

**T.C. İSTANBUL KÜLTÜR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HESAPLAMA PROGRAMLARININ GRAFİKSEL KULLANICI ARAYÜZÜ (GUI)
HATALARI PERSPEKTİFİNDE İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Alpay ENGİN

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

HAZİRAN 2005

**T.C. İSTANBUL KÜLTÜR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HESAPLAMA PROGRAMLARININ GRAFİKSEL KULLANICI ARAYÜZÜ (GUI)
HATALARI PERSPEKTİFİNDE İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Alpay ENGİN

(0309050002)

**Tezin Enstitüye Verildiği Tarih : 7 Haziran 2005
Tezin Savunulduğu Tarih : 9 Haziran 2005**

Tez Danışmanı : Doç.Dr. Servet BAYRAM

Diğer Jüri Üyeleri : Yard.Doç.Dr. Kemal Yüksek

Yard.Doç.Dr. Ertuğrul Saatçi

HAZİRAN 2005

İSTANBUL KÜLTÜR ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü
BİLGİSAYAR MÜHENDİSLİĞİ

HESAPLAMA PROGRAMLARININ
GRAFİKSEL KULLANICI ARAYÜZÜ
(GUI) HATALARI
PERSPEKTİFİNDE İNCELENMESİ

Yüksek Lisans Tezi

ALPAY ENGİN

No: 0309050002

Haziran-2005

**T.C. İSTANBUL KÜLTÜR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HESAPLAMA PROGRAMLARININ GRAFİKSEL KULLANICI ARAYÜZÜ (GUI)
HATALARI PERSPEKTİFİNDE İNCELENMESİ**

YÜKSEK LİSANS TEZİ

ALPAY ENGİN

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

Tez Danışmanı : Doç. Dr. Servet BAYRAM

HAZİRAN 2005

ÖNSÖZ

Bu tez çalışması sırasında bana her türlü konuda yardımcı olan, deneyimlerini ve tecrübelerini aktararak bana yol gösteren, geçen günler ve aylar içerisinde çalışma ruhunu aşilayarak "*çalışma sırasında disiplin*" ilkesiyle beni de eğitmeye çalışan; her sorunumla yakından ilgilenip, çözen; her yol ayrımında sağduyusu ve deneyimini kullanıp beni yönlendiren; "*bilim adamlığının bir meslek değil, yaşam biçimi olduğu*" fikrini somutlaştıran ve tezde büyük emeği bulunan sevgili hocam ve tez danışmanım Doç. Dr. Servet BAYRAM'a yardımlarından dolayı teşekkürü bir borç bilirim.

Tez çalışmasına ilişkin bilgilerimi temellendiren; her sorunumla değerli zamanını esirgemeksizin ilgilenen, bilgi ve tecrübelerini aktararak tezin her aşamasında yardımcı olan ve tezin hazırlanmasında emeği geçen sevgili hocam Yrd. Doç. Dr. Kemal YÜKSEK'e içtenlikle teşekkür ederim.

Tezin oluşumunda ve eğitim hayatımda büyük katkıları bulunan, okuduğum bölüm olan İstanbul Kültür Üniversitesi Bilgisayar Mühendisliği Bölümü'ne, özellikle Bilgisayar Mühendisliği Bölüm Başkanı Prof. Dr. Murat TAYLI başta olmak üzere, bütün öğretim üyelerine teşekkür gönül borcumdur.

Çok sevdiğim üniversitem, bir ferdi olmaktan çok mutlu olduğum ikinci yuvam İstanbul Kültür Üniversitesi'ne, oluşturduğu altyapı ve sağladığı laboratuvar olanakları için minnettarım.

Tezin başından sonuna kadar bana destek olan, her türlü sorunumda her zaman yanımda olan, kendilerine ait olan zamanlarını bana ayıran değerli aileme; başta babam Şenol ENGİN, annem Ülkü ENGİN ve sevgili babaannem Gülsüm ENGİN olmak üzere, abim Ali ENGİN ve kardeşim Mert ENGİN'e teşekkürlerimi ve şükranlarımı sunarım.

İstanbul, Haziran 2005

Alpay ENGİN

İÇİNDEKİLER

1. GİRİŞ	1
2. LİTERATÜR	11
2.1. Hesaplama Programları	11
2.2. MATLAB Programı	11
2.2.1 MATLAB'ın Karakteristik Özellikleri	15
2.2.2 MATLAB'ın Kullanım Alanları.....	16
2.2.3 MATLAB'ın Yapısı	16
2.2.4 MATLAB Programı'nın Kullanılma Amacı	18
2.2.5 MATLAB Elemanları ve Bileşenleri	19
2.2.5.1 MATLAB Programlama Dili.....	19
2.2.5.2 MATLAB Çalışma Alanı	19
2.2.5.3 Grafik Yönetimi.....	20
2.2.5.4 Modelleme & Simülasyon	20
2.2.5.5 MATLAB Araç Kutuları (Toolboxes)	20
2.2.6 MATLAB Platformu/Arayüzü'nün Genel Görünümü	21
2.2.6.1 Çalışma Alanı Penceresi (Workspace).....	22
2.2.6.2 Geçerli Dizin Penceresi (Current Directory Window).....	22
2.2.6.3 Yeni Bir Eylem Başlatma Padi (Launch Pad).....	22
2.2.6.4 Geçmişteki Komutlar Penceresi (Command History Window).....	23
2.2.6.5 Komut Penceresi (Command Window).....	23
2.2.6.6 Yardım Penceresi (Help Window)	24
2.2.6.7 Şekil Penceresi (Figure Window)	24
2.2.7 MATLAB'te Programlama Kavramı	24
2.2.7.1 MATLAB'te Programlamaya Genel Bakış.....	24
2.2.7.2 MATLAB'te Programlamanın Yapısı.....	25
2.2.7.3 MATLAB'te Programlama ile İlgili Özellikler.....	25
2.2.7.4 MATLAB'te Operatörler (Operators)	26
2.2.7.4.1 Aritmetik Operatörler	26
2.2.7.4.2 Mantıksal Operatörler	26
2.2.7.4.3 Karşılaştırma Operatörleri	27
2.2.7.5 MATLAB'te Programlama ile İlgili Kurallar	27
2.2.7.6 İfadeler (Expressions)	27
2.2.7.6.1 Sayılar	27
2.2.7.6.2 Değişkenler	28
2.2.7.6.3 Fonksiyonlar.....	28
2.2.7.7 MATLAB Temel Yazım Notasyonu (Syntax)	28
2.2.7.8 Diziler (Array).....	29
2.2.7.9 Açıklayıcı Yorumlar (Comment)	29
2.2.8 MATLAB'te Grafikselle Kullanıcı Arayüzü (GUI) Oluşturma.....	29
2.2.9 MATLAB Anket Araştırma Sonuçları	30
2.3. MathCAD Programı	39
2.3.1 MathCAD Programı'nın Önemi.....	43
2.3.2 MathCAD Programının Genel Özellikleri.....	44
2.3.3 MathCAD'ın Kullanım Alanları	45
2.3.4 MathCAD Arayüz Çalışma Alanı.....	46

2.3.5	MathCAD Araç Çubukları (Toolbars)	47
2.3.5.1	Grafik Araç Çubuğu	47
2.3.5.2	Hesaplama Araç Çubuğu.....	48
2.3.5.3	Değerlendirme Araç Çubuğu	48
2.3.5.4	Aritmetik Araç Çubuğu	48
2.3.5.5	Boolean Araç Çubuğu	48
2.3.5.6	Vektör (vector) ve Matris (matrix) Araç Çubuğu	48
2.3.5.7	Programlama Araç Çubuğu	48
2.3.5.8	Sembolik Araç Çubuğu	48
2.3.6	MathCAD Programı'nın Temel Yapısı.....	48
2.3.6.1	Matematiksel İşlemler	49
2.3.6.2	Grafik Oluşturma	49
2.3.6.3	Programlama	49
2.3.6.4	Mühendislik Uygulamaları Geliştirme	49
2.3.6.5	Birimler.....	50
2.3.6.6	Denklemlerin Çözümü	50
2.3.6.7	Fonksiyonlar Oluşturma ve Çözme	50
2.3.6.8	Sembolik Hesaplamalar	50
2.3.6.9	Matrix (Matris) / Vector (Vektör) İşlemleri.....	51
2.3.6.10	Veri Analizi	51
2.3.6.11	Model Oluşturma / Dizayn	51
2.3.6.12	Fonksiyonel Özellikler	51
2.3.6.13	Kullanışlılık Özellikleri.....	52
2.3.7	MathCAD'te Programlama Geliştirmek	52
2.3.8	MathCAD Programı'nın Tercih Edilme Sebepleri.....	54
2.3.9	MathCAD Kullanıcısı Olmanın Avantajları	54
2.3.10	MathCAD Anket Araştırma Sonuçları.....	56
2.4.	Grafiksel Kullanıcı Arayüzü (GUI)	58
2.4.1	Grafiksel Kullanıcı Arayüzü (GUI) Dizaynının Amacı.....	60
2.4.1.1	Kullanıcı açısından önemi.....	61
2.4.1.2	Arayüzü geliştiren kişi ya da kuruluşlar açısından önemi	61
2.4.2	Grafiksel Kullanıcı Arayüz (GUI) Modeli.....	62
2.4.2.1	Birinci Kat : Arayüzün Yapısı.....	63
2.4.2.1.1	<i>Birinci Katman : Yapısal Model</i>	63
2.4.2.1.2	<i>İkinci Katman : Görev Akışı ve Çizelgesi</i>	63
2.4.2.1.3	<i>Üçüncü Katman : Organizasyon Modeli</i>	63
2.4.2.2	İkinci Kat : Arayüzün İşleyişi	63
2.4.2.2.1	<i>Dördüncü Katman : Görünüm</i>	64
2.4.2.2.2	<i>Beşinci Katman : Düzeltme ve İdare</i>	64
2.4.2.2.3	<i>Altıncı Katman : Kullanıcı Yardımı</i>	64
2.4.2.3	Üçüncü Kat : Arayüz Sunumu	64
2.4.2.3.1	<i>Yedinci Katman : Düzen</i>	65
2.4.2.3.2	<i>Sekizinci Katman : Biçim</i>	65
2.4.2.3.3	<i>Dokuzuncu Katman : Metin</i>	65
2.4.2.4	Grafiksel Kullanıcı Arayüz (GUI) Modeli'nin Önemli Noktaları..	65
2.4.2.4.1	<i>Esneklik</i>	65
2.4.2.4.2	<i>Kullanıcı Bilinçsizliği</i>	65
2.4.2.4.3	<i>Kullanışlılık</i>	66
2.4.3	Kullanıcı Arayüz Dizaynının Temel İskeleti ve Mimarisi.....	66
2.4.3.1	Görsel Dizayn	66

2.4.3.2	Arayüz Dizaynı	66
2.4.3.3	Bilgi Dizaynı	67
2.4.3.4	Etkileşim Dizaynı.....	67
2.4.3.5	Bilgi Mimarisi	67
2.4.3.6	Fonksiyonel Tanımlamalar	67
2.4.3.7	İçerik İstekleri.....	67
2.4.3.8	Kullanıcı İhtiyaçları	67
2.4.4	Kullanıcı Arayüz Dizaynının Yapısı.....	67
2.4.4.1	Kullanılabilirlik	67
2.4.4.2	Görsellik	68
2.4.4.3	Fonksiyonellik	68
2.4.4.4	Ulaşılabilirlik.....	68
2.4.5	Grafiksel Kullanıcı Arayüz Elemanları.....	68
2.4.5.1	Düğme (Button).....	68
2.4.5.2	İkon (icon)	69
2.4.5.3	Menü (menu).....	69
2.4.5.4	Denetim Kutusu (Check Box).....	69
2.4.5.5	Seçenek Düğmesi (OptionButton).....	70
2.4.5.6	Grup Kutusu (Combo Box)	70
2.4.5.7	Listeleme Kutusu (List Box)	70
2.4.5.8	Metin Alanı (TextField).....	71
2.4.5.9	Etiket (Label)	71
2.4.6	Grafiksel Kullanıcı Arayüz (GUI) Dizayn Teknikleri.....	71
2.4.6.1	Kullanıcının Etkinliği.....	72
2.4.6.2	Renk Körlüğü Problemi.....	72
2.4.6.3	Arayüzde Tutarlılık.....	73
2.4.6.4	Arayüzün Kullanıcı Hassasiyeti	73
2.4.6.5	Arayüzde Kullanıcıya Yol Gösterici Olmak	73
2.4.6.6	Arayüzde Açıklık ve Sadelik İlkesini Benimsemek	74
2.4.6.7	Tekrarlama ve Yineleme'lerden Kaçınmak.....	74
2.4.6.8	Kullanım Kolaylığı	75
2.4.6.9	Kullanıcı Profili'ni Belirlemek	75
2.4.6.10	Ekran Yönetimi	76
2.4.6.11	Ekran Organizasyonu	76
2.4.6.12	Mesajları Uygun Bir Şekilde İfade Etmek.....	77
2.4.6.13	Arayüz Elemanlarını Doğru Kullanmak.....	77
2.4.6.14	Renk Kullanımına Dikkat Etmek	77
2.4.6.15	Tezatlık Kuralını Benimsemek	78
2.4.6.16	Metin Fontlarının Uygunluğunu Sağlamak.....	78
2.4.6.17	Arayüz Elemanlarının Düzen ve Hizalarına Önem Vermek	78
2.4.6.18	Kalabalık Arayüz Ekranı Yaratmaktan Kaçınmak	79
2.4.6.19	Klavyeden (Keyboard) Kısa Yollar Sunmak	79
2.4.6.20	Dalgınlıklardan Kaçınmak.....	80
2.4.6.21	Uyarı ve Hata Mesajları Kullanmak	80
2.4.6.22	Bilgiye Geri Dönüşüm Sağlamak.....	80
2.4.6.23	Ses, Resim, Grafik ve Animasyon Kullanımı.....	81
2.4.6.24	Eş Anımlı Kelimeler & Benzerlikler'den Kaçınmak	81
2.4.7	Grafiksel Kullanıcı Arayüz (GUI) Dizayn Prensipleri.....	82
2.4.7.1	İhtiyaçlar Aşaması.....	84
2.4.7.1.1	Analiz	84

2.4.7.1.2 Arayüz Elemanlarının Dizaynı ve Oluşturulması.....	85
2.4.7.2 Yapısal Dizayn	85
2.4.7.3 Mantıksal Dizayn	86
2.4.7.4 Fiziksel Dizayn	86
2.4.7.5 Oluşturma.....	86
2.4.7.5.1 Programlama Dili.....	87
2.4.7.6 Kullanışlılık Testi.....	87
2.4.8 Grafikselle Kullanıcı Arayüzü (GUI) Hataları	89
2.4.8.1 Grafikselle Kullanıcı Arayüz Elemanlarında Yapılan Hatalar	89
2.4.8.2 Kullanıcı Arayüzünün Düzen ve Görünüşünde Yapılan Hatalar ..	90
2.4.8.3 Metinsel Hatalar	90
2.4.8.4 Etkileşim Hataları.....	91
3. METODOLOJİ.....	94
3.1. Kalite Ölçüm Kontrol Listesi.....	94
3.1.1 Kaliteyi Etkileyen Faktörler	95
3.1.2 Kalite Ölçüm Kriterleri.....	95
3.2. GUI Hataları Ölçüm Kontrol Listesi.....	97
4. SONUÇLAR	99
4.1. MATLAB Kalite Ölçüm Kontrol Listesi Sonuçları	99
4.2. MathCAD Kalite Ölçüm Kontrol Listesi Sonuçları.....	101
4.3. MATLAB & MathCAD GUI Hataları Ölçüm Kontrol Listesi	
Sonuçları.....	102
4.4. MATLAB'teki GUI Hataları & Hataların Doğruları.....	104
4.4.1 MATLAB'te GUI Elemanlarında Yapılan Hatalar (GUI Component	
Bloopers)	105
4.4.2 MATLAB'teki Düzen ve Görünüm Hataları (Layout & Appearance	
Bloopers)	116
4.4.3 MATLAB'teki Metinsel Hatalar (Textual Bloopers)	143
4.4.4 MATLAB'teki Etkileşim Hataları (Interaction Bloopers).....	152
4.5. MathCAD'teki GUI Hataları & Hataların Doğruları	164
4.5.1 MathCAD'te GUI Elemanlarında Yapılan Hatalar (GUI Component	
Bloopers)	164
4.5.2 MathCAD'teki Düzen ve Görünüm Hataları (Layout & Appearance	
Bloopers)	174
4.5.3 MathCAD'teki Metinsel Hatalar (Textual Bloopers).....	188
4.5.4 MathCAD'teki Etkileşim Hataları (Interaction Bloopers)	193
4.6. Genel Değerlendirme.....	200
5. KAYNAKÇA	202

ŞEKİL LİSTESİ

Şekil 1 : MATLAB'ın Yapısı.....	17
Şekil 2 : MATLAB Platformu / Arayüzü.....	21
Şekil 3 : Grafikselsel Kullanıcı Arayüz Geliştirme Alanı (GUIDE)	30
Şekil 4 : MATLAB Anket Araştırma Sonucu-1	30
Şekil 5 : MATLAB Anket Araştırma Sonucu-2	31
Şekil 6 : MATLAB Anket Araştırma Sonucu-3	31
Şekil 7 : MATLAB Anket Araştırma Sonucu-4	31
Şekil 8 : MATLAB Anket Araştırma Sonucu-5	32
Şekil 9 : MATLAB Anket Araştırma Sonucu-6	32
Şekil 10 : MATLAB Anket Araştırma Sonucu-7	32
Şekil 11 : MATLAB Anket Araştırma Sonucu-8	33
Şekil 12 : MATLAB Anket Araştırma Sonucu-9	33
Şekil 13 : MATLAB Anket Araştırma Sonucu-10	33
Şekil 14 : MATLAB Anket Araştırma Sonucu-11	34
Şekil 15 : MATLAB Anket Araştırma Sonucu-12	34
Şekil 16 : MATLAB Anket Araştırma Sonucu-13	34
Şekil 17 : MATLAB Anket Araştırma Sonucu-14	35
Şekil 18 : MATLAB Anket Araştırma Sonucu-15	35
Şekil 19 : MATLAB Anket Araştırma Sonucu-16	35
Şekil 20 : MATLAB Anket Araştırma Sonucu-17	36
Şekil 21 : MATLAB Anket Araştırma Sonucu-18	36
Şekil 22 : MATLAB Anket Araştırma Sonucu-19	36
Şekil 23 : MATLAB Anket Araştırma Sonucu-20	37
Şekil 24 : MATLAB Anket Araştırma Sonucu-21	37
Şekil 25 : MATLAB Anket Araştırma Sonucu-22	37
Şekil 26 : MATLAB Anket Araştırma Sonucu-23	38
Şekil 27 : MATLAB Anket Araştırma Sonucu-24	38
Şekil 28 : MATLAB Anket Araştırma Sonucu-25	38
Şekil 29 : MathCAD Çalışma Alanı	47
Şekil 30 : MathCAD Anket Araştırma Sonucu-1.....	56
Şekil 31 : MathCAD Anket Araştırma Sonucu-2.....	56
Şekil 32 : MathCAD Anket Araştırma Sonucu-3.....	57

Şekil 33 : MathCAD Anket Araştırma Sonucu-4.....	57
Şekil 34 : MathCAD Anket Araştırma Sonucu-5.....	57
Şekil 35 : İnsan - Bilgisayar Etkileşimi	60
Şekil 36 : Kullanıcı Arayüz Modeli	62
Şekil 37 : Kullanıcı Arayüz Dizayn'ının Temel İskeleti ve Mimarisi.....	66
Şekil 38 : Düğme (Button)	68
Şekil 39 : İkon (icon)	69
Şekil 40 : Menü (menu).....	69
Şekil 41 : Denetim Kutusu (Check Box)	70
Şekil 42 : Seçenek Düğmesi (OptionButton).....	70
Şekil 43 : Grup Kutusu (Combo Box)	70
Şekil 44 : Listeleme Kutusu (List Box).....	71
Şekil 45 : Metin Alanı (TextField)	71
Şekil 46 : Etiket (Label)	71
Şekil 47 : Waterfall (Şelale) Modeli.....	82
Şekil 48 : Sürekli Tekrar Halindeki Dizayn İşlemleri	83
Şekil 49 : Yapısal Dizayn	85
Şekil 50 : MATLAB GUI Hatası-1	105
Şekil 51 : MATLAB GUI Hatası-2	106
Şekil 52 : MATLAB GUI Hatası-3	107
Şekil 53 : MATLAB GUI Hatası-3'ün Doğrusu.....	108
Şekil 54 : MATLAB GUI Hatası-4	109
Şekil 55 : MATLAB GUI Hatası-4'ün Doğrusu.....	109
Şekil 56 : MATLAB GUI Hatası-5	110
Şekil 57 : MATLAB GUI Hatası-5'in Doğrusu.....	111
Şekil 58 : MATLAB GUI Hatası-6	112
Şekil 59 : MATLAB GUI Hatası-7'nin Birinci Arayüzü.....	113
Şekil 60 : MATLAB GUI Hatası-7'nin İkinci Arayüzü	114
Şekil 61 : MATLAB GUI Hatası-8	115
Şekil 62 : MATLAB GUI Hatası-8'in Doğrusu.....	115
Şekil 63 : MATLAB GUI Hatası-9	116
Şekil 64 : MATLAB GUI Hatası-9'un Doğrusu.....	117
Şekil 65 : MATLAB GUI Hatası-10	118
Şekil 66 : MATLAB GUI Hatası-10'un Doğrusu.....	119

Şekil 67 : MATLAB GUI Hatası-11	120
Şekil 68 : MATLAB GUI Hatası-11'in Doğrusu.....	121
Şekil 69 : MATLAB GUI Hatası-12	122
Şekil 70 : MATLAB GUI Hatası-12'nin Doğrusu.....	123
Şekil 71 : MATLAB GUI Hatası-13	124
Şekil 72 : MATLAB GUI Hatası-13'ün Doğrusu.....	125
Şekil 73 : MATLAB GUI Hatası-14	126
Şekil 74 : MATLAB GUI Hatası-14'ün Doğrusu.....	126
Şekil 75 : MATLAB GUI Hatası-15	127
Şekil 76 : MATLAB GUI Hatası-15'in Doğrusu.....	127
Şekil 77 : MATLAB GUI Hatası-16	128
Şekil 78 : MATLAB GUI Hatası-16'nın Doğrusu.....	128
Şekil 79 : MATLAB GUI Hatası-17	128
Şekil 80 : MATLAB GUI Hatası-17'nin Doğrusu.....	129
Şekil 81 : MATLAB GUI Hatası-18	130
Şekil 82 : MATLAB GUI Hatası-18'in Doğrusu.....	131
Şekil 83 : MATLAB GUI Hatası-19	132
Şekil 84 : MATLAB GUI Hatası-19'un Doğrusu.....	132
Şekil 85 : MATLAB GUI Hatası-20	133
Şekil 86 : MATLAB GUI Hatası-20'nin Doğrusu.....	134
Şekil 87 : MATLAB GUI Hatası-21	135
Şekil 88 : MATLAB GUI Hatası-21'in Doğrusu.....	135
Şekil 89 : MATLAB GUI Hatası-22	136
Şekil 90 : MATLAB GUI Hatası-23	137
Şekil 91 : MATLAB GUI Hatası-23'ün Doğrusu.....	138
Şekil 92 : MATLAB GUI Hatası-24	139
Şekil 93 : MATLAB GUI Hatası-24'ün Doğrusu.....	140
Şekil 94 : MATLAB GUI Hatası-25	141
Şekil 95 : MATLAB GUI Hatası-26	142
Şekil 96 : MATLAB GUI Hatası-27	143
Şekil 97 : MATLAB GUI Hatası-28	144
Şekil 98 : MATLAB GUI Hatası-28'in Doğrusu.....	144
Şekil 99 : MATLAB GUI Hatası-29	145
Şekil 100 : MATLAB GUI Hatası-29'un Doğrusu.....	146

Şekil 101 : MATLAB GUI Hatası-30	147
Şekil 102 : MATLAB GUI Hatası-31	148
Şekil 103 : MATLAB GUI Hatası-31'in Doğrusu.....	149
Şekil 104 : MATLAB GUI Hatası-32'nin Birinci Arayüzü.....	150
Şekil 105 : MATLAB GUI Hatası-32'nin İkinci Arayüzü	151
Şekil 106 : MATLAB GUI Hatası-33	152
Şekil 107 : MATLAB GUI Hatası-33'ün Doğrusu.....	152
Şekil 108 : MATLAB GUI Hatası-34	152
Şekil 109 : MATLAB GUI Hatası-34'ün Doğrusu.....	153
Şekil 110 : MATLAB GUI Hatası-35	153
Şekil 111 : MATLAB GUI Hatası-36	153
Şekil 112 : MATLAB GUI Hatası-36'nın Doğrusu.....	154
Şekil 113 : MATLAB GUI Hatası-37	154
Şekil 114 : MATLAB GUI Hatası-37'nin Doğrusu.....	154
Şekil 115 : MATLAB GUI Hatası-38	155
Şekil 116 : MATLAB GUI Hatası-38'in Doğrusu.....	156
Şekil 117 : MATLAB GUI Hatası-39	157
Şekil 118 : MATLAB GUI Hatası-40	158
Şekil 119 : MATLAB GUI Hatası-41	159
Şekil 120 : MATLAB GUI Hatası-41'in Doğrusu.....	160
Şekil 121 : MATLAB GUI Hatası-42	160
Şekil 122 : MATLAB GUI Hatası-42'nin Arayüzü	161
Şekil 123 : MATLAB GUI Hatası-42'nin Doğrusu.....	161
Şekil 124 : MATLAB GUI Hatası-43	162
Şekil 125 : MATLAB GUI Hatası-43'ün Doğrusu.....	162
Şekil 126 : MATLAB GUI Hatası-44	163
Şekil 127 : MathCAD GUI Hatası-1.....	164
Şekil 128 : MathCAD GUI Hatası-1'in Doğrusu	164
Şekil 129 : MathCAD GUI Hatası-2.....	165
Şekil 130 : MathCAD GUI Hatası-3'ün Birinci Arayüzü	166
Şekil 131 : MathCAD GUI Hatası-3'ün İkinci Arayüzü	167
Şekil 132 : MathCAD GUI Hatası-4.....	168
Şekil 133 : MathCAD GUI Hatası-5.....	169
Şekil 134 : MathCAD GUI Hatası-5'in Doğrusu	170

Şekil 135 : MathCAD GUI Hatası-6'nın Birinci Arayüzü	171
Şekil 136 : MathCAD GUI Hatası-6'nın İkinci Arayüzü	171
Şekil 137 : MathCAD GUI Hatası-7.....	172
Şekil 138 : MathCAD GUI Hatası-8.....	172
Şekil 139 : MathCAD GUI Hatası-8'in Doğrusu	173
Şekil 140 : MathCAD GUI Hatası-9.....	173
Şekil 141 : MathCAD GUI Hatası-9'un Doğrusu	173
Şekil 142 : MathCAD GUI Hatası-10.....	174
Şekil 143 : MathCAD GUI Hatası-10'un Doğrusu	175
Şekil 144 : MathCAD GUI Hatası-11.....	176
Şekil 145 : MathCAD GUI Hatası-11'in Doğrusu	177
Şekil 146 : MathCAD GUI Hatası-12.....	178
Şekil 147 : MathCAD GUI Hatası-13.....	179
Şekil 148 : MathCAD GUI Hatası-13'ün Doğrusu	179
Şekil 149 : MathCAD GUI Hatası-14.....	180
Şekil 150 : MathCAD GUI Hatası-14'ün Doğrusu	180
Şekil 151 : MathCAD GUI Hatası-15.....	181
Şekil 152 : MathCAD GUI Hatası-15'in Doğrusu	181
Şekil 153 : MathCAD GUI Hatası-16.....	182
Şekil 154 : MathCAD GUI Hatası-16'nın Doğrusu	182
Şekil 155 : MathCAD GUI Hatası-17.....	183
Şekil 156 : MathCAD GUI Hatası-17'nin Doğrusu	183
Şekil 157 : MathCAD GUI Hatası-18.....	184
Şekil 158 : MathCAD GUI Hatası-18'nin Doğrusu	184
Şekil 159 : MathCAD GUI Hatası-19.....	184
Şekil 160 : MathCAD GUI Hatası-19'un Doğrusu	185
Şekil 161 : MathCAD GUI Hatası-20.....	185
Şekil 162 : MathCAD GUI Hatası-20'nin Doğrusu	185
Şekil 163 : MathCAD GUI Hatası-21.....	186
Şekil 164 : MathCAD GUI Hatası-21'in Doğrusu	187
Şekil 165 : MathCAD GUI Hatası-22.....	188
Şekil 166 : MathCAD GUI Hatası-22'nin Doğrusu	189
Şekil 167 : MathCAD GUI Hatası-23.....	189
Şekil 168 : MathCAD GUI Hatası-23'ün Doğrusu	190

Şekil 169 : MathCAD GUI Hatası-24'ün Birinci Arayüzü	190
Şekil 170 : MathCAD GUI Hatası-24'ün İkinci Arayüzü	191
Şekil 171 : MathCAD GUI Hatası-24'ün Üçüncü Arayüzü	191
Şekil 172 : MathCAD GUI Hatası-25'in Birinci Arayüzü	192
Şekil 173 : MathCAD GUI Hatası-25'in İkinci Arayüzü	192
Şekil 174 : MathCAD GUI Hatası-26.....	193
Şekil 175 : MathCAD GUI Hatası-26'nın Doğrusu	193
Şekil 176 : MathCAD GUI Hatası-27.....	193
Şekil 177 : MathCAD GUI Hatası-27'nin Doğrusu	194
Şekil 178 : MathCAD GUI Hatası-28.....	194
Şekil 179 : MathCAD GUI Hatası-28'in Doğrusu	195
Şekil 180 : MathCAD GUI Hatası-29'un Birinci Arayüzü	196
Şekil 181 : MathCAD GUI Hatası-29'un İkinci Arayüzü	197
Şekil 182 : MathCAD GUI Hatası-30.....	198
Şekil 183 : MathCAD GUI Hatası-30'un Doğrusu	199

TABLO LİSTESİ

Tablo 1 : Aritmetik Operatörler ve Sembolleri	26
Tablo 2 : Mantıksal Operatörler ve Sembolleri	26
Tablo 3 : Karşılaştırma Operatörleri ve Sembolleri	27
Tablo 4 : Kullanıcı Arayüz Dizayn Aşamaları ve Tanımlamaları	84
Tablo 5 : Kalite Ölçüm Kontrol Listesi.....	97
Tablo 6 : GUI Hataları Ölçüm Kontrol Listesi.....	98
Tablo 7 : MATLAB Kalite Ölçüm Kontrol Listesi.....	100
Tablo 8 : MathCAD Kalite Ölçüm Kontrol Listesi.....	101
Tablo 9 : MATLAB & MathCAD GUI Hataları Ölçüm Kontrol Listesi.....	103
Tablo 10 : MATLAB & MathCAD GUI Hata Türleri Sayısını Gösteren Liste.	200

Enstitüsü : Fen Bilimleri
Anabilim Dalı : Bilgisayar Mühendisliği
Programı : Bilgisayar Mühendisliği
Tez Danışmanı : Doç. Dr. Servet Bayram
Tez Türü ve Tarihi : Yüksek Lisans – Haziran 2005

KISA ÖZET

HESAPLAMA PROGRAMLARININ GRAFİKSEL KULLANICI ARAYÜZÜ (GUI) HATALARI PERSPEKTİFİNDE İNCELENMESİ

ALPAY ENGİN

Bu tez çalışmasının konusu; hesaplama programlarındaki Grafiksel Kullanıcı Arayüzü (GUI) hatalarının incelenmesidir. Tezde, hesaplama programları olarak MATLAB 6.1 ve MathCAD 12 yazılımları ele alınmıştır. Hedef, arayüz tasarımında büyük çoğunlukla ne gibi hatalar yapıldığını incelemek ve bu hataların doğrularının nasıl olması gerektiğini göstermektir.

Tez çalışmasında, hatalarından arındırılmış iyi bir grafiksel kullanıcı arayüzü nasıl olmalı konusunda bilgiler vermek amaçlanmaktadır. Tez, grafiksel kullanıcı arayüzü oluşturulması konusunda arayüz tasarımcılarına yol göstermesi ve yardım etmesi açısından da önemlidir. Bu tez çalışmasında, grafiksel kullanıcı arayüzü tasarımında dikkat edilmesi gereken arayüz dizayn prensiplerinin ve yöntemlerinin neler olduğu ve bunların önemi vurgulanmaktadır.

Bu çalışma tezinin ana konularından biri olan Grafiksel Kullanıcı Arayüzü (GUI), insan-bilgisayar iletişimini ve bilgisayar ile kullanıcı arasındaki etkileşimi sağlayan bir araçtır. GUI, kullanıcıların çeşitli komutları grafiksel gösterimler (menüler, düğmeler, grafikler v.b) aracılığıyla gerçekleştirmelerini sağlayan bir sistem tasarımıdır. Tezde ele alınan GUI hata türleri dört tanedir : 1. GUI Elemanlarında Yapılan Hatalar, 2. Düzen ve Görünüm Hataları, 3. Metinsel Hatalar ve 4. Etkileşim Hataları.

Tezde ele alınan hesaplama programları; matematiği temel alan, bilgisayar ortamında cebirsel her türlü hesaplama için olanak tanıyan ve mühendislik alanında kullanılan yazılımlardır. Tezdeki hesaplama programlarından biri olan MATLAB, temel olarak nümerik hesaplama, grafiksel veri gösterimi, GUI tasarımı ve programlamayı içeren matris (matrix) tabanlı bir yazılımdır. Tezde seçilen uygulama programlarından bir diğeri olan MathCAD, matematik temelli işlemlerde ve sembolik hesaplamalarda kullanılan bir yazılımdır. MathCAD, bir programlama dilinden çok, kelime işlemci mantığıyla, matematiksel işaretlerin kullanımına olanak tanımaktadır. MATLAB ve MathCAD programlarının her ikisinin de ortak özelliği; mühendislik programları olmalarıdır.

Bu tez çalışması, dünyada büyük bir popülaritesi olan ve inanılmaz boyutlarda kullanıcı kitlesine sahip MATLAB ve MathCAD programlarında pek çok grafiksel kullanıcı arayüzü (GUI) hatasının olduğunu göstermiştir. Tezde, bu hatalar tek tek incelenmiştir.

Anahtar Sözcükler : MATLAB Programı, MathCAD Programı, Hesaplama Programları, Grafiksel Kullanıcı Arayüzü (GUI), GUI Hataları.

University : Istanbul Kültür University
Institute : Institute of Pure and Applied Sciences
Department : Computer Engineering
Programme : Computer Engineering
Supervisor : Assoc. Prof. Dr. Servet Bayram
Degree Awarded and Date : MS – June 2005

ABSTRACT

INVESTIGATION OF CALCULATION PROGRAMS WITHIN GUI (GRAPHICAL USER INTERFACE) BLOOPERS' PERSPECTIVE

ALPAY ENGİN

The subject of this thesis is to investigate the Graphical User Interface (GUI) bloopers of calculation programs. This study deals with MATLAB 6.1 and MathCAD 12 programs as calculation programs. Aim of the thesis is to investigate what GUI bloopers are made in frequently in the interface design and show how these corrects of GUI bloopers should be in MATLAB and MathCAD programs.

This thesis adopts to give an information how a good Graphical User Interface (GUI) design with no bloopers should be. The research is important to help and to guide to the interfece designers for the Graphical User Interface (GUI) design. Also, GUI principles and methods in the GUI design are set out in the thesis.

One of the main topics of the thesis is Graphical User Interface (GUI). GUI is a graphical method of controlling how a user interacts with a computer to perform various tasks. GUI is a method of interacting with a computer through a metaphor of direct manipulation of graphical images and widgets in addition to text. A user interface based on graphics (icons, pictures, menus etc.) instead of text; uses a mouse as well as a keyboard as an input device. There are four GUI bloopers investigation in the thesis : 1. GUI Component Bloopers, 2. Layout and Appearance Bloopers, 3. Textual Bloopers and 4. Interaction Bloopers.

Calculation programs are used for the algebraic calculation in mathematics and in computer engineering. One of the calculation programs is MATLAB in thesis. MATLAB is a tool for doing numerical computations with matrices and vectors. MATLAB (MATrix LABORatory) is an interactive software system for graphics GUI design and programming. The other calculation program is MathCAD in thesis. MathCAD is used for the mathematical operations and symbolic calculations. More than just a programming language, MathCAD works as a word-processor.

The study has revealed that, although MATLAB and MathCAD programs are popular in the world, there are a lot of GUI bloopers in these programs. These GUI bloopers investigated one by one in the thesis.

Key Words : MATLAB Program, MathCAD Program, Calculation Programs, Graphical User Interface (GUI), GUI Bloopers.

1. GİRİŞ

“Hesaplama programlarından *MATLAB 6.1* ve *MathCAD 12* Programlarının Grafiksel Kullanıcı Arayüzü (*Graphical User Interface - GUI*) Hataları Perspektifinde İncelenmesi” konulu bu tez çalışmasında arayüz tasarımında genel olarak ve sıkça yapılan hataların bulunması incelenmiştir.

MATLAB 6.1 ve *MathCAD 12* programlarında bulunan *Grafiksel Kullanıcı Arayüzü (GUI)* hatalarını belirlemek ve bu hataların doğrularının neler olması gerektiğinin gösterilmesi ile ilgili çalışmanın amaçları genel olarak şu şekilde sıralanabilir :

1. Arayüz tasarımında genellikle yapılan hataları bulmak ve incelemek.
2. Arayüz tasarımında yapılan hataların doğrularının neler olması gerektiğini belirtmek.
3. Doğru ve uygun bir arayüz tasarımı konusunda kullanıcı arayüzünü oluşturmak isteyen kişi ya da kurumlara rehber olmak ve arayüz tasarımının nasıl olması gerektiğini söylemek.

Bu tez çalışmasında uygulama programı olarak seçilen *MATLAB*, son yıllarda popülaritesi yüksek olan ve bütün dünya da matematik, fen ve bilişim sektörü alanlarında kullanılmaya başlayan bir programdır. Ayrıca bu program, arayüz tasarımını incelemek ve arayüz tasarımında yapılan genel hataları analiz etmek için çok uygun bir yazılımdır. Tezde, *MATLAB*'in tercih edilmesinin en önemli nedeni; *MATLAB* programının sayısal hesaplamalar, grafik çizimleri, programlama ve simulasyon uygulamaları için geliştirilmiş olan bilgisayar programlarına göre daha özel ve daha üstün bir dil olmasıdır. Yüksek performanslı bir teknik programlama dili olan *MATLAB*, bir çok sahada kullanılan ve kullanıcılarına denklem yazma kolaylığında program yazma imkanı sağlayan sınıfının en gelişmiş programıdır. *MATLAB*, tıbbi araştırmalardan uzay araştırmalarına kadar çok geniş yelpazede kullanılmakta ve kullanıcıya, güvenebileceği hızlı bir analiz ve tasarım ortamı sağlamaktadır.

MATLAB, adını “*Matrix*” kelimesinin **MAT** eki ile “*Laboratory*” kelimesinin **LAB** ekinden almış olan ve *Mathworks* firması tarafından sayısal hesaplamalar, matematiksel problem çözümleri, grafik tasarımı ve programlama geliştirmek için

oluşturulmuş bilimsel bir yazılım programıdır. MATLAB, birçok sayısal problemi matris ve vektör formülleri yardımı ile çözer. MATLAB'in çok güçlü ve çok kapsamlı bir fonksiyon yapısı vardır. MATLAB'te bazı fonksiyonlar daha önceden yerleştirilmiş, bazıları ise sonradan dışarıdan yerleştirilmiş M-dosyalarından oluşan MATLAB kütüphanelerinde tanımlanmışlardır. MATLAB'te yüzlerce hazır fonksiyonla veya programcı tarafından yazılacak fonksiyonlarla işlemler son derece kolaylaştırılabilir. Temelini matris teorisi, lineer cebir ve sayısal işlemlerin oluşturduğu MATLAB'in çok geniş bir kullanıcı kitlesi vardır.

MATLAB, teknik ve bilimsel hesaplamalar için yazılmış yüksek performansa sahip bir yazılımdır. MATLAB'i diğer programlama dillerinden ayıran ve kullanıcının gözünde vazgeçilmez yapan iki önemli özelliği vardır : Birincisi, çeşitli uygulamalarda kullanıcının işlemleri daha kolay yapmasını sağlayan *Araç Kutuları (Toolbox)*'tir. İkincisi ise, her türlü sistemin modellenmesi ve simülasyon işlemleri için kullanılan *Simulink* programıdır. MATLAB'te, sinyal ve görüntü işleme, haberleşme/iletişim, kontrol dizayn, test ve ölçüm, finansal modelleme ve analiz ile biyoloji konularında da uygulamalar geliştirilebilir. MATLAB, sayısal ve simgesel hesaplama, gerçek ortamların simülasyonu ve analizi için Simulink; uygulamaya özel istatistik, optimizasyon, finans, yapay sinir ağları, bulanık (fuzzy) mantık, kontrol, görüntü ve işaret işleme "Toolbox"ları ve "Blockset"leriyle eğitim ve araştırmaya yönelik gereksinimleri karşılayacak olan teknik programlama ortamı sayesinde birçok alanda kullanıcının isteklerine özgü uygulamalar geliştirmesini sağlar.

FORTRAN ve C dili gibi yüksek seviyeden programlama dilleri ile yapılabilen hesaplamaların pekçoğunu MATLAB ile yapmak mümkündür. Buna karşılık MATLAB'in fonksiyon kütüphanesi sayesinde, FORTRAN ve diğer programlama dillerine göre MATLAB'te daha az sayıda komutla çözümler üretmek mümkündür. Gerçekte MATLAB, M-dosyaları (M-files) olarak bilinen pek çok sayıda fonksiyon dosyalarından (altprogramlardan) ibarettir. M-dosyaları, ASCII formatında olup, okunabilirliği olan MATLAB programlama dili kodlarından oluşmuştur. Bu yüzden M-dosyaları kütüphanesine, kullanıcı tarafından müdahale edilebilir. Ancak bu kodlara zorunlu olunmadıkça müdahale edilmemesi, değiştirilmemesi gerekir.

Bu tez çalışmasında diğ er bir uygulama programı olarak seçilen MathCAD, grafik, formül ve denklem editörü olan, özellikle nümerik metodlar için kullanılan ve Mathsoft Inc. firması tarafından dizayn edilmiş bir yazılım programıdır. Tezde, MathCAD programının tercih edilme sebeplerinin başında; hesaplama programları arasında en yaygın kullanıma sahip olma ve en çok sevilen hesaplama programlarından biri olarak görülmesi özelliklerinin bulunması gelmektedir. Ayrıca MathCAD programı, grafiksel kullanıcı arayüz (GUI) hatalarını bulma konusunda elverişli bir yazılımdır. MathCAD; eşitsizliklerin çözülmesini, grafik fonksiyonlarının oluşturulması ile grafiklerin çizilmesini ve sembolik hesaplamaların yapılmasını sağlayan bir programdır. MathCAD'in en büyük avantajı : Öğrenilmesi ve kullanılması kolay bir program olmasıdır. MathCAD programının en önemli özelliklerinden biri : Grafik çiziminde; grafiği oluşturan değişkenlerin değerlerinin kolayca değiştirilmesi ve yeni değişkenler atanması sonucu, ortaya çıkan yeni grafiğin kolayca oluşturulması ve kullanıcıya sunulmasıdır.

MathCAD yazılımı, **MATH** ekini "*Matematik*" kelimesinden, **CAD (Computer-Aided Design)** ekini ise "*Bilgisayar yardımlı dizayn*" kelimelerinden almış olan, bir Windows programıdır. MathCAD bilgisayar programı, hem MATLAB, Mathematica ve Maple gibi matematik temelli işlemler için kullanılan paket programlar gibi; hem de C, Visual Basic ve FORTRAN programlama dilleri gibi işlev görür. Gerek sayısal ve sembolik işlemleri yapmak, gerekse birkaç değişkenli fonksiyonların çizimi gibi pek çok matematiksel işlemleri yapmak amacıyla geliştirilen, matematiksel yazılımların en önemlilerinden biri, MathCAD'tir. MathCAD yazılım programı, *CAS (Computer Algebra System)* ailesi içinde olup, ilköğretim düzeyinden lisansüstü düzeye kadar birçok alanda matematik öğretimi ve matematik araştırmaları gibi konularda kullanılmaktadır. MathCAD ile birlikte diğ er CAS yazılımlar içerisinde en çok bilinenleri ve kullanılanları : MATLAB, Mathematica, Maple, Derive, Converge ve Theorist'tir. Grafik ve CAS teknolojisi ile donatılmış olan MathCAD ile matematik öğretimi için zengin bir ortam yaratmak olasıdır.

MathCAD, mühendislik ile ilgili bilgilerin ve verinin kolayca paylaşılmasını ve yönetilmesini gerçekleştirmek için kullanılır. MathCAD ile hesaplama ve dizayn işlemleri geliştirilir, mühendislik tabanlı uygulamaların ve çeşitli işlerin kalitesi arttırılır, hata riski ve yanlışlıklar en aza indirilir. Ayrıca, harcanan enerji ve güç en

aza indirilerek kullanıcılar için analiz olanakları sağlanır. MathCAD yazılımı ile, C ya da C++ programlarında uygulamalar ve kullanıcı tabanlı fonksiyonlar yaratılır. MathCAD yazılım programı, her türlü işlem ve uygulamada mühendislik ve bilimsel notasyonları kullanır. MathCAD'in, endüstrideki işlemler için mühendislere yardım etmek amacı ile etraflı ve genel bir dizayn alanı vardır. MathCAD'te her türlü işlem için dizayn aşamaları oluşturulur. MathCAD, mühendislerin genel amaçlı kullandıkları, Mathsoft patentli, elektronik matematik teknolojisidir. Bütün mühendisler için, teknik problemlerin çözümü ve analizini sağlamak amacı ile, güçlü özelliklere sahip, ideal bir araçtır. MathCAD, mühendisler için tek bir alanda formülleri, metinleri ve grafikleri birleştirme özelliği sağlayan bir programdır.

MathCAD; matematik, grafik ve programlama gibi üç temel kavramın bütünleşmesi ve birleşmesinden oluşan bir bilgisayar yazılım programıdır. MathCAD arayüz platformu; matematiksel hesaplamaların tekrar tekrar yapılmasını sağlayan, standart matematiksel notasyonlardan ve kolay kullanımlı denklem fonksiyonlarından faydalanılmasına olanak sunan ve birimlerin birbirine dönüştürülmesi özelliğini içinde bulunduran bir alandır. MathCAD'in kullanıcı kitlesi; mühendislik, fen ve matematik alanındaki öğrenciler, mühendisler ve profesyonel kişilerdir.

MathCAD, ekonomi ve iş alanında faydalı olan bir programdır. Ekonomi ve iş alanındaki son eğilimleri göstermek ve bu sektördeki önemli birçok modeli analiz etmek için kullanılan MathCAD, bilimsel bir programdır. MathCAD, ekonomi ve iş dünyasındaki matematiksel engelleri aşmak ve bu alandaki insanlara her türlü yardımı sağlamak için son derece yararlı bir programdır. MathCAD yazılım programı, ekonomi sektöründe ve iş alanında matematikten sıkılan ve yorulan, sistem modelleri içerisinde boğulan insanların sağ kolu ve en önemli destekçisidir. Finans konusunda MathCAD, kullanıcılarına çeşitli imkanlar sağlar. MathCAD, faizlerin bugünkü değerlerini ve gelecekteki değerlerini içeren çeşitli fonksiyonlara sahiptir. Ayrıca, MathCAD'in finans ile ilgili modellerin oluşturulması ve bu modellerin simülasyonu için çeşitli özellikleri vardır.

MathCAD, nümerik işlemlerin gerçekleştirilmesi ve sayısal integral hesaplamaları için birbirinden farklı birçok algoritma kullanır. MathCAD'te sembolik hesaplamaların gerçekleştirilmesi de oldukça kolaydır. MathCAD, boolean işlemlerini

gerçekleştiren sembolleri içinde bulunduran bir programdır. Boolean mantıksal operatörleri, boolean işlemleri için MathCAD kullanıcılarına çeşitli kolaylıklar sağlar.

Grafiksel kullanıcı arayüzü (GUI), bilgisayar kullanıcısı ile bilgisayardaki işlemler ve uygulamalar arasında kolay ve etkili bir etkileşim sağlayan grafiksel tabanlı bir arabirimdir. Grafiksel kullanıcı arayüzü (GUI) dizaynının amacı; bilgisayar arayüzünü kullanan kişilere arayüzü, kolay, anlaşılır, onları hiç sıkmadan ve hoşlarına gidecek bir biçimde sunmaktır. Grafiksel kullanıcı arayüzü (GUI), iki önemli noktanın üzerinde büyük bir titizlikle durur :

1. Kullanıcının bilgisayardan bilgi almasına yardım etmek için, bilgisayar ile kullanıcı arasında iletişim sağlar.
2. Kullanıcı ile bilgisayar arasında bir bağ kurmak için kullanıcı ile bilgisayar arasında etkileşim sağlar.

Arayüz dizayn işlemine başlamadan önce zaman kaybı, dizayn için gerekli olan para zorunluluğu ve gelişim problemleri göz önünde bulundurulmalıdır. Bununla birlikte, arayüzün geniş bir kullanıcı topluluğuna hitap edebileceğide unutulmamalıdır. Bu da arayüzü geliştiren kişi ya da kişilerin sorumluluklarının iki kat daha fazla artmasına neden olacaktır. Bu yüzden arayüz dizaynını belirleyen faktörler üzerinde titizlikle durulması gerekir. Grafiksel kullanıcı arayüz dizaynının oluşmasında, mutlaka göz önünde bulundurulması gereken bazı noktalar vardır. Bu noktalar aynı zamanda arayüz dizaynının ana öğeleridirler. Kullanıcı arayüz dizaynının ana öğeleri, arayüz dizaynının mantığının iyi kurulması ve dizayn için iyi bir model hazırlanması için gereklidirler. Grafiksel kullanıcı arayüz dizaynını belirleyen faktörler ve bu faktörlerin özellikleri şunlardır :

1. Renk :

Renk, arayüzün görselliğinin ön planda tutulmasını ve arayüzün etkileyici bir görünümünün olmasını sağlayan bir kullanıcı arayüz dizayn öğesidir. Renk, arayüz ekranının okunabilir olmasını etkileme gibi bir özelliği ile arayüz dizaynında son derece önemli bir yere sahiptir. Örneğin; arayüz ekranında siyah rengin kullanılması, kullanıcının arayüz ekranındaki bilgileri okumasını engeller. Kullanıcının odaklanması istenilen yerlerde, arayüz dizaynının ilk ana öğesi olan renklerden yardım alınır. Rengin arayüz üzerinde etki ettiği elemanlar şunlardır : Metin rengi, grafik ve şekil rengi, arayüz elemanlarının rengi ve arayüz arka plan rengi.

2. Ekran Görünümü :

Arayüz dizayn edilirken arayüzün görselliğini direkt olarak etkileyen en önemli arayüz dizayn öğesidir. Ekran görünümü, arayüz ile kullanıcı arasındaki iletişimi ve etkileşimi sağlamanın en etkili yoludur. Arayüz elemanlarının ekran üzerindeki düzen ve hizaları, ekran görünümünün içinde yer alır. Ayrıca ekran görünümü içerisinde metin, şekil, resim, grafik, ikon ve görsel animasyonlar da bulunur. Ekran görünümü, estetik bir bütünlüğü sağlamak için yararlıdır.

3. Metin :

Arayüzün yapısını ve içeriğini anlatmak için kullanılan bir arayüz dizayn öğesidir. Kullanıcıya bilgiyi iletmenin en iyi ve en kolay yoludur. Metin ile ilgili üç önemli unsur şunlardır : Metin tipi, metin genişliği ve metin sitali. Metin, arayüz dizayn standartları oluşturulurken ilk akla gelen konudur.

Kaliteli ve nitelikli bir arayüz dizayn etmek ve geliştirmek sanıldığı kadar kolay değildir. Biri diğerinden daha iyi olan arayüzler olabilir ama bu asla o arayüzün harika olduğu anlamına gelmez. Kullanıcı arayüzünün kalitesini gösteren etkenler arasında 1) Kullanım kolaylığı 2) Kullanıcının verimliliğini arttırmak sayılabilir. Kullanıcı arayüzünde kaliteyi arttırmanın en iyi yolu, arayüzün belirli bir sıra halinde iyi tanımlanmış ve tasarlanmış dizayn işlemlerinden oluşmasıdır.

İyi bir grafiksel kullanıcı arayüzünün en önemli karakteristik özelliği, kullanıcının yapmak istediği işlemlere hızlı bir şekilde yanıt verebilmesidir. Bu durum, arayüzün faydalı ve başarılı olduğunu gösterir. Faydalı grafiksel kullanıcı arayüzleri, kullanıcının bilgisine önem verir ve kullanıcının her türlü bilgiyi alması için büyük bir çaba harcar. Bir arayüz uygulamasının kabulü, kullanıcı memnuniyetine bağlıdır. Bunu sağlayan arayüz başarılıdır. Yüksek performans gösteren, kullanıcıyı bekletmeyen arayüz, kullanıcı için faydalıdır.

Arayüz dizaynının her safhasında kullanıcı kitlesi göz önünde bulundurulmalıdır. Örneğin; gözleri görmeyen kimseler ile gözleri görenler arasında fark vardır. Bu nedenle arayüz dizaynı yapılırken, arayüz kullanıcılarının birbirine karıştırılmaması ve birtakım ince noktalara dikkat edilmesi gerekir. Dizayn edilen arayüz kullanıma sunulduğunda, arayüzü geliştiren kişi ya da kişiler tarafından farkına varılmamış olan

bir takım hatalar olabilir. Bu nedenle arayüz, kullanıcının istek ve beklentilerini karşılamanın yanında, kullanıcının zamanını da boşa harcatmamalıdır. Öğrenilmesi, kullanılması kolay, hatalardan arınmış ve basit dizayn edilmiş bir arayüzü geliştirmek temel prensip olmalıdır.

Kullanıcı arayüz dizaynındaki en büyük sorun, dizayn sırasında düşünülen fikirlerin gerçekleştirilmesi için gerekli olan eğitimin, arayüzü dizayn eden kişi ya da kişilerde olmamasıdır. Grafikselle kullanıcı arayüzünün dizayn işlemleri, arayüzün dizayn edilme amacına göre yapılmalıdır. Kullanıcı arayüz dizaynından önce standart bir dokümantasyon oluşturulmalı, dizayn bu dokümantasyondaki işlemlere sadık kalınarak yapılmalıdır. Arayüz dizaynının başarılı bir görsel yapısı olması için bir deneme modeli oluşturulmalıdır. Kullanıcı arayüzü belli standartlara ve prensiplere bağlı kalınarak oluşturulmalıdır. Kullanıcı arayüzünün dizayn edilmesi sırasında bu sayılan şartları sağlamayan arayüzler için zaman ve para boşa harcanmış olur. Sadece ticari amaçlı oluşturulmuş, kullanıcı memnuniyetini sağlamayan ve belirli kuralları içermeyen arayüzlerin, iyi dizayn edilmiş bir arayüz olma olasılığı yok denecek kadar azdır. Bu bakımdan bu tez çalışması, iyi ve doğru bir arayüz oluşturmak isteyen kişi ya da kurumlara kılavuzluk etmesi ve örnek olması açısından önemlidir. Bu tez çalışması ile arayüzü oluşturan kişi ya da kurumlar, arayüz oluşturulurken yapılan genel hataları görmüş olacak ve bu hataların doğrularının nasıl olması gerektiği konusunda fikir sahibi olmuş olacaklardır.

Bilgisayarı ilk defa kullanan, acemi kullanıcılar; arayüzün iyi dizayn edilip edilmediği hakkında ilk anda fikir yürütemeyebilirler. Ancak, daha sonra arayüzde çıkan sorunlar ve problemler yüzünden kullanıcıların arayüze karşı olan tutumlarında negatif yönde azalma olacaktır. Arayüz dizayn edilirken, arayüzü dizayn eden kişi eden kişilerin düşünmesi gereken bazı sorular olmalıdır. Örneğin; *arayüz kimin için dizayn ediliyor?* ya da *arayüz dizayn edilirken hangi prensipler göz önünde bulundurulmalı?* gibi. Hem arayüzü dizayn eden kişi ya da kurumların hem de arayüzü satın alacak olan kişilerin dikkat etmeleri gereken ilk ve en önemli nokta : Arayüzün sadece görselliğine önem verilmemesi gerektiğidir.

Arayüz hem görselliği, hem kullanım kolaylığı hem de verimliliği açısından değerlendirilmelidir. Ayrıca, arayüz karmaşık bir yapıda olmamalıdır. Arayüz dizayn

edilirken gereken hassasiyet gösterilmezse, onca emek ve çalışılan işlerin hepsi, boşa gitmiş olacaktır. Arayüzü dizayn eden kişilerin boşa zaman harcamaları için arayüz dizayn tekniklerine göre arayüzü dizayn etmeleri, bununla birlikte; arayüzü satın alan kişilerinde paralarının boşa gitmemesi için, arayüz hakkında ve arayüzü dizayn eden kişi ya da kurumların eğitim ve tecrübeleri hakkında iyi bir araştırma yapmaları ve daha sonra satın almaları tavsiye edilmektedir. Ayrıca, kullanıcılar arayüzü satın almadan önce arayüzü test ederek, arayüzün kalitesi hakkında bilgi sahibi olabilirler. Satın alınmak istenen bu arayüzü, daha önce kullanan kişilerin fikirleri alınıp, bu görüşler doğrultusunda arayüz hakkında fikir elde edilmelidir. Kullanıcıların arayüz seçiminde bulunurken, kendi istek ve beklentilerini arayüzün tam olarak karşılıyor olmasına dikkat etmeleri gerekir. Arayüz dizayn etmek ve oluşturmak büyük bir disiplin, titizlik ve sabır gerektiren ciddi bir iştir.

Tüm bu bilgiler ekseninde genel olarak MATLAB, matematiksel hesaplama, çözümlenme, görsellik ve algoritma geliştirme araçlarını içeren mühendislik ve bilimsel uygulamalara yönelik güçlü bir yazılım sistemidir.

MATLAB'i genel hatları ve bilinen özellikleri ile kısaca şöyle tanımlayabiliriz :

- ❑ MATLAB adı **MAT**rix **LAB**oratory'den gelir.
- ❑ The Mathworks, Inc. firması tarafından geliştirilmiştir.
- ❑ Matris tabanlı bir programdır.
- ❑ Mühendislikte problem çözümü konusunda çok etkilidir.
- ❑ Etkili ve faydalı bir çalışma platformu vardır :
 - sayısal hesaplamalar için
 - sembolik hesaplamalar için
 - bilimsel çalışmalar için
- ❑ *Simulink* programı ile sistem modelleme ve simülasyon oluşturma özelliği vardır.
- ❑ *Araç Kutuları (Toolbox)* ile uygulamalar ve işlemler geliştirme özelliği vardır.
- ❑ Veri analizi konusunda son derece başarılıdır.
- ❑ Bir programlama dilidir.
- ❑ Grafik çizimleri sağlar.
- ❑ *GUIDE* ile *Grafiksel kullanıcı arayüzü (GUI)* dizayn etme özelliği vardır.

Genel olarak MathCAD,

- Matematiksel notasyon merkezli, güçlü bir teknik hesaplama alanı
- Esnek ve geniş özellikli bir kelime işlemci
- Matematiksel çözümler üreten grafik kapasiteli bir yazılım
- Nümerik hesaplama ile ileri grafik ve görselliği birleştirip entegre bir teknik hesaplama ortamı sağlayan yüksek programlama dili
- Matematiksel hesaplama ve programlama yazılımı

olma özelliklerinin bir kombinasyonudur.

MathCAD matematiksel hesaplama aracı, aynı MATLAB programı gibidir. MathCAD'in, MATLAB programı ile benzer yönleri çoktur. MathCAD'in geniş özellikli, güçlü bir yazılım olması ve büyük çaplı ya da herhangi bir matematiksel işlemin kontrolü, veri analizi, grafik çizimi, dizayn ve modelleme işleri için kullanıcılar tarafından kullanılması gibi yönleri, MATLAB ile olan benzerliğinin en büyük kanıtıdır.

Matematiksel işlemleri ve hesaplamaları gerçekleştirmek amacı ile kullanılan hesap makinesi ile, çeşitli yazıları ve raporları oluşturmak için kullanılan kelime işlemci programlarının yerini alan MathCAD, yazılım sektörünün son yıllarda geliştirdiği en büyük matematiksel bilgisayar programıdır. Matematik uygulama ve işlemlerini daha kolay bir hale getiren MathCAD; kullanırken eğiten, eğitirken de sevdiiren özelliği ile kullanıcı dostu bir programdır. Linear algebra (lineer cebir) işlemlerini kolaylaştırıp, kullanıcı için kolay anlaşılır bir hale getiren MathCAD, bu alandaki en önemli kaynaklardan biridir. MathCAD, mühendislik ve matematik alanındaki çalışmalarda kalem ve kağıdı ortadan kaldırıp, bu alandaki işleri daha kısa bir zamanda ve daha etkili bir biçimde yapmak için dizayn edilmiştir. MathCAD'te işlemlerin ve uygulamaların çoğu basit ve kullanıcıların anlayabileceği bir şekilde yerine getirilir. MathCAD; aritmetik işlemlerden, programlamaya kadar birçok konudaki herhangi bir problemi, farklı komut satırlarıyla yerine getirme ve çözüme gücüne sahip bir yazılım programıdır.

Sonuç olarak; bu tez çalışması, herhangi bir programın grafiksel kullanıcı arayüzünü (GUI) oluşturmak isteyen kişi ya da kurumlara bu konuda kılavuzluk etmek, yol göstermek ve bu kişi ya da kurumların iyi bir arayüz nasıl olmalı konusunda fikir

sahibi olmalarını saęlaması aısından nemlidir. Tezde, arayüz tasarımında genellikle yapılan hataların neler olduęunun gösterilmesi ve bu hataların doęrularının nasıl olması gerektięinin anlatılması amaçlanmaktadır. Ayrıca bu tez, kullanıcıların arayüzü kullanırken zorlanmayacakları ve kullanıcıları memnun edecek doęru bir grafiksel kullanıcı arayüzü (GUI) nasıl oluşturulabilir onu göstermesi aısından önemlidir. Bu alıřma tezinin amacı; hangi řartlarda ve hangi durumlarda kullanıcı arayüz dizayn prensiplerine, arayüz dizayn yöntemlerine ve arayüz dizayn standartlarına daha uygun bir arayüz oluşturulabileceęini aıklamak ve MATLAB ile MathCAD programları rnek alınarak, bu programlardaki grafiksel kullanıcı arayüz hatalarını göstermek kořulu ile bu hataların doęrularının nasıl olması gerektięini inceleyerek, arayüz tasarımcılarına bu bilgiler ışığında tavsiyelerde bulunmaktır. Tez, iyi bir grafiksel kullanıcı arayüzü oluşturmak için ne gibi kriterler gerekli olduęu konusunda aıklayıcı bilgiler vermeyi saęlamaktadır.

2. LİTERATÜR

Tezin bu bölümünde, MATLAB ve MathCAD programlarının tanıtımı yapılmış, grafiksel kullanıcı arayüzü (GUI) hakkında genel bilgiler verilmiştir. MATLAB ile ilgili olarak, MATLAB'in içeriği, genel yapısı, özellikleri, elemanları, ve kullanım alanları gibi konular ele alınmıştır. MathCAD ile ilgili olarak, MathCAD'in nasıl bir program olduğu, genel özellikleri, yapısı ve kullanıcıya sağladığı faydalar gibi konular ele alınmıştır. Grafiksel kullanıcı arayüzü (GUI) ile ilgili olarak, GUI'nin ne olduğu, önemi, kullanıcı arayüz dizaynının amacı, yapısı, elemanları, arayüz dizayn teknikleri ve yöntemleri gibi konular ele alınmıştır.

2.1. Hesaplama Programları

Hesaplama programları, temelini matematiğin oluşturduğu, elektronik ortamda kolay ve az bir zamanda hesaplama imkanı sağlayan programlardır. Bu yazılımlar, matematiksel problemlere çözüm getirmek, matematik bilgilerini paylaşmak ve matematik dışında mühendislikle ilgili her türlü hesaplama ve işlemin yapılmasına olanak sağlayan programlardır. Bu programlar, bilgisayar destekli cebir sistemlerdir. Hesaplama programları, matematiği sadece teori olmaktan kurtaran, matematiğin görselliğini ön plana çıkaran programlardır. Bu yazılımlar, kullanıcıyı sorgulamaya, araştırmaya ve keşfetmeye yönlendirirler. Matematiğin temel yapısını ve içeriğini kullanıcıya sevdirek aktaran programlardır. Her türlü hesaplama işlemleri için kullanılan, matematik yazılımlarının en popüler ve en çok kullanılanları arasında; MATLAB ve MathCAD vardır. Bu tez çalışmasında hesaplama programları olarak MATLAB 6.1 ve MathCAD 12 yazılımları ele alınıp, incelenmiştir.

2.2. MATLAB Programı

MATLAB (MATrix LABoratory); ilk defa 1985 yılında, C.B Moler tarafından matematik ve özellikle de matris esaslı matematik ortamında kullanılmak üzere geliştirilmiş etkileşimli bir paket programlama dilidir. İlk sürümleri FORTRAN diliyle hazırlanmış olan MATLAB, daha sonraları C programlama diliyle geliştirilmiştir. MATLAB mühendislik alanında (hesaplamalarında); sayısal hesaplama, veri çözümleri ve grafik işlemlerinde kullanılabilecek genel amaçlı bir program olmakla beraber, özel amaçlı modüler paketlere de sahiptir. CONTROL TOOLBOX, SIGNAL TOOLBOX gibi paket programlar, CACSD (Bilgisayar

Destekli Denetim Sistemi Tasarımı) paketler olup bunlar denetim sistemlerinin tasarımında çok etkili araçlardır. Ayrıca WINDOWS ortamında çalışan SIMULINK, etkileşimli benzetim programlarının hazırlanması ve çalıştırılmasında büyük kolaylıklar sağlamaktadır [1].

MATLAB, birçok alandaki çeşitli işlem ve uygulamaları kolayca dönüştürmek ve zamandan tasarruf sağlamak için kullanılan temel araçlardan biridir. “MATrix LABoratory” kelimesinin kısaltması olan MATLAB, adında da anlaşılacağı üzere matrisler (matrix) yani diğer bir deyişle diziler (array) ile çalışır. MATLAB’in matrisler ile çalışması demek; verileri matrisler biçiminden değerlendirmesi ve çözümleri matris esasına göre yapması ve sonuçları da matris biçiminde düzenlenmesi demektir. Özellikle mühendislik alanındaki sistemlerin analizinde kullanılan MATLAB, yapay sinir ağları (artificial neural networks), 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 kullanıcılar için mükemmel bir ortam sunar [2].

MATLAB, Mathworks firması tarafından yazılmış ticari bir yazılım paketidir. Özellikle vektör ve matris işlemlerinin yapılmasında kullanılan üst düzey bir programlama dilidir. MathCAD, mühendislik ve fen bilimleri alanlarındaki sayısal hesaplamaların yanında, gözde canlandırılarak ya da sezgisel olarak meydana getirilen grafiklerin oluşumunda ve tasarımında kullanılır. MATLAB’te işlemleri kolaylaştırmak için birçok toolbox vardır. MATLAB, sayısal ve sembolik hesaplamalar, veri çözümlemesi, gerçek ortamda test ve ölçüm yapabilme, çok gelişmiş çizim işlemleri, algoritma geliştirme, ileri seviye programlama, C/C++ ile tümleşik çalışabilmesi, mühendislik ve bilimsel uygulamaları ile tüm dünya da bir çok alanda yaygın olarak kullanılan bir yazılımdır [3].

MATLAB (*MATrix LABoratory*), yüksek performanslı bir hesaplama dilidir. Mühendisliğin olmazsa olmaz programlarından olan MATLAB, FORTRAN programlama diline alternatif olarak yazılmıştır. MATLAB programı, hesaplamayı, görselliği ve programlamayı kullanımı kolay bir ortamda birleştirmiştir. Problemler ve çözümleri, bilinen matematiksel gösterimler ile ifade edilmiştir. MATLAB,

algoritmaları ve uygulamaları geliştirmek ve analiz etmek için kullanılan üst düzey bir dil ve geliştirme aracıdır. MATLAB aynı zamanda C, C++ dillerinin sahip olduğu aritmetik operatörler, veri yapıları, veri tipleri, nesneye dayalı programlama (Object-Oriented Programming) gibi özelliklerin hepsini içerir . MATLAB özellikle matris işlemleri (lineer denklem sistemlerinin ve vektör işlemlerinin çözülmesi gibi) için oluşturulmuş ve dizayn edilmiştir. Lineer olmayan denklem sistemleri ve diferansiyel denklemlerin çözümleri için de kullanılır. Yazılan programlar, MATLAB'ın kendi programlama dili ile oluşturulur. MATLAB'te case-sensitive (büyük-küçük harf duyarlılığı) özelliği bulunur. Örneğin; `solveUT` ile `solveut` aynı değildir [4].

MATLAB, bilimsel ve mühendislik hesaplamaları için kullanılan matris tabanlı bir sistemdir. MATLAB, kompleks sayısal problemlerin program yazmadan çözülmesini sağlar. MATLAB'te problemler, komut satırında çalışan programlar yazarak çözülebileceği gibi; MATLAB GUI geliştirme aracını kullanarak, formlar ve butonlar gibi nesnelere oluşan görsel yazılımlar da oluşturulabilir.

MATLAB'ın ilk orijinal versiyonu The MathWorks, Inc. Firmasının Fen Bölümü Başkanı Dr. Cleve Moler tarafından 1970 yılında yazılmıştır. MATLAB'ın yazılışındaki esas amaç, LINPACK ve EISPACK projelerindeki matris yazılım gelişimine kolaylık sağlamaktır. MATLAB'ın temelini matris teorisi, lineer cebir ve sayısal hesaplamalar oluşturur. Bugün MATLAB, 100'ün üstünde birçok ülkede ve 400.000'in üzerinde birçok insan tarafından kullanılmaktadır [5].

MATLAB, mühendislik ve fen bilimleri alanındaki hesaplamalar ile matematik alanındaki uygulamalar için geliştirilmiş güçlü bir yazılım programıdır. MATLAB'ın en önemli yanı lineer cebir ile olan benzerliğidir. Lineer cebir, MATLAB'ın ruhu ve kalbidir. MATLAB, iki boyutlu ve üç boyutlu grafik oluşturulması konusunda da son derece faydalı bir programdır. Ayrıca yüz veya daha fazla satırdan oluşan kodların oluşturulması konusunda da son derece başarılıdır. Diğer sayısal tabanlı diller (C++ ve FORTRAN gibi) ile kıyaslandığında anlaşılması ve kullanılması daha kolay olmasının yanında, büyük bir kütüphaneye sahip olması en büyük avantajıdır. Buna rağmen, büyük çaplı hesaplamalarda yüksek performans göz önüne alındığında en iyi yazılım programı değildir [6].

2005 yılı itibariyle, 500.000 den fazla akademisyen, arařtırmacı, bilim adamı ve öđrenci tarafından kullanılan MATLAB, içinde gömülü pek çok ara yüzü ile bilgisayar dünyasının en gelişmiş teknik ve bilimsel problem çözme ve uygulama geliştirme araçlarından biridir. MATLAB, kullanımı kolay, görselliđi zengin bir programdır. MATLAB, konuşmuş gibi yazılan bir dildir. Yapılan işlemler ile oluşturulmak istenen anlatım ve ifadeler konuşma diliyle aynı olacak şekilde yazılan bir bilgisayar programıdır. MATLAB'ın nasıl bir yazılım olduğunu anlamak için onu çok gelişmiş özellikleri olan, programlanabilen bir bilimsel hesap makinesine benzetebiliriz. MATLAB'te yazılan programlar, MATLAB'in kendine özgü dili kullanılarak yazılır ve MATLAB içinden çalıştırılır. Ayrıca yazılan programlar DLL ve EXE olarak oluşturulabildiđi gibi C/C++ kodlarına da çevirilebilir [7].

MATLAB etkili bir arayüze sahip, güvenilir algoritmaların oluşturulduđu ve insanların sayısal hesaplamaları yapmak için kullandığı bir bilgisayar programıdır. MATLAB'in asıl amacı problemleri matris tabanlı teknikler kullanarak çözmektir. MATLAB, birtakım özel uygulamaları ve fonksiyonları yerine getirmek için biraraya gelmiş özel fonksiyonlar topluluđu (toolbox)'lara sahiptir. MATLAB'in asıl amacı, problem çözümlerini C ve FORTRAN gibi programlama dillerinde olduđu gibi program yazma ile deđil matris tabanlı teknikler ile çözmektir [7].

Sayısal problemleri hızlı, zorlanmadan ve eğlenceli bir şekilde çözenin en önemli yollarından biri MATLAB'i kullanmaktır. MATLAB'te bilimsel ve matematiksel hesaplamalar, standart programlama dilleri'nden (FORTRAN, C, JAVA vb.) daha çabuk çözülür. MATLAB'in bu özelliđi insan gücünden tasarruf sağladığı gibi kullanıcının zamanını da en aza indirir. MATLAB, gerçek hayattaki problemlerin incelenmesinde inanılmaz kolaylıklar sağladığından kullanıcı dostu bir bilgisayar programıdır. Sayısal hesaplamalar dışında grafik çizimlerinde sağladığı kolaylıklar, verdiđi sonuçlar ve çizilen cisimler ile ilgili ilginç özellikler sunma başarısı ile faydalı bir programdır. Daha önce bilimsel kitaplarda çözülmüş olan matematik alanındaki soruları MATLAB'te tekrardan çözüp sonuçlarını karşılaştırabiliriz. Çıkan sonuçlardaki doğruluk bizim MATLAB'e olan inancımızı ve güvenimizi boşa çıkartmayacaktır. Büyük ölçekli projelerde MATLAB hiç zorlanmaz. Aksine kullanıcıya sürekli kolaylıklar sağlar. MATLAB, genel olarak temel aritmetik işlemleri çözme becerisi, matris (matrix) – vektör (vector) işlemlerinin çözümü,

grafiksel fonksiyonların çizimi ile programlama ve yazılım geliştirme özellikleri ile bilinir [8].

2.2.1 MATLAB'in Karakteristik Özellikleri

MATLAB'i yalnız programlama dili olarak ya da bilimsel, matematiksel ve bilgisayar alanındaki problemlerin çözümündeki faydaları ile bilmek yetmez. MATLAB'i bunların dışında çeşitli yönleri ile de bilmek gerekir. MATLAB'in kendine has, kendi iç yapısını ve dünyasını anlatan birtakım karakteristik özellikleri vardır. Bu özellikler şunlardır [9]:

- MATLAB, sayısal hesaplamalar için kullanılan gelişmiş bir hesap makinesidir.
- MATLAB'ten matematiksel problemlerin çözümleri, mühendislik alanındaki metodların uygulanması konularında faydalanılır.
- Öğrenciler, öğretmenler ve bilim adamları başta olmak üzere çok geniş bir kullanıcı kitlesi vardır.
- Bir veya iki komut ile harika uygulamalar yapma imkanı sunar.
- İlk versiyonu sayısal analizci Cleve Moler tarafından 1970 yılında yazılan MATLAB, sayısal analizler ve matris işlemleri yapmak amacı ile geliştirildi. Daha sonraları ticari amaçlı bir yazılım paketi haline geldi.
- Birçok uygulama ve fonksiyonu yerine getirmek için çok sayıda toolbox'a (çeşitli fonksiyonların ve metodların biraraya gelmesi ile oluşan yapılar) sahiptir.
- Matematik ile grafik fonksiyonlarının kombinasyonundan oluşan MATLAB'in temelini lineer cebir işlemleri ve çok etkili sonuçlar verdiği matematiksel işlemler oluşturur.
- Mathematica ve Maple ile birlikte matematik alanında kullanılan en yaygın üç matematiksel yazılım programından biridir.
- Maple bilgisayar programı gibi sembolik hesaplamalar için kullanılır.
- Grafiksel arayüzündeki kullanım kolaylığı ile matematik ve mühendislik alanındaki kişilere rahatlık sağlar.
- Verinin görsel olarak canlandırılması ve oluşturulması için 2-D ve 3-D fonksiyonlar içerir.
- MATLAB'i anlamının en iyi yolu matrisleri anlamaktır. Çünkü MATLAB'in temeli matrislerdir.
- MATLAB'in orijinal versiyonu FORTRAN programlama dili ile yazıldı. Fakat MATLAB, daha sonra C programlama dili ile tekrardan yazıldı.

- MATLAB, istatistik, optimizasyon ve finans problemlerinin çözümünde son derece faydalıdır.
- MATLAB, algoritmalar ve fonksiyonlar konusunda C, C++, FORTRAN, JAVA, COM ve Microsoft Excel gibi programlama dilleri ve uygulama programları ile etkileşim ve paylaşım içerisindedir.

2.2.2 MATLAB'in Kullanım Alanları

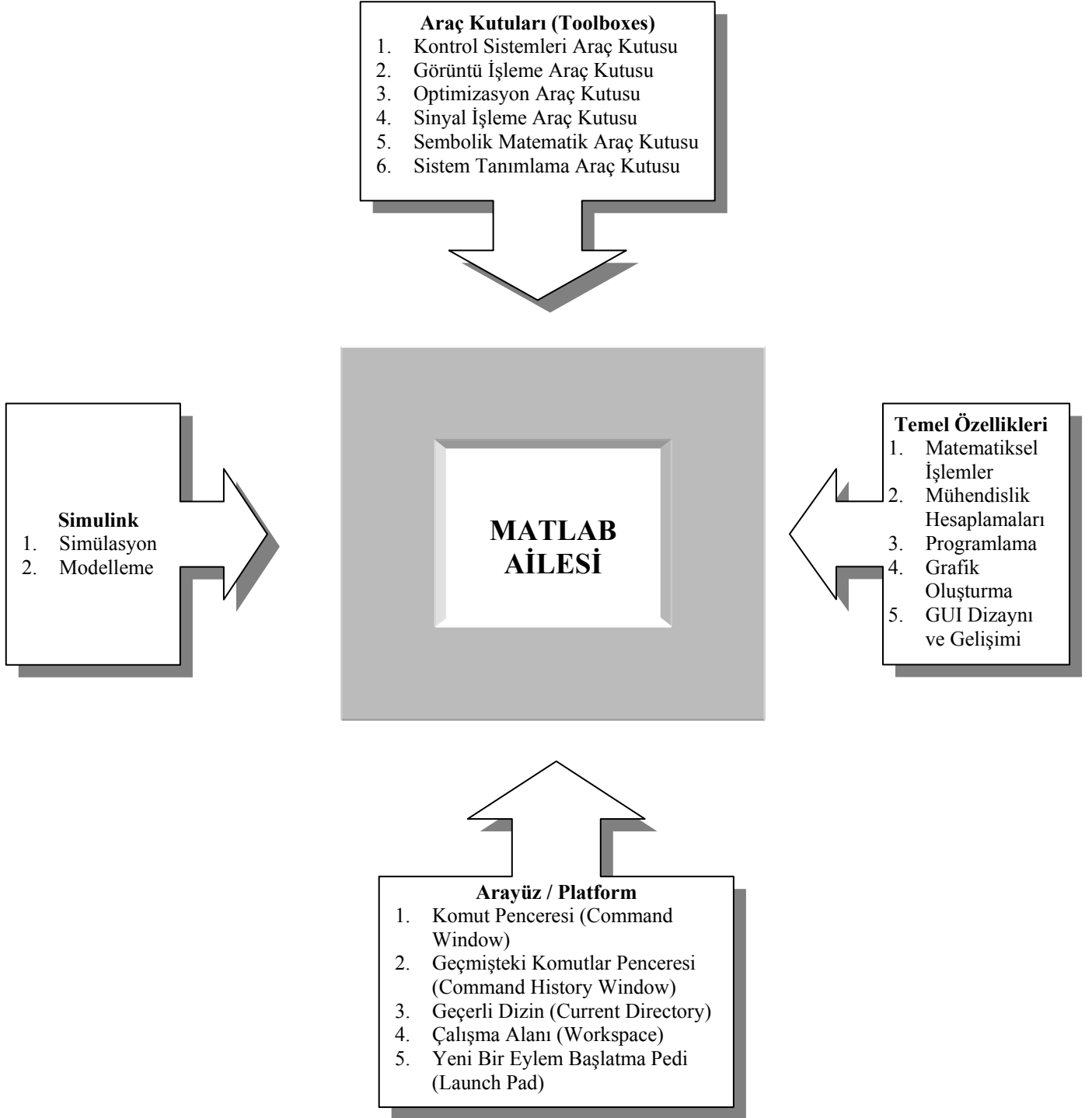
MATLAB, ilk başlarda bilim adamlarına problemlerin çözümüne matris temelli teknikleri kullanarak yardımcı olmaktaydı. Daha sonraları ise geliştirilen yerleşik kütüphanesi, uygulama ve programlama özellikleri ile gerek üniversite ortamlarında (başta matematik ve mühendislik olmak üzere tüm bilim dallarında) gerekse sanayi çevresinde yüksek verimli araştırma, geliştirme ve analiz aracı olarak yaygın bir kullanım alanı bulmuştur. Ayrıca işaret işleme, kontrol, fuzzy, sinir ağları gibi bir çok alanda ortaya koyduğu *Toolbox* adı verilen yardımcı alt programlarla da özelleştirilmiş ve kolaylaştırılmış imkanlar sağlamış ve sağlamaya da devam etmektedir. MATLAB, matematiksel hesaplamalar ve programlama başta olmak üzere çeşitli alanlarda kullanılır. MATLAB'in özünü ve içeriğini oluşturan bu alanlar aşağıda belirtilmiştir [10]:

- Matematik ve hesaplamalar (Maths and Computation)
- Algoritma geliştirme (Algorithm Development)
- Modelleme & Simülasyon (Modeling & Simulation)
- Veri analizi (Data Analysis)
- Grafikler (Graphics)
- Görüntü işleme (Image Processing)
- Sinyal işleme (Signal Processing)
- Grafikselle kullanıcı arayüzü (GUI) geliştirme
- Mühendislik ve fen alanında uygulamalar geliştirme
- Filtreleme

2.2.3 MATLAB'in Yapısı

MATLAB'i matematiksel işlemler ve mühendislik hesaplamaları yapmak, programlama geliştirmek, grafik oluşturmak ve grafikselle kullanıcı arayüz dizaynını geliştirmek gibi temel özellikleri; farklı amaçlı ve farklı görevli beş pencereden oluşan platformu; simülasyon ve modelleme için kullanılan simulink programı ve çeşitli amaçlar için biraya gelmiş fonksiyonlar topluluğu olan araç kutuları (toolboxes) ile bir

bütün halinde düşünmeliyiz. Aşağıdaki şekil MATLAB Ailesini'nin yapısını göstermektedir [11]:



Şekil 1 : MATLAB'in Yapısı

2.2.4 MATLAB Programı'nın Kullanılma Amacı

MATLAB, tıbbi arařtırmalardan uzay arařtırmalarına kadar çok geniř yelpazede kullanılmaktadır. Kullanıcıya, güvenebileceđi hızlı bir analiz ve tasarım ortamı sađlar. Yüzlerce hazır fonksiyonla veya programcı tarafından yazılacak fonksiyonlarla iřlemler son derece kolaylařtırılabilir. Örneđin, yazılan bir MATLAB programını C/C++ dillerine dönüřtürebilir, 20. dereceden bir denklemin kökleri bulunabilir, 100x100'lük bir matrisin tersi alınabilir, bir elektrik motorunun denetimi gerçekleştirilebilir [12].

MATLAB, bütün genel mühendislik ve fen alanındaki çalışmalar ve iřlemler için gerekli olan matematiksel, istatistiksel ve mühendislik fonksiyonlarının hepsini içerir. MATLAB dilinin temelini oluřturan bu fonksiyonlar, matematikteki uzman kiřiler tarafından oluřturulmuř ve geliřtirilmiřtir. MATLAB, matematiksel iřlemleri gerçekleştirmede veri tipleri için (double, single, integer) geniř bir aralık sađlar. MATLAB, matematiksel iřlemleri gerçekleřtirmek ve veri analizini yapabilmek için ařađdaki fonksiyon tiplerini sunar [13]:

- Matrisler (Matrix functions)
- Vektörler (Vector functions)
- Lineer cebir (Linear Algebra)
- Matematiksel hesaplamalar (Dört İřlem)
- Trigonometrik fonksiyonlar
- Logaritmik ve üstel fonksiyon iřlemleri
- Denklem sistemlerinin çözümü
- Diferansiyel denklemler
- Polinomlar
- Olasılık / İstatistik
- Optimizasyon
- Türev / İntegral

MATLAB, sayısal hesaplamaların, denklem sistemlerinin çözümleri ve lineer cebir iřlemlerinin dıřında iki boyutlu fonksiyonlar ile üç boyutlu fonksiyonların oluřturulması ve bu fonksiyonların grafiklerinin çizilmesi konusunda çeřitli imkanlara sahiptir. MATLAB'in içindeki iki boyutlu ve üç boyutlu fonksiyonlar řunlardır [14]:

- **2-D (İki Boyutlu Fonksiyonlar) :** MATLAB, iki boyutlu fonksiyonların oluşturulması konusunda son derece başarılı bir programdır. MATLAB'in içindeki iki boyutlu fonksiyonlar şunlardır:
 - Grafikler
 - Histogramlar
 - Poligonlar
 - Animasyonlar
- **3-D (Üç Boyutlu Fonksiyonlar) :** MATLAB, üç boyutlu fonksiyonların oluşturulması ve bu fonksiyonların grafiklerinin çizilmesi konusunda kullanıcıya çeşitli imkanlar sunar. MATLAB'in içindeki üç boyutlu fonksiyonlar şunlardır:
 - Koni
 - Küre
 - Silindir
 - Yüzey Çizimleri

2.2.5 MATLAB Elemanları ve Bileşenleri

MATLAB'i elemanları ve bileşenleri ile bir bütün olarak, bir sistem yapısı içinde düşünmeliyiz. Bu sistemin çeşitli bölümleri ve bu bölümlerinde kendilerine ait özellikleri vardır. MATLAB'i genel bir sistem olarak düşünürsek; bu sistemi çeşitli yönleri ile ele almak gerekir. MATLAB programı yedi ana bölümden meydana gelir [15]:

2.2.5.1 MATLAB Programlama Dili

MATLAB, fonksiyonları, veri yapıları, nesneye dayalı programlama özellikleri ile matris/array(dizi) tabanlı bir programlama dilidir. Küçük programları kısa bir zaman diliminde, büyük çapta programları ise çabuk bir şekilde ve etkili sonuçlar vererek çalıştırma özelliğine sahiptir.

2.2.5.2 MATLAB Çalışma Alanı

MATLAB hem bir programlama dili hemde çeşitli hesaplamaların ve uygulamaların yapılmasına olanak tanıyan bir çalışma alanına ve platforma sahip bir bilgisayar programıdır. MATLAB bu platformda veri girişi ile veri çıkışı işlemlerini yönetir, M-dosyalarını ve uygulamalarını geliştirmek ve yönetmek için çeşitli araçlar kullanır. MATLAB platformu fonksiyonların, işlemlerin ve uygulamaların yerine getirilmesi için sağladığı özellikler ve araçlar ile kullanıcı ve programcılarının çalışmalarına

yardımcı olur. Ayrıca grafiksel kullanıcı arayüzü geliştirme konusunda da son derece faydalı bir platformdur. MATLAB platformunun en önemli kısmı komut penceresidir. Komut penceresi MATLAB ile ilgili her türlü uygulamanın yapıldığı, MATLAB'in ana kısmıdır.

2.2.5.3 Grafik Yönetimi

MATLAB çok güçlü ve yararlı bir grafik sistemine sahiptir. MATLAB grafik sistemi, verilerin hazırlanmasında ve görselleştirilmesinde çok değişik ve kendine has özellikleriyle kullanıcılara büyük kolaylıklar sağlamaktadır. Grafik yönetiminin asıl amacı; kullanıcılar için grafik tabanlı arayüzler oluşturmak, modelleme, simülasyon, histogram, animasyon gibi uygulamalar geliştirmek, görüntü işleme (image processing) ve grafik çizimleri meydana getirmek, koni, küre ve silindir gibi üç boyutlu matematiksel cisimler yaratmaktır. Matris ve vektörlerin grafiksel gösterimide MATLAB'in sağladığı faydalardan biridir. MATLAB, yüksek seviyede grafik oluşturma yeteneğine sahip bir bilgisayar programıdır. Çeşitli şemalar, figürler, animasyonlar, doğrular, histogramlar, iki ve üç boyutlu grafikler oluşturmak için kolay bir yapısı vardır.

2.2.5.4 Modelleme & Simülasyon

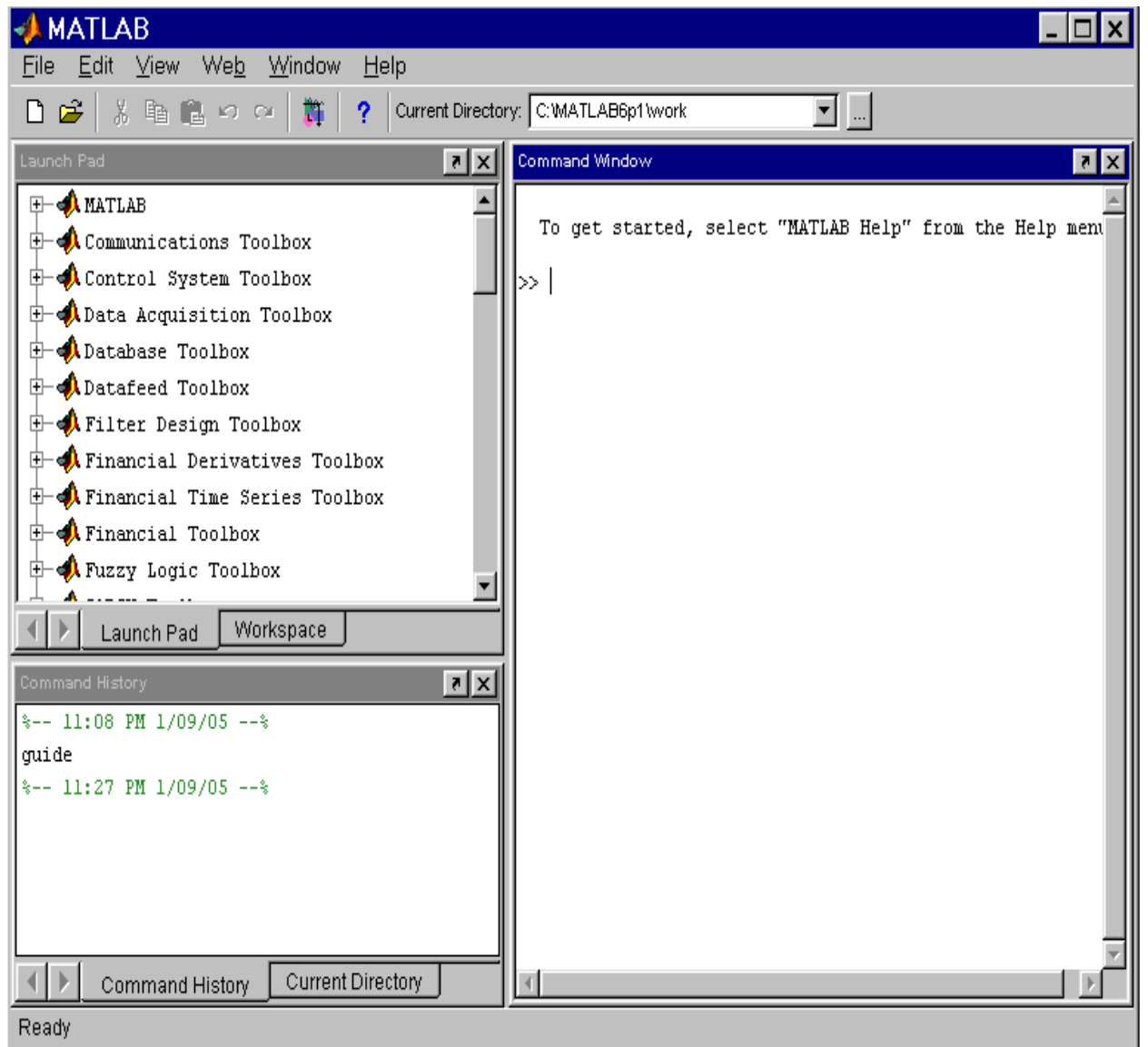
MATLAB model oluşturmada, özel uygulamalar geliştirmede ve dinamik sistemlerin simülasyonunda kullanılmak üzere faydalı ve yardımcı birçok Toolbox'a (çeşitli fonksiyonların ve metodların biraraya gelmesi ile oluşan yapılar) sahiptir. Ayrıca MATLAB, *Simulink* adlı özel bir uygulamaya sahiptir. Simulink, sistem modelleme ve dinamik sistemlerin simülasyonu için kullanılan faydalı bir uygulamadır. Simulink, bir dinamik sistemin modellenmesi için blok şemalar sunar.

2.2.5.5 MATLAB Araç Kutuları (Toolboxes)

Araç kutuları (toolboxes), çeşitli bilimsel alan ve konularda yazılan MATLAB fonksiyon dosyalarından oluşurlar. Araç kutuları, MATLAB'in kullanılabilirliğini arttırdığı gibi çok yönlü olmasında sağlarlar. Araç kutuları (toolboxes) özel amaçlı uygulamalarda sundukları yardımlar ile hem MATLAB'in önemli bir bilgisayar programı olmasında başrol oynamış , hemde MATLAB kullanıcılarının en önemli yardım kaynağı olmuşlardır. Araç kutuları (toolboxes), MATLAB'teki özel uygulamalara ve işlemlere getirdikleri çözümler ile MATLAB'in sağ kolu olarak adlandırılırlar.

2.2.6 MATLAB Platformu/Arayüzü'nün Genel Görünümü

MATLAB, her türlü fonksiyon ve uygulamanın oluşturulması, yönetilmesi ve kaydedilmesi; program organizasyonu konusunda algoritma oluşturulması, programın çalıştırılması ve hatalarının bulunması; veri depolama, veri giriş-çıkış işlemleri; her türlü MATLAB ile ilgili konuda yardım alınması, ve metinsel yazılar ile grafiklerin oluşturulması konusunda etkili bir platforma sahiptir. MATLAB, grafiksel arayüz tasarımı için bağımsız bir platform ve dizayn metodları sağlar. MATLAB platformu/arayüzü sayısal hesaplamaların yapılması, algoritmaların geliştirilmesi ve grafiklerin oluşturulması için son derece elverişli ve uygundur. MATLAB platformu/arayüzü aşağıdaki şekilde gösterilmiştir :



Şekil 2 : MATLAB Platformu / Arayüzü

MATLAB programı başlatıldığı zaman çeşitli menüler ve pencereler açılır. MATLAB platformu birtakım işleri ve uygulamaları yürütmek ve uygulamaya sokmak için bazı bölümlere ayrılmıştır. Bu bölümler ve işlevleri şunlardır [16]:

2.2.6.1 Çalışma Alanı Penceresi (Workspace)

Komut penceresinde (command window) yazılmış olan kod ile ilgili bilgilerin tutulduğu yerdir. Kod içinde kullanılan değişkenlerin listesi ve bu değişkenlerin çeşitli özellikleri (adı, büyüklüğü, kaç byte olduğu vs. gibi) burda tutulur. Çalışma alanı penceresi, komut penceresinde yapılan hesaplamalara ya da uygulamalara ait bütün elemanları ve nesnelere bünyesinde bulunduran ve muhafaza eden yerdir.

2.2.6.2 Geçerli Dizin Penceresi (Current Directory Window)

Bu pencere, MATLAB içindeki dosyaların bulunduğu yeri bulmak ve değiştirmek için kullanılır. Bulunan dosyalar komut penceresi (command window) içinde kullanılabilir. Geçerli dizin penceresi (current directory window); komut penceresinde yapılan hesaplamaların, işlemlerin ya da uygulamaların nereye kaydedildiğini gösteren, eski dosyaların yerini bulmamızı kolaylaştıran bölümdür. Bu pencere aradığımız dosyaların yerini bulmamıza, istediğimiz zaman bu dosyalara yeni birşeyler eklememize, dosyanın yerini değiştirmemize, dosyayı yeniden adlandırmamıza ya da dosyayı silmemize olanak tanır.

2.2.6.3 Yeni Bir Eylem Başlatma Padi (Launch Pad)

MATLAB platformunun sol üst kısmında bulunur. Yeni bir eylem başlatılacağını gösteren, içinde özel uygulama yerleri (toolbox) olan ve MATLAB'in en özel uygulaması Simulink'ide içinde bulunduran yerdir. Bütün genel işlemler ve uygulamalar burada toplanır. MATLAB'in içinde bulunan bütün fonksiyonlara buradan ulaşabiliriz. Bu pencerenin içinde MATLAB'in fonksiyonlarına ulaşmayı sağlayan birtakım bölümler vardır. Bu bölümler şunlardır [17]:

- **Control System Toolbox (Kontrol Sistemleri Araç Kutusu) :** Sistemin kontrolünün yapıldığı yerdir. Kontrol sistemleri araç kutusu, durum uzayı tekniklerini kullanarak kontrol mühendisliği ve sistemler teorisi ile ilgili fonksiyonlardan oluşmaktadır.
- **Image Processing Toolbox (Görüntü İşleme Araç Kutusu) :** Görüntü işleme yeridir.

- **MATLAB Compiler (MATLAB Kod Derleyeci)** : MATLAB'te yazılan kodun derlenmesinin yapıldığı ve sonuçlarının görüntülenmesinin sağlandığı yerdir.
- **Optimization Toolbox (Optimizasyon Araç Kutusu)** : Optimizasyon işlemlerinin yapıldığı yerdir.
- **Signal Processing Toolbox (Sinyal İşleme Araç Kutusu)** : Sayısal işaret işleme yeridir. Sinyal işleme araç kutusu, 1-boyutlu ve 2-boyutlu sayısal sinyal işleme (zaman serilerinin analizi) ile ilgili fonksiyonlardan oluşmaktadır. Ayrıca sinyal işleme araç kutusu, sayısal filtreler için geliştirilen analiz ve tasarım fonksiyonları ile güç spektrumu analizine ilişkin fonksiyonları da içermektedir.
- **Symbolic Math Toolbox (Sembolik Matematik Araç Kutusu)** : Matematiksel sembollerin bulunduğu yerdir.
- **System Identification Toolbox (Sistem Tanımlama Araç Kutusu)** : Sistem kimliklendirme ve tanımlama yeri. Sistem tanımlama araç kutusu, parametrik modelleme ve sistem tanımlama ile ilgili fonksiyonlardan oluşmaktadır.

2.2.6.4 Geçmişteki Komutlar Penceresi (Command History Window)

Komutların bütün hepsinin listesini düzenli bir şekilde, sıralı olarak bulunduran penceredir. Yazılmış olan kodlar kaydedildiği zaman bu pencerede tutulur. Bu pencereden, daha önce yazılmış olan komutların yerini öğrenebiliriz. Geçmişteki komutlar penceresi (command history window), MATLAB platformunun sol alt kısmında bulunur. İçinde daha önce komut penceresinde oluşturulmuş olan uygulama ve komut kümelerini barındırması bakımından önemli bir penceredir. MATLAB içerisinde programlar oluşturulup geliştirildiği zaman program ile ilgili her türlü detayı en ince ayrıntısına kadar görme imkanı sunar. Örneğin; program içinde kullanılan nesnelere, değişken isimleri, veri tipleri gibi.

2.2.6.5 Komut Penceresi (Command Window)

Bu pencere, MATLAB masaüstünün ana penceresidir. Bu ana pencere komut ve yorumları yazmak, sonuçlarını görmek için kullanılır. MATLAB açıldığında karşımıza gelen pencere, MATLAB'in komut penceresidir. Komut penceresi, kullanıcı ile MATLAB komut yorumlayıcısı arasında iletişimi sağlayan bir ara

yüzdür. Yorumlayıcı hazır hale geldiğinde, (>>) iletisi karşımıza gelir. Bu ileti, MATLAB'e komut ya da komut dizileri girilebileceğini gösterir. MATLAB'in candamarı olan yerdir. Komut penceresi (command window); program oluşturulması, geliştirilmesi ve çalıştırılması, komutların yorumlanması, sonuçlarının görülmesi ve çeşitli konularda her türlü uygulamanın yapılmasına olanak tanıyan bir bölümdür. MATLAB içerisinde program oluşturulurken yapılan her türlü hatayı kullanıcıya mesaj verme yoluyla duyurma özelliği ile kullanıcı dostu bir penceredir [18].

2.2.6.6 Yardım Penceresi (Help Window)

Kullanıcıya MATLAB ile ilgili her türlü konuda yardım eden ve aranılan konular ile ilgili kullanıcıya ekstra bilgiler veren bir penceredir. MATLAB içindeki konuları belli bir düzen ve sırada tutar. Böylece kullanıcının bulmak istediği şeylere daha rahat bir şekilde ulaşma imkanı sağlanmış olur. MATLAB ile ilgili her türlü konu ve öğrenilmek istenen yeni bilgiler burda aranabilir.

2.2.6.7 Şekil Penceresi (Figure Window)

Grafiksel nesnelerin görüntülenmesinde kullanılan bir penceredir. MATLAB, tüm grafik çıktılarını “şekil penceresi” olarak adlandırılan ayrı grafik pencerelerinde görüntüler. Bununla birlikte şekil penceresi, yaratılan figür ile ilgili özelliklere, eksen bilgilerine ve figür içindeki nesnelerin bilgilerinede ulaşmayı sağlar.

2.2.7 MATLAB'te Programlama Kavramı

MATLAB'i sayısal problemleri çözme becerisinin dışında birde programlama dili olma özelliği ile de incelemek gerekir. MATLAB, birçok işlemi yerine getirmek amacı ile biraraya gelen güçlü fonksiyonlar kümesinden meydana gelir. MATLAB, diğer programlama dilleri (C, C++, JAVA, FORTRAN gibi) ile karşılıklı olarak etkileşim içinde olmanın yanı sıra, yazılım verimliliği ve yazılım etkinliği sunma özellikleri ile yazılım güvenilirliği sağlama konusundada son derece başarılıdır [19] [20] [21].

2.2.7.1 MATLAB'te Programlamaya Genel Bakış

Her programlama dilinin olduğu gibi MATLAB'inde kendine özgü fonksiyon, kelime, sayı, sembol ve sözdizimi yapısı vardır. Buda, MATLAB'i diğer programlama dillerinden farklı kılar. MATLAB'te program yaratmak için çeşitli uygulamalar geliştirilir. Bu uygulamalar programlama kodu için MATLAB'e özgü sözdizimi yapısını kullanır. Eğer bu sözdizimi yapısı kullanılmazsa program içinde

kullanıcıların hiç sevmediği ve görmek istemediği hatalar oluşur. MATLAB'te geliştirdiğimiz yazılımlar, yordam (procedure) veya fonksiyon (function) şeklindedir. MATLAB'te yordamlar ve fonksiyonlar, M-dosyaları (M-files) olarak adlandırılır ve .m uzantısını alır. MATLAB'te, küçük-büyük harf farkı olduğundan program kodunun yapısına çok dikkat edilmesi gerekir. Aksi halde program kodu içinde çeşitli hatalar oluşacaktır. MATLAB programının, C ve FORTRAN programlama dilleri ile benzer yanları vardır. MATLAB, çeşitli fonksiyonların birleşmesi ve biraraya gelmesi ile oluşmasına rağmen, anlaşılması zor bir programlama dili değildir. MATLAB, diğer programlama dilleri ile etkileşim kurabilmek için alt yöntemler ve metodlar geliştirir. MATLAB'te programlama elemanları olarak döngüler, kontrol yapıları ve akış diyagramları kullanılır.

2.2.7.2 MATLAB'te Programlamanın Yapısı

MATLAB ile yazılan programda, yapılanları satır satır görme ve test etme olanağı olduğundan programı oluşturmak ve çalıştırmak oldukça kolaydır. MATLAB programlama dili, anlamak ve uygulama geliştirmek için oldukça kolay ve anlaşılır bir dildir. MATLAB'te her türlü algoritmaya uygun yapıda program kodu yazılabilir. MATLAB platformu bağımsız bir arayüzdür. Bu yüzden başka dillerin fonksiyonlarına ve uygulamalarına kolaylık ve uygunluk sağlar. " >>" işareti "*command prompt*"olarak adlandırılır ve bu kısma çalıştırılacak olan komutlar yazılır. Bu işaretin hemen yanında yanıp sönen I şeklindeki işaret, komut ve metin yazma cursor'u yani imlecidir. Bu işaretin olduğu yerde klavyeden giriş yapılabilir demektir. MATLAB yazılan bu komutu işletip sonucu verir. Daha sonra yine kullanıcı tarafından komut yazılmasını bekler. Böylelikle kullanıcı MATLAB'e istediği kadar komutu interaktif olarak girip çalıştırabilme imkanı bulacaktır. MATLAB, komut temelli bir programdır. MATLAB'te kodların yazılıp çalıştırıldığı dosyalar, m dosyaları (m-files) olarak adlandırılır ve genel olarak *.m şeklinde gösterilir. m kelimesi MATLAB'in ilk harfinden gelir. Tıpkı C++ dilindeki .c veya Microsoft Word programındaki .doc gibi.

2.2.7.3 MATLAB'te Programlama ile İlgili Özellikler

MATLAB'in çeşitli uygulamaları ve fonksiyonları yerine getirmek amacı ile kendine özgü bir programlama dili vardır. Bu programlama dilinde kendine has birtakım özellikleri vardır. Bu özellikler şunlardır :

- MATLAB'in bazı veri tipleri C programlama dilinden farklıdır. Bu yüzden C programlama dili ile yazılmış olan bir kodu MATLAB programlama diline dönüştürürken hataları önlemek için bu veri tiplerinide MATLAB'e uygun hale getirmek gerekir.
- MATLAB, kullanıcıya program içindeki bazı fonksiyonlar için C programlama dilinde olduğu gibi çeşitli kütüphaneler sunar.
- Program geliştirme zamanı, C ve FORTRAN programlama dilleri ile kıyaslandığında daha kısadır.
- MATLAB, C programlama dili ile yazılmış program kodunu etkili bir şekilde kendine özgü şekle dönüştürme olanağı sağlar.
- MATLAB, C ve FORTRAN programlama dilleri ile kıyaslandığında oldukça yavaştır.

2.2.7.4 MATLAB'te Operatörler (Operators)

MATLAB'te sayısal hesaplamalar ile ilgili kod yazmak için bazı operatörleri kullanmak gereklidir. MATLAB'te kullanılan 3 tür operatör vardır :

2.2.7.4.1 Aritmetik Operatörler

MATLAB'te 5 tane temel matematiksel operatör vardır . Bu temel aritmetik operatörler cebirsel işlemler ve matris işlemlerinde kullanılırlar. Aritmetik operatörler ve sembolleri aşağıdaki tabloda gösterilmiştir :

Operatör Adı	Operatör Sembolü
Toplama	+
Çıkarma	-
Çarpma	*
Bölme	/
Üs alma	^

Tablo 1 : Aritmetik Operatörler ve Sembolleri

2.2.7.4.2 Mantıksal Operatörler

MATLAB'te 3 tane temel mantıksal operatör vardır . Mantıksal operatörler ve sembolleri aşağıdaki tabloda gösterilmiştir :

Operatör Adı	Operatör Sembolü
AND	&
OR	
NOT	~

Tablo 2 : Mantıksal Operatörler ve Sembolleri

2.2.7.4.3 Karşılaştırma Operatörleri

MATLAB'te 6 tane temel karşılaştırma operatörü vardır . Karşılaştırma operatörleri ve sembolleri aşağıdaki tabloda gösterilmiştir :

Operatör Adı	Operatör Sembolü
Küçük	<
Küçük eşit	<=
Büyük	>
Büyük eşit	>=
Eşit	==
Eşit değil	~=

Tablo 3 : Karşılaştırma Operatörleri ve Sembolleri

2.2.7.5 MATLAB'te Programlama ile İlgili Kurallar

MATLAB'te program geliştirilirken uyulması zorunlu olan birtakım kurallar vardır. Dikkat edilmesi gereken bu kurallar şunlardır :

- Biten komut satırının sonuna noktalı virgül (;) konulur.
- MATLAB'te değişken tanımlama yoktur.
- Genel olarak virgül ve noktadan sonra boşluk bırakılmalıdır. Bu işlem program kodunun anlaşılabilir olması açısından önemlidir.
- Bir operatörden sonra boşluk bırakılmalıdır. Bu işlem program kodunun etkinliğini artırır.
- İki satır aynı hizada olmalıdır. Bu işlem program içinde düzenin sağlanması ve görselliği etkilemesi açısından önemlidir.

2.2.7.6 İfadeler (Expressions)

MATLAB diğer programlama dillerinde olduğu gibi bir giriş (input) olarak çeşitli matematiksel ve metinsel ifadeler sağlar. İfadeleri üç ana madde altında inceleyebiliriz :

1. Sayılar (Numbers)
2. Değişkenler (Variables)
3. Fonksiyonlar (Functions)

2.2.7.6.1 Sayılar

Sayılar, hem matematiksel işlemlerde hemde program kodu yazımında sağladıkları faydalar ile önemli bir yere sahiptirler. Diğer programlama dillerinde (C, C++, FORTRAN) olduğu gibi MATLAB'te de sayılar önemli bir yer tutar. MATLAB'te üç tip sayı türü vardır :

- Doğal Sayılar
- Reel Sayılar
- Karmaşık (Complex) Sayılar

2.2.7.6.2 Değişkenler

MATLAB, program kodunu oluştururken değişkenler kullanarak kullanıcının programı daha rahat anlamasını sağlar. MATLAB'te değişkenler ile ilgili kurallar şunlardır :

- Değişkenler küçük harf ile başlamalıdır. Aksi halde bu durum program içinde karışıklığa neden olacaktır.
- Değişken adı iki ya da daha fazla kelimedenden oluşturulacaksa kelimeler arasına alt çizgi (_) konulmalıdır.
- Değişkenler hem hafızada fazla yer tutulmaması hemde hatalara sebep olmaması bakımından tekrar tekrar kullanılmamalıdır.
- MATLAB yeni bir değişken ismi ile karşılaştığında, otomatik olarak ans isminde bir değişken oluşturur ve uygun bir bellek miktarı ayırır.
- Küçük-büyük harf farklılığı vardır. Örneğin; BYTE ile Byte aynı şey değildir.
- MATLAB'te en fazla 31 karakter kullanılabilir.
- C ve FORTRAN da olduğu gibi bütün değişkenler bir değere atanmak zorundadır. Bütün matematik işlemleri 16 haneli rakama kadar yapılır.

2.2.7.6.3 Fonksiyonlar

MATLAB'in çok güçlü ve çok kapsamlı bir fonksiyon yapısı vardır. MATLAB'te bazı fonksiyonlar daha önceden yerleştirilmiş bazıları ise sonradan dışarıdan yerleştirilmiş M-dosyalarından oluşan MATLAB kütüphanelerinde tanımlanmışlardır.

MATLAB'te fonksiyonlar ile ilgili kurallar şunlardır :

- Değişken ve fonksiyon adı, bir sayı ile değil harf ile başlamak zorundadır.
- İki veya daha fazla kelimenin birleşmesi ile oluşturulacak fonksiyon isimleri arasına altçizgi (_) konulmalıdır.

2.2.7.7 MATLAB Temel Yazım Notasyonu (Syntax)

MATLAB'te herşey bir dizi (array) olarak işleme konur. Bir dizi; skaler, vektör, matris veya metinsel dizge (karakter dizisi) olabilir.

- 1x1 dizisi skaler (scaler) veya metin dizgesi (string) gösterir. $a = 3$, $b = -20.56$, $c=3e-4$, $d = 2+5*i$, $e = \text{"Bir tamsayı giriniz :"}'$ gibi. Metin dizgeleri (text string), `'...'` arasında verilir.
- nx1 veya 1xn dizisi bir vektör (vector) gösterir. $x=[0, 2, 4, 6]$ ifadesinde x , 1x4 boyutunda bir vektördür. `[]` köşeli parantezler içindeki sayıların arasında virgül (,) veya en az bir veya daha fazla boşluk olmalıdır.
- nxm veya mxn dizisi bir matris (matrix) gösterir. `[]` köşeli parantezler içindeki sayı grupları arasında noktalı virgül (;) olmalıdır. ; işareti matrisin sütunlarını ayırır. Sayı gruplarında virgül (,) veya en az bir veya daha fazla boşluk olabilir.

2.2.7.8 Diziler (Array)

MATLAB'in en temel işlem elemanı ve veri tipi, dizilerdir. Dizi, en genel matematiksel tanımı ile nümerik ve metinsel değerler topluluğudur. MATLAB'te herşey bir dizi olarak işleme konur. MATLAB'te üç tip dizi ifadesi bulunmaktadır :

1. Reel ile kompleks sayıları ifade eden çift kat veya nümerik diziler (double veya numeric array)
2. Genelleştirme ve çeşitli tipleri ifade eden n-boyutlu diziler (n-dimensional array)

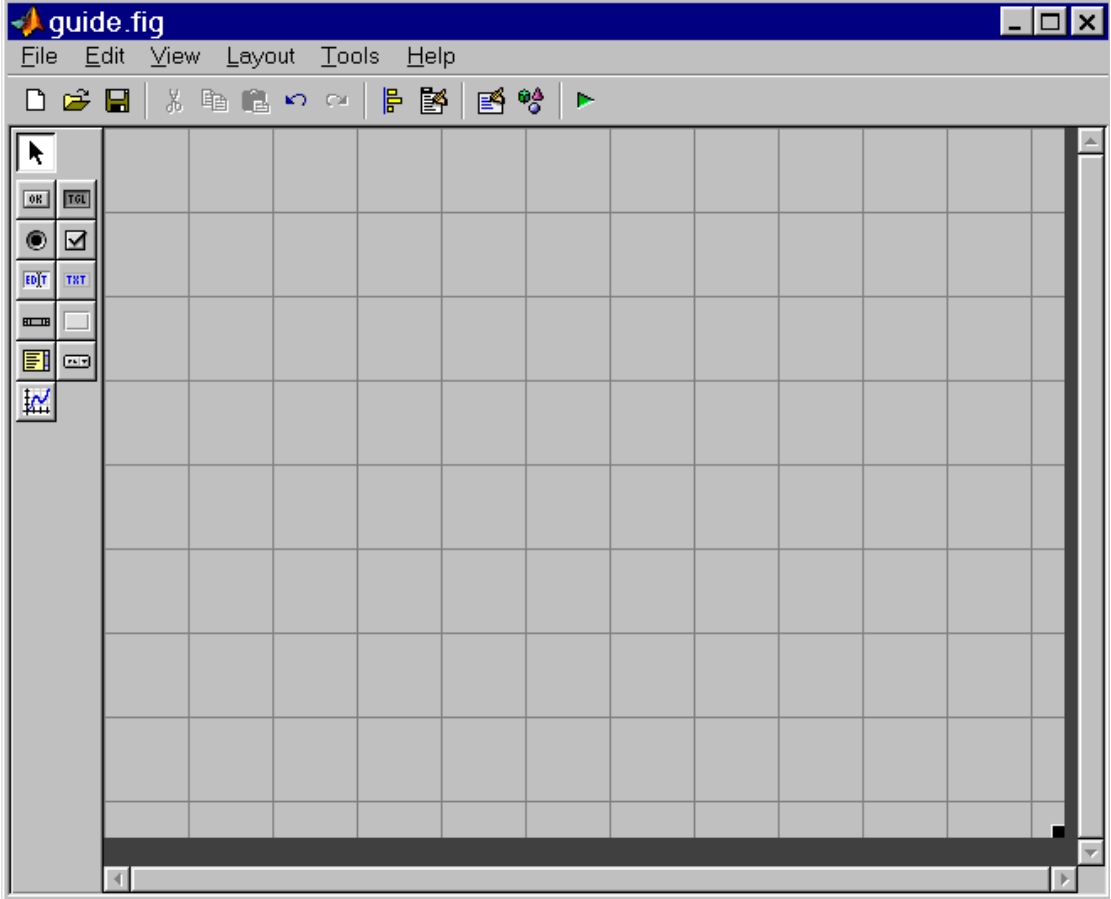
2.2.7.9 Açıklayıcı Yorumlar (Comment)

MATLAB'te açıklayıcı yorumlar (comment), program koduna açıklayıcı bilgiler eklemek için kullanılırlar. Açıklayıcı yorumlar ile ilgili özellikler şunlardır :

- Açıklamalar büyük harfle ve yüzde (%) işaretinden sonra başlamalıdır.
- Açıklamalar program satırının ne yapmak istediğini anlatacak tarzda yazılmalıdır.
- Açıklama kullanıcıya program hakkında kolaylık sağlamalıdır.

2.2.8 MATLAB'te Grafikselle Kullanıcı Arayüzü (GUI) Oluşturma

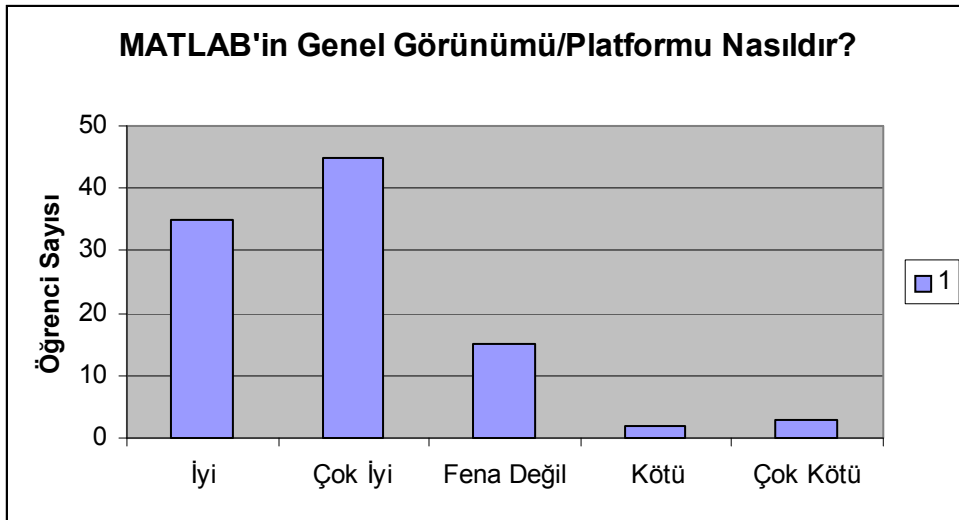
MATLAB'te, grafikselle kullanıcı arayüzü oluşturmak oldukça kolaydır. MATLAB'te kullanıcı arayüzünü oluşturmak, dizayn etmek ve geliştirmek için Grafikselle Kullanıcı Arayüz Geliştirme Alanı (Graphical User Interface Development Environment - GUIDE) kullanılır. GUIDE, bir kullanıcı arayüzündeki dizayn ve düzen işlemlerinin bütün fonksiyonlarını, kavramlarını ve temellerini içinde barındıran bir yapıdır. MATLAB, grafikselle kullanıcı arayüzünün gelişme işlemlerini diğer bilgisayar programlarından daha hızlı ve kolay yapar. Bunun nedeni ise MATLAB'in içinde kullanıcı arayüzünün gelişimine yardım edecek birçok aracın olmasıdır. Aşağıdaki şekil Grafikselle Kullanıcı Arayüz Geliştirme Alanı'nı (GUIDE) göstermektedir [22]:



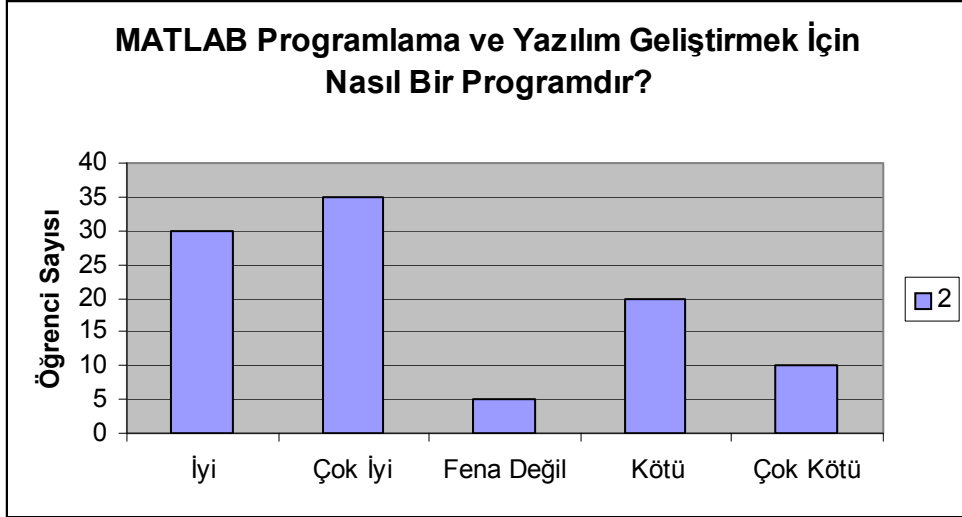
Şekil 3 : Grafiksel Kullanıcı Arayüz Geliştirme Alanı (GUIDE)

2.2.9 MATLAB Anket Araştırma Sonuçları

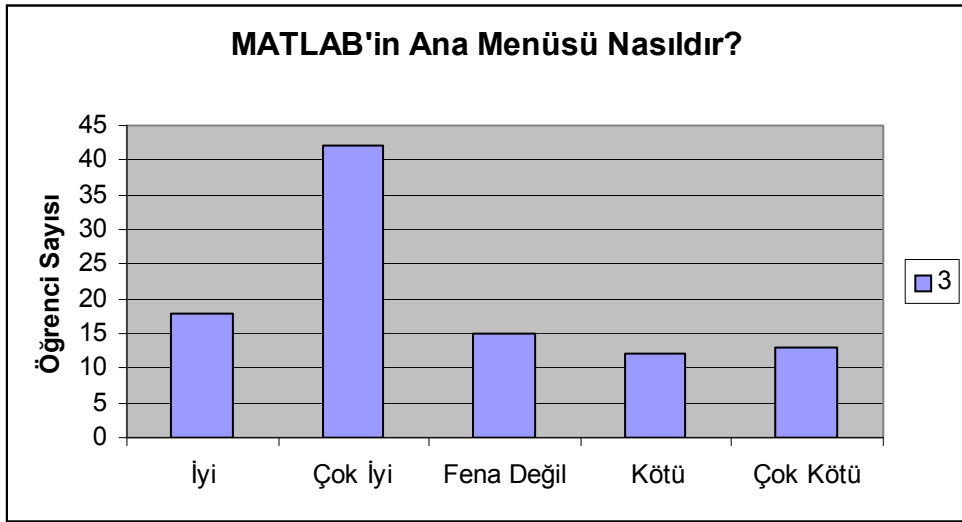
24 Mayıs 1998 yılında Saskatchewan Üniversitesinin Elektrik Mühendisliği Bölümü'nün MATLAB ile ilgili olarak 100 öğrenci üzerinde yapmış olduğu yirmibeş soruluk anket araştırma sonuçlarına göre şu veriler elde edilmiştir [23]:



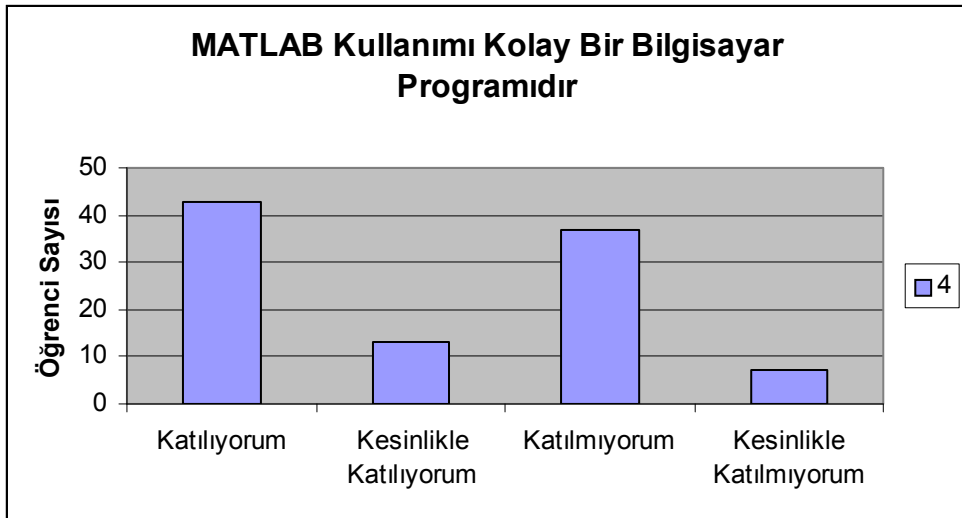
Şekil 4 : MATLAB Anket Araştırma Sonucu-1



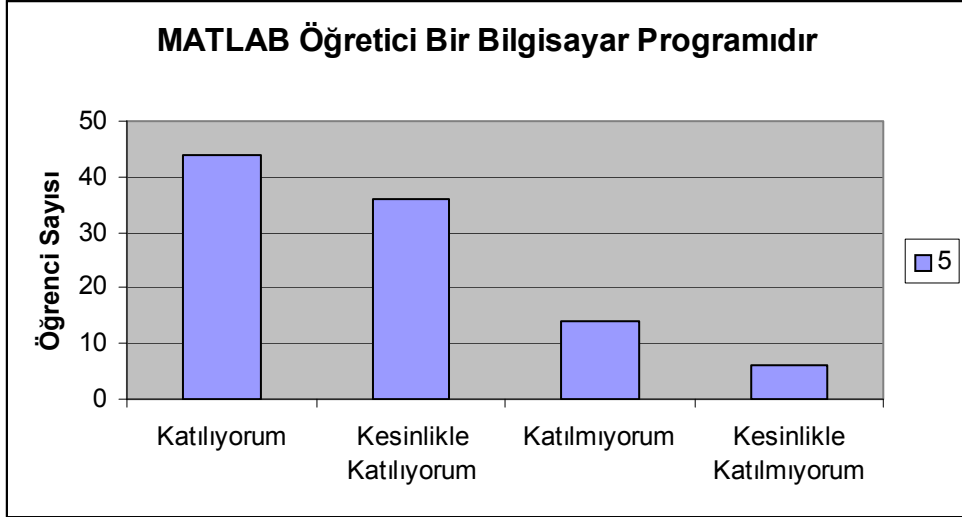
Şekil 5 : MATLAB Anket Araştırma Sonucu-2



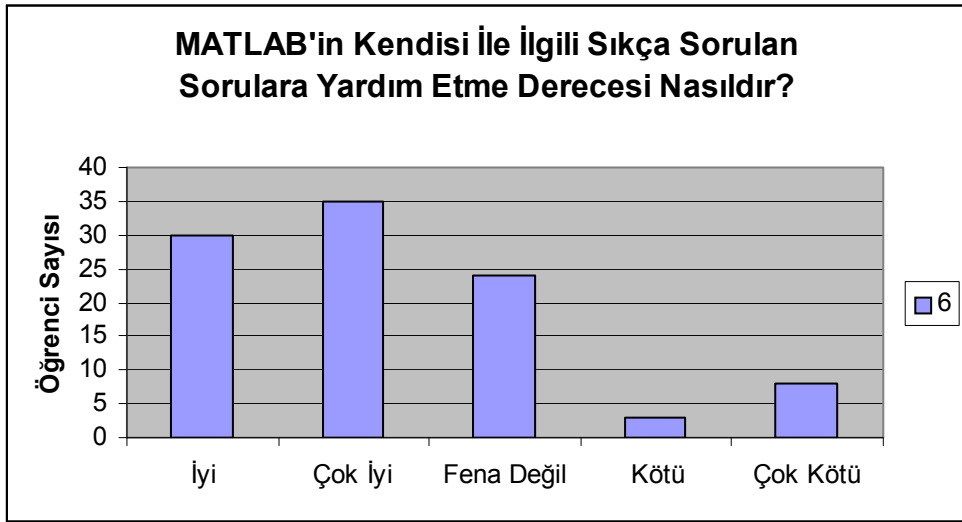
Şekil 6 : MATLAB Anket Araştırma Sonucu-3



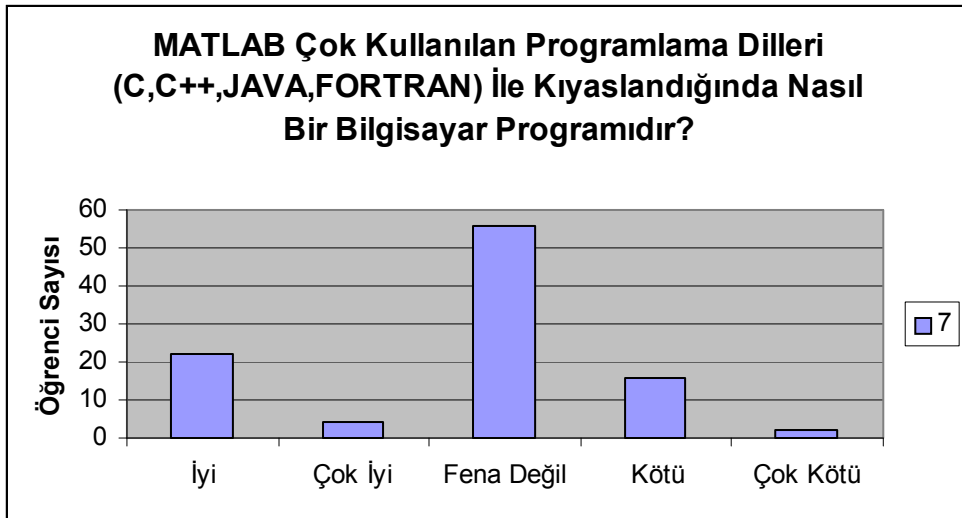
Şekil 7 : MATLAB Anket Araştırma Sonucu-4



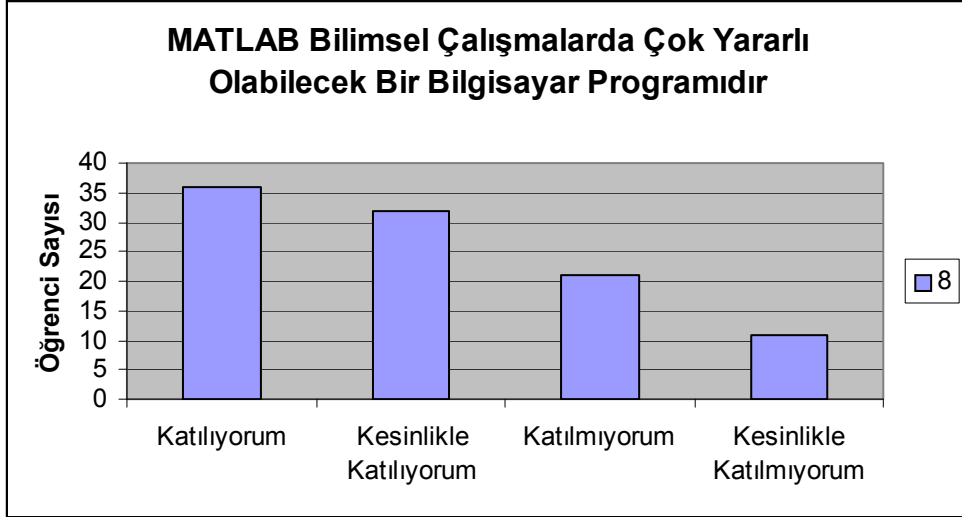
Şekil 8 : MATLAB Anket Araştırma Sonucu-5



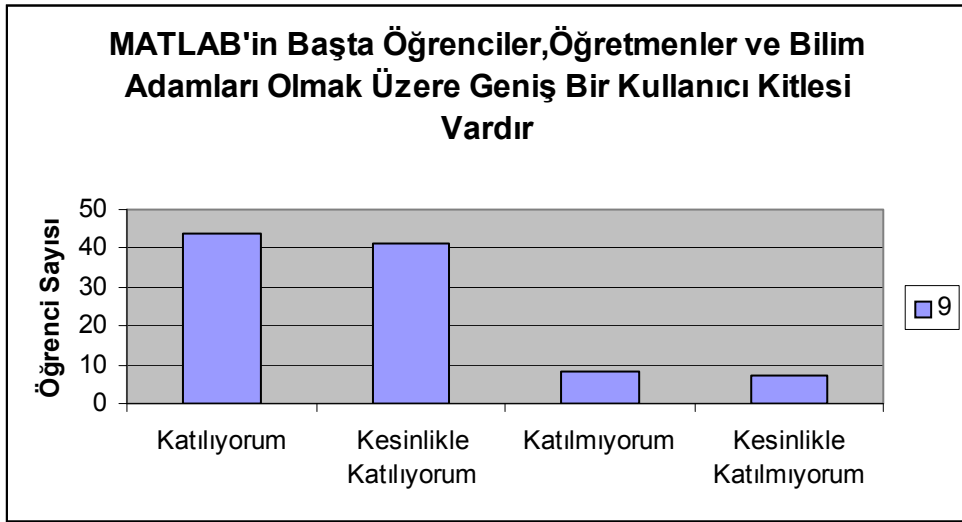
Şekil 9 : MATLAB Anket Araştırma Sonucu-6



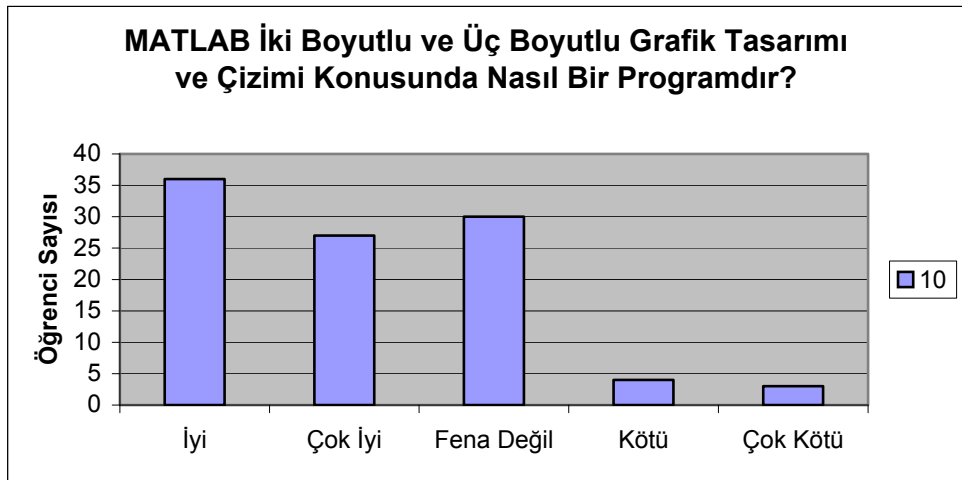
Şekil 10 : MATLAB Anket Araştırma Sonucu-7



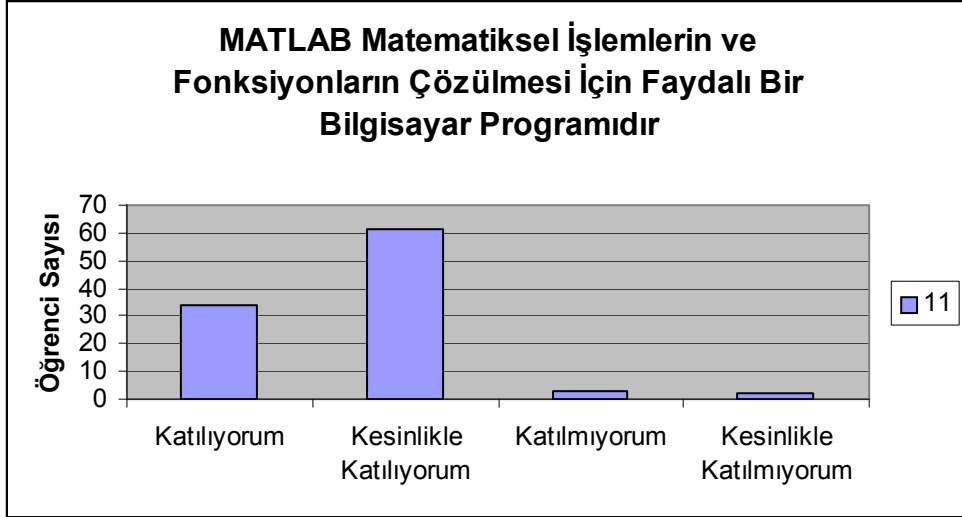
Şekil 11 : MATLAB Anket Araştırma Sonucu-8



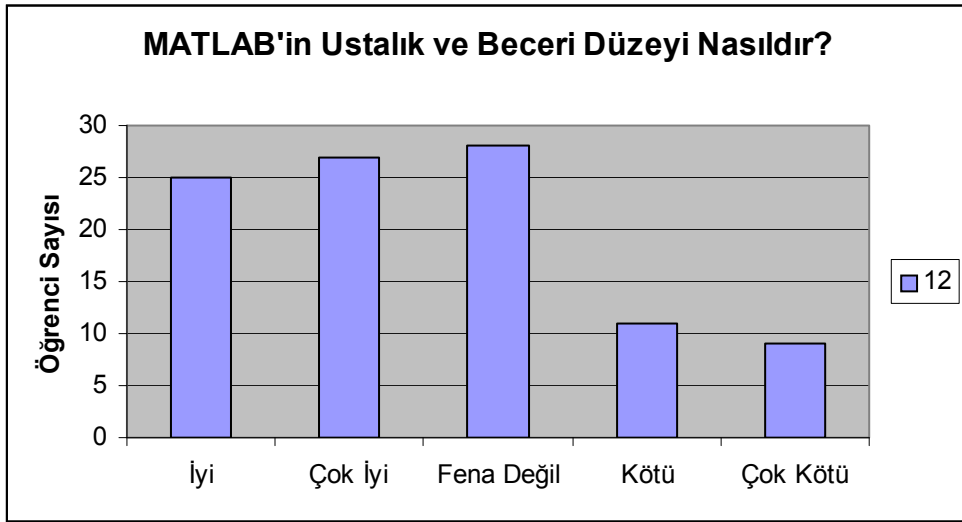
Şekil 12 : MATLAB Anket Araştırma Sonucu-9



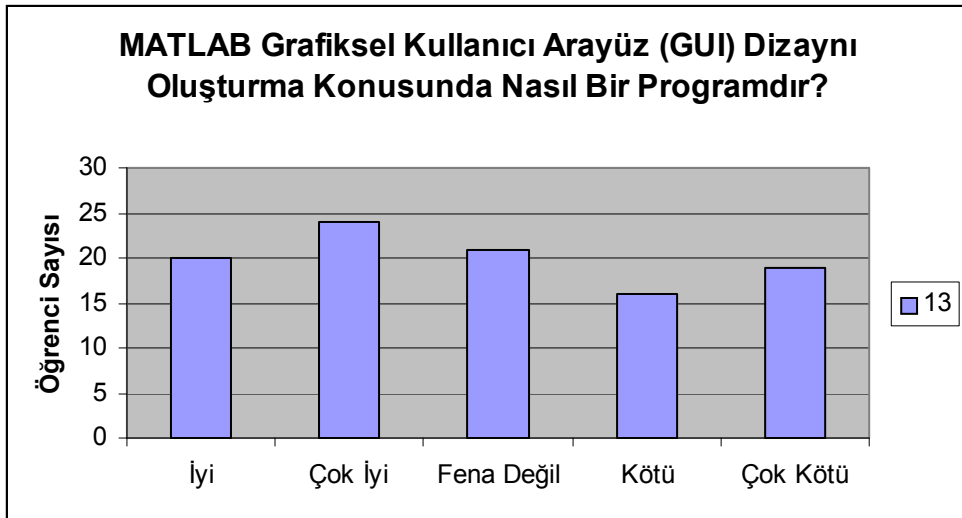
Şekil 13 : MATLAB Anket Araştırma Sonucu-10



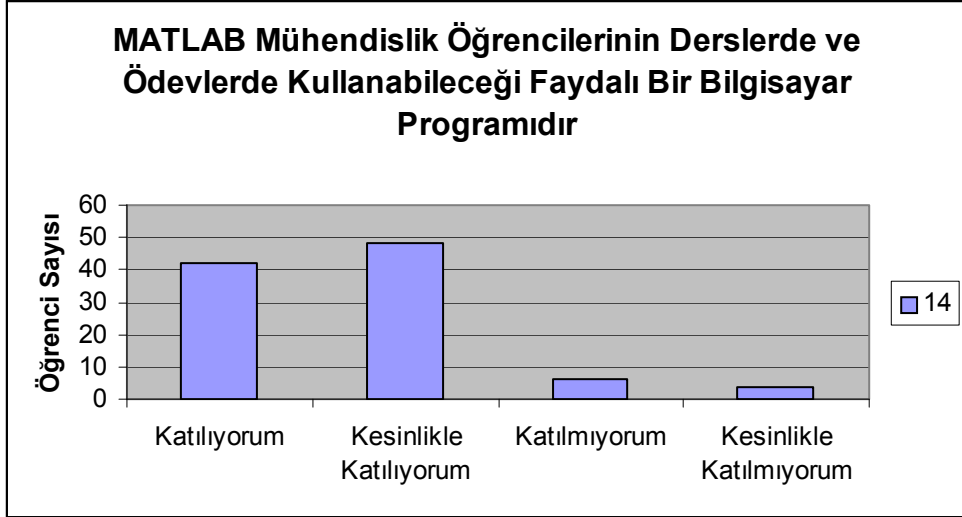
Şekil 14 : MATLAB Anket Araştırma Sonucu-11



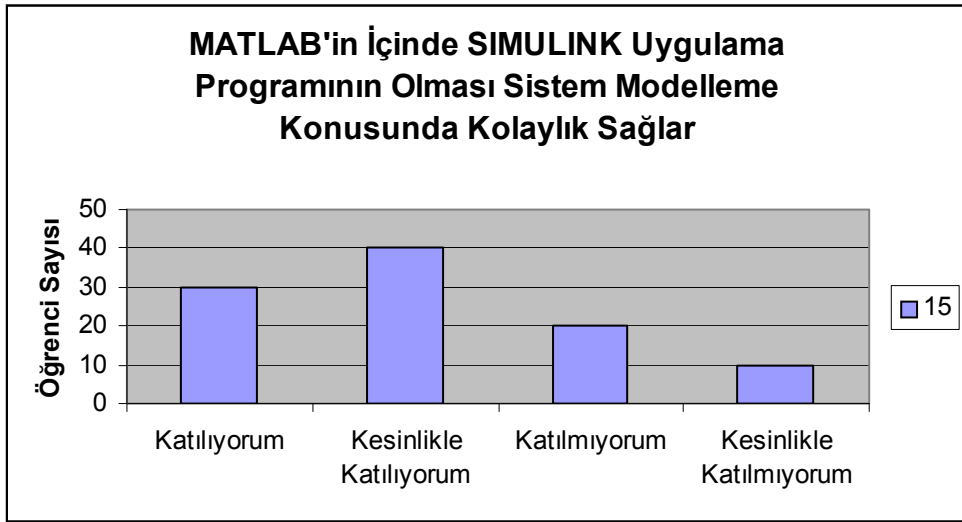
Şekil 15 : MATLAB Anket Araştırma Sonucu-12



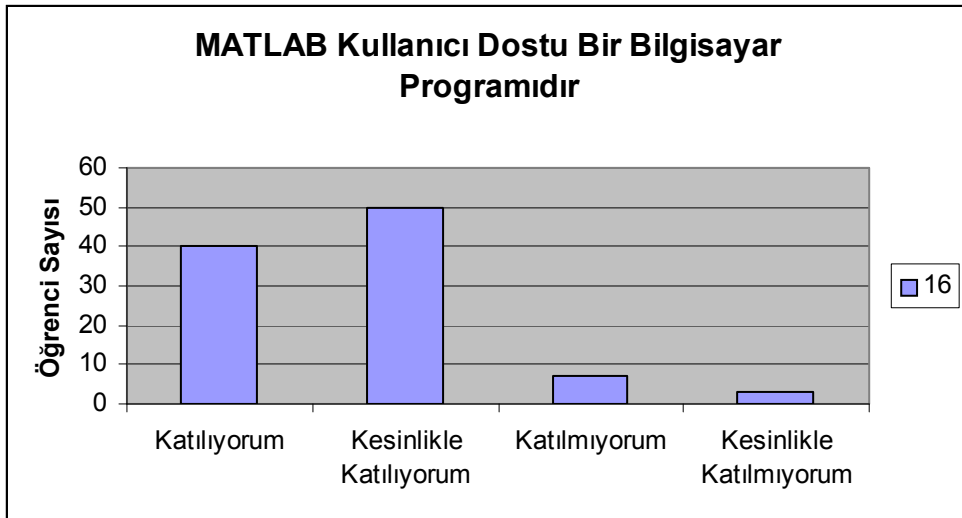
Şekil 16 : MATLAB Anket Araştırma Sonucu-13



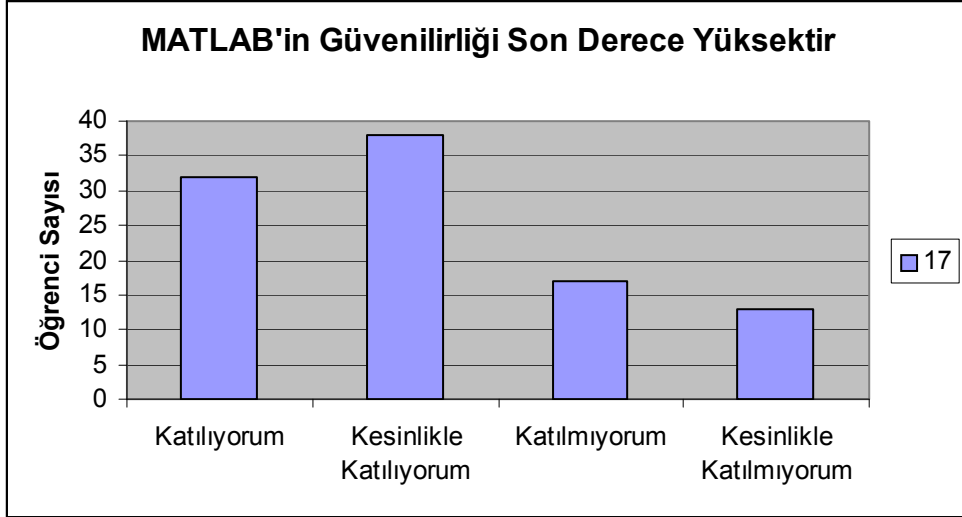
Şekil 17 : MATLAB Anket Araştırma Sonucu-14



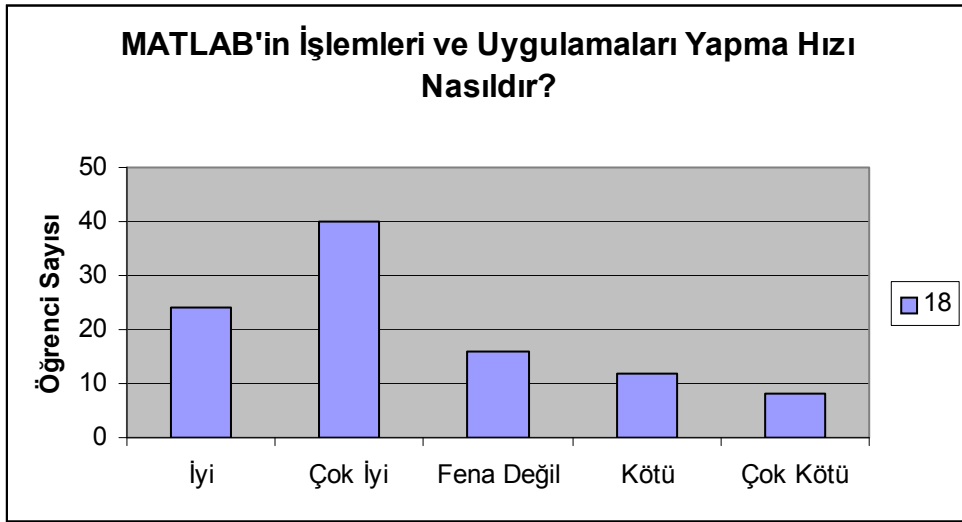
Şekil 18 : MATLAB Anket Araştırma Sonucu-15



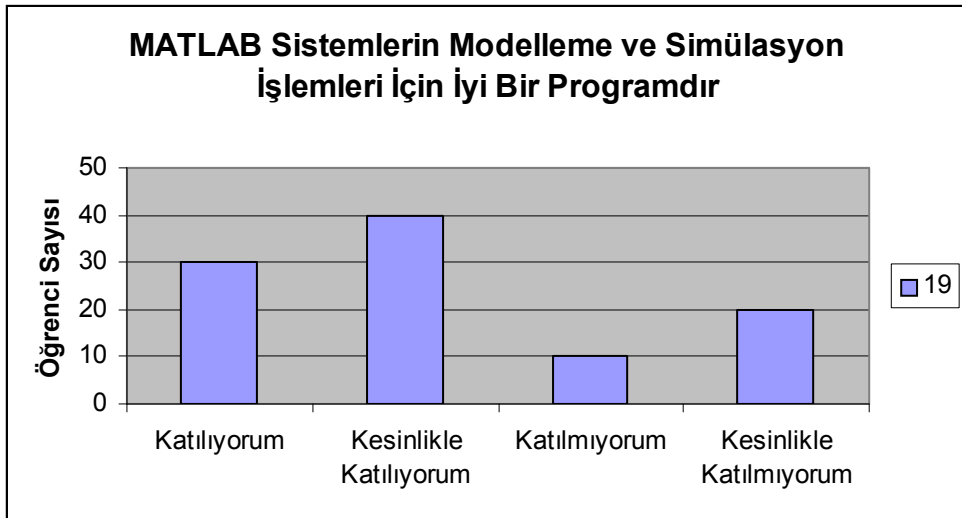
Şekil 19 : MATLAB Anket Araştırma Sonucu-16



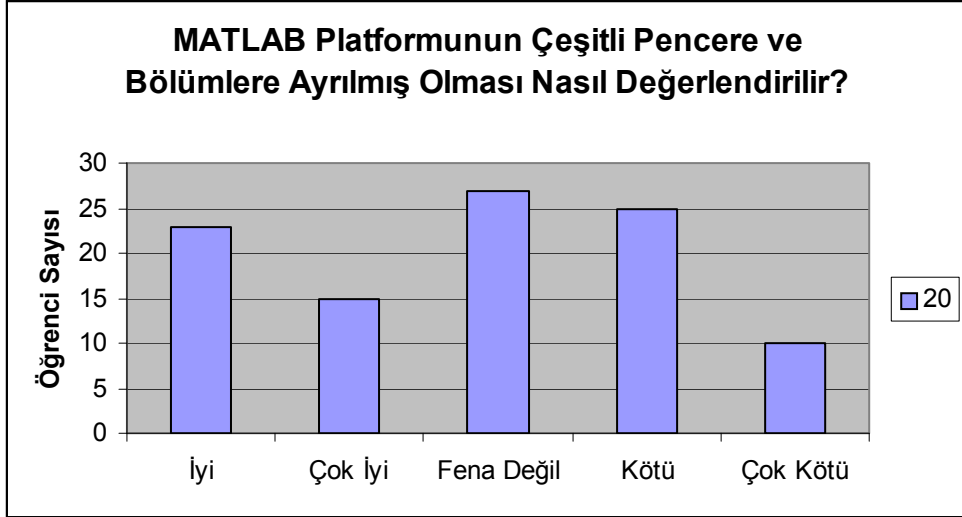
Şekil 20 : MATLAB Anket Araştırma Sonucu-17



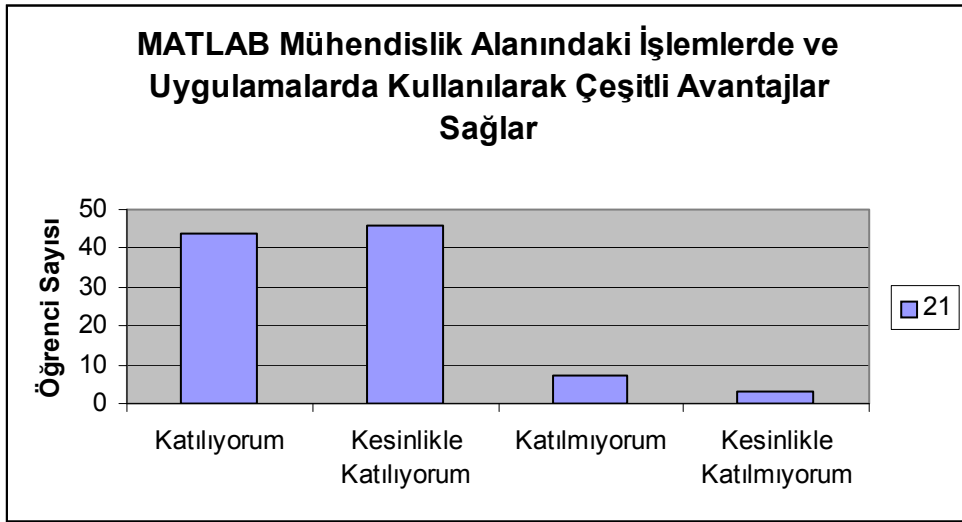
Şekil 21 : MATLAB Anket Araştırma Sonucu-18



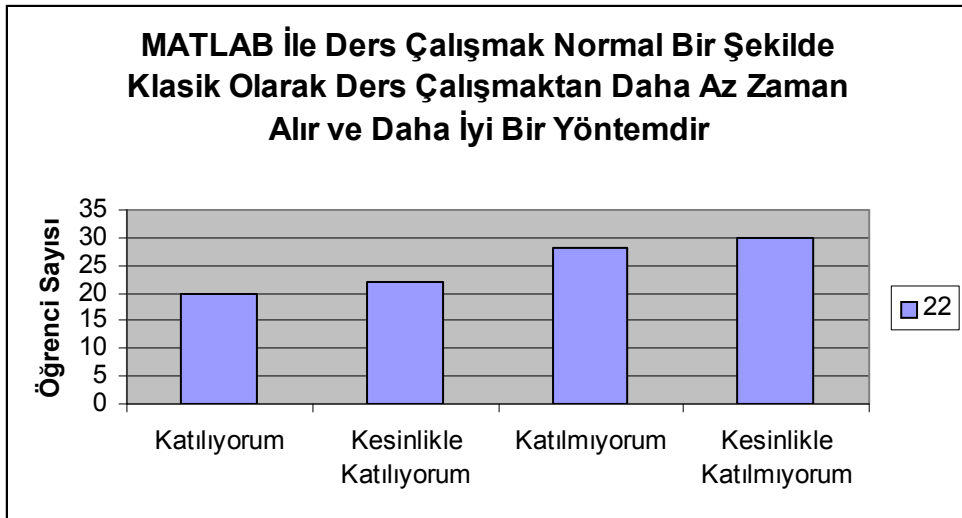
Şekil 22 : MATLAB Anket Araştırma Sonucu-19



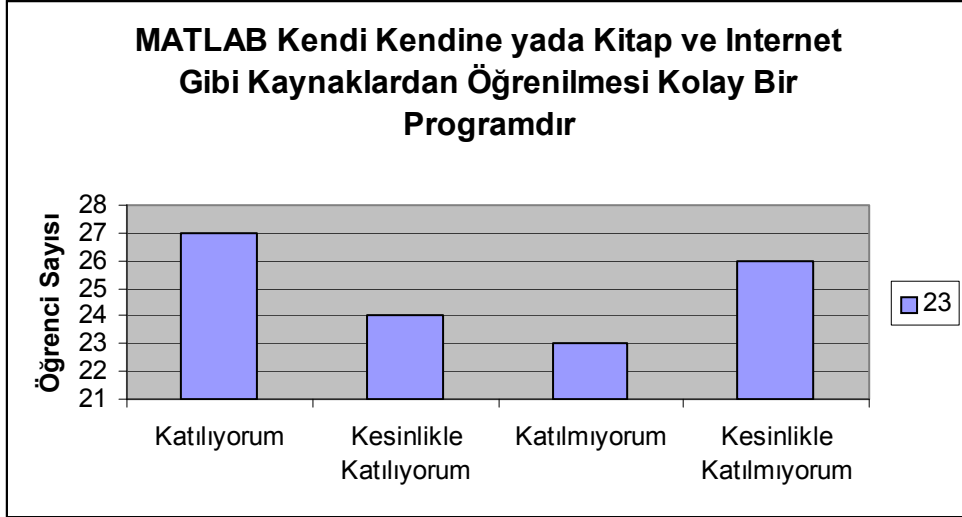
Şekil 23 : MATLAB Anket Araştırma Sonucu-20



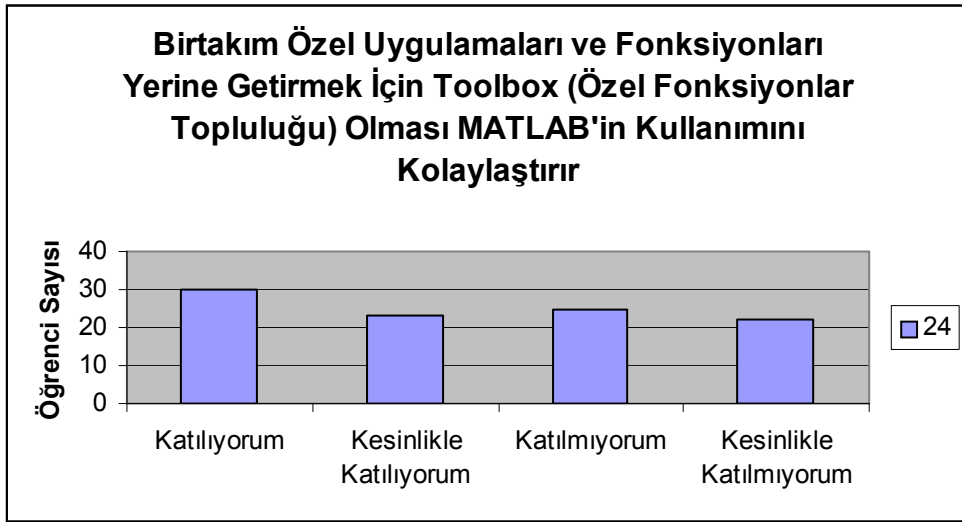
Şekil 24 : MATLAB Anket Araştırma Sonucu-21



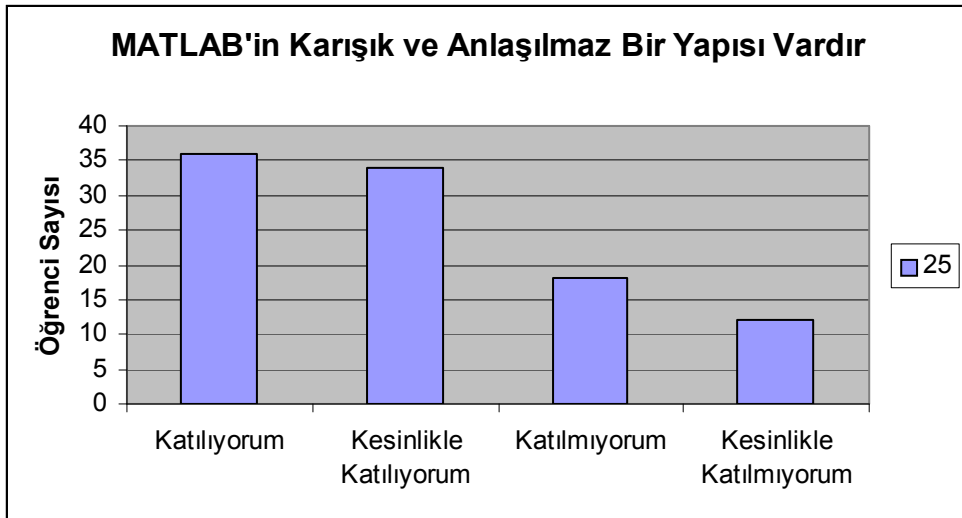
Şekil 25 : MATLAB Anket Araştırma Sonucu-22



Şekil 26 : MATLAB Anket Araştırma Sonucu-23



Şekil 27 : MATLAB Anket Araştırma Sonucu-24



Şekil 28 : MATLAB Anket Araştırma Sonucu-25

MATLAB programını yakından tanımak, özelliklerini görmek ve sağladığı faydaları bilmek için yapılan yukarıdaki anket sonuçlarına göre; aşağıdaki veriler elde edilmiştir :

- Kullanılması oldukça kolaydır.
- Anlaşılması kolay bir programlama dilidir.
- Programlama fonksiyonları mükemmeldir.
- Kullanıcısına geniş fonksiyonlar sunar.
- Öğrenme kolaylığı sağlar.
- Kullanıcı dostudur.
- Grafik tasarımı ve çizimi konusunda kolaylıklar sağlar.
- Grafikselle arayüz tasarımı ile ilgili çeşitli özellikler sunar.
- Program yardım sistemi vardır. Bu yardım sistemi, MATLAB ile ilgili sıkça sorulan birçok soruda kullanılan, faydalı bir yöntemdir.
- Simulink programı, sistem modelleme ve simülasyon işlemleri için kolaylıklar sağlar.
- İşlemleri ve uygulamaları yapma hızı fevkalade iyidir.
- Oldukça güçlü bir programdır ve yüksek bir kaliteye sahiptir.
- Güvenilirliği son derece yüksektir.

2.3. MathCAD Programı

Matematiksel ifadeleri; matematiksel notasyonlar kullanarak çözen, bilgiler arasındaki iletişimi kolayca sağlayan MathCAD yazılım programı, MathSoft Inc. firmasının bir ürünüdür. MathCAD; dünya da 1.5 milyonun üzerinde kullanıcısı ile, teknik profesyoneller ve akademisyenler için endüstri standardında hesaplama paketi olarak bilinir. MathCAD, bu kullanıcıların deneyimleri ve istekleri doğrultusunda, mühendisler, bilim adamları ve teknik kullanıcılar için en güçlü, en hızlı ve en kararlı masaüstü çözümünü sunmak üzere yapılmıştır. Bununla birlikte, kullanıcıları daha verimli kılabilmek için heyecan verici özellikler de sunar. Üniversite öğrencisinden profesöre; teknikerden mühendise kadar geniş bir kitleye hitap eden MathCAD, farklı işleri bir arada yapabilecek benzersiz bir ortam sunmaktadır.

MathCAD programı, 1988 yılında matematiksel işlemlerin nümerik çözümleri için dizayn edilmiştir. 2005 yılı itibariyle 17 yıllık mazisi olan MathCAD, kullanıcıların

matematik gelişimine katkıda bulunmak amacı ile, bu alanda oluşturulmuş en eski ve en geniş kapsamlı yazılımlardan biridir. MathCAD; matematikteki, sayı (aritmetik,geometri) ve sembol (cebir, matematiksel analiz) kavramlarının karışımıdır. MathCAD kullanıcılarının, MathCAD hakkında sahip oldukları en olumsuz düşünceler : MathCAD'in, ticari amaçlı bir yazılım olarak oluşturulduğunun ve fonksiyonlarının sadece gösteri amaçlı olduğunun düşünülmesidir [24].

Elle yapılacak birçok işlemi bilgisayar üzerinde yapma kolaylığı sağlayan MathCAD, çok yönlü ve kullanıcı dostu bir program olup, kullanıcının sıkılmamasına özen gösteren bir kullanıcı arayüzüne sahiptir. MathCAD, bilim dünyasında, matematik ve mühendislik öğrencileri tarafından sıcak bakılan ve rabet görülen, dünya çapında bir programdır. MathCAD'in, kullanıcıların doğal etkileşim içerisinde bulunacakları bir çalışma alanı vardır [25].

MathCAD sembolik matematik yazılım programı, Maple programından esinlenerek ve Maple programı örnek alınarak yaratılmıştır. MathCAD, Maple programının parçalarını birleştiren popüler bir programdır. Bu yüzden MathCAD'in yapısı, Maple programına çok benzemektedir. Bir başka deyişle, Maple programını kullanan bir kişinin MathCAD'i kullanması ve anlaması oldukça kolaydır [26].

MathCAD, spreadsheet (tablolama programı) bir programdır. MathCAD yazılımının, Microsoft Excel, Lotus ve Gnumeric programları gibi spreadsheet (tablolama programı) özelliği vardır. Bu özelliğinin olması, MathCAD'in en önemli güçlerinden ve avantajlarından biridir. MathCAD'in spreadsheet bir program olmasının iki anlamı vardır [27]:

1. Hesaplama işlerine yarayan program
2. Zor ve karmaşık olan finansal ve mühendislik problemlerine ve tablolarına verilen genel tanım

MathCAD'in spreadsheet bir program olmasından dolayı sahip olduğu çeşitli özellikler vardır. Bu özellikler şunlardır :

1. MathCAD'in platformu/çalışma alanı, tablo şeklindedir.
2. MathCAD'te her bir tablo, hücrelerden oluşmaktadır.
3. MathCAD'te her bir hücre, sayı, metin ya da formül içermektedir.

4. MathCAD’te hesaplamalar, tekrar tekrar yapılma olanağına sahiptir.

İstatistiksel ve sayısal hesaplamaların çözümünde kullanılan MathCAD, bugün öğrencilerin ödevlerini yararken kullandıkları en önemli kaynaklardan biridir. Üniversitelerde bilgisayar mühendisliği bölümünde, liselerde ise matematik ve bilgisayar öğretmenlerinin derslerde, mühendislik, matematik ve bilimsel alandaki işlemlerin çözümünde kullandığı MathCAD, nümerik analiz ve istatistik alanlarındaki başarıları ve sağladığı faydalar ile ün yapmış bir matematik bilgisayar yazılım programıdır. Ayrıca; MathCAD, matematiksel problemlerin çözüm yollarının öğretilmesinde de faydalıdır. MathCAD, sembolik hesaplamaların yapılması, nümerik analiz metodlarının kullanılması ve matematiksel problemlerin çözümü konularında kullanılan EUREKA, MATLAB, MATHEMATICA yazılım programları gibi yararlı olan popüler bir programdır [28].

MathCAD, basitten karmaşığa birçok bilimsel ve teknik problemin çözümü ve analizi için çeşitli fonksiyonlara ve özelliklere sahiptir. İleri düzey matematiksel fonksiyonları ve görsel yeterlilikleri içinde bulunduran MathCAD, istenilen hesaplamaların yerine getirilmesini sağlar. Kullanıcıya problemleri daha kolay anlama olanağı sunar. MathCAD, işlemleri hızlı yapma beceresi olan ve verimliliği en üst seviyede kullanıcıya sağlayan bir programdır. MathCAD, mühendislik ve bilimsel alandaki uygulamalarda ve işlemlerde sağladığı faydalarla kullanışlı bir programdır. İstatistik ve finans alanında kullanılması son derece uygun bir bilgisayar programı olan MathCAD, veri analizi konusunda da yeterli beceriye ve kaliteye sahiptir [29].

Matematiksel kaynaklar ve belgeler için en iyi yazılım programı MathCAD’tir. Neredeyse eksiksiz bir matematik yazılım paketi olan MathCAD, matematikçilerden bilim adamlarına; mühendislerden öğrencilere kadar matematikle ilgili tüm kullanıcılara hitap etmektedir. Bir programlama dilinden çok, kelime işlemci mantığıyla matematiksel işaretlerin kullanımına olanak tanıyan MathCAD programı, hesaplama işlemlerini otomatik olarak yapmaktadır. Uzun bir süredir piyasada olan MathCAD’in birçok farklı sürümü vardır. MathCAD programının içinde bulunan canlandırma özelliği, herhangi bir değişkenin değiştirilmesi durumunda oluşabilecek sonuçları, kullanıcıya canlandırılmış olarak anlatabilmektedir. Bunun için her adımı değerlendiren MathCAD, oluşabilecek sonuçları bir AVI dosyası olarak sunmaktadır.

Çalışma sayfalarının MathML içeren HTML dosyaları olarak kaydedilebilmesiye, sunulan başarılı özelliklerden bir diğeri oluşturmaktadır. Özelleştirilmiş bir XML biçimi olan MathML, matematik ifadelerinin gösterilmesini kolaylaştırmaktadır. Bilgisayar dünyasında yeni ve daha önceleri piyasaya sunulan matematik tabanlı bilgisayar programları göz önüne alındığında; MathCAD hala matematiksel belgeler oluşturmak için dizayn edilmiş en iyi programdır. Ayrıca, yapılan çalışmaları daha fazla kullanıcı ile paylaşmayı sağlayacak olan web özellikleri de, programın en önemli artlarından biridir. MathCAD programının en büyük dezavantajı : Karmaşık problemlerin çözümünde yavaş kalmasıdır [30].

MathCAD, birçok teknik problemin çözümü için çeşitli yapılarda fonksiyonlar ve yüzlerce operatör kullanır. MathCAD programı, matematik ile ilgili işlemleri ve uygulamaları gerçekleştirmek ve bu işlemlerin iletişimini sağlamak için entegre edilmiş bir çalışma alanına sahiptir. Çeşitli işlemler ve farklı türdeki değişik birçok uygulamada kullanıcılara hızlı çözümler sağlar. MathCAD'in, her türlü hesaplamalarda, matematiksel ve mühendislik alanındaki cebirsel işlemlerde, grafik oluşturmada ve yazılım geliştirme konusunda yüksek bir performans gücü vardır. MathCAD'in hesaplama konusundaki yeterliliği ile ilgili genel görünümünü oluşturanların başında linear algebra (lineer cebir) işlemleri gelmektedir. Ayrıca; MathCAD, nümerik analiz fonksiyonlarını gerçekleştirme ve istatistiksel hesaplamalar konusunda da son derece faydalı bir programdır. Otomatik olarak birimleri birbirine dönüştürme özelliğine sahip, ölçümler ile ilgili uygulamaları gerçekleştirme beceresi olan MathCAD, scalar (skaler) ve vector (vektörel) işlemleri oluşturmak için kullanılan bir programdır. MathCAD yazılım programı; sistem dizayn, modelleme, analiz ve optimizasyon işlemleri için de kullanılır [31].

MathCAD, bilim ve mühendislik dünyasındaki milyonlarca profesyonel kişinin hesaplama ve programlama ihtiyaçlarını karşılamak için dizayn edilmiş güçlü bir programdır. Gerçek hayattaki problemlerin çözümü için, sembolik ve nümerik matematiksel ve veri analiz özelliklerini kullanarak yeni üretimler sunmaya çalışır. Karmaşık problemleri kolay ve hızlı bir şekilde yapan MathCAD, farklı programlardaki birçok uygulama ile çalışmak için dizayn edilmiştir. Bütün bunların dışında MathCAD, bir iş organizasyonundaki proje yönetimini kolaylaştırmak ve yardımcı olmak amacı ile de kullanılır. MathCAD, 1000'den fazla şirket ve kurum

tarafından kullanılması sebebi ile, endüstri sektöründe büyükçe bir kullanım alanı vardır. NASA'dan, enerji sektörüne kadar geniş bir kullanıcı ağı olan MathCAD, 500'den fazla devlet tarafından kullanılmaktadır. Ayrıca, 2000'den fazla lise, kolej ve üniversitede; öğrenciler ve fakülteler tarafından da kullanılan bir yazılımdır [32].

MathCAD, matematik alanındaki en güçlü yazılım programlarından biridir. Matematiksel eşitlikler, metinler ve diyagramların birleşiminden oluşan MathCAD, uygulama geliştirmek için kullanıcı açısından oldukça kolay ve uygun bir çalışma alanına/platform sahiptir. MathCAD; denklemleri, formülleri, sayıları ve grafikleri değerlendirmek için tek bir metod kullanan yararlı bir yazılım programıdır. MathCAD'te denklemler, analitik ve nümerik olacak şekilde fonksiyonlar kullanılarak çözülebilir. Kolay kullanım özelliği olan MathCAD, işlerin analizi ve organizasyonu konusunda son derece yararlı bir programdır.

MathCAD, 2005 yılı itibariyle mühendislik alanının gelişimi için dünya da 50'den fazla ülkede ve 1.5 milyon kullanıcı tarafından kullanılmaktadır. MathCAD'te yazılım geliştirirken program kodu yazmak güvenilir, kolay ve rahat emin olunabilecek bir iştir. MathCAD'in oluşturulan program kodunu gözden geçiren ve tekrardan inceleyen özelliğinin yanında, güçlü bir mimari yapısı ve gelişmiş bir arayüzü vardır. MathCAD, her zaman daha kararlı ve daha iyi bir bellek yönetimini sağlamaya çalışır. MathCAD'in, ürün gelişimine ve mühendislik dizaynına yardımcı olmak amacı ile, mühendislik metodlarını yöneten ve bu metodları kullanan bir yapısı vardır. MathCAD, birçok organizasyona mühendislik bilgilerini ve bilincini aşlamaya çalışan ve bu konudaki yaratıcılıklarını arttırmayı planlayan bir programdır. Bununla birlikte; MathCAD, kullanıcıların işlerini yaparken daha üretici, daha verimli ve daha yaratıcı olmaları için de uğraş verir [33].

2.3.1 MathCAD Programı'nın Önemi

MathCAD'i bu kadar ilgi çekici hale getiren ve önemli kılan şey : Güçlü hesaplama programlarından daha fazla kendini geliştirmiş olmasıdır. İşleri fazla zaman harcamadan, kullanıcıyı zahmete sokmadan kolayca yapması, MathCAD'e olan inancın artmasına neden olmuştur. Bu durum da, kullanıcı sayısının çoğalmasına yol açmıştır. MathCAD programı ile kullanıcılar, farklı uygulamalardaki verileri, grafiksel elemanları ve hesaplamaları tek bir dokümanda birleştirebilirler. MathCAD,

projelerdeki sıkıcılığı önlemeye, zaman kaybını azaltmaya ve oluşan hataların kaynağını bulmaya çalışır. MathCAD'in etkileşimlilik özelliğinin olması, en büyük avantajıdır. Kullanıcılar, MathCAD ile yaptıkları işlemlerin doğruluğundan emin olabilirler. MathCAD, kullanıcılara bu garantiyi vermektedir. Verilen bu garanti ile, MathCAD'i kullananların yaptıkları işlemlerde verinin takip edilme zorunluluğu ortadan kalktığı gibi, kullanıcıların sorumluluk duygularıda en aza indirilecektir.

MathCAD'in en önemli artlarından biri : Önemli hesaplama programlarından biri olan Mathematica'dan kullanımının daha kolay olması ve bir programlama programı gibi görünmesidir. MathCAD, bir sistemin modeli oluşturulurken dizayn etme, geliştirme, yönetme ve test etme işlemleri için kullanılır. MathCAD'in bu özelliği, özellikle karmaşık sistemler analiz edilirken yararlı olmaktadır. MathCAD'in iletişim, işbirliği ve proje yönetim özellikleri, bir şirket ya da kurumdaki mühendislerin etkinliğini ve üretkenliğini artırır. MathCAD, oluşturulan program kodunu, diğer program geliştiren programlama dillerinin çalışma alanlarına kolayca aktarma özelliğine sahiptir. MathCAD, yapısal dizaynı kolayca yaptığı gibi, kurum ve şirketlere çeşitli konularda rehberlik eder ve üretim metodları sunar. Bununla birlikte, sistemi doğrular ve kontrol eder.

MathCAD, mühendislerin yaratıcılıklarını geliştirmek, onlara yeni ufuklar açmak amacı ile dünya da geniş kullanıma sahip, teknik uygulamalardan biridir. MathCAD, mühendislerin uğraştıkları konu ile ilgili bireysel olarak uzmanlaşmalarına yardımcı olabilmektedir. Zamanın değerini bilenler için MathCAD, işin kalbi olarak tanımlanabilir. Bir bilgisayar programının yapması rüya gibi görülen uygulamalar, MathCAD ile daha kolay yapılır bir hale gelmiştir. Bilgisayarlarda geniş çaplı uygulamalar geliştirmek için dizayn edilmiş programlardan biri olan MathCAD, geniş kullanıcı sayısı ile dünya çapında bir yazılım olarak göze çarpmaktadır [34].

2.3.2 MathCAD Programının Genel Özellikleri

MathCAD programının içeriğini gösteren, yapısını anlatan genel özellikleri şunlardır [35] :

1. MathCAD, basitten karmaşığa, sayısal ya da sembolik her türlü problemin, en iyi çözümünü vermek için uğraşır.

2. MathCAD, her türlü fonksiyon ve matematiksel ifade için geniş bir tanım kümesi aralığı sağlar.
3. MathCAD, “canlı matematik” teknolojisini kullanır. MathCAD’teki bu özellik, işlemlerin yapıldıktan hemen sonra sonucunun görülmesi anlamına gelmektedir.
4. MathCAD, aritmetik işlemleri kendi fonksiyonları ve kendi matematik operatörleri ile yapar.
5. MathCAD’in metin editörü özelliği olduğundan dolayı, elektronik ortamda herhangi bir bilgiyi saklamak ve çeşitli notlar tutmak mümkündür.
6. MathCAD, grafikleri tekrar çizme, grafiklerdeki verileri değiştirme ya da silme özelliklerine sahiptir.
7. MathCAD, hem iyi bir kelime işlemci, hem iyi bir denklem editörü, hem de iyi bir kod geliştiricidir.
8. MathCAD, eşitliklere, fonksiyonlara ya da denklemlere, açıklayıcı not ve bilgiler ekleme özelliğine sahiptir.
9. MathCAD, herhangi bir programlama dili gibi programlama yazım notasyonunu kullanmak yerine, kullanıcıya matematik dilini kullanarak kolaylık sağlar.
10. MathCAD’te, spreadsheet (tablolama programı) özelliği olduğundan, denklemler farklı bir türde ve şekilde yazılır. Örneğin;
 - Bir programlama dilinde denklem eşitliği, $x=(-B+\text{SQRT}(B^2-4*A*C))/(2*A)$ şeklinde yazılırken;
 - Yukarıdaki denklem eşitliği MathCAD’te :
 $+(-B1+\text{SQRT}(B1*B1-4*A1*C1))/(2*A1)$ şeklinde yazılır.

2.3.3 MathCAD’in Kullanım Alanları

MathCAD’in; matematik, fen bilimleri, bilgisayar ve işletim dünyası başta olmak üzere birçok kullanım alanı vardır. MathCAD altında, birçok farklı konuyu açıklayan ve çözümler sunan ek paketler mevcuttur. Örnek olarak : İnşaat mühendisliği, makine mühendisliği, elektrik/elektronik mühendisliği paketleri, finansal analiz, istatistik, fizik, akışkanlar mekaniği, dalgalar, hidrolik, mekanik titreşimler, termodinamik, elektromanyetizma, sinyal işleme, görüntü işleme, simülasyon, ileri matematik, sonlu elemanlar yöntemine giriş, yazılım güvenliği, veri analizi ve CRC kimya.

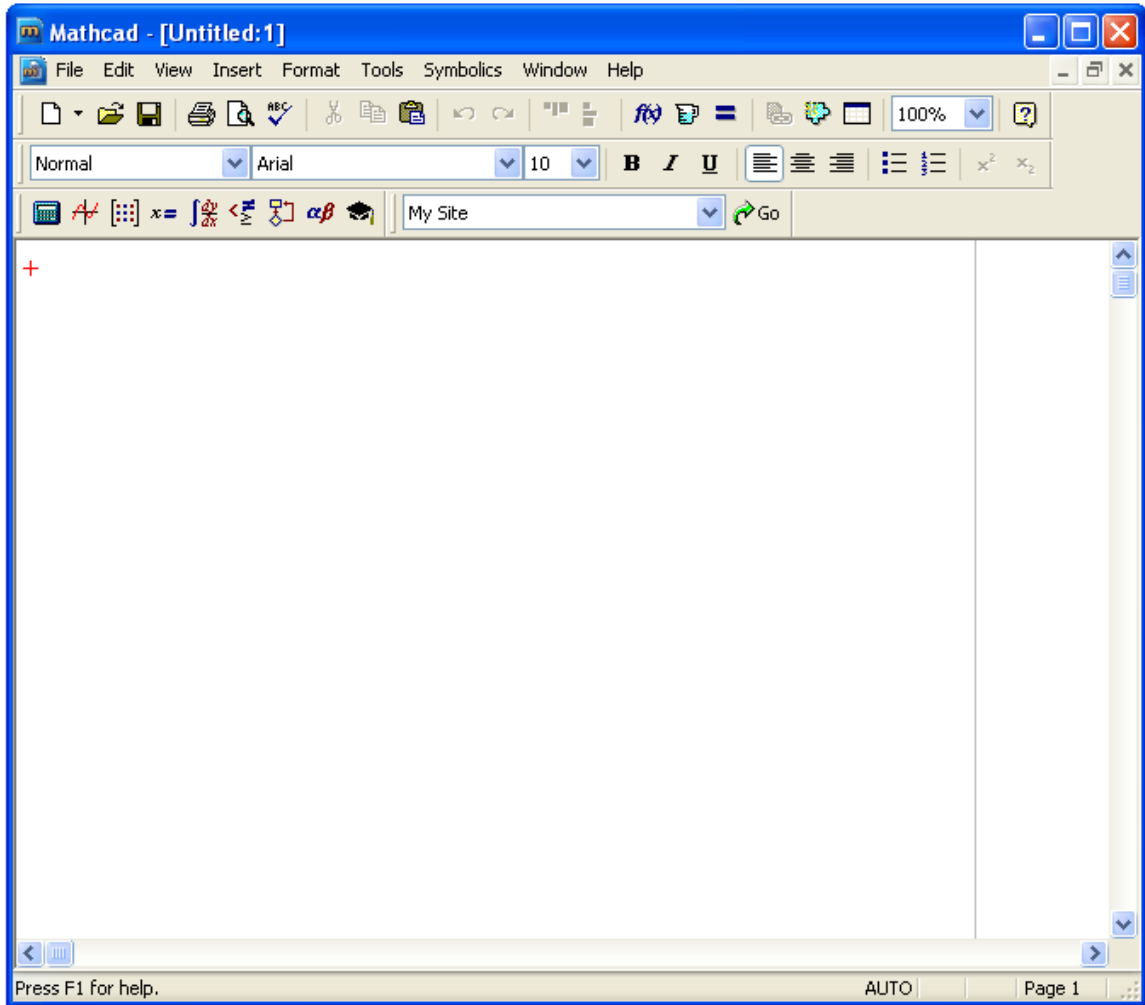
Fizik ve kimya laboratuvarlarındaki deneylerin ölçümlerinden elde edilen sayısal çözümleri sağlama ve bu deneylerin teknik raporlarını yazma beceresi MathCAD’in

en iyi özelliklerinden biridir. MathCAD; çeşitli deneylerin oluşturulması ve yapılması ile ilgili olarak kimya mühendisliği alanında; birtakım formülleri içeren problemlerin çözümü ile ilgili olarak da fizik mühendisliği alanında ve bunlara ilave olarak kalite kontrolünde, makine mühendisliği alanında, biyoloji biliminde, sosyal bilimlerde ve iş akış analizi konularında kullanılan bir programdır. Deneylerin sonuçlarını analiz etmek, öğrencilerin fizik ve kimya derslerindeki ödevlerine yardım etmek, problemlerine çözüm getirmek gibi uygulamaları ile MathCAD, fizik ve kimya alanında da kendini gösteren bir yazılım programıdır [36].

2.3.4 MathCAD Arayüz Çalışma Alanı

MathCAD çalışma alanı, diğer veri analiz programlarından (Excel gibi) çok farklıdır. MathCAD'in çalışma platformunda bulunan denklemler, fonksiyonlar, veri tabloları, grafikler ve şemalar bir uygulamanın geliştirilmesi için son derece yararlı elemanlardır. Sunduğu bu imkanlar, MathCAD'in en önemli artlarından ve pozitif yararlarından biridir. MathCAD'in kullanıcıların rahatlığı ve kolaylığı için sağladığı basit, sade ve anlaşılır bir çalışma alanı vardır. MathCAD'in çalışma alanının en iyi özellikleri : Hesaplama, veri elde etme, sembolik işlemler yapma ve teknik raporlar yazma gibi imkanlar sağlamasıdır. Ayrıca MathCAD'in, SI birimleri oluşturma ve birbirine dönüştürme ile 2-D (2 boyutlu) ve 3-D (3 boyutlu) grafikler meydana getirme gibi becerileri de vardır. MathCAD'in problem çözümleri için zengin bir çalışma platformu vardır. Kullanıcıya her türlü iş, sorun veya yardım için geniş bir seçenek imkanı sunar. MathCAD, bir iş veya uygulamada analiz ve görsel teknikleri kullanma olanağı ile son derece kullanışlı bir programdır. MathCAD'i diğer teknik hesaplama yazılımlarından ayıran bir diğer önemli özelliği : Bütün hesaplama işlemlerine birimleri entegre etmesidir. MathCAD, çok yönlü hesaplamalar yapma imkanı bakımından en popüler platforma sahip programlardan biridir. MathCAD ile; problemlerde, formüle edilmiş yapılarda, veri analizinde ve model oluşturma işlemlerinde en iyi çözümü bulmak ve sonuçlar arasındaki iletişimi sağlamak gibi birbirinden farklı yapıdaki olaylar gerçekleştirilebilir. Matematiksel işlemlerde ya da herhangi bir mühendislik hesaplamalarında kaliteyi arttırmayı gerçekleştirmek MathCAD ile mümkündür. MathCAD arayüz platformu; grafiklere, denklemlere ya da fonksiyonlara veri girme işlemlerini sağladığı gibi, bu yapılardaki verilerin, değişkenlerin ya da denklemlerin değiştirilmesi ve bu yapılara yeni veriler eklenmesi

gibi işlemleri de gerçekleştirilebilir. Aşağıdaki şekil MathCAD programının çalışma alanını göstermektedir [37]:



Şekil 29 : MathCAD Çalışma Alanı

2.3.5 MathCAD Araç Çubukları (Toolbars)

MathCAD programının içinde matematiksel formülleri ve sembolik ifadeleri girmeyi sağlayan, grafik çizimi ve programlama gibi işlemleri yapmak için geliştirilmiş olan araç çubukları (toolbars) vardır. MathCAD'in kullandığı bu araç çubukları ve yaptıkları işlevler şunlardır [38]:

2.3.5.1 Grafik Araç Çubuğu

MathCAD'te grafik araç çubuğu, 2D (2-boyutlu) ve 3D (3-boyutlu) grafikler yaratmak için kullanılır.

2.3.5.2 Hesaplama Araç Çubuğu

MathCAD’te hesaplama araç çubuğu, matematiksel, bilimsel ve mühendislik alanındaki hesaplamaları gerçekleştirmek için kullanılır.

2.3.5.3 Değerlendirme Araç Çubuğu

MathCAD’te değerlendirme araç çubuğu, eşitleme operatörünü “:=” , genel değerlendirme operatörünü “ ’ ” ve değerlendirme operatörünü “=” içinde bulunduran bölümdür.

2.3.5.4 Aritmetik Araç Çubuğu

MathCAD’te aritmetik araç çubuğu, aritmetiksel işlemler olan toplama, çıkarma, çarpma, bölme ile üstel fonksiyon işlemlerini ve karekök alma işlemlerini gerçekleştirmek için kullanılır.

2.3.5.5 Boolean Araç Çubuğu

MathCAD’te boolean araç çubuğu, mantıksal işlemleri ve boolean aritmetiğini gerçekleştirmek için kullanılır. Boolean araç çubuğu içinde eşitlik işareti “=” , eşit değil işareti “≠” , küçük eşit “≤” işareti ve büyük eşit “≥” işareti gibi mantık operatörleri bulunur.

2.3.5.6 Vektör (vector) ve Matris (matrix) Araç Çubuğu

MathCAD’te vektör (vector) ve matris (matrix) araç çubuğu, vektör ve matris işlemlerini oluşturmak için kullanılır.

2.3.5.7 Programlama Araç Çubuğu

MathCAD’te programlama araç çubuğu, yazılım geliştirmek ve program kodu oluşturmak için kullanılır.

2.3.5.8 Sembolik Araç Çubuğu

MathCAD’te sembolik araç çubuğu, sembolik hesaplamaları yapmak için kullanılır. Sembolik ifadeleri (türev, integral, polinomlar gibi) değerlendirme işlemleri, sembolik araç çubuğunda gerçekleştirilir.

2.3.6 MathCAD Programı’nın Temel Yapısı

MathCAD, yüzlerce operatör ve fonksiyon kütüphaneleri ile en basitten en karmaşığına kadar bütün teknik problemlere, profesyonel çözümler sunan bir yazılım programıdır. MathCAD’in kendine özgü özellikleri vardır. MathCAD denilince akla

gelen ve en çok üzerinde durulan konular ile MathCAD'in yapısını anlatan bu özellikler şunlardır [39] [40] [41]:

2.3.6.1 Matematiksel İşlemler

Matematiksel ifadelerin çözümlenmesi için, MathCAD yazılım programı kullanılabilir. MathCAD, sayısal işlemler için operatörler ve fonksiyonlar kullanır. Ayrıca bu program, nümerik sistemlerin çözümünü ve minimizasyonunu sağlar. Hataların takip ve tesbit için geliştirilmiş "Hata İzleme" penceresi vardır. Çeşitli matematiksel işlemler MathCAD'te kolayca yapılır. MathCAD'te gerçekleştirilen bu matematiksel işlemler şunlardır :

- Aritmetik işlemler (toplama, çıkarma, çarpma, bölme)
- Karmaşık sayıların hesaplanması
- Karekök alma
- Logaritma
- Mutlak değer
- Trigonometrik fonksiyonlar
- Üstel işlemler

2.3.6.2 Grafik Oluşturma

MathCAD, farklı tipte birçok grafik çizimi yeteneğine sahip bir bilgisayar programıdır. MathCAD'te grafik konusunda geliştirilen uygulamalar şunlardır :

- Veri şemaları
- 2-boyutlu (2D) ve 3-boyutlu (3D) grafiklerin oluşturulması
- Animasyon
- Katı cisimlerin yüzey çizimleri
- Geliştirilmiş histogram fonksiyonu

2.3.6.3 Programlama

MathCAD'te yazılım geliştirilirken; ifadelerin ve metinlerin özgürce, kullanıcının istediği şekilde, keyfe göre kullanılması mümkündür. Bu durum, kullanıcıya program kodu oluşturulurken birtakım kolaylıklar sağlamaktadır.

2.3.6.4 Mühendislik Uygulamaları Geliştirme

MathCAD, farklı mühendislik alanları ile ilgili her türlü uygulama ve işlemin yapılmasını sağlar.

2.3.6.5 Birimler

MathCAD, standart birimleri tanır ve birimler arası deęişiklikleri hesaplar. MathCAD programı, genel olarak kullanılan bilimsel birimlerin çoęunu kabul eder (Örneęin; enerji, zaman, hacim vs. birimleri gibi). MathCAD programında, terimlerin yapılarında otomatik olarak hazır halde *SI* birim sistemi kullanılır. MathCAD programı, *SI* ve dięer metrik birimleri destekler ve otomatik olarak birimleri birbirlerine çevirir. MathCAD, bilimsel ve matematiksel hesaplamalar ve işlemlerde birimleride sonuçlara dahil eder. MathCAD’te birimlerin kullanılması oldukça kolaydır. MathCAD programında, fizik ya da kimya alanındaki herhangi bir problemin çözümünde çeşitli formüllerin kullanılmasından sonra ortaya çıkan sonuçların yanına birimlerinin de yazılması oldukça kolaydır. Örneęin; MathCAD’te bir deęişken tanımlandığında ifadenin yanına “kilometre” birimi basitçe “km” olarak yazılır.

2.3.6.6 Denklem Çözümü

MathCAD’te, iki ve üç deęişkenli denklem sistemlerinin çözümü oldukça basittir. MathCAD, kullanıcılara denklem çözümü konusunda yardımcı olup, çeşitli olanaklar sunar. MathCAD’te; iki bilinmeyenli denklem çözümleri, sayısal deęerlerin bulunması, denklem sistemlerinde bilinmeyenlerin belirlenmesi ve sonuçlarının bulunması, fonksiyon çözümleri ve polinom denklemlerinin köklerinin bulunması işlemleri yapılır.

2.3.6.7 Fonksiyonlar Oluşturma ve Çözme

MathCAD’te fonksiyonel işlem ve uygulamaların oluşturulması ve çözülmesi oldukça basittir. MathCAD’in çözüme kavuşturduğu fonksiyonel yapılar şunlardır :

- Doğrusal ve doğrusal olmayan sistem çözümleri
- Optimizasyon işlemleri
- Fonksiyonların sayısal hesaplamaları
- Kök bulma algoritmaları
- Program fonksiyonları

2.3.6.8 Sembolik Hesaplamalar

MathCAD; fizik ve kimya alanındaki problemlerde, içinde sembollerin bulunduğu formüllerin hesaplanmasında son derece faydalıdır ve iyi sonuçlar verir. MathCAD’te sembolik hesaplamalar denince genel olarak sözü edilen konular şunlardır :

- Boolean Algebra (Boolean Cebri)
- Diferansiyel Denklemler
- İntegral
- Türev

2.3.6.9 Matrix (Matris) / Vector (Vektör) İşlemleri

MathCAD, matris ve vektör işlemlerinin gerçekleştirilmesi ve çözülmesi konusunda oldukça kullanışlı ve kullanıcı için faydalı bir programdır. MathCAD, yüzlerce değişkenli matrix (matris) sistemlerinin çözümünü de sağlar.

2.3.6.10 Veri Analizi

MathCAD, bütün işlemlerdeki verilerin analizini yapma yeteneğine sahip bir programdır.

2.3.6.11 Model Oluşturma / Dizayn

MathCAD, modelleme ve simulasyon için güçlü bir araçtır. Dinamik sistem simulasyonları yaratmak ve geliştirmek için faydalı bir programdır. VisSim LE, iletişim ve sinyal işleme sistemlerinin MathCAD içinde eksiksiz olarak modellenmesini, simule ve kontrol edilmesini sağlar. Bu güçlü simulasyon paketinde hızlı ve doğru sonuç veren bir zaman - tanım alanı simulasyon motoru da vardır.

2.3.6.12 Fonksiyonel Özellikler

MathCAD programının sahip olduğu, kendine has fonksiyonel özellikleri şunlardır :

- Resim yönetim özellikleri ve dosya formatları ile bir resim görüntüleyici
- Veri kurgusunu bir grafik tipinden diğer bir grafik tipine çevirmenizi sağlayan yeni haritalama fonksiyonları
- Vektörler ve matrisler içindeki değerleri bulmak için yeni arama fonksiyonları
- Sonuçları mühendislik notasyonunda görüntüleyen yeni bir otomatik seçenek
- Daha iyi bir görüntü sağlayan ve MathCAD'in kullanılmasını kolaylaştıran Windows 2000 desteği ve güncellenmiş ara birim
- Hesaplamalardaki hataları takip etmek ve belirlemek için MathCAD'in hata izleme özelliği vardır.
- Sayısal hesapları semboller ile entegre etme özelliğine sahiptir. MathCAD dünyanın en gelişmiş sembolik cebir motoruna sahiptir. Maple Sembolik hesaplamaları, mutlak doğruluk sağlar.

- IntelliMath özelliđi denklemlerin ve ifadelerin girilmesini, düzeltilmesini ve problem çözümünü kolaylaştırarak, kullanıcıların yapması gereken işleri ve oluşturması gereken fonksiyonları otomatize eder, kullanıcıya zaman kazandırır.

2.3.6.13 Kullanışlılık Özellikleri

MathCAD kullanışlılık özellikleri ile kullanıcıların yardımcısı olur. MathCAD'in sahip olduđu bu özellikler şunlardır :

- MathCAD, işlemlerin otomatik olarak tekrardan hesaplanmasını sağlar.
- Algoritmaların oluşturulması konusunda, kullanıcı için faydalı bir programdır.
- Her konuda kullanıcıya uyarı, bilgi ve hata mesajları ile yardım eder.
- Program oluşturulurken hataları bulma ve kullanıcıya gösterme beceresi ile son derece yardımcı bir programdır.
- Her türlü matematiksel işlemin kolayca yapılması konusunda faydalı bir programdır.
- Kolayca iş yapabilme imkanı sunan bir arayüzü vardır.
- MathCAD, kullanıcılara teknik çizimleri için gerekli hesaplamaları gerçekleştirip "SmartSketch" programı altında çizdirme imkanı sunar SmartSketch ile kullanıcıların çizimleri MathCAD veri blokları ile değiştirilebilir, yenilenebilir ve çizim özellikleri MathCAD hesaplamaları ile kontrol edilebilir.
- MathCAD, Microsoft Windows standartlarına uyumlu menü ve araç çubukları ile alışagelmış bir pencerede çalışma imkanı sunar.
- MathCAD, AXUM ile daha kapsamlı 2 ve 3 boyutlu grafikler yaratma imkanı sağlar.

2.3.7 MathCAD'te Programlama Geliştirmek

MathCAD; çeşitli eşitlikleri ya da eşitsizlikleri deneyerek, bunların nasıl görüldüğünü kontrol etmek ve matematiksel modeller kullanmak için uygun ve özel bir programdır. MathCAD programı, algoritma geliştirmek ve yazılım oluşturmak için kullanılan yararlı programlardan biridir. Uzmanlaşmış birçok ticari simülasyon programı ve daha özel maksatlı FORTRAN programının yapmış olduđu birçok işin ve analiz işlemlerinin, MathCAD tarafından da yapılıyor olması, MathCAD'in kullanıcılar için doğru ve uygun bir program olduğunun kanıtıdır.

MathCAD ile çalışmanın güzelliklerinden biri şudur : MathCAD’te program kodu oluşturmak, FORTRAN ya da diğer programlama dillerinden çok daha farklı ve basittir. Böylece o programlama dillerindeki hataların aynısı yapılmamış olur ve program kodundaki hatalar okunarak bulunabilir. MathCAD’te mantıksal hataları bulmak ve hataları düzeltmek oldukça kolaydır. MathCAD’te programlama geliştirilirken bağlı kalınan ve uyulması gereken bazı kurallar vardır. Bu kurallar şunlardır [42] [43]:

- MathCAD’te, programın başında sabit değerler ve değişkenler belirtilir. Daha sonra, yapılan işleme göre fonksiyonlar ya da formüller tanımlanır.
- “y:x^2# -7” eşitliğinde “#” işareti boşluk satırı koyma anlamı taşır.
- Bir sayıyı tanımlamak için “:=” işareti kullanılır. Örneğin; R:=8310 gibi
- Bir sayıya değişken atamak “=” için işareti kullanılır. Örneğin; x= 12 gibi

Programlama geliştirilirken, MathCAD programının sahip olduğu kendine has birtakım özellikleri vardır. Bu özellikler şunlardır :

- MathCAD arayüzü, farklı bilgisayar programlarındaki uygulama, işlem ve fonksiyonlara giriş olanağı sağlar. MathCAD için, programlama dilleri C, C++ ve FORTRAN’da kullanıcı amaçlı fonksiyonlar yaratılabilir.
- MathCAD, case-sensitive (büyük-küçük harfin farklı olmasına duyarlı olma) özelliğine sahip bir programdır. Bundan dolayı MathCAD’te, x ve X farklı değer ve değişkenlerdir.
- MathCAD’te otomatik olarak tekrardan değişken atama özelliği vardır.
- Fonksiyonel programlama için çeşitli operatörlere sahiptir.
- Kullanıcılara sembolik denklemlerle programlama olanağı sunar.
- MathCAD’te programlama, kolayı zora çevirme özelliği ile bilinir.
- MathCAD, birçok programlama dili ile ilişki içerisindedir.
- MathCAD, program kodu geliştirilirken hataları bulma ya da en aza indirme özelliği ile ön plana çıkar.
- MathCAD, program oluşturulurken kullanıcıya kendi değişkenlerini ve kendi fonksiyonlarını yaratma ve tanımlama imkanı sağlar.
- MathCAD’te programlama operatörleri kullanılarak çok satırlı yapılar oluşturulabilir.

- MathCAD’te program kodu geliştirilirken oluşturulan komutlar, uygulamanın basitleştirilmesini, genişletilmesini, bütünleştirilmesini ve diğer uygulamalardan farklı olmayı sağlar.

2.3.8 MathCAD Programı’nın Tercih Edilme Sebepleri

Dünya da geniş bir kullanıcı kitlesi olan MathCAD yazılım programının tercih edilmesinin sebepleri şunlardır [44]:

- MathCAD yazılım programı, matematiğin kullanıldığı tüm alanları kapsayan çözümleri hem çok daha pratik bir şekilde, hem de üniversitelere çok ekonomik koşullarla sunmaktadır.
- MathCAD, MS Windows altındaki bir çok yazılım ile veri alışverişinde bulunabilme imkanına sahiptir. Bu sayede bir grup çalışmasında farklı işler, farklı yazılımlar ile çözülür ve bir tek ortamda : MathCAD ortamında birleşir. Örnek olarak Excel’de hazırlanmış bir veri tablosunu MathCAD okur, hesaplama yapar ve hesap sonucu bazı verileri AutoCAD’e göndererek AutoCAD çizimini yeniler.
- Öğrenciler açısından öğrenilmesi ve kullanımı en kolay matematiksel hesaplama yazılımıdır.
- MathCAD ile yaratılan çalışmalar aynı zamanda rapor veya kitapçık olarak basılabilmektedir. MathCAD içinde her türlü metnin yazımı ve düzenlenmesi için kolay seçenekler mevcuttur. Hatta Microsoft Word yazılımından herhangi bir metin yazısı seçilip, kopyalanarak kolayca MathCAD içine yapıştırılabilir. Bu sayede tek bir ortamda açıklayıcı metin yazılarını, hesaplamaları, grafikleri ve varsa simülasyonları gerçekleştirmiş ve raporu da anında alınmış olur.

2.3.9 MathCAD Kullanıcısı Olmanın Avantajları

MathCAD’i yalnız çeşitli özellikleri ile değil sağladığı faydalar ile de bilmek gerekir. MathCAD programının kullanıcılara çeşitli yararları ve avantajları vardır. Bu avantajlar şunlardır [44]:

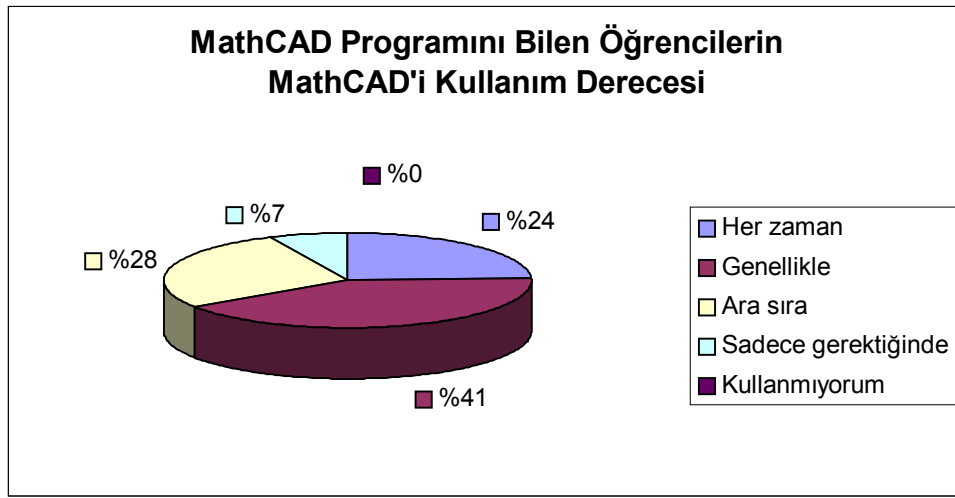
- Windows uyumlu arayüz (interface), daha fazla matematiksel güç, uygulamaları birleştirici yeni teknolojiler, yeni web altında çalışma olanakları ile MathCAD, hesaplama programları arasında yeni bir standart oluşturmaktadır. MathCAD; grup çalışmalarında, etkileşimli projelerde, mühendislik, tasarım ve ArGe

çalışmalarındaki toplam verimliliği ve doğruluğu ile kalitesini açıkça ortaya koymaktadır.

