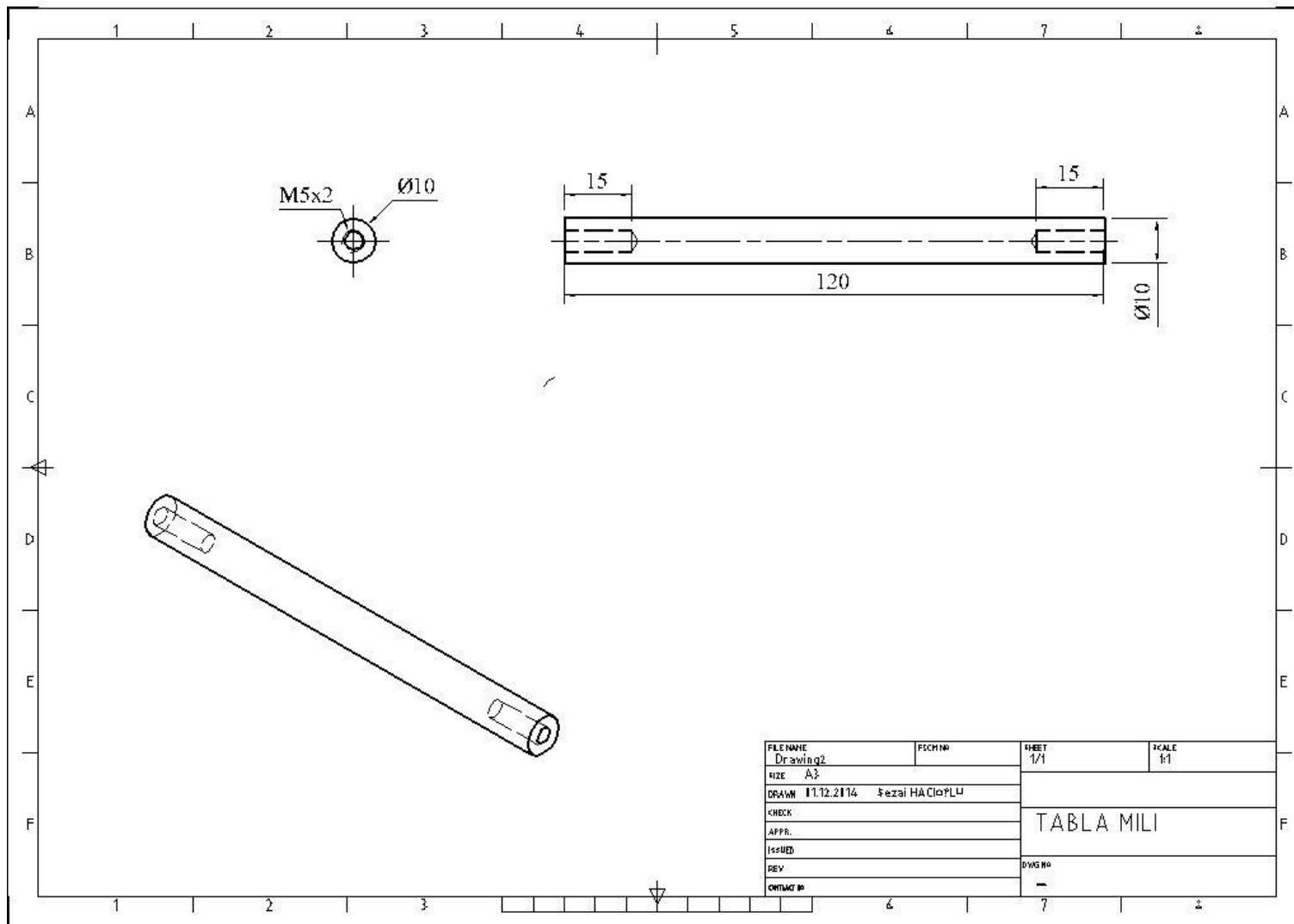


4	TABLA MILI	3
3	M5 X 16	3
2	DESEN TABLASI	1
1	NESNE TAKIP DESENI	1
Item	Name	Qty

FILENAME HARBETLİ PLATFORM KOMPLE	FSCH NO	SHEET 1/1	SCALE 1:1
SIZE A3			
DRAWN 11.12.2114 Fezal HACIÖĞLU			
CHECK	NESNE TAKIP PLATFORMU		
APPR.			
ISSUED			
REV	DWS NO		
CONTIG NO	-		







ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Sezai HACIOĞLU
Doğum Yeri : Tolbuhin
Medeni Hali : Evli
E-posta : shacioglu26@hotmail.com
Adresi : EMKO Sanayi Sitesi Blok B10 No:8, 26110, Eskişehir

Eğitim

Lisans : Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi,
Makine Mühendisliği Bölümü
Yüksek Lisans: Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Makine
Mühendisliği Bölümü Yüksek Lisans Programı
Doktora : -

Yabancı Dil ve Düzeyi

İngilizce, İyi

İş Deneyimi

Klas Makine Sanayi A.Ş. Ar-Ge Mühendisi (2010 - ...)
Hiscar Otomotiv End.ve Tic. A.Ş. Proje Mühendisi (2008 – 2010)
Hiscab Kabin ve Zirai Ekip. End. San. ve Tic. A.Ş. Tasarım Mühendisi(2007-2008)

Deneyim Alanları Tasarım, Proje, Araştırma – Geliştirme, Otomotiv, Gıda Makinaları

Tezden Üretilmiş Projeler ve Bütçesi -----

Tezden Üretilmiş Yayınlar -----

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar -----