

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MEMs İVME SENSÖRLÜ ATALETSEL SEYRÜSEFER SİSTEMİ İÇİN ARAYÜZ
TASARIMI**

DOĞAN AYDENİZ

**YÜKSEK LİSANS TEZİ
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI
MAKİNE TEORİSİ VE KONTROL PROGRAMI**

**DANIŞMAN
YRD. DOÇ. DR. VASFİ EMRE ÖMÜRLÜ**

İSTANBUL, 2011

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**MEMs İVME SENSÖRLÜ ATALETSEL SEYRÜSEFER SİSTEMİ İÇİN ARAYÜZ
TASARIMI**

Doğan AYDENİZ tarafından hazırlanan tez çalışması 07.07.2011 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Vasfi Emre ÖMÜRLÜ
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Vasfi Emre ÖMÜRLÜ
Yıldız Teknik Üniversitesi

Doç. Dr. Haluk GÖRGÜN
Yıldız Teknik Üniversitesi

Yrd. Doç. Dr. Selçuk ARSLAN
Yıldız Teknik Üniversitesi

ÖNSÖZ

Geçmişten günümüze teknolojik gelişmeler her zaman daha iyi ve daha maliyetsiz üretime doğru olmuştur. Bu açıdan yakın geçmişte MEMs tabanlı sensörlerin gelişmesi yeni teknolojik yapılandırmalara olanak sağlamıştır.

Bu çalışma kapsamında Jiroskopsuz Ataletsel Yönlendirme Sistemleri konusu işlenecektir. Öncelikle daha önceden yapılmış olan çalışmalara kısaca bir göz atılacak daha sonra Jiroskopsuz Ataletsel Yönlendirme Sistemlerinin çeşitli yapılandırmalarından ve matematiksel denklemleri üzerinde durulacak. Bunu takiben ivme sensörlerine değinilecek ve bu çalışma kapsamında yazılmış olan GUI programından bahsedilecektir.

Çalışmalarım sırasında, bana her zaman destek olan aileme, değerli yardımlarını esirgemeyen Ahmet KIRLI'ya ve Sayın Hocam Yrd. Doç. Dr. Vasfi Emre ÖMÜRLÜ'ye teşekkürlerimi sunarım.

Mayıs, 2011

Doğan AYDENİZ

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	vii
KISALTMA LİSTESİ	viii
ŞEKİL LİSTESİ.....	ix
ÖZET.....	x
ABSTRACT	xi
BÖLÜM 1.....	1
GİRİŞ.....	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	4
1.3 Orjinal Katkı.....	4
BÖLÜM 2.....	5
İVME SENSÖRLERİ	5
2.1 MEMs Tabanlı İvme Sensörleri	5
2.2 Bir İvme Sensörünün Genel Yapısı	6
2.3 Bir İvme Sensörünün Matematiksel Modeli	8
BÖLÜM 3.....	10
JİROSKOPTAN BAĞIMSIZ BİR ATALETSEL SEYRÜSEFER SİSTEMİNİN PARAMETRİK TASARIMI	10
3.1 Giriş	10
3.2 Ataletsel Yönlendirme Sistemlerinin Çalışma Prensipleri	10
3.3 İvme Bilgilerinin Hesaplanması	11
BÖLÜM 4.....	17
YAPILANDIRMA HATALARININ ÇIKIŞA ETKİLERİ	17

4.1	MEMs Tabanlı İvme Sensörleri	17
4.1.1	Yönlendirme Hataları.....	18
4.1.2	Yerleştirme Hataları.....	19
BÖLÜM 5	20
ARAYÜZ TASARIMI	20
5.1	Tasarlanan GUI Programının Çalışma Prensipleri ve İçeriği	20
5.2	GF-INS Tasarım Örneği ve Rota Tasarımı	29
BÖLÜM 6	33
SONUÇ VE ÖNERİLER	33
KAYNAKLAR	34
EK-A	36
GUI PROGRAM ÖRNEĞİ	36
EK-B	52
PROGRAM KODU	52
ÖZGEÇMİŞ	81

SİMGE LİSTESİ

Θ	: Yunuslama açısı [radyan]
\emptyset	: Yalpalama açısı [radyan]
\vec{V}	: Gövde çizgisel hız vektörü
$\vec{\Omega}$: Gövde açısal hız vektörü
e	: Hareket hatası
\dot{e}	: Hareket hatasının türevi
a_F	: Aracın ivmesi
g	: Yerçekimi ivmesi
k_p	: Oransal Kontrol Katsayısı
k_i	: İntegral Kontrol Katsayısı
k_d	: Türevsel Kontrol Katsayısı
$L_{x,y,z}$: x, y ve z eksenlerindeki açısal momentumlar
$\dot{\omega}$: Açısal İvme
P	: Lineer İvme
ω	: Açısal Hız
u_e	: İvme Sensörünün yerleştirme Hatası
θ_e	: Çalışma Doğrultusundaki Sapma
ω	: Açısal Hız
$\dot{\omega}$: Açısal İvme
P	: Lineer İvme

KISALTMA LİSTESİ

INS	Inertial Navigation Systems
ASS	Ataletsel Seyrüsefer Sistemi
JB-ASS	Jiroskoptan Bağımsız Ataletsel Seyrüsefer Sistemi
IMU	Inertial Measurement Unit (Ataletsel Ölçme Sistemi)
PID	Oransal-İntegral-Türev denetleyici
GPS	Global Positioning System (Global Konumlama Sistemi)
LQR	Linear Quadratic Regulator
GUI	Graphical User Interface (Grafiksel Kullanıcı Arayüzü)
DC	Direct Current
VDC	Volt Direct Current
rpm	revolutions per minute (devir/dakika)
A	Amper
Hz	Hertz
kg	kilogram
m	metre
cm	santimetre
mm	milimetre
sn	saniye
nb	negatif büyük
nm	negatif orta
ns	negatif küçük
z	sıfır

ŞEKİL LİSTESİ

	Sayfa
Şekil 1.1	9 İvme Sensörü Kullanılarak Oluşturulmuş Bir JB-ASS [18] 3
Şekil 2.1	Bir İvme Sensörünün Genel Yapısı [9]..... 7
Şekil 2.2	Üç Boyutlu Katı Cisim Hareketi [9]..... 8
Şekil 3.1	Tek Eksenli 6 İvme Sensörü Kullanılmış Bir JB-ASS Yapılandırması [10] .. 11
Şekil 5.1	Ana Menü..... 20
Şekil 5.2	Ana Menü Komut Penceresi 21
Şekil 5.3	İvme Sensör Konum Penceresi - 1 21
Şekil 5.4	İvme Sensör Konum Penceresi - 2 22
Şekil 5.5	İvme Sensör Konum Komut Penceresi..... 23
Şekil 5.6	İvme Sensör Çalışma Doğrultu Seçim Penceresi..... 24
Şekil 5.7	İvme Sensör Çalışma Doğrultu Seçimi Komut Penceresi 25
Şekil 5.8	Ana Menü (2) 26
Şekil 5.9	Ana Menü Komut Penceresi 26
Şekil 5.10	İvme Sensör Değerleri Penceresi 27
Şekil 5.11	Grafik Seçim Penceresi 28
Şekil 5.12	İvme Grafik Penceresi 29
Şekil 5.13	Açısal İvme Grafiği 30
Şekil 5.14	Doğrusal İvme Grafiği 30
Şekil 5.15	Doğrusal Hız Grafiği 31
Şekil 5.16	Doğrusal Konum Grafiği..... 31
Şekil Ek.1	Komut Penceresi 45
Şekil Ek.2	GUI Başlangıç Penceresi 45
Şekil Ek.3	Ana Pencere 46
Şekil Ek.4	GUI Program Örneği..... 46
Şekil Ek.5	Tasarım Aşaması - 1 47
Şekil Ek.6	Tasarım Aşaması - 2 48
Şekil Ek.7	Komut Penceresi - 1 49
Şekil Ek.8	Komut Penceresi - 2 49
Şekil Ek.9	Komut Penceresi - 3 50
Şekil Ek.10	Komut Penceresi - 4 50
Şekil Ek.11	Test Aşaması ve Sonuç..... 50

MEMs TABANLI İVME SENSÖRÜ YARDIMIYLA ATALETSEL YÖNLENDİRME SİSTEMİ GELİŞTİRİLMESİ

Dođan AYDENİZ

Makine Mühendisliđi Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Vasfi Emre ÖMÜRLÜ

Ataletsel Seyrüsefer Sistemleri (Inertial Navigation System - INS), uçuş bilgilerinin elde edilmesi açısından, hava araçlarında geniş çaplı olarak kullanılmakla beraber, daha ucuz maliyet amacıyla, bu cihazların jiroskop tabanlı olanları yerine, mikro elektromekanik sistemler (Micro Electromechanical Systems - MEMs) kullanılarak tasarlanması üzerine çalışmalar son yıllarda hız kazanmıştır. Bu sayede sağlanacak maliyet düşüklüğü ve boyut küçüklüğü, yer araçları dâhil olmak üzere, farklı uygulamalara da kapı açmaktadır. Bu araştırmada, MEM tabanlı ataletsel seyrüsefer sistemlerinin tasarım değişkenlerine ve detaylı olarak MEM algılayıcısı, algılayıcı konum ve yönelimi üzerine benzetim çalışması yapılarak, farklı tasarım değişkenlerinin girilebileceđi bir tasarım arayüzü oluşturulması amaçlanmaktadır. INS'lerde kullanılan MEMs'lerin, INS üzerindeki konumlandırılmalarından kaynaklanan geometrik hataların ve algılayıcının kendi iç hatalarının sebep olduđu açısız ve doğrusal konum hatalarının asgariye indirilmesi bu çalışmanın konuları arasındadır.

Anahtar Kelimeler: GF-INS, MEMs, GUI, INS, ataletsel seyrüsefer sistemi, ivme sensörü

YILDIZ TEKNİK ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

**THE DESIGN OF AN INERTIAL NAVIGATION SYSTEM BY MEANS OF MEMs-
BASED ACCELARATION SENSORS**

Dođan AYDENİZ

Department of Mechanical Engineering

MSc. Thesis

Advisor: Assist. Prof. Dr. Vasfi Emre ÖMÜRLÜ

Although, gyro-based inertial navigation systems (INS) is extensively used on air vehicles in order to obtain flight information, MEMs-based INS (GF-INS) has been studied exponentially to reduce the costs of these devices in recent years. Resulting cost-effective and small designs of INS sensors would lead to extended usage of the sensors covering land vehicles. In this research, design parameters of the INS systems, specifically, the ones that are related to the MEMs sensors, their location and directions on an INS, are aimed to be optimized so that several near-ideal systems can be realized and the system trials can be performed. Simulation studies aim to minimize the effects of the MEMs locational, directional and internal errors on the overall INS system.

Key words: GF-INS, MEMs, GUI, INS, inertial navigation systems, accelerometer

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

1.1 Literatür Özeti

ASS, bir aracın hızını ve konumunu, algılayıcılar vasıtasıyla, başka bir dış referansa bağlı kalmadan takip eden ölçüm sistemidir ve hava araçları, uzay araçları, balistik roketler, otomatik tarım, inşaat araçları ve robotik gibi askeri ve sivil amaçlar için kullanılmaktadır[1]. Yakın zamana kadar sadece üç ivme sensörü ve üç jiroskoptan oluşan bu ataletsel seyrüsefer sistemleri, gelişen MEMs algılayıcıları sayesinde, altı ya da daha fazla MEMs ivme sensörü kullanılarak, jiroskopsuz ASS (JB-ASS) olarak son yıllarda tasarlanmaya başlanmıştır. ASS'lerde ivme sensörleri doğrusal ivme ölçümleri yaparken jiroskoplar ise açısal ivme ölçümünden sorumludurlar. JB-ASS'lerde ise hem doğrusal hem de açısal ivme ölçümlerinden ivme sensörleri sorumludur. Fakat bu görevi yerine getirebilmeleri için belirli şekillerde yapılandırılmaları gereklidir. Jiroskopların, MEMs ivme algılayıcılarına göre, daha karmaşık, büyük ve pahalı olduğu göz önüne alınırsa, jiroskopsuz tasarlanan ASS'lerin daha küçük boyutlarda ve daha ucuza imal edilebileceği anlaşılmaktadır[2-3-4-5].

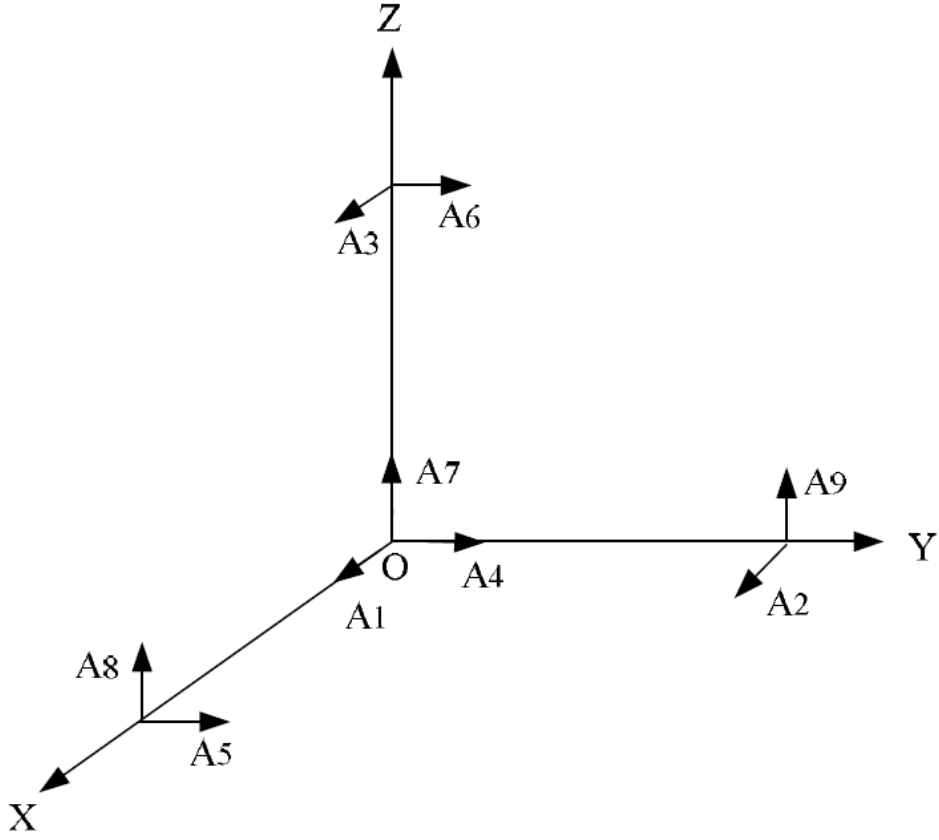
Fakat bu sistemlerin bir zayıf noktası olarak, uzun süreli kullanımlarda, konumlamada sapmalar oluşabilmektedir. Bu sebeple bu sistemler ile eş çalışan Küresel Konumlama Sistemleri (GPS), ASS'lerin uzun süreli kullanımında oluşabilecek sapmaları gidermek veya azaltmak için önerilmektedirler.[6-7].

JB-ASS ile ilgili ilk alıřmalar XX. yzyılın sonlarında otomobil uygulamalarıyla gerekleřtirilmiřtir ki burada, halihazırda otomobil iinde kullanılmakta olan farklı ivme algılayıcılarının koordineli kullanımıyla, seyrsefer bilgilerinin eldesi ve optimizasyon amalanmıřtır[8]. Ardından, konuyla ilgili detaylı analizler, algılayıcıların yerleřtirilmesi, ynelimi, sistemde oluřabilecek sensr yerleřimlerinden kaynaklanabilecek geometrik hatalar ve sensr bazlı hatalar zerine yoęunlařılmıřtır[9-10]. eřitli sayılarda ivme sensrleri ve bu sensrler iin eřitli konumlamalar kullanılarak JB-ASS'ler retilmiřtir. Bir JB-ASS'in tasarımında kullanılabilecek en az sayıda, tek eksenli ivme sensr sayısı 6'dır[11]. Minimum sayıda ivme sensr kullanmak her ne kadar maliyet ve byklk aısından avantajlı olsa da yapılan deneyler ortaya koymuřtur ki bu durum oluřan hataları arttırmaktadır. Hatalar, ivme sensrlerinin konum ve yerleřtirilmelerindeki sapmalardan kaynaklanmaktadır ve bunlar tamamen olmasa bile bir dereceye kadar giderilebilir[12-13-14].

JB-ASS zerine arařtırmalar, bu cihazların GPS destekli kullanımı, sensrlerin geometrik ve i hatalara karřı kalibrasyonu, sistem verimlilięi ve daha ok da tasarımı konularında yoęunlařmaktadır. Jiroskop tabanlı klasik ASS cihazlarında olduęu gibi, birikimli sensr hatalarının belli aralıklarla sıfırlanması maksadıyla, JB-ASS sistemlerde de GPS destekli lmler zerine alıřmalar yapılmıřtır[15]. Gene bu cihazların alıřma kalitesi aısından, algılayıcı, toplam cihaz ıkıř kalitesi ve alan deęerlendirmesi bazında, cihazın alıřma performansı ltleri zerine alıřmalar mevcuttur[16].

JB-ASS cihazlarının, genel otomatik kalibrasyonu zerine yapılan arařtırmalar ise tasarımıla ilgili olanların ardından ikinci sırada gelmektedir ki bunun sebebi, sensrlerin yerleřtirilirkenki konumlama ve ynelim hatalarının sıfırlanamaması ve sistem ıkıřına etkisidir[17-18].

Kullanılabilecek ivme algılayıcısı sayısı[19] , bunların konumları, ynelimleri ve lebildikleri eksen sayıları[20] , algılayıcı zellikleri, btnyle tasarım ltlerine girmektedir ki bunların seimleri zerine farklı varyasyonlar dřnlerek ok sayıda alıřma yapılmıřtır. Bu alıřmalardan bir tanesi olan 9 ivme sensrl yapılandırma Őekil-1.1'de grlmektedir.[21-22].



Şekil 1.1 9 İvme Sensörü Kullanılarak Oluşturulmuş Bir JB-ASS [18]

Bu çalışmada, JB-ASS cihazlarının bilinen hesaplama yöntemleri kullanılarak, en genel manada bir tasarım arayüzü oluşturulması amaçlanmıştır. Cihazın giriş-çıkış arasındaki kinematik matematiksel altyapısı öncelikle verildikten sonra, program algoritması ve arayüzü tanıtılacak ve sonuçlar tartışılacaktır.

1.2 Tezin Amacı

Ataletsel Seyrüsefer Sistemi (Inertial Navigation System - INS), uçuş bilgilerinin elde edilmesi açısından, hava araçlarında geniş çaplı olarak kullanılmakla beraber, daha ucuz maliyet amacıyla, bu cihazların jiroskop tabanlı olanları yerine, küçük elektromekanik algılayıcılar (Micro Electromechanical Devices - MEMs) kullanılarak tasarlanması üzerine çalışmalar son yıllarda hız kazanmıştır. Bu sayede sağlanacak maliyet düşüklüğü ve boyut küçüklüğü, yer araçları dâhil olmak üzere, farklı uygulamalara da kapı açmaktadır. Bu araştırmada, MEM tabanlı ataletsel yönlendirme sistemlerinin tasarım değişkenleri üzerine ve detaylı olarak MEM algılayıcısı, algılayıcı konum ve yönelimi üzerine benzetim çalışması yapılmıştır. Benzetim çalışması sonucunda, ASS lerde kullanılan MEMs lerin, ASS üzerindeki konumlandırılmalarından kaynaklanan geometrik hataların ve algılayıcının kendi iç hatalarının sebep olduğu açisal ve doğrusal konum hatalarının asgariye indirilmesi amaçlanmaktadır.

1.3 Orjinal Katkı

Havacılıkta kullanılan bütün sistemlerin maliyetli sistemler olması dolayısıyla, öncelikle bu tarz bütünleşik (entegre) sistemlerin teknolojik altyapılarının ülkemizde oluşturulması, bilgi birikimi ile ancak mümkündür. Dolayısıyla, mevcut kullanılan jiroskop tabanlı ataletsel seyrüsefer sistemlerinin daha ucuz maliyet ve yüksek hassasiyetle eldesini sağlayacak MEM destekli yeni bir yapılanmayı amaçlamaktadır. Teknolojik altyapıyı, bilgi birikimini beslemesi ve ayrıca kısmen yeni bir teknolojiyi amaçlaması bakımından tez önem arz etmektedir.

Tezin organizasyonu şu şekilde olacaktır; Tezin ikinci bölümünde MEMs tabanlı ivme sensörleri üzerine bilgiler verilecek, bir sonraki kısımda JB-ASS'lerin parametrik tasarımlarından bahsedilecek, dördüncü kısımda yapılandırmadan kaynaklanan hatalara yer verilecek, beşinci kısımda oluşturulan arayüz programı tanıtılacak ve en son kısım olan altıncı kısımda ise sonuçlar ve gelecekte yapılması planlanan çalışmalardan bahsedilecektir.

BÖLÜM 2

İVME SENSÖRLERİ

Tez kapsamında yapılan çalışmada MATLAB altında çalışan bir GUI programı tasarlanmıştır ve bu programın istenen sonuçları verebilmesi için belirli bir ASS konfigürasyonunun seçilmesi gerekmektedir. Şu ana kadar yapılmış çalışmalar da bazı konfigürasyonlar denenmiştir, ki bunlar 6 ivme sensörlü kübik yapılandırmalı, 9 ivme sensör yapılandırmalı, 4 adet 3 eksenli ivme sensörü kullanılarak oluşturulan yapılandırmalardır. [9-13-18]

Önceki çalışmalara baktığımızda karşımıza en çok çıkan INS konfigürasyonu kübik yapıda olmaktadır. Bu konfigürasyonu takiben çeşitli sayılardaki ivme sensörleriyle değişik şekillerdeki yapılandırmalar ile karşılaşmaktayız.

2.1 MEMs Tabanlı İvme Sensörleri

Bu bölümde, günümüze kadar kullanılan konfigürasyonlardan bahsederken ASS'in hem mekanik hem de mekatronik yapılarından bahsedilecektir. Böylece bu sistemler hakkında okuyucuya daha sonraki bölümlerin daha iyi anlaşılmasına imkan verecek bilgilerin verilmesi amaçlanmıştır.

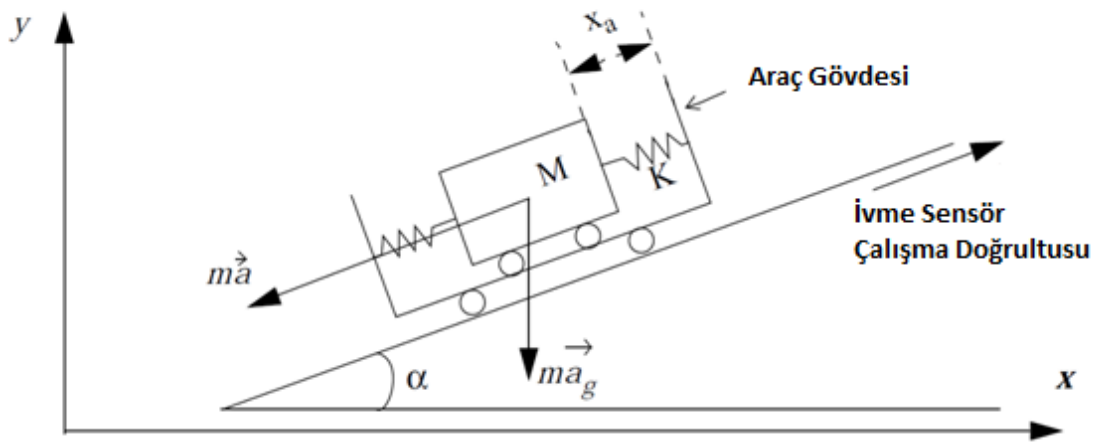
Mikroelektro-mekanik sistemler (MEMs) günümüzde kullanılan mekanik ve elektrik sistemlerinin mikron boyutlarında tasarımı yapılmış türevleridir. Bu sistemler mikron boyutlarındadır fakat bu sistemler aynı zamanda nano boyutlarda tasarlanıp nanoteknoloji alanında da kullanılmaktadır. MEMs adı ilk olarak 1987 yılında kullanıma girmiştir. Fakat MEMs kavramının ortaya çıkması esas olarak entegre devre

çalışmalarında yaşanan gelişmeler ışığında olmuştur. Mikro boyutlarda tasarımların yapılması fikrini ilk olarak fizikçi Richard Feynman tarafından 1959 yılında yapılan "There's plenty of room at the bottom" isimli konuşmada ortaya atılmıştır. Mikro-elektromekanik sistemlerin boyutları 1 ile 100 mikrometre arasında değişim göstermektedir ve günümüzde bilindiği gibi bu boyutlarda evrensel fizik kanunları geçerliliğini kaybetmektedir . MEMs yapılarında hacimin yüzey alanına oranı oldukça düşüktür ve bunun neticesinde yüzey etkileri (elektrostatik kuvvetler) hacim etkilerine (eylemsizlik, ısı kütlesi) baskın gelir. Mikro elektro-mekanik sistem yapıları üç bölümden oluşur. Bu bölümler mekanik bölüm, mekanik bölümü çalıştıran tahrik bölümü ve mekanik hareketin davranışını inceleyen algılama bölümü olarak ayrılabilir. İkinci bölümde bahsedilen MEMs tahrik mekanizmaları verilen tahrik tipine göre farklılık gösterir. MEMS yapıları ısı, elektrostatik, manyetik, pnömatik ve optik olarak tahrik edilebilir. Algılama işlemi ise genellikle optik ve elektronik sinyaller vasıtası ile yapılır. MEMs, Makina-Malzeme-Elektronik başta olmak üzere, temelde tüm mühendislik dalları arasında sıklıkla kullanılmaktadır.

Ayrıca statik ivmenin düzeyini ölçme kabiliyeti bulunan MEMS tipi ivmeölçerler de bazı uygulamalarda vazgeçilmezdirler.

2.2 Bir İvme Sensörünün Genel Yapısı

Tek eksenli bir ivme sensörünün genel yapısı Şekil 2.1'de görüldüğü gibidir. Bu yapıdan yola çıkılarak x_a sıkıştırma miktarını, M kütleyi, K yay katsayısını, a_g ise yerçekimi ivmesini göstermektedir.



Şekil 2.1 Bir İvme Sensörünün Genel Yapısı [9]

Burada ivme sensörüne etki eden iki kuvvet yerçekimi kuvveti ve araçtan kaynaklanan kuvvettir. Newton'un ikinci kanununa göre;

$$\vec{F} = m\vec{a} + m\vec{a}_g = m\vec{a}_{eq} \quad (2.1)$$

olmaktadır.

\vec{a}_{eq} yerçekimi ivmesi ve araçtan kaynaklanan ivmeler vektörel toplamı olan eşdeğer ivmeyi ifade etmektedir. Bu noktada yay kuvveti $F = kx_a$ kullanılarak;

$$\vec{a}_{eq} = \ddot{x}_a = \frac{kx_a}{m} \quad (2.2)$$

olarak bulunabilir. Kapasitans tipi bir ivme sensöründe ise x_a yerine;

$$C = \epsilon_0 \frac{A}{x_a} \quad (2.3)$$

yazılacak olursa;

$$\vec{a}_{eq} = \frac{kA\epsilon_0}{mC} \quad (2.4)$$

olarak bulunabilir. Yukarıdaki denklemde görülebileceği gibi kapasitans tipi bir ivme sensörünün eşdeğer ivmesi elde edilmiştir.

$$A = \langle \ddot{r}_I - a_g + \delta \ddot{\theta}_I, \theta_I \rangle \quad (2.5)$$

olarak bulunur. Bu denklemde δ yaklaşık 10^{-7} m mertebesindedir ve bu sebepten dolayı ihmal edilebilir. Denklem yeniden düzenlenirse;

$$A = \langle \ddot{r}_I - a_g, \theta_I \rangle \quad (2.6)$$

İvme sensörü çıkışı bu şekilde yazılabilir. Denklemden de görülebileceği gibi ivme sensörü çıkışı, rijit cismin kütle merkezinin ve yerçekimi ivmelerinin ivme sensörü çalışma doğrultusuna izdüşümü olarak ortaya çıkmaktadır.

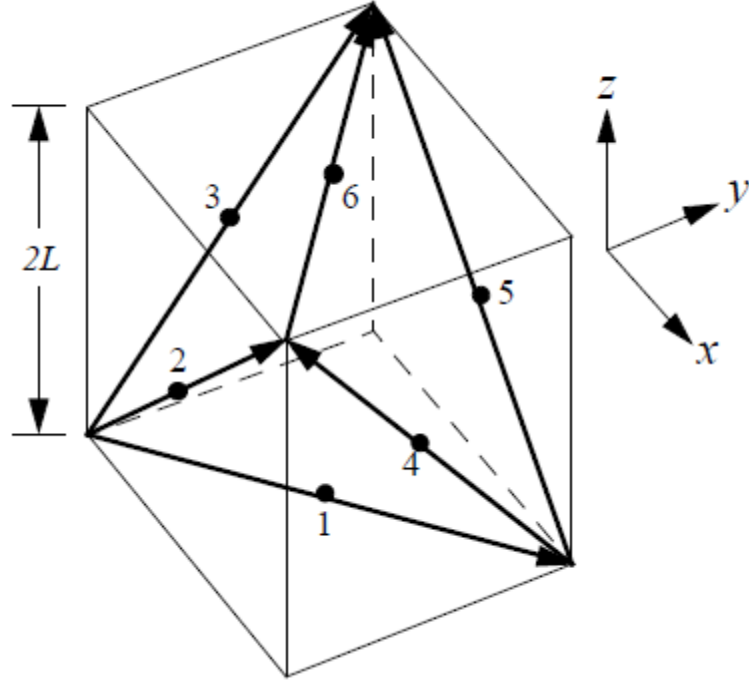
JİROSKOPTAN BAĞIMSIZ BİR ATALETSEL SEYRÜSEFER SİSTEMİNİN PARAMETRİK TASARIMI

3.1 Giriş

Gerçekleştirilmesi planlanan bu tezin içeriğinde, MATLAB programı altında çalışan grafiksel arayüz ve Simulink programları yardımıyla çeşitli varyasyonların denenebildiği bir arayüz hazırlanıp daha sonra seçilen bir yapılandırmanın testlerinin gerçekleştirilmesi bulunmaktadır. Testleri yapılacak sistemde, seçilen bir yapılandırmaya göre stratejik olarak yerleştirilecek belirli sayılardaki ivme sensörleri çıkışlarının bilgisayar ekranından okunarak denemeler yapılabilmesi planlanmaktadır. Böylece JB-ASS'in son halinin uygulamalar için hazır hale getirilmesi planlanmaktadır.

3.2 Ataletsel Yönlendirme Sistemlerinin Çalışma Prensibi

Şekil-3.1'de görülen yapılandırmada 6 adet tek eksenli ivme sensörü kullanılarak oluşturulmuş bir JB-ASS tasarımı görülmektedir. Bu sistem, üzerine yerleştirildiği aracın ivme bilgilerini kullanıcının aracı harekete geçirmesiyle birlikte sisteme stratejik olarak yerleştirilmiş ivme sensörleri vasıtasıyla ölçmekte ve mikroişlemciye göndermektedir. Yazılan algoritma sayesinde her bir ivme sensöründen gelen veriler ışığında düzeneğin doğrusal ve açısal ivmeleri bulunmaktadır. Bulunan doğrusal ve açısal ivmelerin hız ve konuma dönüştürülmesiyle birlikte istenilen çıkış bilgileri elde edilmiş olmaktadır.



Şekil 3.1 Tek Eksenli 6 İvme Sensörü Kullanılmış Bir JB-ASS Yapılandırması [10]

3.3 İvme Bilgilerinin Hesaplanması

Bir JB-ASS'in oluşturulması sırasında kullanılan denklemler şu şekildedir;

İlk önce ivme sensörlerinin belirlenecek bir referansa göre konumları belirlenmelidir ki, ivme sensör konumları denklemlerde U matrisi ile belirtilecektir. Şekil-3.1'deki yapılandırma göz önüne alınacak olursa U matrisi aşağıdaki şekilde düzenlenebilir;

$$U = [u_1, u_2, u_3, u_4, u_5, u_6] = L \begin{bmatrix} 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Denklem düzenlenirken koordinat sisteminin küpün merkezinde seçildiğine dikkat etmek gerekir. U matrisi kullanılan ivme sensör sayısına göre benzer şekilde oluşturulmaktadır. U matrisi oluşturulduktan sonra yapılması gereken işlem, konumları belirlenen ivme sensörlerinin çalışma yönlerinin tayinidir. Örneğin Şekil-3.1'deki kübik yapılandırmada gösterilen ok yönleri ivme sensörlerinin çalışma doğrultularıdır ve J_2 matrisi ile gösterilmektedir. Her bir ivme sensörü için çalışma doğrultuları $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ belirlenmelidir. Bu durumda J_2 matrisi aşağıdaki şekilde oluşturulabilir;

$$J_2 = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & 0 & 1 & -1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (3.2)$$

Aynı şekilde J_2 matrisi kullanılan ivme sensör sayısına göre benzer şekilde oluşturulabilir. Daha sonra oluşturulan U ve J_2 matrislerinin vektörel çarpımlarıyla oluşturulan J_1 matrisi bulunmalıdır, bu matris daha sonra lineer ivme ve açısal ivmelerin bulunmasında kullanılacaktır.

$$J_1 = [u_1 \times \theta_1, \dots, u_N \times \theta_N] = \frac{L}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & 0 & 1 & -1 \\ -1 & 0 & 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & -1 & 1 & 0 \end{bmatrix} \quad (3.3)$$

(3.4) ve (3.5) numaralı denklemlerde aracın açısal ivme ve doğrusal ivmeleri, açısal hız ve ivme çıkışlarının fonksiyonlarıdır.

$$\dot{\omega} = f(\omega, A) \quad (3.4)$$

$$P = g(\omega, A) \quad (3.5)$$

ve

$$\frac{1}{2} \begin{bmatrix} J_1 \frac{1}{L^2} \\ J_2 \end{bmatrix} \begin{bmatrix} -L J_2^T \begin{bmatrix} \omega_2 \omega_3 \\ \omega_1 \omega_3 \\ \omega_1 \omega_2 \end{bmatrix} \end{bmatrix} = - \begin{bmatrix} O_3 \\ LI_3 \end{bmatrix} \begin{bmatrix} \omega_2 \omega_3 \\ \omega_1 \omega_3 \\ \omega_1 \omega_2 \end{bmatrix} \quad (3.6)$$

şekindedir. Buradan; $J_1 J_2^T = O_3$, $J_2 J_2 = 2I_3$, $Q = J^{-1} = \frac{1}{2} \begin{bmatrix} \frac{1}{L^2} J_1 \\ J_2 \end{bmatrix}$ ve

$\hat{\omega} = [\omega_2 \omega_3 \quad \omega_1 \omega_3 \quad \omega_1 \omega_2]^T$ olması kaydıyla;

$$\begin{bmatrix} \dot{\omega} \\ P \end{bmatrix} = QA - Q \begin{bmatrix} \theta_1^T \Omega^2 u_1 \\ \vdots \\ \theta_N^T \Omega^2 u_N \end{bmatrix} = \frac{1}{2} \begin{bmatrix} J_1 \frac{1}{L^2} \\ J_2 \end{bmatrix} \begin{bmatrix} A_1 \\ \vdots \\ A_6 \end{bmatrix} + L \begin{bmatrix} O_3 \\ \hat{\omega} \end{bmatrix} \quad (3.7)$$

şeklinde düzenlenebilir. Yukarıdaki denklemde O_3 3x3 sıfır matris, I ise 3x3 birim matrisi temsil etmektedir. Bu denklemde açısız ve doğrusal ivme ayrı ayrı ele alınarak çözüme ulaşılabilir. Şöyle ki;

$$\dot{\omega} = \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \dot{\omega}_3 \end{bmatrix} = \frac{1}{2L\sqrt{2}} \begin{bmatrix} A_1 - A_2 + A_5 - A_6 \\ -A_1 + A_3 - A_4 - A_6 \\ A_2 - A_3 - A_4 + A_5 \end{bmatrix} \quad (3.8)$$

ve

$$P = \frac{1}{2\sqrt{2}} \begin{bmatrix} -A_1 + A_2 - A_5 - A_6 \\ A_1 + A_3 - A_4 + A_6 \\ A_2 + A_3 + A_4 + A_5 \end{bmatrix} + L \begin{bmatrix} \omega_2 \omega_3 \\ \omega_1 \omega_3 \\ \omega_1 \omega_2 \end{bmatrix} \quad (3.9)$$

Sadece ivme çıkışlarına (A_1, \dots, A_N) bağılı olan açısız ivme ($\dot{\omega}$) bu denkleme göre hesaplanabilir. Burada hesaplanan her bir eksendeki açısız ivme aracılığıyla açısız hızlar hesaplanarak doğrusal ivme denkleminde kullanılmalıdır. Bu sayede doğrusal ivme hesaplanabilir. Fakat bu noktada karşılaşılan problemlerden birisi açısız ivmeden açısız hıza geçişlerde karşımıza çıkan kaymadır. Bu durumda oluşacak olan kayma bir dereceye kadar bertaraf edilmelidir.

Bu durumda kullanılan yöntemlerden birisi Kalman Filtreleme yöntemidir. Yapılan çalışmalar ve simülasyonlar sonucunda bu yöntemle elde edilen cevaplar girişteki ölçüm hatalarının sabit bileşen içerip içermemesine göre değişim göstermektedir. Bu yöntem ayrıca ASS'lerde kullanılan yöntemler arasında önde gelen bir yöntemdir.[6-13]

Bir diğer yöntem ise Parçacık Filtreleme yöntemidir. Bu yöntemde elde edilen cevaplar girişteki ölçüm hatalarının sabit bileşenler içerip içermemesine göre değişim göstermez.

Ölçüm hatalarının sabit bileşenler içermediği durumlarda Kalman Filtreleme yöntemi daha iyi sonuçlar vermektedir.

Ayrıca ivme sensörlerinin yerleştirilmesi ve çalışma doğrultusunun ayarlanması sırasında oluşacak olan montaj hataları da göz önünde bulundurulması gereken hata çeşitlerindedir. Bu hataların giderilmesi alınacak cevap üzerinde büyük oranda iyileştirme sağlar. Bu duruma bağlı olan çıkıştaki hatalar lineer ve açısal hareketin büyüklüğüne göre değişim gösterir. İdeal bir ivme sensörü cevabı aşağıda gösterildiği şekildedir. Terimleri kısaca açıklamak gerekirse bu denklemden A ivme sensörü çıkışı, u ivme sensörünün konumu, θ ivme sensörünün çalışma doğrultusu P lineer ivme, Gu ise açısal ivme olarak karşımıza çıkmaktadır.

$$A(u, \theta) = \langle P, \theta \rangle + \langle Gu, \theta \rangle \quad (3.10)$$

Fakat ivme sensörlerinin yerleştirilmesi ve çalışma doğrultusunun ayarlanması sırasında oluşan hatalardan dolayı ivme sensörünün gerçek konumu u_r ve gerçek çalışma doğrultusu ise θ_r 'dir. Denklem bu şekilde ele alınacak olursa aşağıdaki şekilde tekrar düzenlenebilir.

$$A_r(u_r, \theta_r) = \langle L, \theta_r \rangle + \langle Gu_r, \theta_r \rangle \quad (3.11)$$

Böylece ivme sensörünün gerçek çıkışı yukarıdaki denklemden görüldüğü gibi A_r 'dir. İvme sensörlerinde arzu edilen çıkış ile gerçek çıkış arasındaki fark ise $A_e = A_r - A$, yerleştirmedeki sapma $u_e = u_r - u$, çalışma doğrultusundaki sapma $\theta_e = \theta_r - \theta$ şeklindedir ve ivme sensörü çıkışındaki hata şu şekilde hesaplanabilir.

$$A_e(u_r, \theta_r; u, \theta) = [(ux\theta_e + u_e x \theta_e)^T \theta_e^T] \begin{bmatrix} \dot{\omega} \\ L \end{bmatrix} + [\theta_e^T \Omega^2 u + \theta_r^T \Omega^2 u_e] \quad (3.12)$$

Hassasiyet, denklemin konum ve çalışma doğrultusuna göre kısmi türevleri alınarak yerleştirme hataları basit bir şekilde yerçekimi ivmesi kullanılarak düzeltilebilir. Sistem hareketsiz iken ivme sensörünün vereceği tek çıkış yerçekimi ivmesidir. Eksen takımını standart olarak $\{e_1, e_2, e_3\}$ şeklinde kabul edersek ve buna göre A_r gerçek ivme sensör çıkış denklemi tekrar düzenlenerek;

$$A_r(u, \theta_r) = - \langle F^T g, \theta_r \rangle = |g| \langle F^T e_3, \theta_r \rangle \quad (3.13)$$

elde edilebilir. Denklem eksen takımlarına göre düzenlenir ve buradan θ_r çekilirse;

$$\theta_r = \frac{1}{|g|} (-A_r^{(2)} e_1 + A_r^{(1)} e_2 + A_r^{(3)} e_3) \quad (3.14)$$

İvme sensörünün gerçek çalışma doğrultusu hesaplanabilir.

Konumlama hataları gövde döndürülmedikçe belirlenemezler bu sebeple gövde, konum hatalarını hesaplamak için atalet düzlemine bağlı olarak tüm yüzeylerde ayrı ayrı döndürülmelidir. Bütün döndürme işlemlerinde kullanılan eksen yerçekimi ivmesi yönüne paralel şekilde seçilmelidir. Örneğin Şekil-3.1'deki yapılandırma ele alınırsa önce küpün her bir düzlemi üzerinde bulunan ivme sensörü ile birlikte numaralandırılmalı ve bütün ivme sensörleri için bu yöntem ayrı ayrı uygulanmalıdır. İvme sensörünün gerçek konumu j ise;

$$A_{ri}^{(j)} = \theta_{ri}^T \hat{F}^{(j)} + \theta_{ri}^T (\Omega^{(j)})^2 u_{ri}$$

$$\omega^{(1)} = \omega_0 [\cos\beta \ 0 \ \sin\beta]^T, \quad \hat{F}^{(1)} = |g| [\cos\beta \ 0 \ \sin\beta]^T$$

$$\omega^{(2)} = \omega_0 [0 \ \cos\beta \ \sin\beta]^T, \quad \hat{F}^{(2)} = |g| [0 \ \cos\beta \ \sin\beta]^T$$

$$\omega^{(3)} = \omega_0 [\cos\beta \ \sin\beta \ 0]^T, \quad \hat{F}^{(3)} = |g| [\cos\beta \ \sin\beta \ 0]^T$$

Denklemlerde ω_0 sabit dönüş hızını $\omega(j)$ ise gövde düzleminin dönüş eksenini ifade etmektedir ve ivme sensörünün gerçek konumu;

$$u_{ri} = \begin{bmatrix} \theta_{ri}^T (\Omega^{(1)})^2 \\ \theta_{ri}^T (\Omega^{(2)})^2 \\ \theta_{ri}^T (\Omega^{(3)})^2 \end{bmatrix}^{-1} \begin{bmatrix} A_{ri}^{(1)} - \theta_{ri}^T \hat{F}^{(1)} \\ A_{ri}^{(2)} - \theta_{ri}^T \hat{F}^{(2)} \\ A_{ri}^{(3)} - \theta_{ri}^T \hat{F}^{(3)} \end{bmatrix} \quad (3.15)$$

olarak bulunabilir.

Bu kısımda, bir JB-ASS'nin parametrik denklemlerinin nasıl elde edildiği ve bir JB-ASS'nin imalatı yapılırken yerleştirilen ivme sensörlerinin kalibrasyonlarının nasıl yapılması gerektiğinden bahsedilmiştir.

YAPILANDIRMA HATALARININ ÇIKIŞA ETKİLERİ

Tezin bu kısmında tasarım sırasında oluşabilecek olan ivme sensöründeki konum ve yönlendirme hatalarından dolayı oluşabilecek hatalar ve bu hataların etkilerinden bahsedilecek, daha sonra ise bu hataların giderilme yöntemleri üzerinde durulacaktır.

4.1 MEMs Tabanlı İvme Sensörleri

İvme sensör çıkışını yanlış yönde saptırabilecek birkaç hata çeşidi vardır. Bunlar; yapılandırma hataları ve ivme sensörünün kendisinden kaynaklanan hatalardır. Yapılandırma hataları yerleştirme ve yönlendirme hataları olarak ayrılabilir. MEMs ivme sensör hata kaynakları ölçüm faktör hatası, sapma ve gürültüdür. Bütün bu hatalar tanımlanmalı ve kalibre edilmelidir. Bu kısımda bu hatalardan yapılandırma hataları ele alınacaktır. İdeal bir ivme sensörünün konumu u ve yönlendirmesi θ olarak bölüm 3'te belirtilmiştir ve buna göre ivme sensör çıkışı hatalarla birlikte, sensör konumu ve yönlendirmesi u_r ve θ_r olarak tanımlanırsa gerçek ivme sensör çıkışı için (3.11) numaralı denklemde;

$$A_r(u_r, \theta_r) = \langle L, \theta_r \rangle + \langle G u_r, \theta_r \rangle$$

$u_e = u_r - u$ ve $\theta_e = \theta_r - \theta$ olarak tanımlanırsa,

Çıkıştaki değişim için (3.12) numaralı denklemde, $A_e = A_r - A$ olarak tanımlanırsa;

$$A_e(u_r, \theta_r; u, \theta) = [(u_x \theta_e + u_e x \theta_e)^T \theta_e^T] \begin{bmatrix} \dot{\omega} \\ L \end{bmatrix} + [\theta_e^T \Omega^2 u + \theta_r^T \Omega^2 u_e]$$

Bu denklemden bütün çıkış hatalarının doğrusal hareket (L) ve açısal değere (ω) bağlı olduğu görülebilir.

Hata denkleminin kısmi türevini aldığımızda, yerleştirme ve yönlendirme hataları hesaplanabilir. Bu denklemler aşağıda gösterilmiştir.

$$\frac{\partial A_e}{\partial \theta_e} \Big|_{(u,\theta)} = (\dot{\omega} \times u)^T + L^T + u^T \Omega^2 \quad (4.1)$$

$$\frac{\partial A_e}{\partial u_e} \Big|_{(u,\theta)} = -(\dot{\omega} \times \theta) + \theta^T \Omega^2 \quad (4.2)$$

Eğer hizalama hatası yoksa ivme sensör çıkışı şu şekildedir. $A=[A_1, \dots, A_n]^T$. Temel algoritma ile doğru açısal çevrim $\omega(t)$ hesaplanabilir fakat hizalama hataları u_{ei} ve θ_{ei} varsa, gerçek ivme sensör çıkışı i , A_{ri} şeklindedir ve gerçek açısal çevrim $\omega(t)$ olur. Buna göre;

$$\frac{\partial \hat{\omega}}{\partial \theta_{ei}} = \frac{\partial \hat{\omega}}{\partial A_{ei}} * \frac{\partial A_{ei}}{\partial \theta_{ei}} = \frac{1}{2l^2} (u_i \times \theta_i) [(\dot{\omega} \times u_i)^T + L^T + u_i^T \Omega^2] \quad (4.3)$$

$$\frac{\partial \hat{\omega}}{\partial u_{ei}} = \frac{\partial \hat{\omega}}{\partial A_{ei}} * \frac{\partial A_{ei}}{\partial u_{ei}} = \frac{1}{2l^2} (u_i \times \theta_i) [\theta_i^T \Omega^2 - \dot{\omega} \times \theta_i] \quad (4.4)$$

Böylece eğer hizalama hatası varsa ve tanımlanıp kompanse edilmez ise gerçek açısal çevrim $\omega(t)$ harekete ve ivme sensörleri arasındaki uzaklığa bağlı olmaktadır. Bir sonraki kısımda yerleştirme ve yönlendirme hatalarının nasıl tanımlanabileceği tartışılacaktır.

4.1.1 Yönlendirme Hataları

Yönlendirme hatasının olduğunu varsayarsak, gerçek konum ve yönlendirme, u ve $\theta_r = \theta + \theta_e$ şeklinde olur. Bu hata yerçekimi etkisi kullanılarak bulunabilir. Gövde sabit olmak şartıyla ivme sensörüne etkileyen kuvvet yerçekimidir. Buna göre yönlendirme hatasını gövdeyi her üç ekseninde de döndürmek suretiyle bulunabilir. Bu standart eksenler e_1, e_2, e_3 ve $\ddot{R}_I = 0$, $\Omega = 0$ olduğunda $G = 0$, $L = -F^T g$ ve ivme sensör çıkışı;

$$A_r(u, \theta_r) = -\langle F^T g, \theta_r \rangle \geq |g| \langle F^T e_3, \theta_r \rangle \quad (4.5)$$

denkleme indirgenebilir.

F_1, F_2 ve F_3 gövdenin e_1, e_2 ve e_3 eksenlerine bağlı olarak saat yönünün tersi yönündeki dönüşü olsun, buna bağlı olarak ivme sensör çıkışları A_{r1}, A_{r2} ve A_{r3} olursa buna göre $F_1 e_3 = e_2$, $F_2 e_3 = -e_1$ ve $F_3 e_3 = e_2$ olmaktadır. Bu bilgiler ışığında;

$$A_r^1 = |g| \langle e_2, \theta_r \rangle, A_r^2 = -|g| \langle e_1, \theta_r \rangle, A_r^3 = |g| \langle e_3, \theta_r \rangle \quad (4.6)$$

İvme sensör çıkışlarının yanısıra, gerçek yönlendirme elde edilebilir: $\theta_r = \sum_i \langle e_i, \theta_r \rangle e_i$;

$$\theta_r = \frac{1}{|g|} (-A_r^2 e_1 + A_r^1 e_2 + A_r^3 e_3) \quad (4.7)$$

4.1.2 Yerleştirme Hataları

Eğer yönlendirme hatalarının yanı sıra yerleştirme hataları da varsa ivme sensör konumları ve yönlendirmeleri u_{ri} ve θ_{ri} olacaktır. Yerleştirme hataları gövde döndürülmeden tanımlanamaz. Bundan dolayı ataletsel düzleme bağlı olan gövde yüzeyi düzleminin sabit açısal hızı için üç durum göz önüne alınmalıdır. Her üç durumda da dönüş yerçekimi yönüne paralel olmalıdır. Bu üç durum gövdenin (1) x-z düzlemi, (2) y-z- düzlemi ve (3) x-y düzlemdir. Sabit dönüş hızı için $\ddot{R}_T=0$ ve $\dot{\Omega}=0$ 'dır. Bu durumda ivme sensör çıkışı i için ;

$$A_{ri}^{(j)} = \theta_{ri}^T \hat{F}^{(j)} + \theta_{ri}^T (\Omega^{(j)})^2 u_{ri}, 1 \leq j \leq 3$$

$\Omega^{(j)} \leftrightarrow \omega^{(j)}$ olduğunda;

$$\omega^{(1)} = \omega_0 [\cos\beta \ 0 \ \sin\beta]^T, \hat{F}^{(1)} = |g| [\cos\beta \ 0 \ \sin\beta]^T$$

$$\omega^{(2)} = \omega_0 [0 \ \cos\beta \ \sin\beta]^T, \hat{F}^{(2)} = |g| [0 \ \cos\beta \ \sin\beta]^T$$

$$\omega^{(3)} = \omega_0 [\cos\beta \ \sin\beta \ 0]^T, \hat{F}^{(3)} = |g| [\cos\beta \ \sin\beta \ 0]^T$$

Burada ω_0 sabit açısal dönüş ve $\omega^{(j)}$, (j) durumu için gövde düzlemindeki eksenel dönüşü göstermektedir. Buna göre ivme sensörünün asıl konumu şu şekildedir:

$$u_{ri} = \begin{bmatrix} \theta_{ri}^T (\Omega^{(1)})^2 \\ \theta_{ri}^T (\Omega^{(2)})^2 \\ \theta_{ri}^T (\Omega^{(3)})^2 \end{bmatrix}^{-1} \begin{bmatrix} A_{ri}^{(1)} - \theta_{ri}^T \hat{F}^1 \\ A_{ri}^{(2)} - \theta_{ri}^T \hat{F}^2 \\ A_{ri}^{(3)} - \theta_{ri}^T \hat{F}^3 \end{bmatrix} \quad (4.8)$$

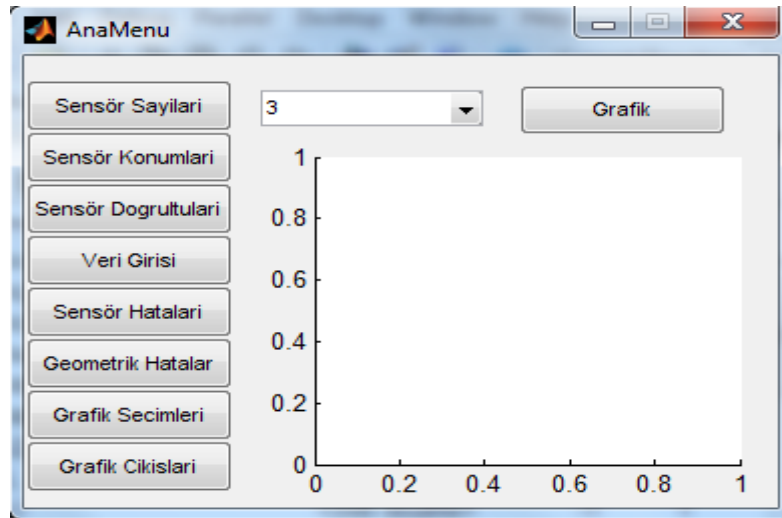
ARAYÜZ TASARIMI

Bu kısımda tezin amacı çerçevesinde hazırlanan MATLAB altında çalışan GUI programıyla hazırlanmış bir arayüz tasarımının çalışma prensibi ve arka planında çalışan program algoritması verilecektir.

5.1 Tasarlanan GUI Programının Çalışma Prensibi ve İçeriği

Tasarlanan programın çalışma prensibi adım adım aşağıdaki gibidir.

1. Ana Menü

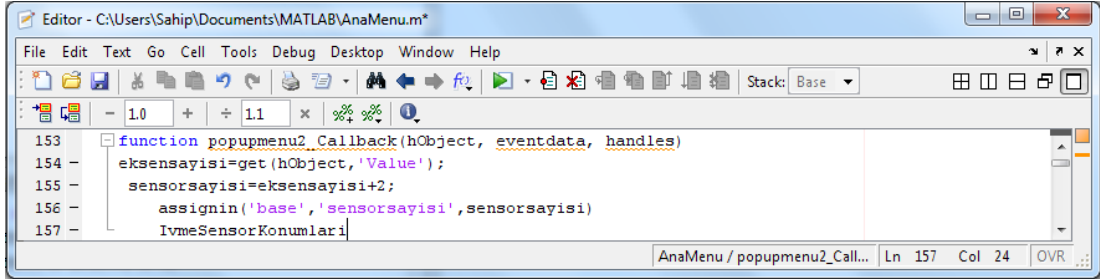


Şekil 5.1 Ana Menü

Şekil 5.1 programın ilk aşaması olup programın ana menüsü görevini görmektedir.

Ayrıca bu pencere programdaki diğer pencerelere kolay erişim sağlamaktadır.

Aynı zamanda ana menünün bir diğer görevi sensör sayısının seçimini sağlamaktır.

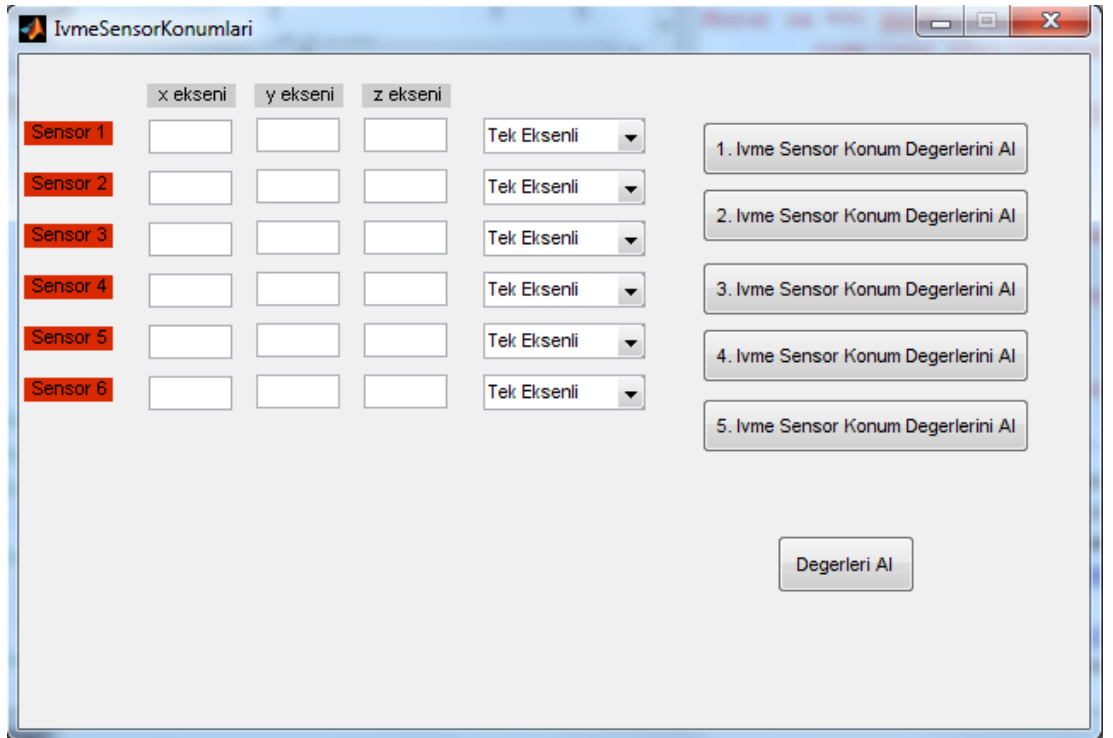


```
153 function popupmenu2_Callback(hObject, eventdata, handles)
154     eksensayisi=get(hObject,'Value');
155     sensorsayisi=eksensayisi+2;
156     assignin('base','sensorsayisi',sensorsayisi)
157     IvmeSensorKonumlar1
```

Şekil 5.2 Ana Menü Komut Penceresi

Şekil 5.2’de görüldüğü üzere ana menüde alınan “sensör sayısı” değeri daha sonraki pencerelerde kullanılmaktadır.

2. İvme Sensör Konumları



	x eksen	y eksen	z eksen		
Sensor 1	<input type="text"/>	<input type="text"/>	<input type="text"/>	Tek Eksenli	1. Ivme Sensor Konum Degerlerini Al
Sensor 2	<input type="text"/>	<input type="text"/>	<input type="text"/>	Tek Eksenli	2. Ivme Sensor Konum Degerlerini Al
Sensor 3	<input type="text"/>	<input type="text"/>	<input type="text"/>	Tek Eksenli	3. Ivme Sensor Konum Degerlerini Al
Sensor 4	<input type="text"/>	<input type="text"/>	<input type="text"/>	Tek Eksenli	4. Ivme Sensor Konum Degerlerini Al
Sensor 5	<input type="text"/>	<input type="text"/>	<input type="text"/>	Tek Eksenli	5. Ivme Sensor Konum Degerlerini Al
Sensor 6	<input type="text"/>	<input type="text"/>	<input type="text"/>	Tek Eksenli	

Degerleri Al

Şekil 5.3 İvme Sensör Konum Penceresi - 1

Ana menüden sensör sayısının seçiminden sonra karşımıza gelen 2. pencere Şekil 5.3’te görüldüğü gibi ivme sensörlerinin konumlarının seçildiği penceredir. Bu pencerede ana

menüden seçilmiş olan ivme sensörü sayısına göre bu ivme sensörlerinin konumları belirlenmektedir. Sağ kısımda görülen düğmeler aracılığı ile daha önceden hazırlanmış konumlandırmalar kullanılabilir.

Bu pencerenin amacı ivme sensörlerinin konumlarının yapılandırma üzerinde belirlenmesidir. Bu sayede yapılandırmanın ana şekli ortaya çıkarılmış olur. GF-INS yapılandırmalarında dikkat edilmesi gereken hususlardan bir tanesi de bu yapılandırmalarda 6 adetten daha az sayıda ivme sensörünün kullanılmasının mümkün olmamasıdır. Bu yönde yapılacak olan denemeler Bölüm 2’de anlatıldığı üzere hatalı sonuçlar verecektir.

	x eksen	y eksen	z eksen	
Sensor 1	0	0	-1	Tek Eksenli
Sensor 2	0	-1	0	Tek Eksenli
Sensor 3	-1	0	0	Tek Eksenli
Sensor 4	1	0	0	Tek Eksenli
Sensor 5	0	1	0	Tek Eksenli
Sensor 6	0	0	1	Tek Eksenli

1. Ivme Sensor Konum Degerlerini Al

2. Ivme Sensor Konum Degerlerini Al

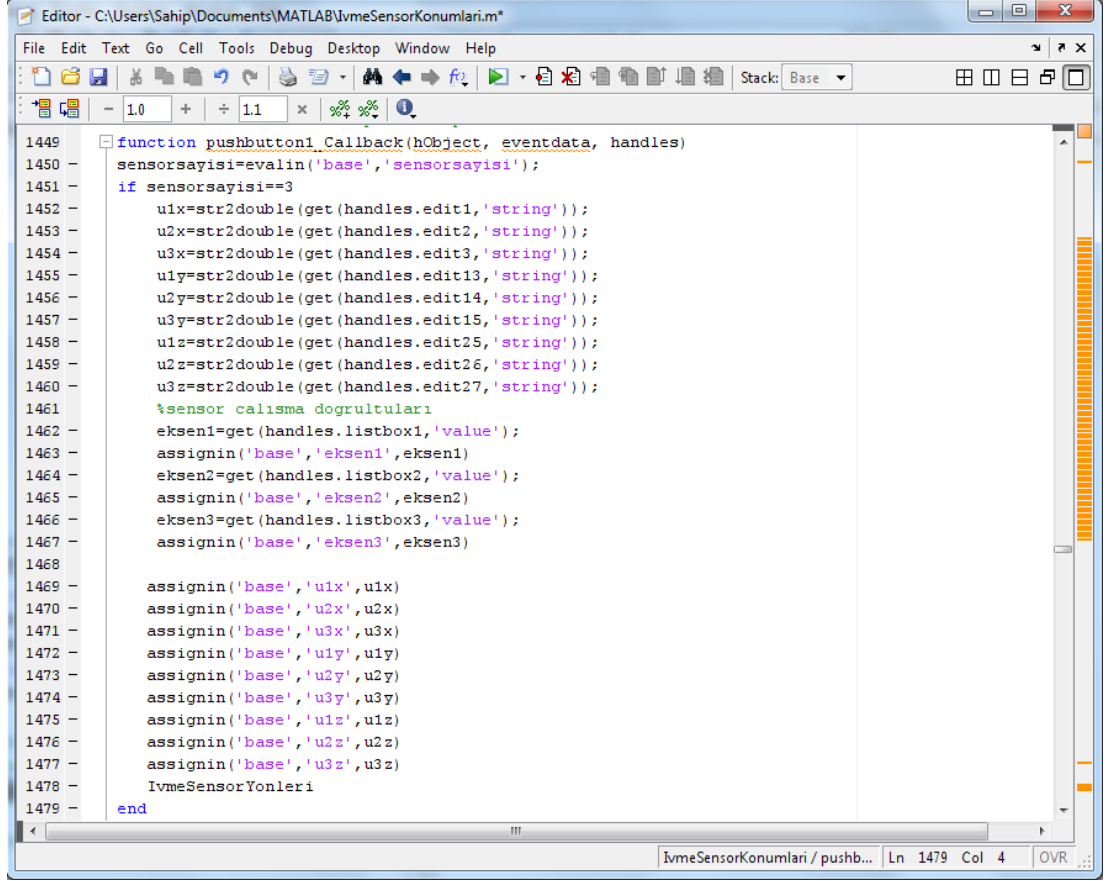
3. Ivme Sensor Konum Degerlerini Al

4. Ivme Sensor Konum Degerlerini Al

5. Ivme Sensor Konum Degerlerini Al

Degerleri Al

Şekil 5.4 İvme Sensör Konum Penceresi - 2



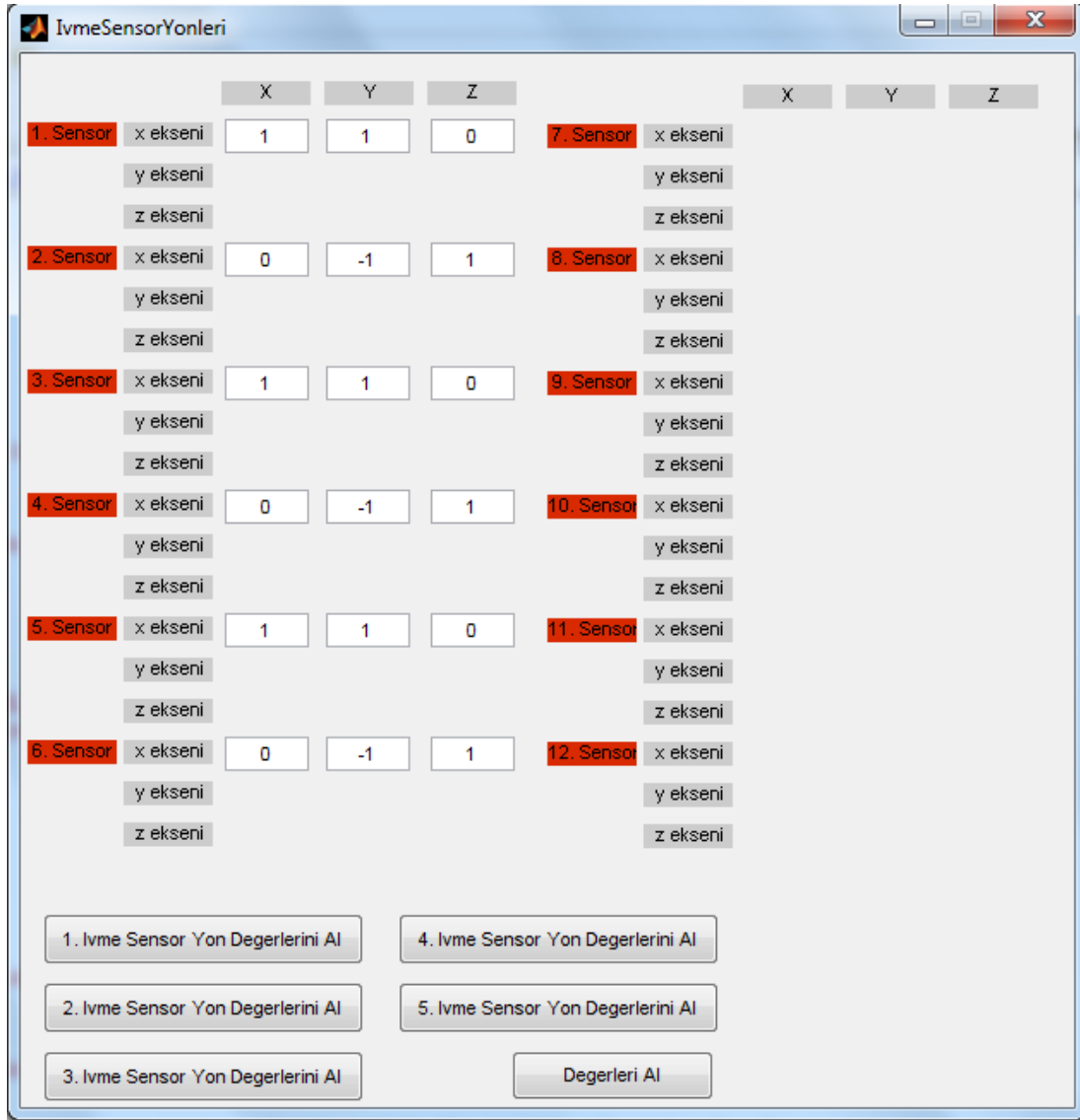
```
1449 function pushbutton1_Callback(hObject, eventdata, handles)
1450     sensorsayisi=evalin('base','sensorsayisi');
1451     if sensorsayisi==3
1452         u1x=str2double(get(handles.edit1,'string'));
1453         u2x=str2double(get(handles.edit2,'string'));
1454         u3x=str2double(get(handles.edit3,'string'));
1455         u1y=str2double(get(handles.edit13,'string'));
1456         u2y=str2double(get(handles.edit14,'string'));
1457         u3y=str2double(get(handles.edit15,'string'));
1458         u1z=str2double(get(handles.edit25,'string'));
1459         u2z=str2double(get(handles.edit26,'string'));
1460         u3z=str2double(get(handles.edit27,'string'));
1461         %sensor calisma dogrultulari
1462         eksen1=get(handles.listbox1,'value');
1463         assignin('base','eksen1',eksen1);
1464         eksen2=get(handles.listbox2,'value');
1465         assignin('base','eksen2',eksen2);
1466         eksen3=get(handles.listbox3,'value');
1467         assignin('base','eksen3',eksen3);
1468
1469         assignin('base','u1x',u1x);
1470         assignin('base','u2x',u2x);
1471         assignin('base','u3x',u3x);
1472         assignin('base','u1y',u1y);
1473         assignin('base','u2y',u2y);
1474         assignin('base','u3y',u3y);
1475         assignin('base','u1z',u1z);
1476         assignin('base','u2z',u2z);
1477         assignin('base','u3z',u3z);
1478         IvmeSensorYonleri
1479     end
```

Şekil 5.5 İvme Sensör Konum Komut Penceresi

Şekil 5.5'te İvme Sensör Konumları komut satırında sensör sayısının 3 seçilmesi durumunda olmaları açıklanan bir dizi komut görüntülenmektedir.

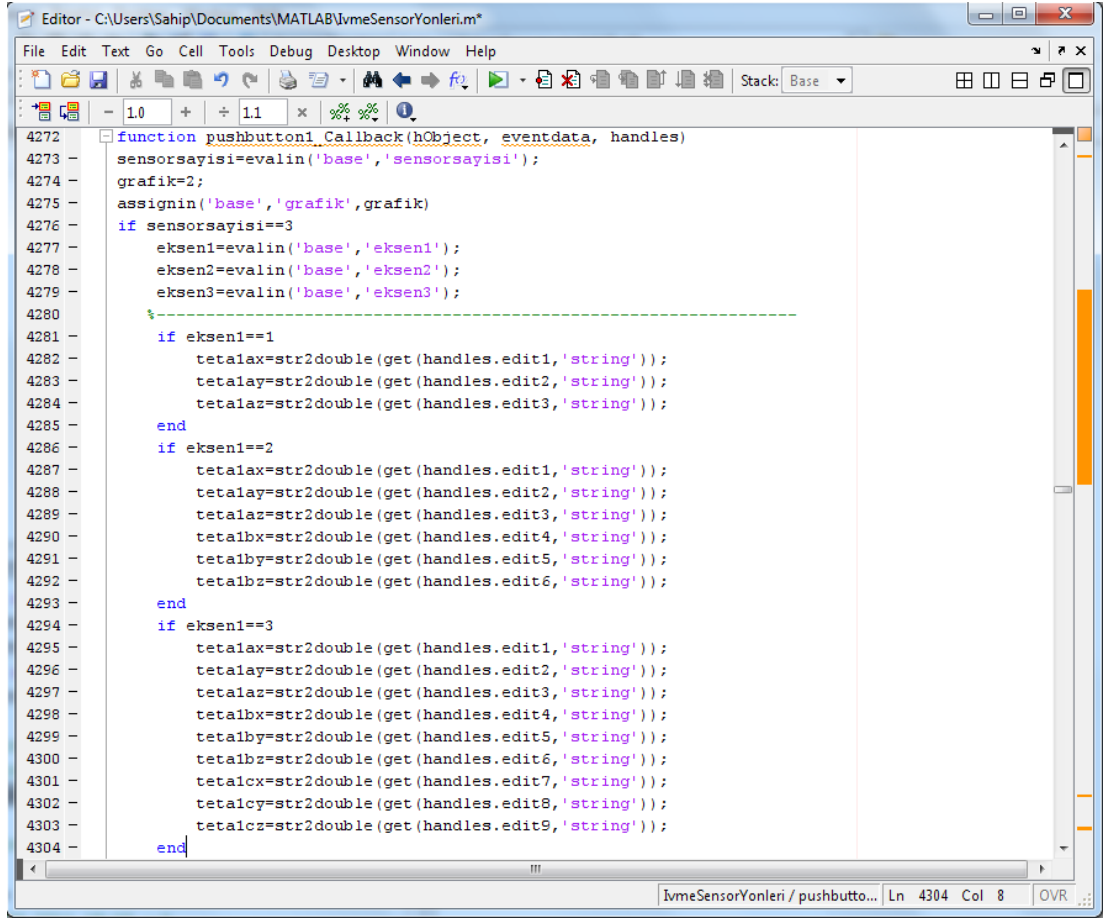
3. İvme Sensör Yönleri

3. aşama olarak ivme sensörlerinin çalışma doğrultularının belirlenmesi gelmektedir. Şekil 5.6'da görülebileceği gibi her bir ivme sensörünün çalışma doğrultusu seçilebilmektedir.



Şekil 5.6 İvme Sensör Çalışma Doğrultu Seçim Penceresi

Şekil 5.7’de bulunan komut satırında gösterildiği üzere 3. adımda ivme sensör çalışma doğrultuları “teta” türevleri olarak MATLAB programının veri tutucusu olarak çalışan workspace’te saklanmaktadır.



```
4272 function pushbutton1_Callback(hObject, eventdata, handles)
4273     sensorsayisi=evalin('base','sensorsayisi');
4274     grafik=2;
4275     assignin('base','grafik',grafik)
4276     if sensorsayisi==3
4277         eksen1=evalin('base','eksen1');
4278         eksen2=evalin('base','eksen2');
4279         eksen3=evalin('base','eksen3');
4280         %-----
4281         if eksen1==1
4282             tetalax=str2double(get(handles.edit1,'string'));
4283             tetalay=str2double(get(handles.edit2,'string'));
4284             tetalaz=str2double(get(handles.edit3,'string'));
4285         end
4286         if eksen1==2
4287             tetalax=str2double(get(handles.edit1,'string'));
4288             tetalay=str2double(get(handles.edit2,'string'));
4289             tetalaz=str2double(get(handles.edit3,'string'));
4290             teta1bx=str2double(get(handles.edit4,'string'));
4291             teta1by=str2double(get(handles.edit5,'string'));
4292             teta1bz=str2double(get(handles.edit6,'string'));
4293         end
4294         if eksen1==3
4295             tetalax=str2double(get(handles.edit1,'string'));
4296             tetalay=str2double(get(handles.edit2,'string'));
4297             tetalaz=str2double(get(handles.edit3,'string'));
4298             teta1bx=str2double(get(handles.edit4,'string'));
4299             teta1by=str2double(get(handles.edit5,'string'));
4300             teta1bz=str2double(get(handles.edit6,'string'));
4301             teta1cx=str2double(get(handles.edit7,'string'));
4302             teta1cy=str2double(get(handles.edit8,'string'));
4303             teta1cz=str2double(get(handles.edit9,'string'));
4304         end
```

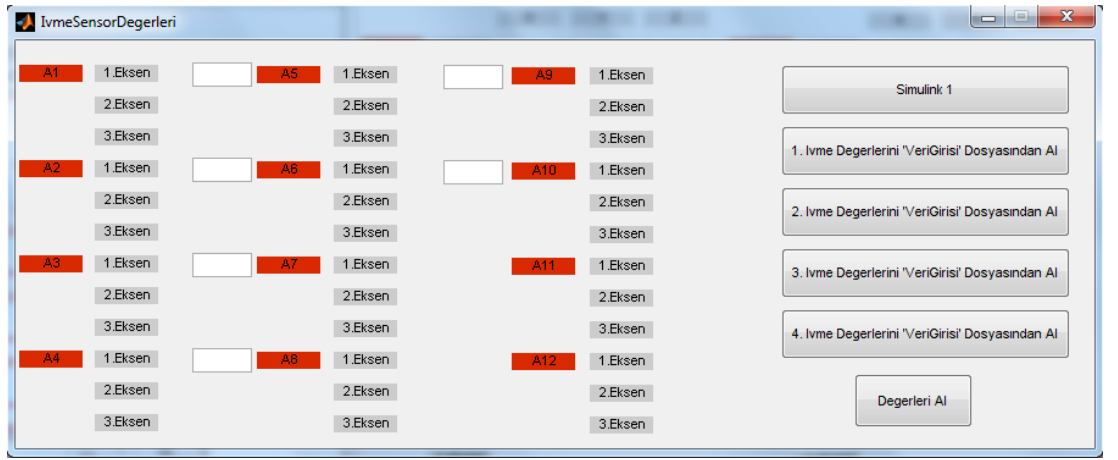
Şekil 5.7 İvme Sensör Çalışma Doğrultu Seçimi Komut Penceresi

4. Ana Menü (2)

Programın bu kısmında yapılan bütün seçimler için her bir ivme sensörünün konumu ve çalışma doğrultusu Şekil-5.8 üzerindeki grafikte görülebilmektedir. Bahsedilen grafik ana menü üzerindeki “Grafik” düğmesi kullanılarak çizilmektedir.

5. İvme Sensör Değerleri

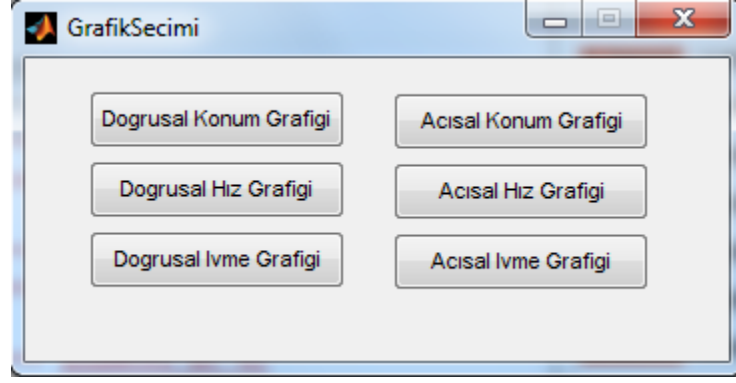
İvme sensörlerinin konumlandırılmalarının ve yönlendirmelerinin doğruluğu kontrol edilip onaylandıktan sonra programa ivme sensörlerinin algıladığı ivme giriş bilgilerinin verilmesi gerekmektedir. Şekil 5.10'da görülebileceği gibi bu işlem hem simülink hemde daha önce hazırlanmış hazır veriler üzerinden yapılabilecek şekilde oluşturulmuştur.



Şekil 5.10 İvme Sensör Değerleri Penceresi

6. Grafik Seçimi

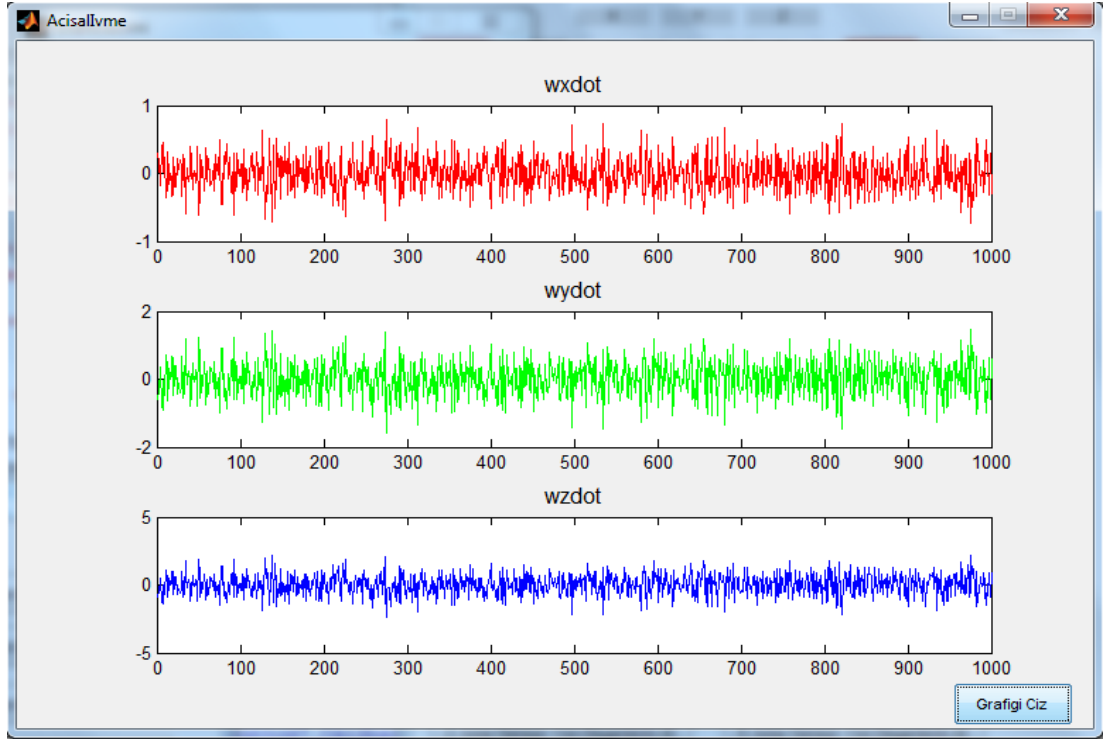
Bütün girişler gerçekleştirildikten sonra ilk önce "Açısal İvme" düğmesi kullanılarak gerekli hesaplamaların MATLAB tarafından yapılması sağlanmalıdır. Bu işlemden sonra Şekil 5.11'de görülebileceği gibi GF-INS'in açısal ivmesi, açısal hızı, açısal konumu ve doğrusal ivmesi, doğrusal hızı, doğrusal konumu elde edilebilmektedir.



Şekil 5.11 Grafik Seçim Penceresi

7. Grafikler

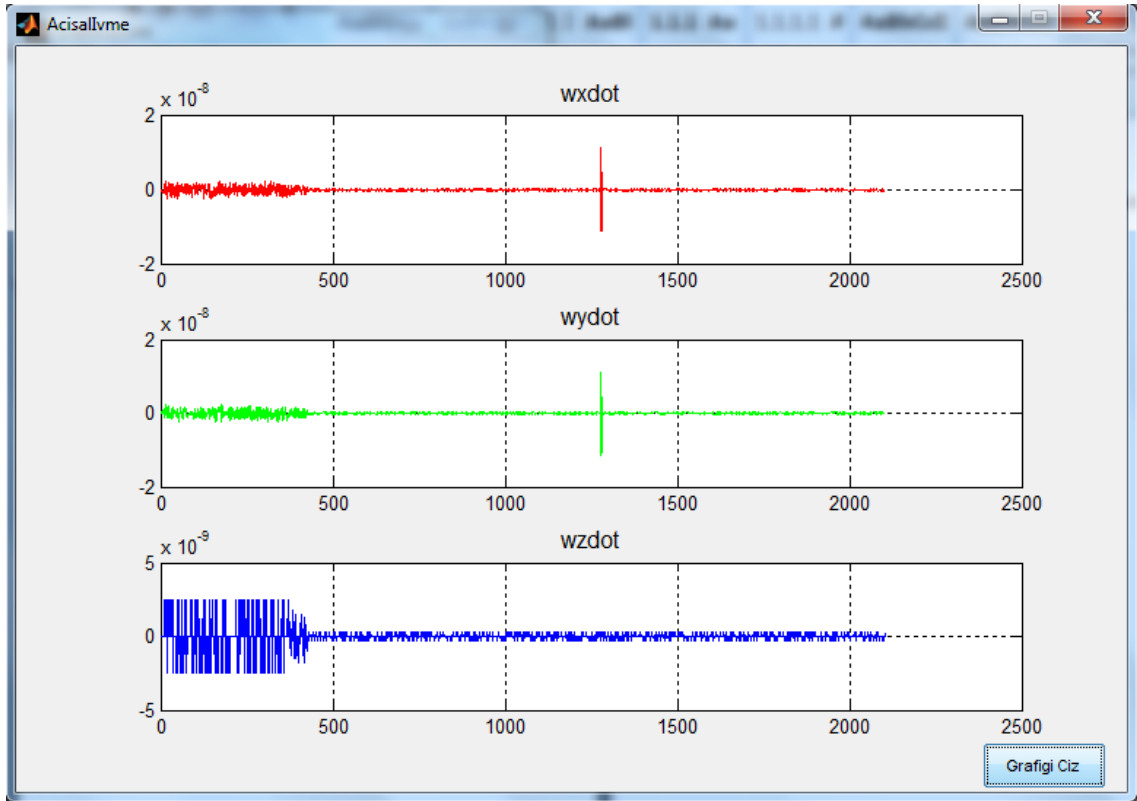
JB-ASS programına, ilk test olarak hem sistemin kararlılığını test etmek amacıyla hem de açısıl dönüşlerden kaçınılacak şekilde ivme girişleri yapılarak sistemin doğruluğunu test etmek amacıyla beyaz gürültü verileri simulink vasıtasıyla verilmiştir. Bu veriler verilirken bir deneme sırasında alınmış olan açısıl ivme grafiklerinin ekran görüntüsü Şekil 5.12’de görülmektedir.



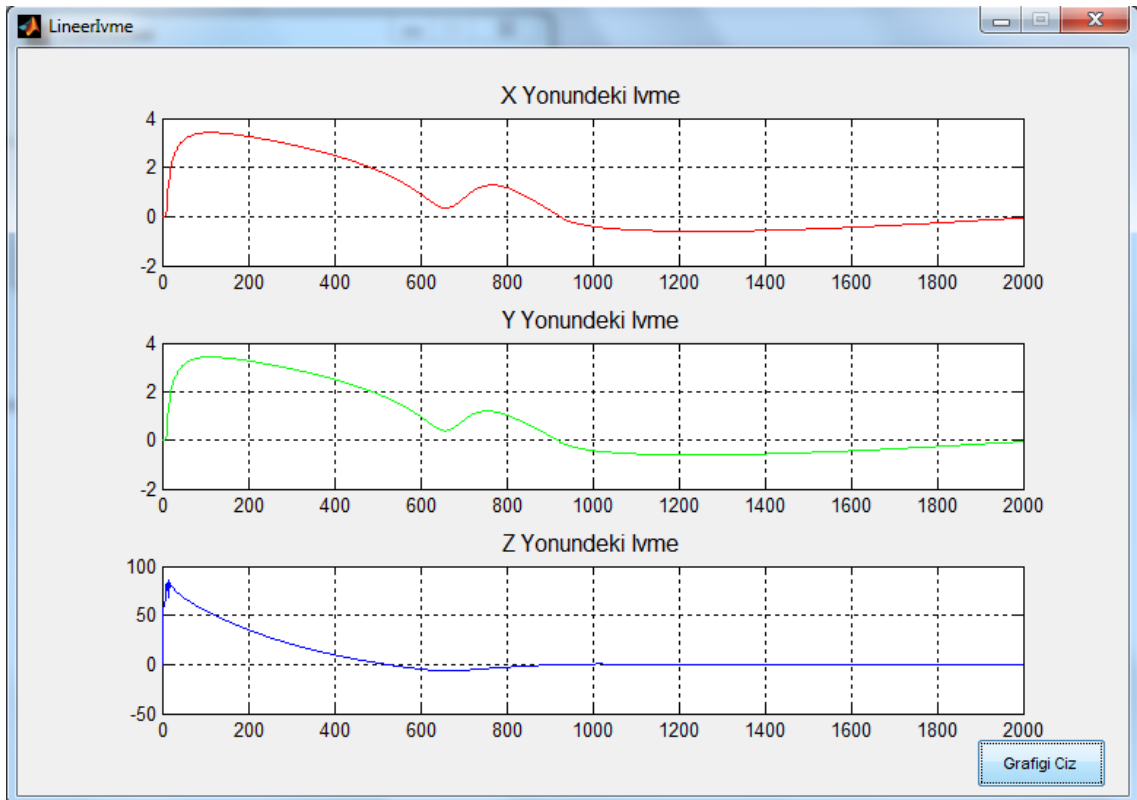
Şekil 5.12 İvme Grafik Penceresi

5.2 GF-INS Tasarım Örneği ve Rota Tasarımı

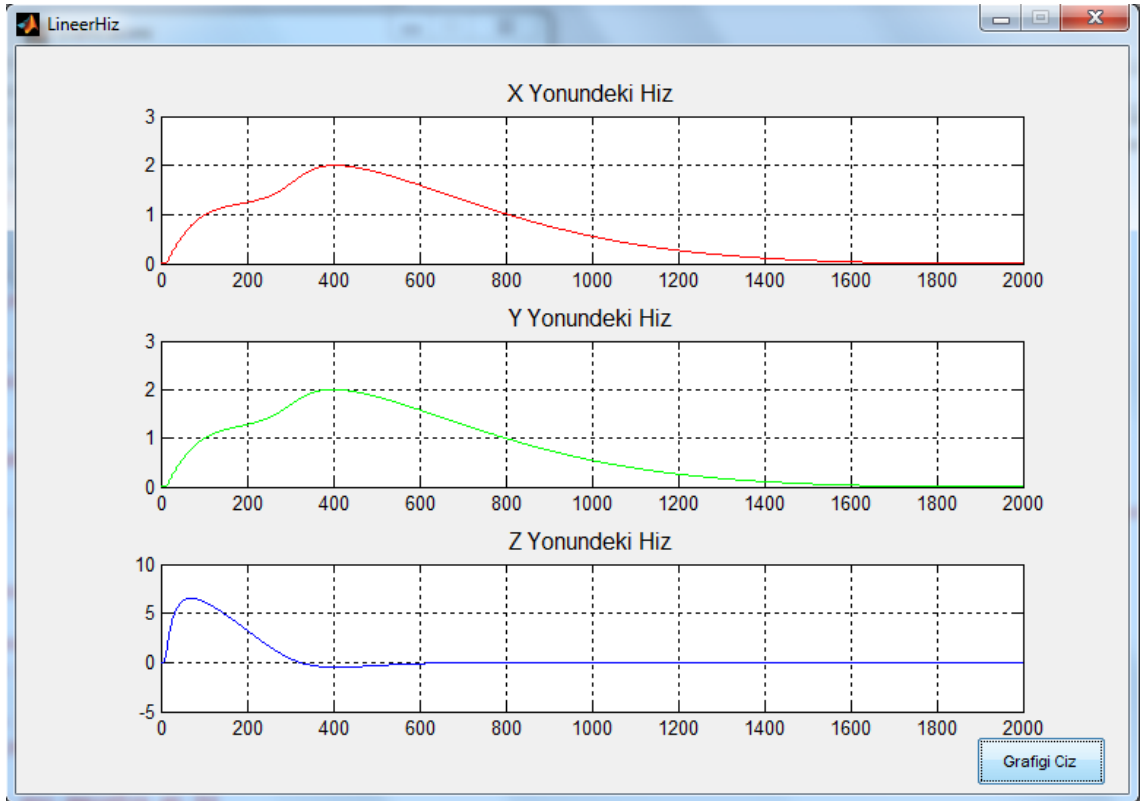
Aşağıda görülmekte olan örnekte bir hava aracı modelinden alınmış ivme değerlerinden yola çıkılarak bu hava aracının konumunun belirlenmesi çalışması yapılmıştır. Hazırlanan programa hava aracının üzerinde bulunduğu varsayılan JB-ASS'den alınan ivme veri beslemesi yapılmış ve bu ivme verileri kullanılarak aracın hem açısal hızı ve konumu elde edilmiştir. Şekil 5.13'te görülmekte olduğu gibi açısal ivme ihmal edilebilecek değerlerde olduğundan dolayı grafik gösteriminde lineer hız ve lineer konum grafiklerine yoğunlaşmıştır. Programdan istendiği takdirde açısal hız ve konum grafikleri de elde edilebilmektedir.



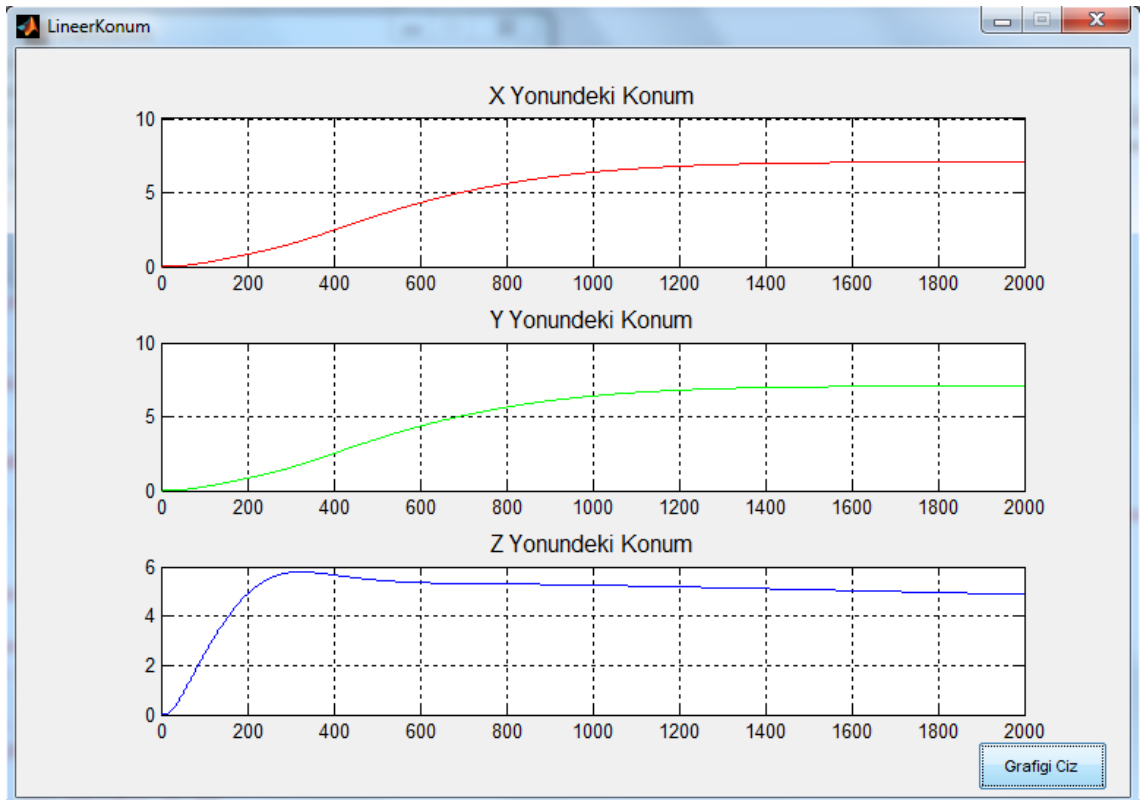
Şekil 5.13 Açısall İvme Grafiği



Şekil 5.14 Doğrusal İvme Grafiği



Şekil 5.15 Doğrusal Hız Grafiği



Şekil 5.16 Doğrusal Konum Grafiği

Şekil 5.14-15 ve 16'da verilen ivme sensör bilgileri ve ivme değerleri için JB-ASS'in programdan elde edilmiş olan ivme, hız ve konum grafikleri görülmektedir. Aynı zamanda elde edilen bu veriler, ivme sensörlerinden ölçülmüş olan ivme değerleri vasıtasıyla bulunmuş olan JB-ASS'nin üzerine yerleştirileceği aracın ivme, hız ve konum değerlerini de ifade etmektedir.

BÖLÜM 6

SONUÇ VE ÖNERİLER

Bu çalışma boyunca JB-ASS'nin çalışma prensiplerinden ve ivme sensörlerinin özelliklerinden bahsedilmiş, daha sonra hazırlanmış olan GUI programı yardımıyla bir JB-ASS yapılandırması oluşturulmuştur. Bu program için bir hava aracı modeli rotası üzerinden ivme sensör verileri elde edilmiş ve bu veriler kullanılarak hava aracının her anındaki hız ve konum verileri bulunmuştur. Hazırlanmış olan bu program kendisinden istenilen görevleri yerine getirebilecek bir program olmakla beraber geliştirilmeye açık bir programdır. Bir sonraki aşama olarak bu program için Kalman filtresi tasarımının gerçekleştirilmesi ve ayrıca ivme sensör hatalarının giderilebileceği bir grafik arayüzünün tasarlanması düşünülmektedir. En son aşama olarak ise gerçek bir JB-ASS tasarımı yapılarak çalışır hale getirilmesi ve testlerinin bu program aracılığıyla yapılması hedeflenmektedir.

KAYNAKLAR

- [1] Zhou J. Q., (2006), *Simplified Analysis of IMU Sensor Corruptions on Existing Pendulation Control System For Ship-Mounted Crane* , Master's Thesis, Virginia Polytechnic Institute and State University.
- [2] White A., (2002), *A Review of Some Current Research in Microelectromechanical Systems (MEMS) with Defence Applications*, Published by DSTO Aeronautical and Maritime Research Laboratory, Australia.
- [3] Lijung P. B, (1997), *Micromachined Gyroscopes With Integrated Electronics*, Doctoral Thesis, University of California, Berkley.
- [4] Patil N., (2006), *Design and Analysis of MEMS Angular Rate Sensors* , Master Thesis, Indian Institute of Science.
- [5] Gebre-Egziabher D., Elkaim G. H., Powell J. D. ve Parkinson B. W., (2000), "A Gyro-Free Quaternion-Based Attitude Determination System Suitable for Implementation Using Low Cost Sensors", *IEEE Position Location and Navigation Symposium*, 185-192.
- [6] Ornedo R. S., Farnsworth K. A. ve Sandhoo G. S., (1998), "GPS and Radar Aided Inertial Navigation Systems for Missile System Applications", *IEEE Position Location and Navigation Symposium*, 614-621.
- [7] Mandapat R. E., (2001), *Development And Evaluation Of Positioning Systems For Autonomous Vehicle Navigation* , Master Thesis, University Of Florida.
- [8] Mostov K. S., Soloviev A. A. ve Koo T. K. J., (1997), "Accelerometer Based Gyro-Free Multi-Sensor Generic Inertial Device for Automotive Applications" ., *IEEE Conference on Intelligent Transportation System*, 1047-1052.
- [9] Tan C., Mostov K. ve Varaiya P., (2000), "Feasibility of a gyroscope-free inertial navigation system for tracking rigid body movement", *PATH Research Report, UCB-ITS-PRR-2000-9*.
- [10] Tan C. ve Park S., (2002), "Design and error analysis of accelerometer-based inertial navigation systems", *PATH Research Report, UCB-ITS-PRR-2002-22*.

- [11] Xiao B., Xu C. ve Xu L., (2009), "An Application of MEMS Sensors in Inertial Navigation System", *Proceedings of the 2009 International Conference on Industrial and Information Systems*, 00:360-363.
- [12] Li Q., Zhang W., Zhang H. ve Xu W., (2006), "Attitude measurement system based on micro-silicon accelerometer array", *Chaos, Solitons & Fractals*, 29(1):141-147.
- [13] Buhmann A., Peters C., Cornils M. ve Manoli Y., (2006), "A GPS aided Full Linear Accelerometer Based Gyroscope-free Navigation System" *IEEE/ION Position, Location, And Navigation Symposium*, 622- 629.
- [14] Niu X., Goodall C., Nassar S. ve El-Sheimy N. , (2006), "An Efficient Method for Evaluating the Performance of MEMS IMUs", *Position, Location, And Navigation Symposium, IEEE/ION*, 766- 771.
- [15] Kamar N. S. ve T. Jann, (2004), *Estimation of attitudes from a low-cost miniaturized inertial platform using Kalman Filter-based sensor fusion algorithm*, Published by Springer India, in co-publication with Indian Academy of Sciences.
- [16] Shu-zhi M., Xiong-zhu B. ve Yong-xin L. , (2006), "Optimization Design and Calibration of Installation Error Coefficients for Gyroscope-Free Strapdown Inertial Measurement Unit" *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, 238-242.
- [17] Akeila E., Salcic Z. ve Swain A., (2008), "Implementation, Calibration and Testing of GFINS Models Based on Six-Accelerometer Cube", *IEEE Region 10 Conference, TENCON 2008*, 1-6.
- [18] Quin F., Xu J. ve Jiang S., (2009), "A New Scheme of Gyroscope Free Inertial Navigation System Using 9 Accelerometers", *ISA 2009 Conference on Intelligent Systems and Applications*.
- [19] Li A., Qin F., Xu J. ve Jiang S., (2009), "Gyroscope Free Strapdown Inertial Navigation System using rotation modulation", *Second International Conference on Intelligent Computation Technology and Automation*, 3:611-614.
- [20] Xudan J., Zheng M. , Fuling W. ve Yali W., (2007), "Research on Gyroscope Free Strapdown Inertial Navigation System Based on 3-axis accelerometer", *8th International Conference on Electronic Measurement and Instruments*, 1-967-1-970.
- [21] Cardou P. ve Angales J., (2005), "A Comparative Study of All-Accelerometer Strapdowns for UAV INS", *Conference Paper, McGill University, RTO-MP-SET-092*.
- [22] Nakamura S., (2005), "MEMS Inertial Sensor Toward Higher Accuracy and Multi-axis Sensing", *IEEE Sensors*, 939-942.
- [23] Akeila E., Salcic Z. ve Swain A., (2008), "Implementation, Calibration and Testing of GFINS Models Based on Six-Accelerometer Cube", *IEEE Region 10 Conference, TENCON 2008*, 1-6.

GUI PROGRAM ÖRNEĞİ

MATLAB, teknik hesaplamalar ve matematiksel problemlerin çözümü ve analizi için tasarlanmış bir yazılım geliştirme aracıdır. “MATrix LABoratory” kelimesinin kısaltması olan MATLAB, adında da anlaşılacağı üzere matrisler (matrix) yani diğer bir deyişle diziler (array) ile çalışır. Özellikle mühendislik alanındaki sistemlerin analizinde kullanılan MATLAB, görüntü işleme (image processing), yapay sinir ağları (artificial neural networks), sayısal işaret işleme (signal processing), optimizasyon (optimization), veri elde etme (data acquisition), veritabanı (database), süzgeç tasarımı (filter design), bulanık mantık (fuzzy logic), sistem kimliklendirme (system identification), dalgacıklar (wavelets) gibi araçları ile herkez için mükemmel bir ortam sunar.

MATLAB'in nasıl bir yazılım olduğunu anlamak için onu çok gelişmiş özellikleri olan, programlanabilen bir bilimsel hesap makinesine benzetebiliriz. MATLAB'de yazılan programlar, MATLAB'in kendine özgü dili kullanılarak yazılır ve MATLAB içinde çalıştırılır. Ayrıca yazılan programlar DLL ve EXE olarak oluşturulabildiği gibi C/C++ kodlarına da çevirilebilir.

Problemler MATLAB'de komut satırında çalışan programlar yazılarak çözülebildiği gibi MATLAB GUI geliştirme aracını da kullanarak, formlar ve butonlar gibi nesnelere oluşan görsel yazılımlar geliştirilebilir.

MATLAB'in şu an için Windows ve Linux ortamlarında çalışan versiyonları mevcuttur.

MATLAB ile ;

Veri elde etme

Veri analizi ve inceleme

Görsellik ve görüntü işleme

Algoritma prototipi oluşturma ve geliştirme

Modelleme ve simülasyon

Programlama ve uygulama geliştirme yapılabilir. Bugün 500.000 den fazla akademisyen, araştırmacı, bilim adamı ve öğrenci tarafından kullanılan MATLAB, içinde gömülü pek çok ara yüzü ile bilgisayar dünyasının en gelişmiş teknik ve bilimsel problem çözme ve uygulama geliştirme aracıdır.

MATLAB'de geliştirdiğimiz yazılımlar yordam veya fonksiyon şeklindedir. Konsolda edit yazıp <enter> tuşuna bastığımızda MATLAB editörü açılacaktır.

Prosedürler;

Düz yazı şeklinde yazılan komutlardır. Arka arkaya çalıştırmak istenilen komutlar yazılır. MATLAB'de yordamlar ve fonksiyonlar M-dosyaları (M-files) olarak adlandırılır ve .m uzantısını alır.

Örneğin aşağıdaki komutlar editörde yazılıp hesapla.m ismi ile diske kaydedilirse, MATLAB içinden bu dosyayı çalıştırmak için hesapla yazıp <enter> tuşuna basmak yeterli olacaktır.

```
hesapla.m
```

```
ort=sum(sum(a));
```

```
hiz=x/t;
```

```
sonuc=ort*hiz;
```

Yalnız burada dikkat edilmesi gereken nokta, MATLAB içinden bu dosya çalıştırılmak istendiğinde, dosyanın çalışma dizininde (work directory) olması gerekir. Çalışma dizinini görüntülemek için cd komutu kullanılır.

```
**** cd
```

C:\matlabR12\work

Çalışma dizinini değiştirmek için de cd komutundan faydalanılır. Örneğin;

**** cd c:\calismadizininim

Ayrıca dikkat edilmesi gereken bir diğer nokta da, M-dosyası çalışırken bazı değişkenlere başvuruyorsa bunların daha önceden tanımlanmış ve değerleri atanmış olması gerekir. Yukarıdaki örnekte `ort=sum(sum(a))` satırında a isminde bir matrisin toplamı hesaplanıp ort değişkenine atanmaktadır. Ancak program çalıştırılmak istendiğinde eğer a dizisi yoksa, M-dosyası hata verecektir. Yukarıdaki M-dosyası çalışabilmek için, x ve t değişkenlerine de ihtiyaç duymaktadır.

M-dosyası çalıştırıldıktan sonra, M-dosyası içinde tanımlanan değişkenler MATLAB çalışma ortamında (workspace) kullanılmaya başlanır.

O an için tanımlanmış değişkenleri görmek için whos komutu kullanılabilir. Her hangi bir değişkeni silmek için `clear degisken_ismi` şeklinde clear komutu kullanılır. MATLAB ortamındaki tüm değişkenleri silmek için clear komutu tek başına kullanılabilir.

Fonksiyonlar

Diğer programlama dillerinde olduğu gibi MATLAB’de de fonksiyon tanımlamak mümkündür. Fonksiyonlar içine değer alabilen ve bir sonuç döndüren program parçalarıdır.

Örneğin MATLAB editöründe,

```
function donendeger=toplakare(x,y)
```

```
%Bu fonksiyon x ve y nin toplamının karesini hesaplar
```

```
%Kullanım : toplakare(3,5) şeklindedir.
```

```
toplam=x+y;
```

```
donendeger=toplam^2;
```

şeklinde bir fonksiyon yazıp, toplakare.m dosyası şeklinde kaydetelim. Fonksiyondaki ilk satır, fonksiyonun isminin toplakare olduğunu, “donendeger” değişkeni ile sonucun

kullanıcıya aktarılacağı, x ve y olarak iki değişkeni dışardan aldığını belirtir. İkinci ve üçüncü satırlar yorum satırlarıdır. Derleyici tarafından icra edilmezler. Fonksiyonlarda ilk satırdan sonra kullanılan yorum satırları, yazılan m-dosyasının yardım metnini oluşturur. Konsolda help toprakare yazıp <enter> tuşuna basarsak, yukarda % işareti ile başlayan iki satır görüntülenecektir. En son satırda “donendeger” değişkenine hesaplanan ifade atanır ve bu şekilde kullanıcıya sonuç yollanmış olur.

MATLAB’de `a=toplakare(3,5)` yazarak, a değişkenine fonksiyondan dönen 64 değeri atanmış olur.

Temel Matematiksel Operatörler

MATLAB’da diziler arası matematiksel operatörler aşağıdaki gibidir. Ancak dikkat edilmesi gereken bazı noktalar vardır. Örneğin iki matris toplanacağı zaman, matrislerin aynı boyutlarda olması gerektiğidir. 3x3 tipindeki bir matris ile 2x2 tipindeki bir matris toplanamaz. Ancak 1x1 tipindeki skaler değerlerde bu problem yoktur. Örneğin 5x5 tipindeki bir matrisin tüm elemanlarına 2 eklemek için `a+2` şeklinde bir komut kullanılabilir.

Toplama

`a+b`

Çıkarma

`a-b`

Çarpma

`a*b`

Bölme

`a/b`

Üs alma

`a^b`

Karşılaştırma Operatörleri

< Küçük

<= Küçük eşit

> Büyük

>= Büyük eşit

== Eşit

~= Eşit değil

Mantıksal Operatörler

And & Ve

Or | Veya

Not ~ Değil

Şartlı Deyimler ve Döngüler

Program içinde bir değişkenin içeriğine bağlı olarak program akışı değiştirilmek isteniyorsa şartlı deyimler kullanılabilir. En çok kullanılan şartlı deyim if-else-end komutudur. Örneğin;

```
if a>5
```

```
disp('a besten büyüktür');
```

```
x=1;
```

```
else
```

```
disp('a besten kücüktür');
```

```
x=0;
```

```
end
```

şeklinde bir M-dosyası olsun. Eğer a değişkenin değeri 5 den büyükse, konsola 'a besten büyüktür' ifadesi yazdırılacak ve x değişkenine 1 değeri atanacaktır. Eğer a, 5'ten küçük veya eşitse konsola 'a besten kücüktür' ifadesi yazdırılacak ve x değişkenine 0 değeri atanacaktır.

En çok kullanılan bir döngü çeşidi olan for döngüsü aşağıdaki gibi tanımlanır.

```
top=0;
```

```
for i=1:100
```

```
top=top+i;
```

```
end
```

1'den 100'e kadar olan sayıların toplamını bulan bu döngüde i döngü indeksidir. 1:100 ifadesi ile 1'den başlayıp 100'e kadar, döngünün birer artması sağlanmıştır.

Bir diğer döngü tanımlaması while komutu ile yapılabilir. Yukarıdaki örneği while komutu ile şu şekilde yapılabilir.

```
top=0;
```

```
i=0;
```

```
while i<100
```

```
top=top+i;
```

```
i=i+1;
```

```
end
```

Grafikler

MATLAB'de veriler ile çalışırken bu verilerin görüntülenmesi gerekebilir. Analizde çok önemli olan grafiklerin oluşturulması için plot komutu ve surf komutu kullanılabilir.

Plot komutu ile $1 \times N$ tipindeki vektör biçimindeki bir veya daha fazla veri için grafik oluşturulabilir.

Örneğin;

```
t=-pi:0.1i;
```

```
y=sin(t);
```

```
plot(y)
```

komutlarını yazarak -3.14 den +3.14'e kadar 0.1 aralıklarla artan bir vektör tanımlanırsa, Sin fonksiyonu ile bu vektörün tüm değerlerinin sinüslerini hesaplayarak plot komutu ile ekrana çizdirelim. (pi ön tanımlı bir değişken olup değerini atamanıza gerek yoktur)

İki vektörü aynı pencere içinde üst üste görüntülemek istersek $1 \times N$ vektörleri $N \times 1$ tipine çevirmek ve bunları yan yana sütun şeklinde yerleştirmek gerekir. Örneğin yukardaki örnekte t ve y 'yi aynı anda görmek için `plot([t' y'])` komutu kullanılmalıdır.

Plot komutu ile ilgili daha fazla bilgi elde etmek için MATLAB konsolda `help plot` yazılabilir.

MATLAB'de $M \times N$ tipindeki matrislerin 3 boyutlu topografik grafiklerini incelemek için `surf` veya `mesh` komutları kullanılır.

```
**** a=randn(20);
```

```
**** surf(a);
```

Yukarıdaki komutlar ile 20×20 tipinde, elemanları 0-1 arasında rasgele değer alan bir verinin 3 boyutlu grafiği çizdirilmiştir.

Bazı Temel Fonksiyonlar ve Komutlar

MATLAB'de kullanılan bazı temel fonksiyonlar ve açıklamaları verilmiştir.

Clear

Workspace'deki değişkenleri siler

Whos

Workspace'deki değişkenleri listeler

Clc

Ekranı temizler

Disp

Ekrana mesaj yazdırır

Sum

Matris elemanları toplamı

Mean

Matris elemanları ortalaması

Max

Matris elemanlarının en büyük değeri olanı

Min

Matris elemanlarının en küçük değeri olanı

Size

Matrisin boyutları

Sqrt

Karekök alma

Cos

Kosinüs Fonksiyonu

Sin

Sinüs Fonksiyonu

Tan

Tanjant Fonksiyonu

Exp

Eksponansiyel (e)

Log

Doğal algoritma (ln)

Log10

10 tabanında logaritma

Abs

Mutlak değer

Round

Yuvarlama

Floor

Tam değer

Load

Workspace'e diskten veri yükleme

Save

Workspace'deki veriyi diske kaydetme

Strcat

String'leri birleştirme

Break

Döngüden çıkma

Rand

0-1 arasında rasgele sayı üretme

Edit

MATLAB Editörü

Plot

2boyutlu grafik çizme

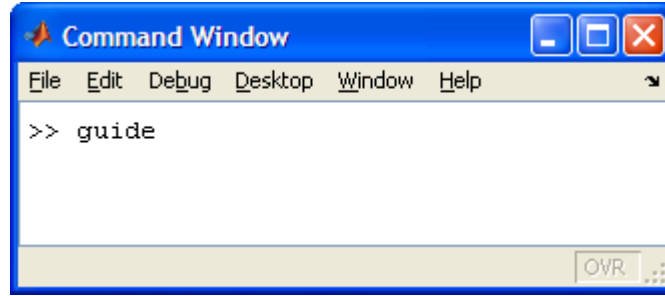
GUI programının MATLAB'de kullanılmasının ana nedeni hazırlanan programın kullanıcılar tarafından kolay ve rahat bir şekilde kullanılmasına olanak sağlamasıdır. GUI, kullanıcıyı komut penceresinin karmaşık ve zor kullanımından kurtararak, görsel araçlar ve pencereler sayesinde programın kullanımını kolaylaştırır. Neredeyse her gün kullandığımız internet tarayıcılar GUI programlarıdır ve bu programların komut penceresi sayesinde kullanılmasını gözönüne getirdiğimizde GUI'nin sağladığı faydaları anlamamız daha kolay olacaktır.

GUI kullanımında nasıl bir program yaratılmak istendiğinde göre bilmemiz gereken komutlar değişiklik gösterir. Basit bir GUI programı yaratımında MATLAB bilgisine dahi gerek duyulmaz iken, karmaşık ve çok pencere etkileşimli GUI programlarında bilmemiz gereken komut sayısı ve bilgisi önem arzeder.

GUI kullanımının bilinmesi bu tezde yazılan programın anlaşılması için gereklidir. Bu nedenden dolayı bir sonraki kısımda bir GUI programının oluşturulmasından bahsedilecektir.

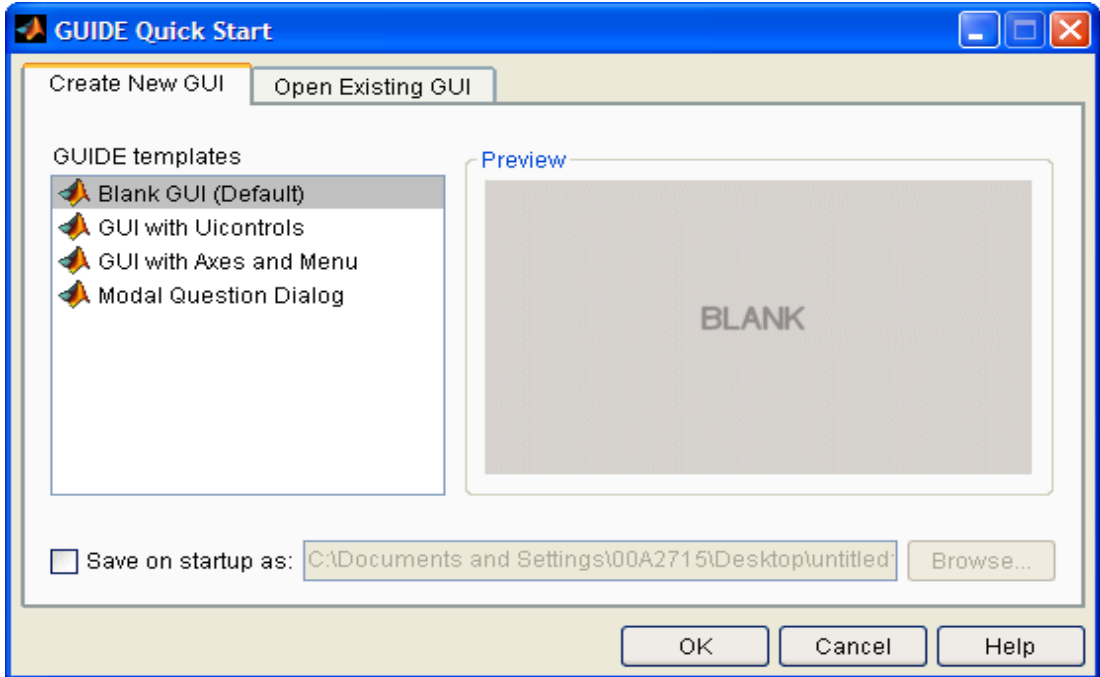
GUI'ye Giriş;

1. MATLAB programında komut satırına "guide" komutunu yazmamız GUI programını başlatmamız için yeterli olacaktır.
2. Şekil Ek.1'de klasik bir komut penceresi görülmektedir.



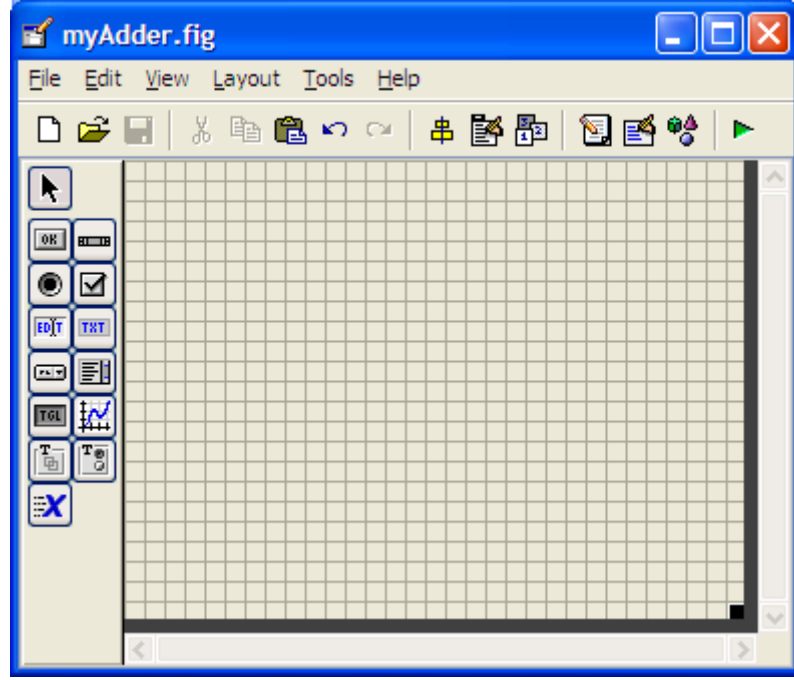
Şekil Ek.1 Komut Penceresi

3. Yeni bir GUI penceresi için ilk seçenek seçilmelidir. [Blank GUI (Default)]



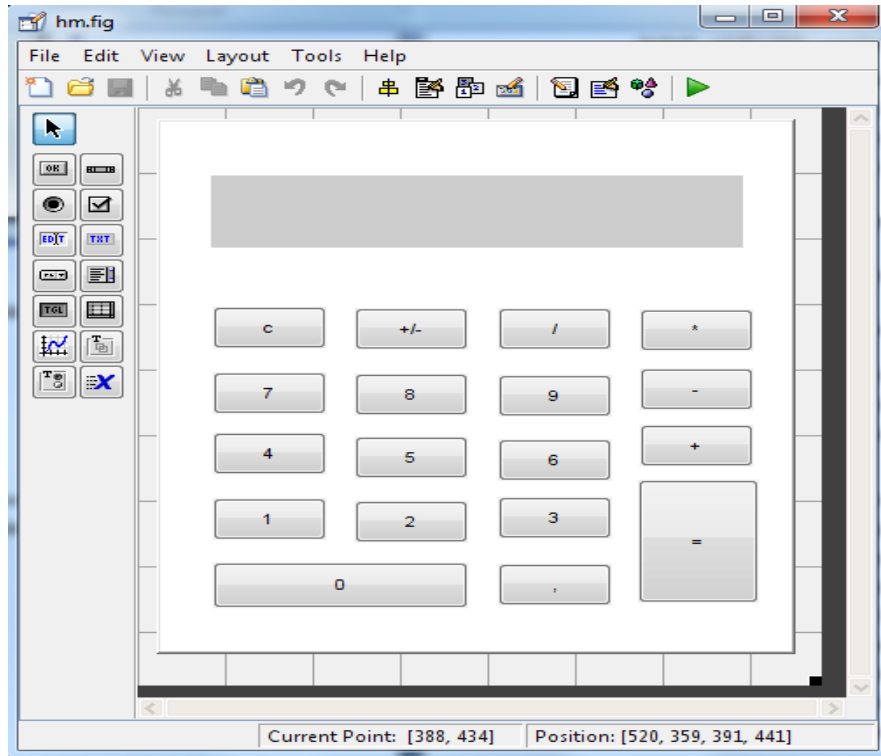
Şekil Ek.2 GUI Başlangıç Penceresi

4. Karşımıza çıkan pencere GUI'nin ana penceresidir. Bu kısımda programın görünümünün tasarlanması önerilir.



Şekil Ek.3 Ana Pencere

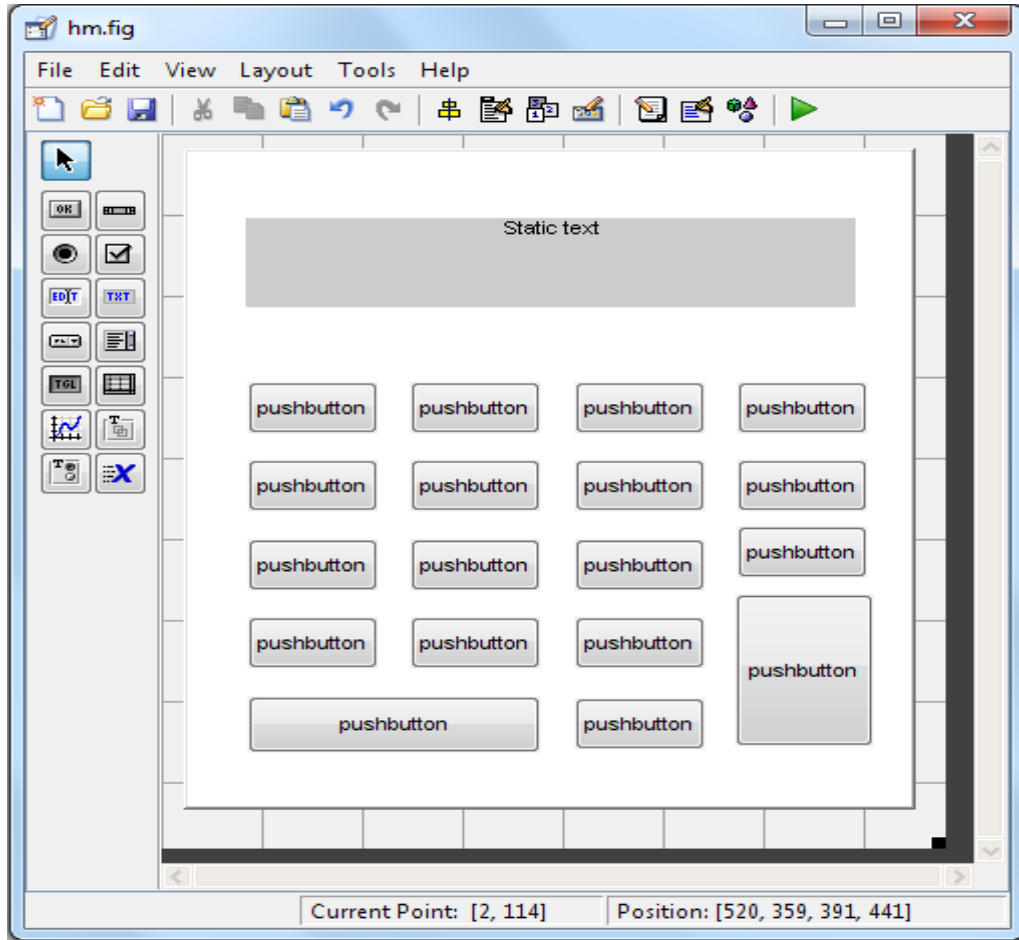
5. Şekil Ek.4'te basit bir GUI programı görülmektedir.



Şekil Ek.4 GUI Program Örneği

Yukarıda anlatılan GUI programına giriş kısmından sonra Ek-A kısmında basit bir GUI programı kullanılarak basit bir hesap makinesinin nasıl oluşturulabileceğinden ve bu esnada kullanılan komutlardan bahsedilmiştir.

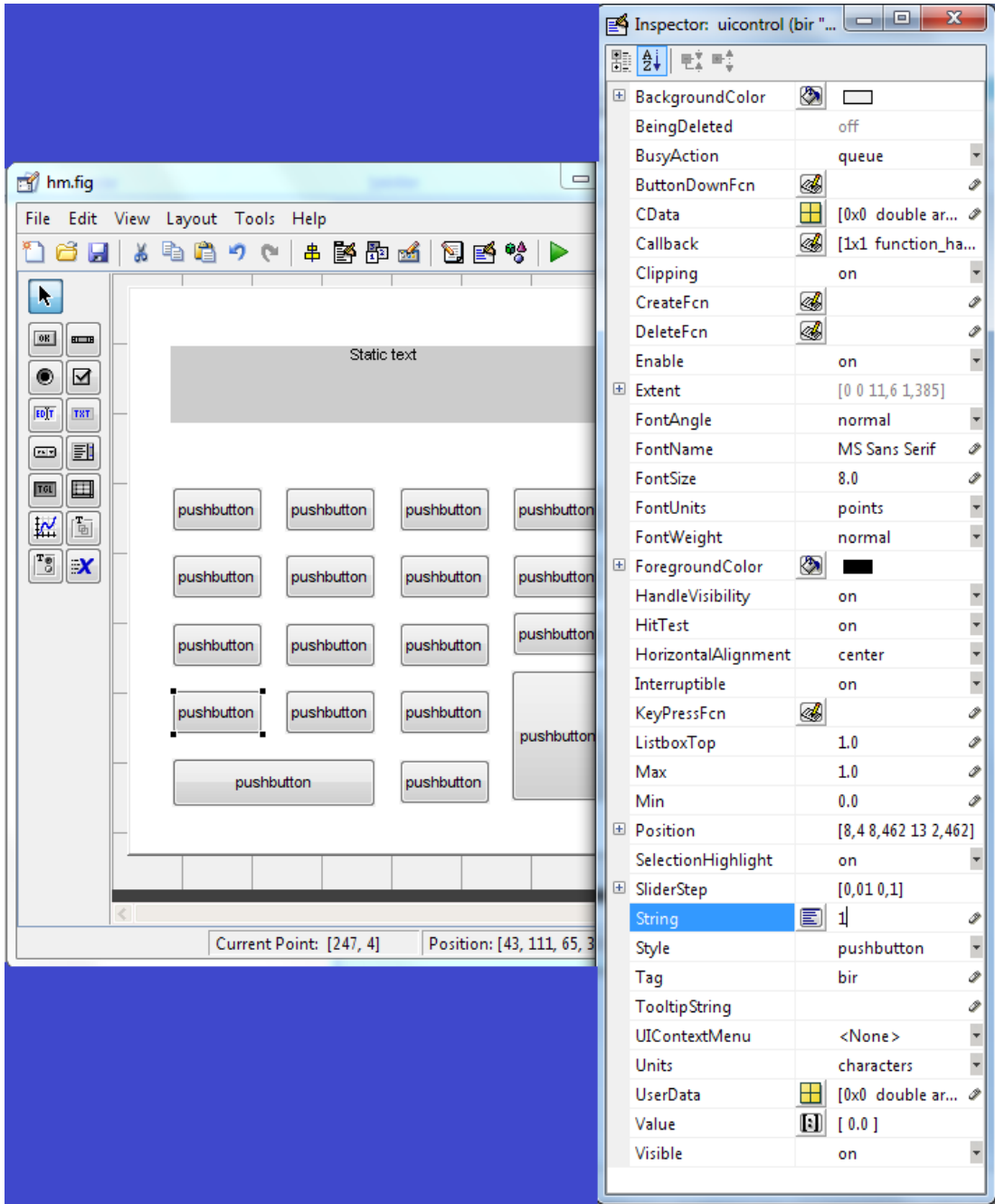
GUI programını daha önce anlatıldığı gibi açtıktan sonra pencere boyutunun ayarlanması program oluşturulurken kolaylık sağlayacaktır. GUI programının en yararlı özelliklerinden bir tanesinde yaratıcılığa açık olmasıdır bu sebepten dolayı bu program pek çok farklı şekilde gerçekleştirilebilir.



Şekil Ek.5 Tasarım Aşaması - 1

Hesap makinesi oluşturulurken yukarıdaki gibi "pushbutton"lar ve "Static Text" özellikleri kullanılmıştır. Bu kısımda yapılması komut satırında ve GUI penceresinde

görünmesini istediğimiz isimleri çift tıklanıldığında açılan özellikler penceresinden ayarlanmasıdır.

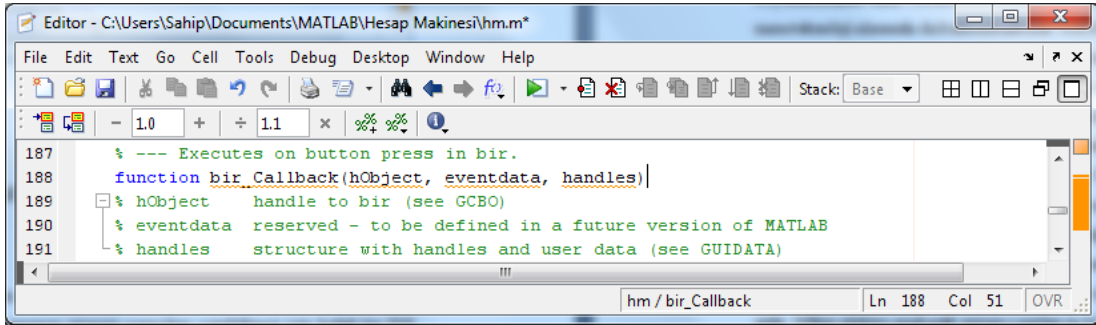


Şekil Ek.6 Tasarım Aşaması - 2

Hesap makinesinde kullanılacak "1" düğmesini ele alalım. Sağ pencerede görülen String kısmındaki değişim sadece GUI penceresini etkiler komut satırında bir değişikliğe neden

olmaz komut satırı deęişiklikleri "Tag" kısmından yapılmalıdır. Her bir düęme için gerekli deęişiklikler bu şekilde yapıldıktan sonra işin GUI kısmını bitirmiş oluruz.

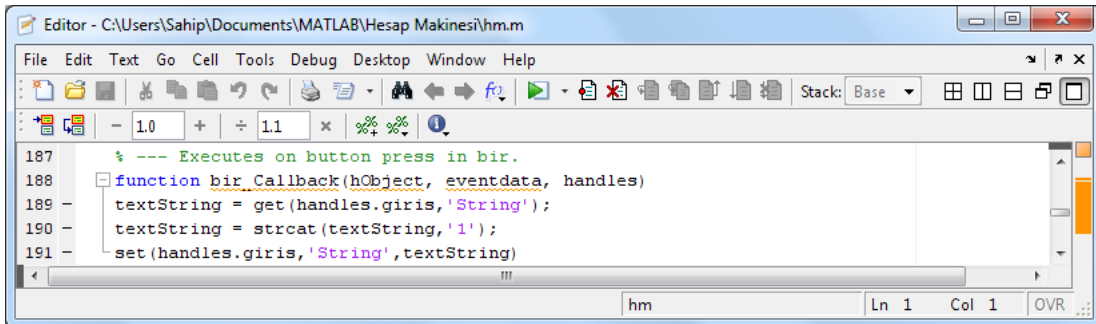
Deęişiklikleri kaydettikten sonra yukarıdaki menüden "M-File Editor" düęmesinden komut penceresini açalım, bu kısımda her atadığımız düęme için bir fonksiyon bloęu oluşmaktadır. Biraz önceki örnekte verdiğimiz "1" düęmesini ele alalım.



```
187 % --- Executes on button press in bir.
188 function bir_Callback(hObject, eventdata, handles)
189 % hObject handle to bir (see GCBO)
190 % eventdata reserved - to be defined in a future version of MATLAB
191 % handles structure with handles and user data (see GUIDATA)
```

Şekil Ek.7 Komut Penceresi - 1

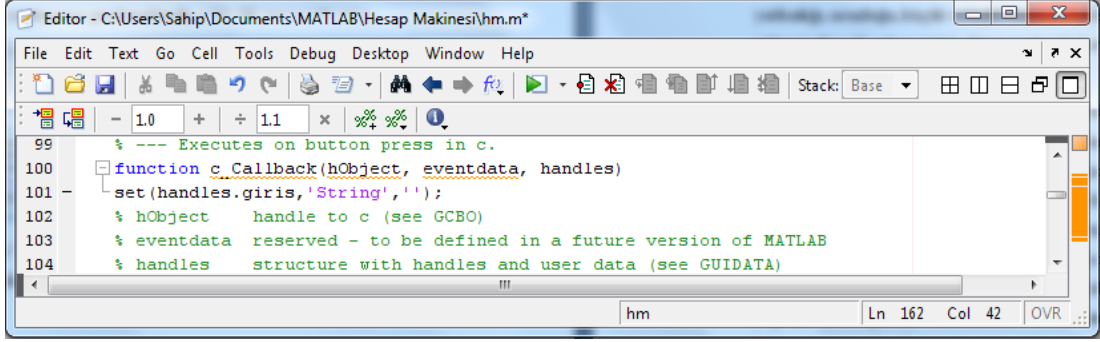
Gördüğümüz "function" bloęu GUI'de atadığımız her düęme için otomatik olarak komut penceresinde yaratılmıştır. Bu düęmenin yapmasını istediğimiz işi bu kısma yazmamız gerekir.



```
187 % --- Executes on button press in bir.
188 function bir_Callback(hObject, eventdata, handles)
189     textString = get(handles.giris, 'String');
190     textString = strcat(textString, '1');
191     set(handles.giris, 'String', textString)
```

Şekil Ek.8 Komut Penceresi - 2

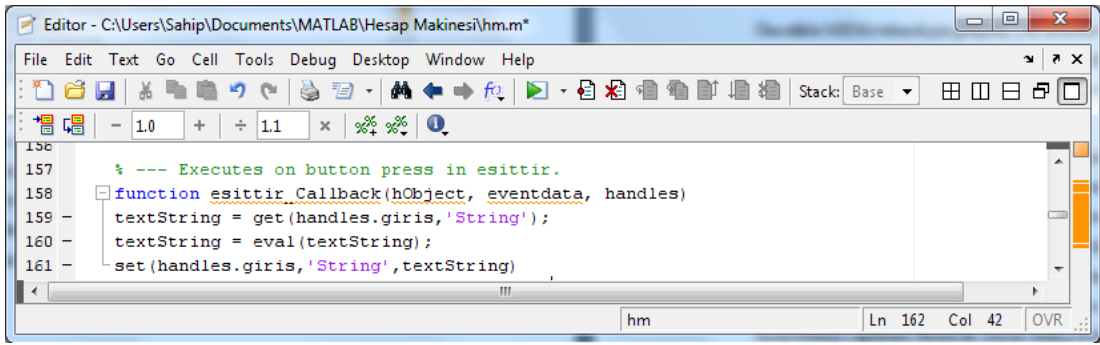
Yazılan bu satırları açıklamak gerekirse, mavi olarak yazılmış "function" satırı otomatik olarak program tarafından eklenen satırdır. İkinci satırda, "giris" olarak adlandırılmış "Static Text" penceresindeki bütün sayıları alıp "textString" olarak saklamasını söylemekteyiz, üçüncü satırda "textString"ın sonuna "1" rakamını ekleyerek tekrar "textString" olarak saklamasını, son satırda ise "textString"ı "giris"e aktarmasını söylemekteyiz. Bu kısımdaki komutları bütün dięer rakamlar ve işlem operativleri için de yapmamız gerekmektedir. ("0"-"9", "+", "-", "*", "/")



```
99 % --- Executes on button press in c.
100 function c_Callback(hObject, eventdata, handles)
101     set(handles.giris,'String','');
102     % hObject    handle to c (see GCBO)
103     % eventdata  reserved - to be defined in a future version of MATLAB
104     % handles    structure with handles and user data (see GUIDATA)
```

Şekil Ek.9 Komut Penceresi - 3

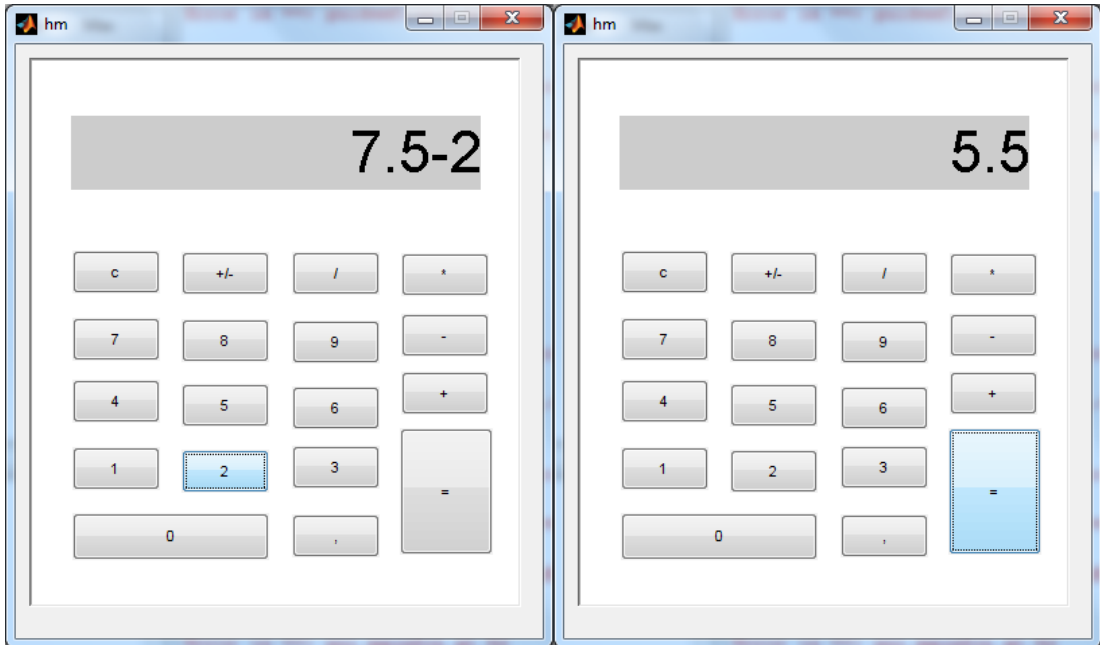
“c” fonksiyonunda yapılmasını istediğimiz işlem text kısmının tamamen boşaltılmasıdır.



```
156
157 % --- Executes on button press in esittir.
158 function esittir_Callback(hObject, eventdata, handles)
159     textString = get(handles.giris,'String');
160     textString = eval(textString);
161     set(handles.giris,'String',textString)
```

Şekil Ek.10 Komut Penceresi - 4

Şekil Ek.10’da görüldüğü gibi “esittir” fonksiyonuna farklı bir komut yazmalıyız. Bu kısımda artık bütün yazılan sayıların işlemlerini yapmasını söylemekteyiz.



Şekil Ek.11 Test Aşaması ve Sonuç

Dikkat edilmesi gereken başka bir husus ise büyük ve küçük harflerin yazımıdır. Örneğin “textString”in başka bir fonksiyonda “textstring” olarak yazılması programın çalışmaması için yeterlidir.

PROGRAM KODU

1. Ana Menü

```

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @AnaMenu_OpeningFcn, ...
                  'gui_OutputFcn', @AnaMenu_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function AnaMenu_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
eksensayisi=get(handles.popupmenu1,'value');
syms grafik;
assignin('base','grafik',grafik);
function varargout = AnaMenu_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
function pushbutton3_Callback(hObject, eventdata, handles)
EksenSayilari
function pushbutton4_Callback(hObject, eventdata, handles)
lvmeSensorDegerleri
function pushbutton5_Callback(hObject, eventdata, handles)
GrafikSecimi
function pushbutton6_Callback(hObject, eventdata, handles)
function pushbutton7_Callback(hObject, eventdata, handles)
function pushbutton8_Callback(hObject, eventdata, handles)
lvmeSensorKonumlari
function pushbutton9_Callback(hObject, eventdata, handles)
lvmeSensorYonleri

```

```

function pushbutton10_Callback(hObject, eventdata, handles)
Sensorhatalari
function pushbutton11_Callback(hObject, eventdata, handles)
Geometrikhatalar
function popupmenu2_Callback(hObject, eventdata, handles)
eksensayisi=get(hObject,'Value');
sensorsayisi=eksensayisi+2;
    assignin('base','sensorsayisi',sensorsayisi)
    lvmeSensorKonumlari
function popupmenu2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function pushbutton12_Callback(hObject, eventdata, handles)
function axes1_CreateFcn(hObject, eventdata, handles)
syms grafik;
assignin('base','grafik',grafik);
grafik=evalin('base','grafik');
function pushbutton19_Callback(hObject, eventdata, handles)
sensorsayisi=evalin('base','sensorsayisi');
grafik=evalin('base','grafik');
if grafik==1;
    cla reset
if sensorsayisi==3
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
if grafik==1;
axis([-1 1 -1 1 -1 1])
plot3(u1x,u1y,u1z,'r*')
grid on
hold on
axis([-1 1 -1 1 -1 1])
plot3(u2x,u2y,u2z,'r*')
grid on
hold on
axis([-1 1 -1 1 -1 1])
plot3(u3x,u3y,u3z,'r*')
grid on
hold on
box on
.
.
.
if eksen12==3
    teta12ax=evalin('base','teta12ax');
    teta12ay=evalin('base','teta12ay');
    teta12az=evalin('base','teta12az');
    teta12bx=evalin('base','teta12bx');
    teta12by=evalin('base','teta12by');

```

```

    teta12bz=evalin('base','teta12bz');
    teta12cx=evalin('base','teta12cx');
    teta12cy=evalin('base','teta12cy');
    teta12cz=evalin('base','teta12cz');
    axis([-1 1 -1 1 -1 1])
    line([u12x u12x+(teta12ax*0.1)],[u12y u12y+(teta12ay*0.1)],[u12z
u12z+(teta12az*0.1)],'Marker','.', 'LineStyle','-')
    hold on
    line([u12x u12x+(teta12bx*0.1)],[u12y u12y+(teta12by*0.1)],[u12z
u12z+(teta12bz*0.1)],'Marker','.', 'LineStyle','-')
    hold on
    line([u12x u12x+(teta12cx*0.1)],[u12y u12y+(teta12cy*0.1)],[u12z
u12z+(teta12cz*0.1)],'Marker','.', 'LineStyle','-')
    hold on
    grid on
    box on
end
end
end

```

2. İvme Sensör Konumları

```

function varargout = IvmeSensorKonumlari(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @IvmeSensorKonumlari_OpeningFcn, ...
    'gui_OutputFcn', @IvmeSensorKonumlari_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function IvmeSensorKonumlari_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
sensorsayisi=evalin('base','sensorsayisi');
if sensorsayisi==3
    set(handles.edit4,'Visible','off')
    set(handles.edit5,'Visible','off')
    set(handles.edit6,'Visible','off')
.
.
.
    set(handles.text5,'Visible','off')
    set(handles.text6,'Visible','off')
    set(handles.text7,'Visible','off')
    set(handles.text8,'Visible','off')
    set(handles.text9,'Visible','off')

```

```

set(handles.text10,'Visible','off')
set(handles.text11,'Visible','off')
set(handles.text12,'Visible','off')
end
.
.
.
function pushbutton1_Callback(hObject, eventdata, handles)
sensorsayisi=evalin('base','sensorsayisi');
if sensorsayisi==3
    u1x=str2double(get(handles.edit1,'string'));
    u2x=str2double(get(handles.edit2,'string'));
    u3x=str2double(get(handles.edit3,'string'));
    u1y=str2double(get(handles.edit13,'string'));
    u2y=str2double(get(handles.edit14,'string'));
    u3y=str2double(get(handles.edit15,'string'));
    u1z=str2double(get(handles.edit25,'string'));
    u2z=str2double(get(handles.edit26,'string'));
    u3z=str2double(get(handles.edit27,'string'));
    eksen1=get(handles.listbox1,'value');
    assignin('base','eksen1',eksen1)
    eksen2=get(handles.listbox2,'value');
    assignin('base','eksen2',eksen2)
    eksen3=get(handles.listbox3,'value');
    assignin('base','eksen3',eksen3)
    assignin('base','u1x',u1x)
    assignin('base','u2x',u2x)
    assignin('base','u3x',u3x)
    assignin('base','u1y',u1y)
    assignin('base','u2y',u2y)
    assignin('base','u3y',u3y)
    assignin('base','u1z',u1z)
    assignin('base','u2z',u2z)
    assignin('base','u3z',u3z)
    lvmeSensorYonleri
if sensorsayisi==4
.
.
.
function pushbutton2_Callback(hObject, eventdata, handles)
M=xlsread('lvmeSensorKonumlari.xlsx');
    u1x=M(1,1);
    u1y=M(2,1);
    u1z=M(3,1);
    u2x=M(1,2);
    u2y=M(2,2);
    u2z=M(3,2);
    u3x=M(1,3);
    u3y=M(2,3);
    u3z=M(3,3);
    u4x=M(1,4);
    u4y=M(2,4);
    u4z=M(3,4);
    u5x=M(1,5);
    u5y=M(2,5);

```

```

u5z=M(3,5);
u6x=M(1,6);
u6y=M(2,6);
u6z=M(3,6);
u7x=M(1,7);
u7y=M(2,7);
u7z=M(3,7);
u8x=M(1,8);
u8y=M(2,8);
u8z=M(3,8);
u9x=M(1,9);
u9y=M(2,9);
u9z=M(3,9);
u10x=M(1,10);
u10y=M(2,10);
u10z=M(3,10);
u11x=M(1,11);
u11y=M(2,11);
u11z=M(3,11);
u12x=M(1,12);
u12y=M(2,12);
u12z=M(3,12);
set(handles.edit1,'String',num2str(u1x));
set(handles.edit13,'String',num2str(u1y));
set(handles.edit25,'String',num2str(u1z));
set(handles.edit2,'String',num2str(u2x));
set(handles.edit14,'String',num2str(u2y));
set(handles.edit26,'String',num2str(u2z));
set(handles.edit3,'String',num2str(u3x));
set(handles.edit15,'String',num2str(u3y));
set(handles.edit27,'String',num2str(u3z));
set(handles.edit4,'String',num2str(u4x));
set(handles.edit16,'String',num2str(u4y));
set(handles.edit28,'String',num2str(u4z));
set(handles.edit5,'String',num2str(u5x));
set(handles.edit17,'String',num2str(u5y));
set(handles.edit29,'String',num2str(u5z));
set(handles.edit6,'String',num2str(u6x));
set(handles.edit18,'String',num2str(u6y));
set(handles.edit30,'String',num2str(u6z));
set(handles.edit7,'String',num2str(u7x));
set(handles.edit19,'String',num2str(u7y));
set(handles.edit31,'String',num2str(u7z));
set(handles.edit8,'String',num2str(u8x));
set(handles.edit20,'String',num2str(u8y));
set(handles.edit32,'String',num2str(u8z));
set(handles.edit9,'String',num2str(u9x));
set(handles.edit21,'String',num2str(u9y));
set(handles.edit33,'String',num2str(u9z));
set(handles.edit10,'String',num2str(u10x));
set(handles.edit22,'String',num2str(u10y));
set(handles.edit34,'String',num2str(u10z));
set(handles.edit11,'String',num2str(u11x));
set(handles.edit23,'String',num2str(u11y));
set(handles.edit35,'String',num2str(u11z));
set(handles.edit12,'String',num2str(u12x));

```

```
set(handles.edit24,'String',num2str(u12y));
set(handles.edit36,'String',num2str(u12z));
```

3. İvme Sensör Yönlendirmeleri

```
function varargout = IvmeSensorYonleri(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @IvmeSensorYonleri_OpeningFcn, ...
    'gui_OutputFcn',  @IvmeSensorYonleri_OutputFcn, ...
    'gui_LayoutFcn',  [], ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function IvmeSensorYonleri_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
sensorsayisi=evalin('base','sensorsayisi');
if sensorsayisi==3
    eksen1=evalin('base','eksen1');
    eksen2=evalin('base','eksen2');
    eksen3=evalin('base','eksen3');
    if eksen1==1
        set(handles.edit4,'Visible','off')
        set(handles.edit5,'Visible','off')
        set(handles.edit6,'Visible','off')
        set(handles.edit7,'Visible','off')
        set(handles.edit8,'Visible','off')
        set(handles.edit9,'Visible','off')
    end
    if eksen1==2
        set(handles.edit7,'Visible','off')
        set(handles.edit8,'Visible','off')
        set(handles.edit9,'Visible','off')
    end
    if eksen2==1
        set(handles.edit13,'Visible','off')
        set(handles.edit14,'Visible','off')
```

```

    set(handles.edit15,'Visible','off')
    set(handles.edit16,'Visible','off')
    set(handles.edit17,'Visible','off')
    set(handles.edit18,'Visible','off')
end
if eksen2==2
    set(handles.edit16,'Visible','off')
    set(handles.edit17,'Visible','off')
    set(handles.edit18,'Visible','off')
end
if eksen3==1
    set(handles.edit22,'Visible','off')
    set(handles.edit23,'Visible','off')
    set(handles.edit24,'Visible','off')
    set(handles.edit25,'Visible','off')
    set(handles.edit26,'Visible','off')
    set(handles.edit27,'Visible','off')
end
if eksen3==2
    set(handles.edit25,'Visible','off')
    set(handles.edit26,'Visible','off')
    set(handles.edit27,'Visible','off')
end
    set(handles.edit28,'Visible','off')
    set(handles.edit29,'Visible','off')
    set(handles.edit30,'Visible','off')
    .
    .
    .
    set(handles.edit61,'Visible','off')
    set(handles.edit62,'Visible','off')
    set(handles.edit63,'Visible','off')
    set(handles.edit106,'Visible','off')
    .
    .
    .
    set(handles.edit107,'Visible','off')
    set(handles.edit108,'Visible','off')
    set(handles.edit106,'Visible','off')
    set(handles.edit107,'Visible','off')
    set(handles.edit108,'Visible','off')
end

function pushbutton1_Callback(hObject, eventdata, handles)
sensorsayisi=evalin('base','sensorsayisi');
grafik=2;
assignin('base','grafik',grafik)
if sensorsayisi==3
    eksen1=evalin('base','eksen1');
    eksen2=evalin('base','eksen2');
    eksen3=evalin('base','eksen3');
    if eksen1==1
        teta1ax=str2double(get(handles.edit1,'string'));
        teta1ay=str2double(get(handles.edit2,'string'));
        teta1az=str2double(get(handles.edit3,'string'));
    end
end

```

```

if eksen1==2
    teta1ax=str2double(get(handles.edit1,'string'));
    teta1ay=str2double(get(handles.edit2,'string'));
    teta1az=str2double(get(handles.edit3,'string'));
    teta1bx=str2double(get(handles.edit4,'string'));
    teta1by=str2double(get(handles.edit5,'string'));
    teta1bz=str2double(get(handles.edit6,'string'));
end
if eksen1==3
    teta1ax=str2double(get(handles.edit1,'string'));
    teta1ay=str2double(get(handles.edit2,'string'));
    teta1az=str2double(get(handles.edit3,'string'));
    teta1bx=str2double(get(handles.edit4,'string'));
    teta1by=str2double(get(handles.edit5,'string'));
    teta1bz=str2double(get(handles.edit6,'string'));
    teta1cx=str2double(get(handles.edit7,'string'));
    teta1cy=str2double(get(handles.edit8,'string'));
    teta1cz=str2double(get(handles.edit9,'string'));
end
%-----
if eksen2==1
    teta2ax=str2double(get(handles.edit10,'string'));
    teta2ay=str2double(get(handles.edit11,'string'));
    teta2az=str2double(get(handles.edit12,'string'));
end
if eksen2==2
    teta2ax=str2double(get(handles.edit10,'string'));
    teta2ay=str2double(get(handles.edit11,'string'));
    teta2az=str2double(get(handles.edit12,'string'));
    teta2bx=str2double(get(handles.edit13,'string'));
    teta2by=str2double(get(handles.edit14,'string'));
    teta2bz=str2double(get(handles.edit15,'string'));
end
if eksen2==3
    teta2ax=str2double(get(handles.edit10,'string'));
    teta2ay=str2double(get(handles.edit11,'string'));
    teta2az=str2double(get(handles.edit12,'string'));
    teta2bx=str2double(get(handles.edit13,'string'));
    teta2by=str2double(get(handles.edit14,'string'));
    teta2bz=str2double(get(handles.edit15,'string'));
    teta2cx=str2double(get(handles.edit16,'string'));
    teta2cy=str2double(get(handles.edit17,'string'));
    teta2cz=str2double(get(handles.edit18,'string'));
end
%-----
if eksen3==1
    teta3ax=str2double(get(handles.edit19,'string'));
    teta3ay=str2double(get(handles.edit20,'string'));
    teta3az=str2double(get(handles.edit21,'string'));
end
if eksen3==2
    teta3ax=str2double(get(handles.edit19,'string'));
    teta3ay=str2double(get(handles.edit20,'string'));
    teta3az=str2double(get(handles.edit21,'string'));
    teta3bx=str2double(get(handles.edit22,'string'));
    teta3by=str2double(get(handles.edit23,'string'));

```

```

    teta3bz=str2double(get(handles.edit24,'string'));
end
if eksen3==3
    teta3ax=str2double(get(handles.edit19,'string'));
    teta3ay=str2double(get(handles.edit20,'string'));
    teta3az=str2double(get(handles.edit21,'string'));
    teta3bx=str2double(get(handles.edit22,'string'));
    teta3by=str2double(get(handles.edit23,'string'));
    teta3bz=str2double(get(handles.edit24,'string'));
    teta3cx=str2double(get(handles.edit25,'string'));
    teta3cy=str2double(get(handles.edit26,'string'));
    teta3cz=str2double(get(handles.edit27,'string'));
end
%=====
if eksen1==1
    assignin('base','teta1ax',teta1ax)
    assignin('base','teta1ay',teta1ay)
    assignin('base','teta1az',teta1az)
end
if eksen1==2
    assignin('base','teta1ax',teta1ax)
    assignin('base','teta1ay',teta1ay)
    assignin('base','teta1az',teta1az)
    assignin('base','teta1bx',teta1bx)
    assignin('base','teta1by',teta1by)
    assignin('base','teta1bz',teta1bz)
end
if eksen1==3
    assignin('base','teta1ax',teta1ax)
    assignin('base','teta1ay',teta1ay)
    assignin('base','teta1az',teta1az)
    assignin('base','teta1bx',teta1bx)
    assignin('base','teta1by',teta1by)
    assignin('base','teta1bz',teta1bz)
    assignin('base','teta1cx',teta1cx)
    assignin('base','teta1cy',teta1cy)
    assignin('base','teta1cz',teta1cz)
end
%-----
if eksen2==1
    assignin('base','teta2ax',teta2ax)
    assignin('base','teta2ay',teta2ay)
    assignin('base','teta2az',teta2az)
end
if eksen2==2
    assignin('base','teta2ax',teta2ax)
    assignin('base','teta2ay',teta2ay)
    assignin('base','teta2az',teta2az)
    assignin('base','teta2bx',teta2bx)
    assignin('base','teta2by',teta2by)
    assignin('base','teta2bz',teta2bz)
end
if eksen2==3
    assignin('base','teta2ax',teta2ax)
    assignin('base','teta2ay',teta2ay)
    assignin('base','teta2az',teta2az)

```

```

    assignin('base','teta2bx',teta2bx)
    assignin('base','teta2by',teta2by)
    assignin('base','teta2bz',teta2bz)
    assignin('base','teta2cx',teta2cx)
    assignin('base','teta2cy',teta2cy)
    assignin('base','teta2cz',teta2cz)
end
%-----
if eksen3==1
    assignin('base','teta3ax',teta3ax)
    assignin('base','teta3ay',teta3ay)
    assignin('base','teta3az',teta3az)
end
if eksen3==2
    assignin('base','teta3ax',teta3ax)
    assignin('base','teta3ay',teta3ay)
    assignin('base','teta3az',teta3az)
    assignin('base','teta3bx',teta3bx)
    assignin('base','teta3by',teta3by)
    assignin('base','teta3bz',teta3bz)
end
if eksen3==3
    assignin('base','teta3ax',teta3ax)
    assignin('base','teta3ay',teta3ay)
    assignin('base','teta3az',teta3az)
    assignin('base','teta3bx',teta3bx)
    assignin('base','teta3by',teta3by)
    assignin('base','teta3bz',teta3bz)
    assignin('base','teta3cx',teta3cx)
    assignin('base','teta3cy',teta3cy)
    assignin('base','teta3cz',teta3cz)
end
end
if sensorsayisi==4
    eksen1=evalin('base','eksen1');
    eksen2=evalin('base','eksen2');
    eksen3=evalin('base','eksen3');
    eksen4=evalin('base','eksen4');
%-----
    if eksen1==1
        teta1ax=str2double(get(handles.edit1,'string'));
        teta1ay=str2double(get(handles.edit2,'string'));
        teta1az=str2double(get(handles.edit3,'string'));
    end
    if eksen1==2
        teta1ax=str2double(get(handles.edit1,'string'));
        teta1ay=str2double(get(handles.edit2,'string'));
        teta1az=str2double(get(handles.edit3,'string'));
        teta1bx=str2double(get(handles.edit4,'string'));
        teta1by=str2double(get(handles.edit5,'string'));
        teta1bz=str2double(get(handles.edit6,'string'));
    end
    if eksen1==3
        teta1ax=str2double(get(handles.edit1,'string'));
        teta1ay=str2double(get(handles.edit2,'string'));
        teta1az=str2double(get(handles.edit3,'string'));
    end
end

```

```

teta1bx=str2double(get(handles.edit4,'string'));
teta1by=str2double(get(handles.edit5,'string'));
teta1bz=str2double(get(handles.edit6,'string'));
teta1cx=str2double(get(handles.edit7,'string'));
teta1cy=str2double(get(handles.edit8,'string'));
teta1cz=str2double(get(handles.edit9,'string'));
end
%-----
if eksten2==1
    teta2ax=str2double(get(handles.edit10,'string'));
    teta2ay=str2double(get(handles.edit11,'string'));
    teta2az=str2double(get(handles.edit12,'string'));
end
if eksten2==2
    teta2ax=str2double(get(handles.edit10,'string'));
    teta2ay=str2double(get(handles.edit11,'string'));
    teta2az=str2double(get(handles.edit12,'string'));
    teta2bx=str2double(get(handles.edit13,'string'));
    teta2by=str2double(get(handles.edit14,'string'));
    teta2bz=str2double(get(handles.edit15,'string'));
end
if eksten2==3
    teta2ax=str2double(get(handles.edit10,'string'));
    teta2ay=str2double(get(handles.edit11,'string'));
    teta2az=str2double(get(handles.edit12,'string'));
    teta2bx=str2double(get(handles.edit13,'string'));
    teta2by=str2double(get(handles.edit14,'string'));
    teta2bz=str2double(get(handles.edit15,'string'));
    teta2cx=str2double(get(handles.edit16,'string'));
    teta2cy=str2double(get(handles.edit17,'string'));
    teta2cz=str2double(get(handles.edit18,'string'));
end
%-----
if eksten3==1
    teta3ax=str2double(get(handles.edit19,'string'));
    teta3ay=str2double(get(handles.edit20,'string'));
    teta3az=str2double(get(handles.edit21,'string'));
end
if eksten3==2
    teta3ax=str2double(get(handles.edit19,'string'));
    teta3ay=str2double(get(handles.edit20,'string'));
    teta3az=str2double(get(handles.edit21,'string'));
    teta3bx=str2double(get(handles.edit22,'string'));
    teta3by=str2double(get(handles.edit23,'string'));
    teta3bz=str2double(get(handles.edit24,'string'));
end
if eksten3==3
    teta3ax=str2double(get(handles.edit19,'string'));
    teta3ay=str2double(get(handles.edit20,'string'));
    teta3az=str2double(get(handles.edit21,'string'));
    teta3bx=str2double(get(handles.edit22,'string'));
    teta3by=str2double(get(handles.edit23,'string'));
    teta3bz=str2double(get(handles.edit24,'string'));
    teta3cx=str2double(get(handles.edit25,'string'));
    teta3cy=str2double(get(handles.edit26,'string'));
    teta3cz=str2double(get(handles.edit27,'string'));

```

end

function pushbutton3_Callback(hObject, eventdata, handles)

M=xlsread('IvmeSensorYonleri.xlsx');

```
teta1ax=M(2,1);
teta1ay=M(3,1);
teta1az=M(4,1);
teta1bx=M(5,1);
teta1by=M(6,1);
teta1bz=M(7,1);
teta1cx=M(8,1);
teta1cy=M(9,1);
teta1cz=M(10,1);
teta2ax=M(2,2);
teta2ay=M(3,2);
teta2az=M(4,2);
teta2bx=M(5,2);
teta2by=M(6,2);
teta2bz=M(7,2);
teta2cx=M(8,2);
teta2cy=M(9,2);
teta2cz=M(10,2);
teta3ax=M(2,3);
teta3ay=M(3,3);
teta3az=M(4,3);
teta3bx=M(5,3);
teta3by=M(6,3);
teta3bz=M(7,3);
teta3cx=M(8,3);
teta3cy=M(9,3);
teta3cz=M(10,3);
teta4ax=M(2,4);
teta4ay=M(3,4);
teta4az=M(4,4);
teta4bx=M(5,4);
teta4by=M(6,4);
teta4bz=M(7,4);
teta4cx=M(8,4);
teta4cy=M(9,4);
teta4cz=M(10,4);
teta5ax=M(2,5);
teta5ay=M(3,5);
teta5az=M(4,5);
teta5bx=M(5,5);
teta5by=M(6,5);
teta5bz=M(7,5);
teta5cx=M(8,5);
teta5cy=M(9,5);
teta5cz=M(10,5);
teta6ax=M(2,6);
teta6ay=M(3,6);
teta6az=M(4,6);
teta6bx=M(5,6);
teta6by=M(6,6);
teta6bz=M(7,6);
teta6cx=M(8,6);
```

teta6cy=M(9,6);
teta6cz=M(10,6);
teta7ax=M(2,7);
teta7ay=M(3,7);
teta7az=M(4,7);
teta7bx=M(5,7);
teta7by=M(6,7);
teta7bz=M(7,7);
teta7cx=M(8,7);
teta7cy=M(9,7);
teta7cz=M(10,7);
teta8ax=M(2,8);
teta8ay=M(3,8);
teta8az=M(4,8);
teta8bx=M(5,8);
teta8by=M(6,8);
teta8bz=M(7,8);
teta8cx=M(8,8);
teta8cy=M(9,8);
teta8cz=M(10,8);
teta9ax=M(2,9);
teta9ay=M(3,9);
teta9az=M(4,9);
teta9bx=M(5,9);
teta9by=M(6,9);
teta9bz=M(7,9);
teta9cx=M(8,9);
teta9cy=M(9,9);
teta9cz=M(10,9);
teta10ax=M(2,10);
teta10ay=M(3,10);
teta10az=M(4,10);
teta10bx=M(5,10);
teta10by=M(6,10);
teta10bz=M(7,10);
teta10cx=M(8,10);
teta10cy=M(9,10);
teta10cz=M(10,10);
teta11ax=M(2,11);
teta11ay=M(3,11);
teta11az=M(4,11);
teta11bx=M(5,11);
teta11by=M(6,11);
teta11bz=M(7,11);
teta11cx=M(8,11);
teta11cy=M(9,11);
teta11cz=M(10,11);
teta12ax=M(2,12);
teta12ay=M(3,12);
teta12az=M(4,12);
teta12bx=M(5,12);
teta12by=M(6,12);
teta12bz=M(7,12);
teta12cx=M(8,12);
teta12cy=M(9,12);
teta12cz=M(10,12);

```

if eksen1==1
    assignin('base','teta1ax',teta1ax)
    assignin('base','teta1ay',teta1ay)
    assignin('base','teta1az',teta1az)
end
if eksen1==2
    assignin('base','teta1bx',teta1ax)
    assignin('base','teta1by',teta1ay)
    assignin('base','teta1bz',teta1az)
    assignin('base','teta1bx',teta1bx)
    assignin('base','teta1by',teta1by)
    assignin('base','teta1bz',teta1bz)
end
if eksen1==3
    assignin('base','teta1ax',teta1ax)
    assignin('base','teta1ay',teta1ay)
    assignin('base','teta1az',teta1az)
    assignin('base','teta1bx',teta1bx)
    assignin('base','teta1by',teta1by)
    assignin('base','teta1bz',teta1bz)
    assignin('base','teta1cx',teta1cx)
    assignin('base','teta1cy',teta1cy)
    assignin('base','teta1cz',teta1cz)
end
%-----
if eksen2==1
    assignin('base','teta2ax',teta2ax)
    assignin('base','teta2ay',teta2ay)
    assignin('base','teta2az',teta2az)
end
if eksen2==2
    assignin('base','teta2bx',teta2ax)
    assignin('base','teta2by',teta2ay)
    assignin('base','teta2bz',teta2az)
    assignin('base','teta2bx',teta2bx)
    assignin('base','teta2by',teta2by)
    assignin('base','teta2bz',teta2bz)
end
if eksen2==3
    assignin('base','teta2ax',teta2ax)
    assignin('base','teta2ay',teta2ay)
    assignin('base','teta2az',teta2az)
    assignin('base','teta2bx',teta2bx)
    assignin('base','teta2by',teta2by)
    assignin('base','teta2bz',teta2bz)
    assignin('base','teta2cx',teta2cx)
    assignin('base','teta2cy',teta2cy)
    assignin('base','teta2cz',teta2cz)
end
%-----

if eksen3==1
    assignin('base','teta3ax',teta3ax)
    assignin('base','teta3ay',teta3ay)
    assignin('base','teta3az',teta3az)
end

```

```

if eksen3==2
    assignin('base','teta3ax',teta3ax)
    assignin('base','teta3ay',teta3ay)
    assignin('base','teta3az',teta3az)
    assignin('base','teta3bx',teta3bx)
    assignin('base','teta3by',teta3by)
    assignin('base','teta3bz',teta3bz)
end
if eksen3==3
    assignin('base','teta3ax',teta3ax)
    assignin('base','teta3ay',teta3ay)
    assignin('base','teta3az',teta3az)
    assignin('base','teta3bx',teta3bx)
    assignin('base','teta3by',teta3by)
    assignin('base','teta3bz',teta3bz)
    assignin('base','teta3cx',teta3cx)
    assignin('base','teta3cy',teta3cy)
    assignin('base','teta3cz',teta3cz)
.
.
.
if eksen12==1
    assignin('base','teta12ax',teta12ax)
    assignin('base','teta12ay',teta12ay)
    assignin('base','teta12az',teta12az)
end
if eksen12==2
    assignin('base','teta12ax',teta12ax)
    assignin('base','teta12ay',teta12ay)
    assignin('base','teta12az',teta12az)
    assignin('base','teta12bx',teta12bx)
    assignin('base','teta12by',teta12by)
    assignin('base','teta12bz',teta12bz)
end
if eksen12==3
    assignin('base','teta12ax',teta12ax)
    assignin('base','teta12ay',teta12ay)
    assignin('base','teta12az',teta12az)
    assignin('base','teta12bx',teta12bx)
    assignin('base','teta12by',teta12by)
    assignin('base','teta12bz',teta12bz)
    assignin('base','teta12cx',teta12cx)
    assignin('base','teta12cy',teta12cy)
    assignin('base','teta12cz',teta12cz)
end

```

4. İvme Sensör Değerleri

```

function varargout = IvmeSensorDegerleri(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @IvmeSensorDegerleri_OpeningFcn, ...

```

```

        'gui_OutputFcn', @lvmeSensorDegerleri_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function lvmeSensorDegerleri_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
sensorsayisi=evalin('base','sensorsayisi');
if sensorsayisi==3
    eksen1=evalin('base','eksen1');
    eksen2=evalin('base','eksen2');
    eksen3=evalin('base','eksen3');
    if eksen1==1
        set(handles.edit220,'Visible','off')
        set(handles.edit221,'Visible','off')
    end
    if eksen1==2
        set(handles.edit221,'Visible','off')
    end
    if eksen2==1
        set(handles.edit223,'Visible','off')
        set(handles.edit224,'Visible','off')
    end
    if eksen2==2
        set(handles.edit224,'Visible','off')
    end
    if eksen3==1
        set(handles.edit226,'Visible','off')
        set(handles.edit227,'Visible','off')
    end
    if eksen3==2
        set(handles.edit227,'Visible','off')
    end
    set(handles.edit228,'Visible','off')
    set(handles.edit229,'Visible','off')
    set(handles.edit230,'Visible','off')
    set(handles.edit231,'Visible','off')
    set(handles.edit232,'Visible','off')
    set(handles.edit233,'Visible','off')
    set(handles.edit234,'Visible','off')
    set(handles.edit235,'Visible','off')
    set(handles.edit236,'Visible','off')
    set(handles.edit237,'Visible','off')
    set(handles.edit238,'Visible','off')
    set(handles.edit239,'Visible','off')
    set(handles.edit240,'Visible','off')
    set(handles.edit241,'Visible','off')
    set(handles.edit242,'Visible','off')

```

```

set(handles.edit243,'Visible','off')
set(handles.edit244,'Visible','off')
set(handles.edit245,'Visible','off')
set(handles.edit246,'Visible','off')
set(handles.edit247,'Visible','off')
set(handles.edit248,'Visible','off')
set(handles.edit249,'Visible','off')
set(handles.edit250,'Visible','off')
set(handles.edit251,'Visible','off')
set(handles.edit252,'Visible','off')
set(handles.edit253,'Visible','off')
set(handles.edit254,'Visible','off')
end
.
.
.
if sensorsayisi==12
    eksen1=evalin('base','eksen1');
    eksen2=evalin('base','eksen2');
    eksen3=evalin('base','eksen3');
    eksen4=evalin('base','eksen4');
    eksen5=evalin('base','eksen5');
    eksen6=evalin('base','eksen6');
    eksen7=evalin('base','eksen7');
    eksen8=evalin('base','eksen8');
    eksen9=evalin('base','eksen9');
    eksen10=evalin('base','eksen10');
    eksen11=evalin('base','eksen11');
    eksen12=evalin('base','eksen12');
    if eksen1==1
        set(handles.edit220,'Visible','off')
        set(handles.edit221,'Visible','off')
    end
    if eksen1==2
        set(handles.edit221,'Visible','off')
    end
    if eksen2==1
        set(handles.edit223,'Visible','off')
        set(handles.edit224,'Visible','off')
    end
    .
    .
    .
    if eksen12==1
        set(handles.edit253,'Visible','off')
        set(handles.edit254,'Visible','off')
    end
    if eksen12==2
        set(handles.edit254,'Visible','off')
    end
end
function pushbutton1_Callback(hObject, eventdata, handles)
sensorsayisi=evalin('base','sensorsayisi');

if sensorsayisi==3

```

```

eksen1=evalin('base','eksen1');
eksen2=evalin('base','eksen2');
eksen3=evalin('base','eksen3');
if eksen1==1
    A1x=str2double(get(handles.edit219,'string'));
end
if eksen1==2
    A1x=str2double(get(handles.edit219,'string'));
    A1y=str2double(get(handles.edit220,'string'));
end
if eksen1==3
    A1x=str2double(get(handles.edit219,'string'));
    A1y=str2double(get(handles.edit220,'string'));
    A1z=str2double(get(handles.edit221,'string'));
end
if eksen2==1
    A2x=str2double(get(handles.edit222,'string'));
end
if eksen2==2
    A2x=str2double(get(handles.edit222,'string'));
    A2y=str2double(get(handles.edit223,'string'));
end
if eksen2==3
    A2x=str2double(get(handles.edit222,'string'));
    A2y=str2double(get(handles.edit223,'string'));
    A2z=str2double(get(handles.edit224,'string'));
end
if eksen3==1
    A3x=str2double(get(handles.edit225,'string'));
end
if eksen3==2
    A3x=str2double(get(handles.edit225,'string'));
    A3y=str2double(get(handles.edit226,'string'));
end
if eksen3==3
    A3x=str2double(get(handles.edit225,'string'));
    A3y=str2double(get(handles.edit226,'string'));
    A3z=str2double(get(handles.edit227,'string'));
end
if eksen1==1
    assignin('base','A1x',A1x)
end
if eksen1==2
    assignin('base','A1x',A1x)
    assignin('base','A1y',A1y)
end
if eksen1==3
    assignin('base','A1x',A1x)
    assignin('base','A1y',A1y)
    assignin('base','A1z',A1z)
end
if eksen2==1
    assignin('base','A2x',A2x)
end
if eksen2==2
    assignin('base','A2x',A2x)

```

```

    assignin('base','A2y',A2y)
end
if eksen2==3
    assignin('base','A2x',A2x)
    assignin('base','A2y',A2y)
    assignin('base','A2z',A2z)
end
if eksen3==1
    assignin('base','A3x',A3x)
end
if eksen3==2
    assignin('base','A3x',A3x)
    assignin('base','A3y',A3y)
end
if eksen3==3
    assignin('base','A3x',A3x)
    assignin('base','A3y',A3y)
    assignin('base','A3z',A3z)
end
end
.
.
.
GrafikSecimi
close(lvmeSensorDegerleri)
function pushbutton3_Callback(hObject, eventdata, handles)
M=xlsread('VeriGirisi.xlsx');
    A1x=M(1,1);
    A1y=M(2,1);
    A1z=M(3,1);
    A2x=M(1,2);
    A2y=M(2,2);
    A2z=M(3,2);
    A3x=M(1,3);
    A3y=M(2,3);
    A3z=M(3,3);
    A4x=M(1,4);
    A4y=M(2,4);
    A4z=M(3,4);
    A5x=M(1,5);
    A5y=M(2,5);
    A5z=M(3,5);
    A6x=M(1,6);
    A6y=M(2,6);
    A6z=M(3,6);
    A7x=M(1,7);
    A7y=M(2,7);
    A7z=M(3,7);
    A8x=M(1,8);
    A8y=M(2,8);
    A8z=M(3,8);
    A9x=M(1,9);
    A9y=M(2,9);
    A9z=M(3,9);
    A10x=M(1,10);

```

```

A10y=M(2,10);
A10z=M(3,10);
A11x=M(1,11);
A11y=M(2,11);
A11z=M(3,11);
A12x=M(1,12);
A12y=M(2,12);
A12z=M(3,12);
set(handles.edit219,'String',num2str(A1x));
set(handles.edit220,'String',num2str(A1y));
set(handles.edit221,'String',num2str(A1z));
set(handles.edit222,'String',num2str(A2x));
set(handles.edit223,'String',num2str(A2y));
set(handles.edit224,'String',num2str(A2z));
set(handles.edit225,'String',num2str(A3x));
set(handles.edit226,'String',num2str(A3y));
set(handles.edit227,'String',num2str(A3z));
set(handles.edit228,'String',num2str(A4x));
set(handles.edit229,'String',num2str(A4y));
set(handles.edit230,'String',num2str(A4z));
set(handles.edit231,'String',num2str(A5x));
set(handles.edit232,'String',num2str(A5y));
set(handles.edit233,'String',num2str(A5z));
set(handles.edit234,'String',num2str(A6x));
set(handles.edit235,'String',num2str(A6y));
set(handles.edit236,'String',num2str(A6z));
set(handles.edit237,'String',num2str(A7x));
set(handles.edit238,'String',num2str(A7y));
set(handles.edit239,'String',num2str(A7z));
set(handles.edit240,'String',num2str(A8x));
set(handles.edit241,'String',num2str(A8y));
set(handles.edit242,'String',num2str(A8z));
set(handles.edit243,'String',num2str(A9x));
set(handles.edit244,'String',num2str(A9y));
set(handles.edit245,'String',num2str(A9z));
set(handles.edit246,'String',num2str(A10x));
set(handles.edit247,'String',num2str(A10y));
set(handles.edit248,'String',num2str(A10z));
set(handles.edit249,'String',num2str(A11x));
set(handles.edit250,'String',num2str(A11y));
set(handles.edit251,'String',num2str(A11z));
set(handles.edit252,'String',num2str(A12x));
set(handles.edit253,'String',num2str(A12y));
set(handles.edit254,'String',num2str(A12z));
.
.
.

```

5. Açısıl İvme (Grafikler)

```

function varargout = Acisallvme(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...

```

```

        'gui_Singleton', gui_Singleton, ...
        'gui_OpeningFcn', @Acisallvme_OpeningFcn, ...
        'gui_OutputFcn', @Acisallvme_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Acisallvme_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = Acisallvme_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
sensorsayisi=evalin('base','sensorsayisi');
if sensorsayisi==3
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
end
if sensorsayisi==4
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
    u4x=evalin('base','u4x');
    u4y=evalin('base','u4y');
    u4z=evalin('base','u4z');
end
if sensorsayisi==5
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
    u4x=evalin('base','u4x');

```

```

    u4y=evalin('base','u4y');
    u4z=evalin('base','u4z');
    u5x=evalin('base','u5x');
    u5y=evalin('base','u5y');
    u5z=evalin('base','u5z');
end
.
.
.
.
if sensorsayisi==6
    if eksen1==1
        L=1;
        assignin('base','L',L)
        U=L*[u1x u2x u3x u4x u5x u6x;u1y u2y u3y u4y u5y u6y; u1z u2z u3z u4z u5z u6z];
        assignin('base','U',U)
        J2=(1/sqrt(2))*[teta1ax teta2ax teta3ax teta4ax teta5ax teta6ax; teta1ay teta2ay teta3ay teta4ay
teta5ay teta6ay; teta1az teta2az teta3az teta4az teta5az teta6az];
        assignin('base','J2',J2)
        J1=[cross(U(:,1),J2(:,1)) cross(U(:,2),J2(:,2)) cross(U(:,3),J2(:,3)) cross(U(:,4),J2(:,4)) cross(U(:,5),J2(:,5))
cross(U(:,6),J2(:,6))];
        t=0;
        for vv=1:1000
wdot(:,vv)=(1/(2*(sqrt(2))*L))*(J1*[A1x(vv);A2x(vv);A3x(vv);A4x(vv);A5x(vv);A6x(vv)]);
            wxdot(vv)=wdot(1,vv);
            wydot(vv)=wdot(2,vv);
            wzdot(vv)=wdot(3,vv);

            t=t+1;
            tt(vv)=t;
        end
        subplot(3,1,1)
        plot(tt,wxdot,'r');
        title('wxdot','FontSize',12)
        hold on;
        subplot(3,1,2)
        plot(tt,wydot,'g');
        title('wydot','FontSize',12)
        hold on;
        subplot(3,1,3)
        plot(tt,wzdot,'b');
        title('wzdot','FontSize',12)
        hold on;
.
.
.

function varargout = GrafikSecimi(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @GrafikSecimi_OpeningFcn, ...
    'gui_OutputFcn', @GrafikSecimi_OutputFcn, ...
    'gui_LayoutFcn', [], ...

```

```

        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function GrafikSecimi_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = GrafikSecimi_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function popupmenu1_Callback(hObject, eventdata, handles)
function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
function pushbutton1_Callback(hObject, eventdata, handles)
DogrusalKonum
function pushbutton2_Callback(hObject, eventdata, handles)
DogrusalHiz
function pushbutton3_Callback(hObject, eventdata, handles)
LineerIvme
function pushbutton6_Callback(hObject, eventdata, handles)
AcisalKonum
function pushbutton7_Callback(hObject, eventdata, handles)
AcisalHiz
function pushbutton8_Callback(hObject, eventdata, handles)
Acisallvme

```

6. Doğrusal İvme (Grafikler)

```

function varargout = Acisallvme(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Acisallvme_OpeningFcn, ...
    'gui_OutputFcn', @Acisallvme_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Acisallvme_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = Acisallvme_OutputFcn(hObject, eventdata, handles)

```

```

varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
sensorsayisi=evalin('base','sensorsayisi');
if sensorsayisi==3
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
end
if sensorsayisi==4
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
    u4x=evalin('base','u4x');
    u4y=evalin('base','u4y');
    u4z=evalin('base','u4z');
end
if sensorsayisi==5
    u1x=evalin('base','u1x');
    u1y=evalin('base','u1y');
    u1z=evalin('base','u1z');
    u2x=evalin('base','u2x');
    u2y=evalin('base','u2y');
    u2z=evalin('base','u2z');
    u3x=evalin('base','u3x');
    u3y=evalin('base','u3y');
    u3z=evalin('base','u3z');
    u4x=evalin('base','u4x');
    u4y=evalin('base','u4y');
    u4z=evalin('base','u4z');
    u5x=evalin('base','u5x');
    u5y=evalin('base','u5y');
    u5z=evalin('base','u5z');
end
.
.
.
if sensorsayisi==6
    if eksen1==1
        L=1;
        assignin('base','L',L)
        U=L*[u1x u2x u3x u4x u5x u6x;u1y u2y u3y u4y u5y u6y; u1z u2z u3z u4z u5z u6z];
        assignin('base','U',U)
    end
end

```

```

J2=(1/sqrt(2))*[teta1ax teta2ax teta3ax teta4ax teta5ax teta6ax; teta1ay teta2ay teta3ay teta4ay
teta5ay teta6ay; teta1az teta2az teta3az teta4az teta5az teta6az];
assignin('base','J2',J2)
J1=[cross(U(:,1),J2(:,1)) cross(U(:,2),J2(:,2)) cross(U(:,3),J2(:,3)) cross(U(:,4),J2(:,4)) cross(U(:,5),J2(:,5))
cross(U(:,6),J2(:,6))];
t=0;
for vv=1:2000
wdot(:,vv)=(1/(2*(sqrt(2))*L))*(J1*[A1x(vv);A2x(vv);A3x(vv);A4x(vv);A5x(vv);A6x(vv)]);
wxdot(vv)=wdot(1,vv);
wydot(vv)=wdot(2,vv);
wzdot(vv)=wdot(3,vv);
t=t+1;
tt(vv)=t;
end
t2=0;
for cc=1:2000
P(:,cc)=(1/(2*(sqrt(2))))*(J2*[A1x(cc);A2x(cc);A3x(cc);A4x(cc);A5x(cc);A6x(cc)]+L*[wydot(cc)*wzdot(cc);
wxdot(cc)*wzdot(cc);wxdot(cc)*wydot(cc)]);
assignin('base','P',P)
Px(cc)=P(1,cc);
Py(cc)=P(2,cc);
Pz(cc)=P(3,cc);
t2=t2+1;
tt(cc)=t2;
end
subplot(3,1,1)
plot(tt,Px,'r');
title('X Yonundeki Ivme','FontSize',12)
grid
hold on;
subplot(3,1,2)
plot(tt,Py,'g');
title('Y Yonundeki Ivme','FontSize',12)
grid
hold on;
subplot(3,1,3)
plot(tt,Pz,'b');
title('Z Yonundeki Ivme','FontSize',12)
grid
hold on;
end
assignin('base','Px',Px)
assignin('base','Py',Py)
assignin('base','Pz',Pz)

end
.
.
.

```

7. Doğrusal Hiz

```

function varargout = LineerHiz(varargin)
gui_Singleton = 1;

```

```

gui_State = struct('gui_Name',    mfilename, ...
                 'gui_Singleton', gui_Singleton, ...
                 'gui_OpeningFcn', @LineerHiz_OpeningFcn, ...
                 'gui_OutputFcn', @LineerHiz_OutputFcn, ...
                 'gui_LayoutFcn', [], ...
                 'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function LineerHiz_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = LineerHiz_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
    Pxx=evalin('base','Pxx');
    Pyy=evalin('base','Pyy');
    Pzz=evalin('base','Pzz');
    t2=0;
    for cc=1:2000
        t2=t2+1;
        tt(cc)=t2;
    end
    subplot(3,1,1)
    plot(tt,Pxx,'r');
    title('X Yonundeki Hiz','FontSize',12)
    grid
    hold on;
    subplot(3,1,2)
    plot(tt,Pyy,'g');
    title('Y Yonundeki Hiz','FontSize',12)
    grid
    hold on;
    subplot(3,1,3)
    plot(tt,Pzz,'b');
    title('Z Yonundeki Hiz','FontSize',12)
    grid

```

8. Doğrusal Konum

```

function varargout = LineerKonum(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                 'gui_Singleton', gui_Singleton, ...
                 'gui_OpeningFcn', @LineerKonum_OpeningFcn, ...
                 'gui_OutputFcn', @LineerKonum_OutputFcn, ...
                 'gui_LayoutFcn', [], ...
                 'gui_Callback', []);
if nargin && ischar(varargin{1})

```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function LineerKonum_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = LineerKonum_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
Pxxx=evalin('base','Pxxx');
Pyyy=evalin('base','Pyyy');
Pzzz=evalin('base','Pzzz');
t2=0;
for cc=1:2000
t2=t2+1;
tt(cc)=t2;
end
subplot(3,1,1)
plot(tt,Pxxx,'r');
title('X Yonundeki Konum','FontSize',12)
grid
hold on;
subplot(3,1,2)
plot(tt,Pyyy,'g');
title('Y Yonundeki Konum','FontSize',12)
grid
hold on;
subplot(3,1,3)
plot(tt,Pzzz,'b');
title('Z Yonundeki Konum','FontSize',12)
grid
hold on;

```

9. Açısız Hız

```

function varargout = LineerHiz(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @LineerHiz_OpeningFcn, ...
    'gui_OutputFcn', @LineerHiz_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});

```

```

else
    gui_mainfcn(gui_State, varargin{:});
end
function LineerHiz_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = LineerHiz_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
function pushbutton1_Callback(hObject, eventdata, handles)
    wdotxx=evalin('base','wdotxx');
    wdotyy=evalin('base','wdotyy');
    wdotzz=evalin('base','wdotzz');
    t2=0;
    for cc=1:2000
        t2=t2+1;
        tt(cc)=t2;
    end
    subplot(3,1,1)
    plot(tt, wdotxx,'r');
    title('X Eksenindeki Hiz','FontSize',12)
    grid
    hold on;
    subplot(3,1,2)
    plot(tt, wdotyy,'g');
    title('Y Eksenindeki Hiz','FontSize',12)
    grid
    hold on;
    subplot(3,1,3)
    plot(tt, wdotzz,'b');
    title('Z Eksenindeki Hiz','FontSize',12)
    grid

```

10. Açısıl Konum

```

function varargout = LineerKonum(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @LineerKonum_OpeningFcn, ...
    'gui_OutputFcn', @LineerKonum_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function LineerKonum_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
function varargout = LineerKonum_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```

```
function pushbutton1_Callback(hObject, eventdata, handles)
    wdotxxx=evalin('base','wdotxxx');
    wdotyyy=evalin('base','wdotyyy');
    wdotzzz=evalin('base','wdotzzz');
    t2=0;
    for cc=1:2000
        t2=t2+1;
        tt(cc)=t2;
    end
    subplot(3,1,1)
    plot(tt, wdotxxx,'r');
    title('X Eksenindeki Konum','FontSize',12)
    grid
    hold on;
    subplot(3,1,2)
    plot(tt, wdotyyy,'g');
    title('Y Eksenindeki Konum','FontSize',12)
    grid
    hold on;
    subplot(3,1,3)
    plot(tt, wdotzzz,'b');
    title('Z Eksenindeki Konum','FontSize',12)
    grid
    hold on;
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Doğan AYDENİZ
Doğum Tarihi ve Yeri :21.08.1983 / İstanbul
Yabancı Dili :İngilizce - Almanca
E-posta :doganaydeniz@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Makine Mühendisliği	Yıldız Teknik Üniversitesi	-
Lisans	Makine Mühendisliği	Karadeniz Teknik Üniversitesi	2008
Lise	Üsküdar Anadolu Lisesi		

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2009	ERMAK Beton ve İnşaat Makineleri San. Ve Tic. LTD. ŞTİ.	Makine Mühendisi

YAYINLARI

Bildiri

1. Modeling a Accelerometer Based Inclinometer by Artificial Neural Networks **2011**