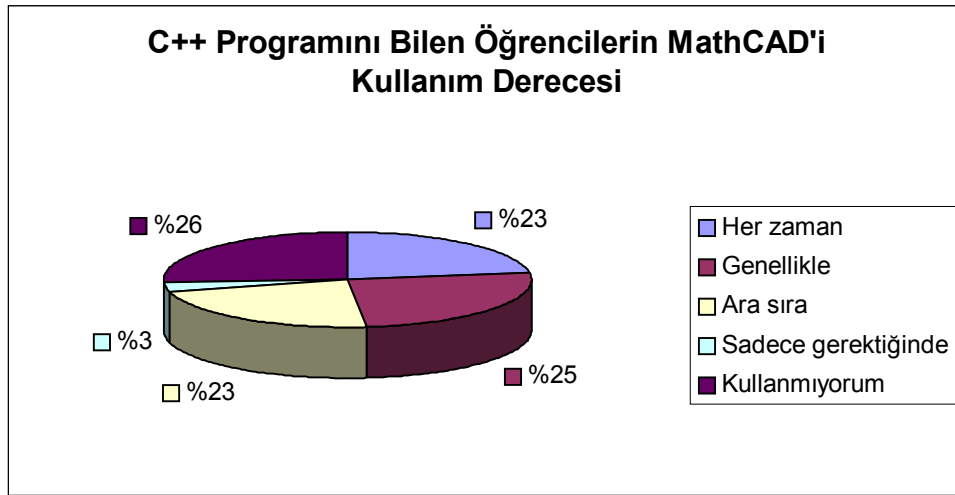
- MathCAD, dünya çapında teknik profesyonellerin birinci tercihidir. Çünkü MathCAD'in problem çözüm ortamı, benzersizdir. MathCAD “çalışma ekranı” ile denklem, grafik ve animasyonları istenilen yerde, gerçek matematiksel notasyonlarla çözüme imkanı sunmaktadır. Ayrıca, MathCAD “canlı arayüz (interface)” ile farklı girdilerin cevapları anında bulunabilir. Kısaca, MathCAD geniş bir yelpazede teknik problemleri inceleyecek ve çözecek eşsiz bir ortam sunar.
- MathCAD ile kullanıcılar, daha fazla problem çözümüne sahiptirler. Türevden, simültane denklem çözümüne; hipotez testinden, curve fitting'e kadar her türlü problem, MathCAD programında incelenebilir. Ayrıca MathCAD'in tüm özelliklerini kullanarak, kullanıcılar kendi programlarını da yaratabilirler.
- MathCAD, bilgiye ulaşmak için kullanıcılara daha fazla seçenek sunar. Aranılan bilgiye anında ulaşmak için MathCAD'in içinde “Resource Center” bölümü vardır. MathCAD, buradaki yüzlerce örnek ve fonksiyonu taşı ve bırak yöntemiyle kullanıcıların kendi dökümanlarına aktarmasını sağlar.
- Çalışma gurubunda standardizasyonu MathCAD ile yakalamak mümkündür. Bilgiyi kavramak MathCAD ile daha kolaydır. Çünkü MathCAD hem hesaplamaları herkesin kavrayabileceği tarzda çözer, hem de çalışmalarını draft kalitesinde rapor ve döküman olarak saklama ve basma olanağı sağlar. Projeler ve hesaplamalar her zaman arşivleme, yeniden inceleme ve tekrar hesaplama için hazırdır.
- MathCAD, herhangi bir organizasyonun oluşumunda ya da bir projenin yönetilmesinde çeşitli faydalar sağlar.
- MathCAD, herhangi bir hesaplama, işlem ya da projeden elde edilen verinin tekrar ve kolayca kullanılmasını, yönetilmesini ve diğer uygulamalara aktarılmasını sağlar.
- MathCAD, hesaplamalardaki detaylar gizlendiğinden dolayı, hesaplamaların yapısına ve veri akışına çok önem veren bir tekniksel hesaplama yazılım paketidir.
- MathCAD, araştırmaların yapılıp, bu araştırmaların sunumu ile yayınlanmasını sağlar.

2.3.10 MathCAD Anket Araştırma Sonuçları

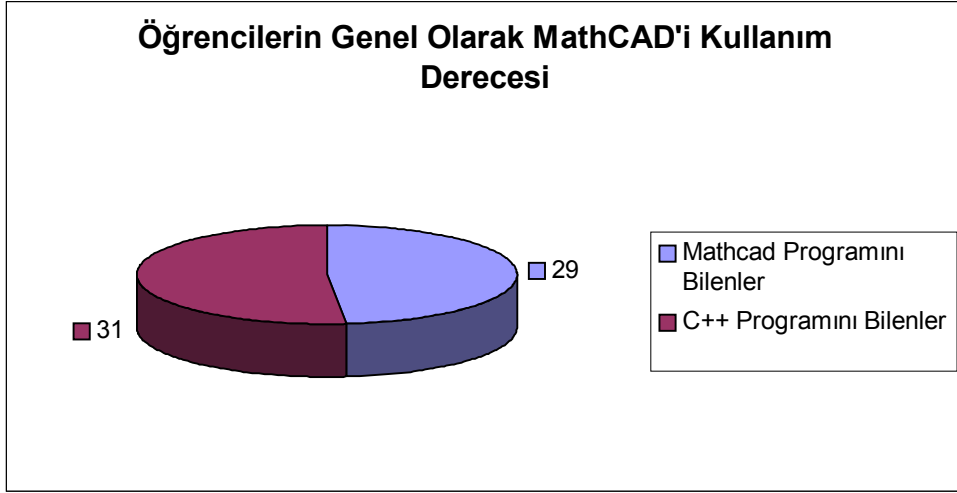
Citadel Üniversitesi, İnşaat Mühendisliği ve Çevre Mühendisliği Bölümlerinde MathCAD ile ilgili olarak bir anket düzenlemiştir. Üniversitenin yapmış olduğu anket araştırmasında; 43 öğrencinin 29 tanesi MathCAD tabanlı bir dersi 1996 yılında almıştır. 37 öğrencinin 31 tanesi de, C++ tabanlı bir dersi 1995 yılında almıştır. Citadel Üniversitesinin 1998 yılında, İnşaat Mühendisliği ve Çevre Mühendisliği Bölümlerinde okuyan öğrenciler üzerinde MathCAD ile ilgili olarak yapmış olduğu anket araştırma sonuçlarına göre, öğrencilerin MathCAD'ten faydalanma hakkındaki görüşlerinde şu veriler elde edilmiştir [45]:



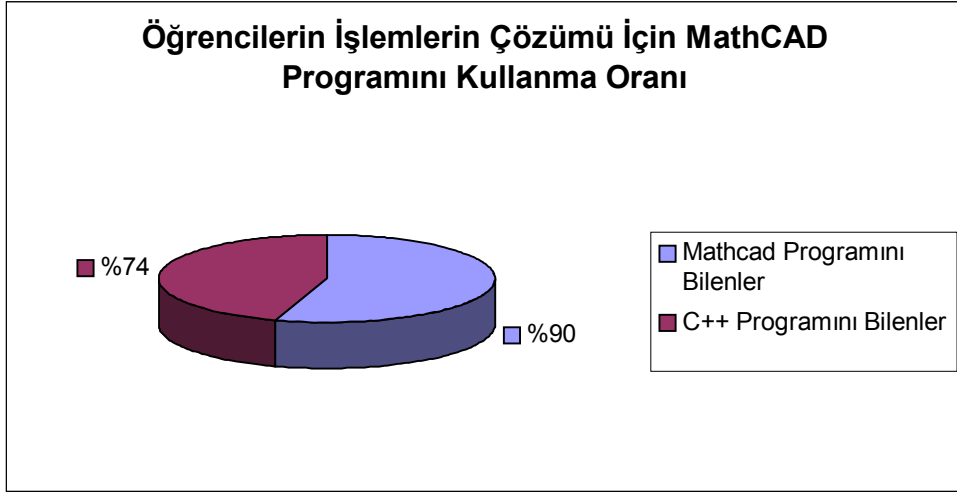
Şekil 30 : MathCAD Anket Araştırma Sonucu-1



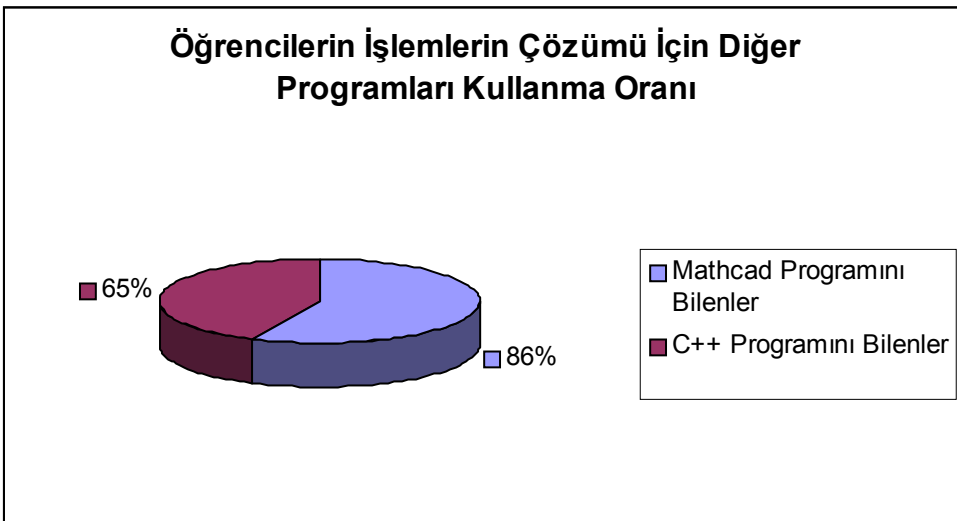
Şekil 31 : MathCAD Anket Araştırma Sonucu-2



Şekil 32 : MathCAD Anket Araştırma Sonucu-3



Şekil 33 : MathCAD Anket Araştırma Sonucu-4



Şekil 34 : MathCAD Anket Araştırma Sonucu-5

MathCAD programını tanımak, sağladığı faydaları bilmek ve MathCAD tabanlı bir programlama dersi ile geleneksel programlama dillerinden biri olan C++ tabanlı bir programlama dersinin kıyaslanması, karşılaştırılması ve aralarındaki farkların belirtilmek istenmesi amacı ile yapılan yukarıdaki anket sonuçlarına göre aşağıdaki veriler elde edilmiştir :

1. MathCAD tabanlı bir dersi alan öğrencilerin %90'ı, bir yıl sonra hala kendi istekleri MathCAD'i problemlerin çözümünde kullanmaktadır.
2. Öğrencilerin MathCAD programını öğrenme isteği, herhangi bir geleneksel programlama dilini (C++, FORTRAN gibi) öğrenme isteğinden daha yüksektir.
3. MathCAD programını öğrenen öğrenciler, programlama konusunda ve algoritma geliştirme işlemlerinde C++ programını öğrenen öğrencilerden daha başarılıdır.
4. C++ dersini almış öğrenciler arasında, işlere bilgisayar tabanlı çözümler sağlamak için MathCAD programı, C++ programından daha sık kullanılmaktadır. Kısacası MathCAD, C++ programından daha fazla ilgi görmektedir.
5. Öğrencilerin %13'ü (2 tanesi MathCAD programını bilen öğrencilerden, 6 tanesi C++ programını bilenlerden oluşmaktadır) işlemlerin çözümü için, C++ ya da diğer geleneksel programlama dillerini kullanmaktadır.
6. C++ programını bilen öğrenciler, MathCAD programını bilen öğrencilerden daha fazla MathCAD programını kullanmaktadır.
7. C++ programı yerine, MathCAD programının mutlaka öğrenilmesi öğrenciler tarafından savunulmaktadır.

2.4. Grafiksel Kullanıcı Arayüzü (GUI)

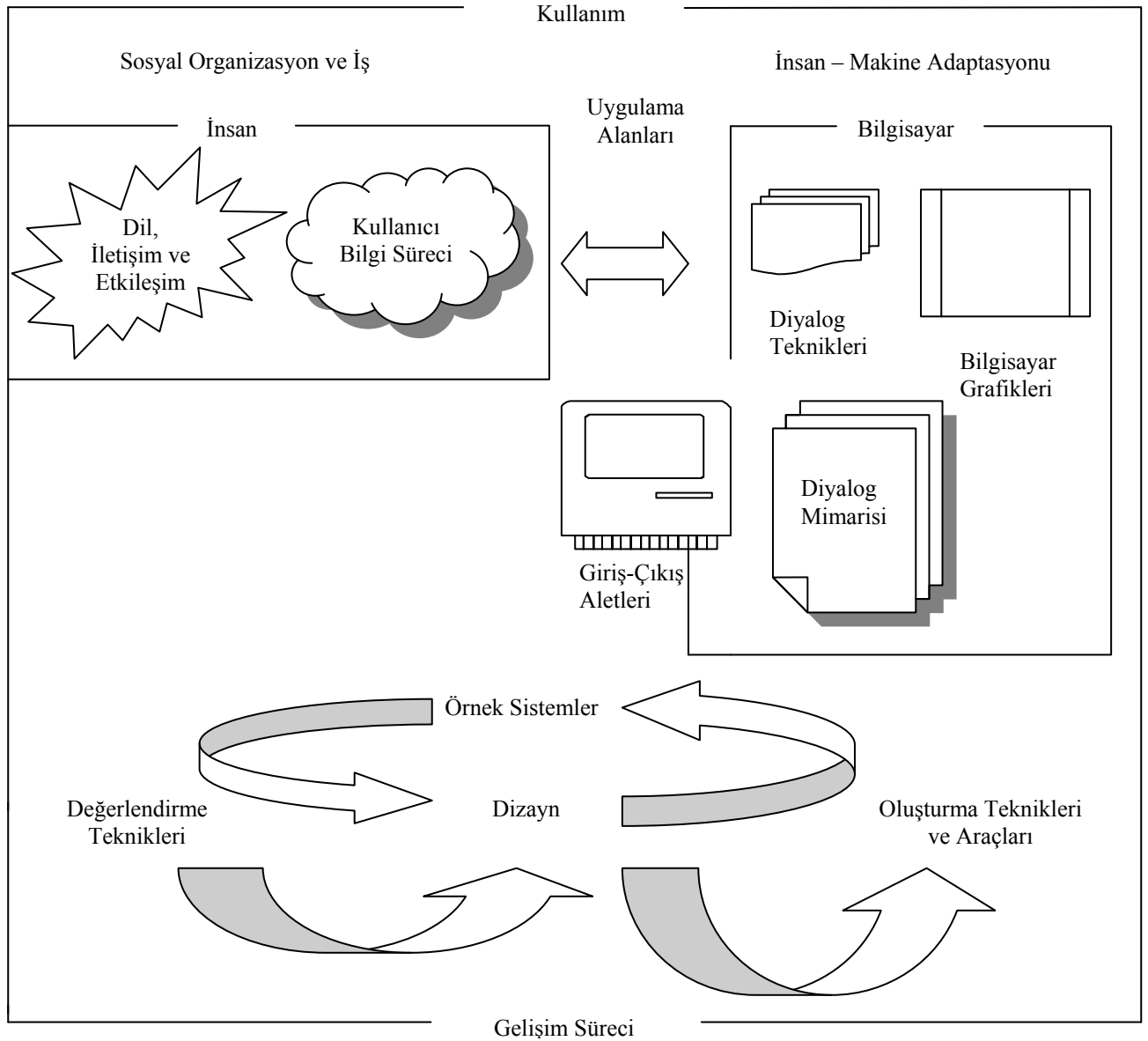
Grafiksel kullanıcı arayüzü (GUI), bilgisayar kullanıcısı ile bilgisayardaki işlemler ve uygulamalar arasında kolay ve faydalı bir etkileşim sağlayan grafiksel tabanlı bir arabirimdir. Genel olarak arayüzü, herhangi birşeyin kullanımını ya da onda etkileşimi sağlayan kısım olarak düşünebiliriz. Kullanıcı arayüzü, bilgisayarda bir yazılımı kullanabilmek için oluşturulan menüler, butonlar ve grafikler ortamıdır. Bir kullanıcı arayüzü, programları, dosyaları grafiksel imgelerle gösterir. Bu imgeler, ikonlar (icons) ve menüler içerir. Kullanıcıda, fare (mouse) ve keyboard (klavye) ile arayüzdeki bu seçenekleri seçer ve çalıştırır. Böylece kullanıcı ile arayüz arasındaki etkileşim sağlanmış olur.

Grafiksel kullanıcı arayüzü (GUI), Microsoft Windows ve MacOS'un örnek gösterilebileceği, metin yerine grafik tabanlı bir bilgisayar arayüzüdür. Bu iki işletim sisteminin arayüz elemanları, bir fare (mouse) arabirimi ile dosya klasörleri olarak görünen görsel dizinler sistemidir. GUI ilk olarak XEROX firması tarafından geliştirildi. Grafiksel kullanıcı arayüzünün geçmişi ve zaman çizgisi 1973 yılında Xerox firmasının araştırma merkezi California'da bilgisayar gelişimi ile ilgili yaptığı araştırmalara dayanır. Bu araştırmalarda kullanıcı arayüzünün varlığına ulaşıldı. Daha sonra Apple firması Macintosh ile, Microsoft ise Windows ile grafiksel arayüz tasarımını benimsedi. GUI, komutlardan çok ikonları (icons) kullanan bir programdır. Windows ve Macintosh işletim sistemleri, grafiksel kullanıcı arayüzü'nü (GUI) kullanırken, Unix ve DOS ise komut tabanlı bir arayüzü kullanır [46].

GUI, grafiklerin fare (mouse) ve klavye (keyboard) ile birlikte kullanıldığı ve bunların aracılığı ile bazı programlara kolay bir arayüz kullanımı sağlamak için, kullanılan tekniktir. GUI, bir işletim sistemi ya da uygulama arayüzünde olduğu gibi; grafiksel gösterimler, resimler, ikonlar (icons), menüler ve pencereler fare (mouse) ve keyboard (klavye) ile kullanılarak kullanıcı ile bilgisayar arasındaki etkileşimi sağlar. Macintosh bilgisayarların çıkışı ve ardından Microsoft'un Windows yazılımının geliştirilmesinden beri birçok insan, arayüz dendiğinde otomatik olarak grafiksel kullanıcı arayüzünü (GUI) düşünmektedirler. Örneğin; kola makinası, satın alma işleminin yapılmasını sağlayan farklı renk ve büyüklüklere sahip düğmelerden oluşan basit bir arayüze sahiptir. Bir araba, gaz ve fren pedalları ile aracı yönlendirmeyi sağlayan direksiyonu içeren daha karmaşık bir arayüze sahiptir. Bir bilgisayarın arayüzü, klavye ve fare ile monitörde gözükken ve bilgisayarı birçok farklı iş için kullanabilmemizi sağlayan yazılımlardan oluşur.

GUI, insan bilgisayar etkileşimi (Human Computer Interaction - HCI) tarafından kullanılan bir methodtur. İnsan bilgisayar etkileşimi (Human Computer Interaction – HCI), insan ile bilgisayar arasında bilgi akışını ve iletişimini sağlamak için kullanılan bir yöntemdir. İnsan bilgisayar etkileşimi'nin (HCI) temel prensibi : “*Basit dizayn iyi dizayn'dır*” düşüncesidir [47].

Aşağıdaki şekil, insan ile bilgisayar arasındaki etkileşimi ve bu etkileşimin gelişim sürecini göstermektedir [48]:



Şekil 35 : İnsan - Bilgisayar Etkileşimi

2.4.1 Grafikselsel Kullanıcı Arayüzü (GUI) Dizaynının Amacı

Grafikselsel kullanıcı arayüzü (GUI) dizaynının amacı, bilgisayar arayüzünü kullanıcılarına sevimli, net ve anlaşılır bir biçimde sunmaktır. Bunun için kullanıcının arayüzden ne istediğini çok iyi bilmek ve kullanıcının istediği işlere ve fonksiyonlara göre bir arayüz oluşturmak gerekir. Kullanıcı arayüz dizaynının ana fikri, kullanıcı ile sistem arasındaki entegrasyonu sağlamaktır. Kullanıcı ile sistem arasındaki entegrasyonu sağlamak için, sistem analiz yöntemlerini, arayüz dizaynını, yazılım mühendisliğini ve sistem değerlendirme metodlarını birleştirmek gerekir. Grafikselsel

kullanıcı arayüzü, kullanıcı ile bilgisayar arasındaki kontak noktasıdır. Genel olarak, grafiksel kullanıcı arayüzünün iki temel amacı vardır :

1. İletişim
2. Etkileşim

Grafiksel kullanıcı arayüzünün iletişim ile sağlamak istediği, kullanıcının bilgisayardan bilgi almasına yardım etmektir. Grafiksel kullanıcı arayüzünün etkileşim ile sağlamak istediği ise, kullanıcı ile bilgisayar arasında bir bağ kurmaktır [49].

Kullanıcı arayüz dizaynının etkili ve kaliteli olması gerekir. Çünkü böyle bir arayüz, hem kullanışlılığı hem de kullanıcı memnuniyetini sağlamış olur. Grafiksel kullanıcı arayüzü'nün (GUI) dizaynı iki açıdan önemlidir [50]:

1. Kullanıcı açısından
2. Arayüzü geliştiren kişi ya da kuruluşlar açısından

2.4.1.1 Kullanıcı açısından önemi

Kullanıcı arayüzünün, kullanıcı memnuniyetini sağlamak için dizayn edildiğini düşünürsek, arayüz dizaynının kullanıcı açısından önemli olduğu çeşitli noktalar vardır. İyi dizayn edilen bir arayüzde, kullanıcıyı ilgilendiren bu noktalar şunlardır :

- Kullanıcıya kolaylık ve rahatlık sağlanmış olur.
- Uygulamanın, kullanıcı için kullanışlılığı arttırılmış olur.
- Kullanıcıya az zamanda çok iş yapma imkanı sağlanmış olur.
- Kullanıcının arayüz ile ilgili sorunları azaltılmış olur.
- Uygulamanın kalitesi arttırılmış olur.

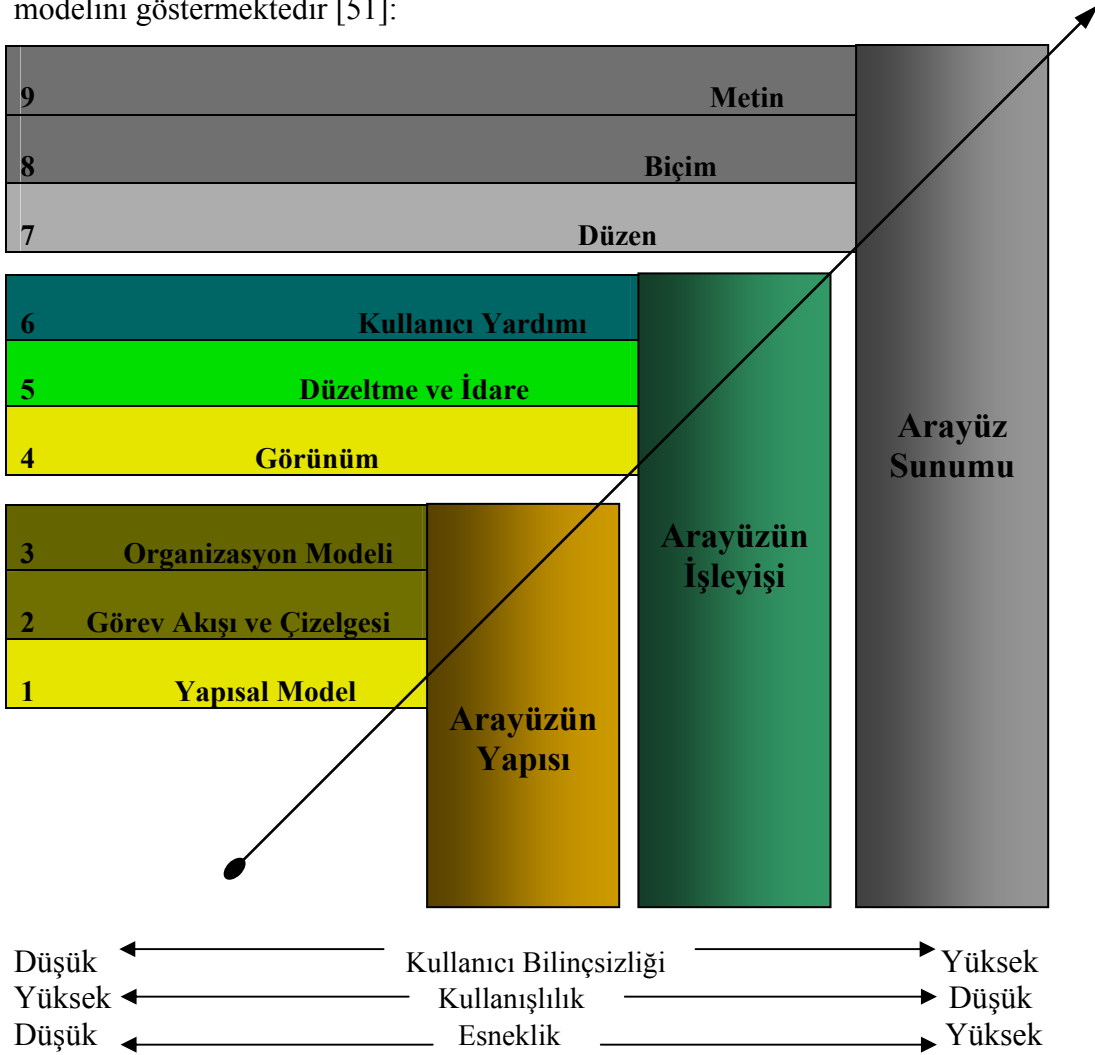
2.4.1.2 Arayüzü geliştiren kişi ya da kuruluşlar açısından önemi

Kullanıcı arayüzünün kişi ya da kuruluşlar tarafından dizayn edildiğini düşünürsek, arayüz dizaynının kişi ya da kuruluşlar açısından önemli olduğu çeşitli noktalar vardır. İyi dizayn edilen bir arayüzde, kişi ya da kuruluşları ilgilendiren bu noktalar şunlardır :

- Kuruluşun müşteri memnuniyeti sağlanmış olur.
- Kurumun itibar ve saygınlığı arttırılmış olur.
- Arayüzü geliştiren kişinin üretkenliği arttırılmış olur.
- Başarılı ve faydalı bir arayüz üretilmiş olur.

2.4.2 Grafiksel Kullanıcı Arayüz (GUI) Modeli

Grafiksel kullanıcı arayüzünün dizaynı yapılmadan önce arayüzün içeriğini ve yapısını anlatan bir arayüz mimari modeli oluşturulmalıdır. Bu modelde öğeler özelliklerine göre bölümlere ayrılırlar. Arayüz dizaynı sırasında hangi öğe daha önemli ise, arayüz modelinde onun önceliği fazladır. Arayüz modeli, kullanıcı arayüzünün temel öğelerini ve bu öğelerin öncelik sırasını gösterir. Modelde, bütün arayüz öğeleri birbirinden bağımsız bölümlerde bulunurlar. Aşağıdaki şekil ve şekile ait açıklamalar, arayüzün yapısını ve mimarisini gösteren grafiksel kullanıcı arayüz modelini göstermektedir [51]:



Şekil 36 : Kullanıcı Arayüz Modeli

Grafiksel kullanıcı arayüz (GUI) modeli, 3 kat ve 9 katmandan oluşur. Bu kat ve katmanların özellikleri şunlardır :

2.4.2.1 Birinci Kat : Arayüzün Yapısı

Kullanıcı arayüzünün en alt kademesi olan *arayüzün yapısı* katı üç kısımdan oluşur :

- Yapısal Model
- Görev Akışı ve Çizelgesi
- Organizasyon Modeli

Arayüzün yapısı katı, içinde bulundurduğu üç bölüm (yapısal model, görev akışı ve çizelgesi ile organizasyon modeli) ile arayüzün temelini oluşturur. Arayüzün yapısını oluşturan bu üç kısım oluşturulmaları büyük bir tecrübe gerektirdiğinden ve arayüz dizaynının temelini oluşturmaları açısından dizaynın en önemli bölümleridirler.

2.4.2.1.1 Birinci Katman : Yapısal Model

Yapısal model, arayüzün en temel görünümünü oluşturur. Arayüzün birinci katmanını oluşturan yapısal model, kullanıcı ile arayüz arasındaki ilişkiyi açıklar. Yapısal model'in amacı, kullanıcının geçmiş tecrübelerinden yaralanmak ve kullanıcının arayüzün temel işlemlerini ve fonksiyonlarını anlamasını sağlamaktır.

2.4.2.1.2 İkinci Katman : Görev Akışı ve Çizelgesi

Görev akışı ve çizelgesi katmanı, sistemin görevlerini belirlemek ve bu görevlerin belirli bir sıra ve düzen içerisinde olmasını sağlamak için kullanılır. Görev akışı ve çizelgesi, direkt olarak ürünün çalışma alanı platformuna bağlıdır.

2.4.2.1.3 Üçüncü Katman : Organizasyon Modeli

Organizasyon modeli, sistemin içeriğini ve fonksiyonlarını bir sıra ve şema halinde gösterir. Sistem fonksiyonlarını düzenler ve sınıflandırır. Organizasyon modeli, sistemin bilgi mimarisi olarakta bilinir. Organizasyon modeli, sistemin hem sınıflandırma şemasını hemde ilişkisel modelini içinde bulundurur. Ayrıca, sistemin planında organizasyon modeli ile yapılır.

2.4.2.2 İkinci Kat : Arayüzün İşleyişi

Arayüzün işleyişi, grafiksel kullanıcı arayüz modelinin orta katıdır. Kullanıcı arayüzünün ortadaki kademesi olan *arayüzün işleyişi* katı üç kısımdan oluşur :

- Görünüm
- Düzeltme ve İdare
- Kullanıcı Yardımı

Arayüzün işleyişi katı, arayüzün yapısal mimarisinin ve görsel şeklinin aksine kullanıcı arayüzünün kalitesini arttırmayı amaçlar. Kullanıcı arayüz dizaynı ile ilgili problemlerin önceden bulunup çözüme kavuşturulmasına çalışılan bölüm, arayüzün işleyişi katıdır.

2.4.2.2.1 Dördüncü Katman : Görünüm

Görünüm katmanı, arayüzün işleyişi katının büyük bir bölümünü kaplar. Arayüzün işlemlerini ve görüntüsünü belirler. Görünüm katmanı, arayüzün kullanıcı üzerindeki etkisini arttırmak için kullanılır. Arayüzün kullanıcıya sunulması için çok önemli bir bölümdür.

2.4.2.2.2 Beşinci Katman : Düzeltme ve İdare

Kullanıcı arayüzünde sürekli değişiklikler olacağından düzeltme ve idare katmanı, arayüz dizaynında önemli bir yere sahiptir. Düzeltme ve idare katmanında, arayüze metin eklenmesi ve arayüzden metin silinmesi ile arayüzdeki metin üzerinde değişiklik ve düzeltmeler yapılması gibi işlemler yerine getirilir. Ayrıca, düzeltme ve idare katmanı, arayüzde kullanılacak olan arayüz elemanlarının idaresinden sorumludur.

2.4.2.2.3 Altıncı Katman : Kullanıcı Yardımı

Arayüz modeli içerisindeki kullanıcı yardım katmanı ile uygulamanın durumu ve işlevleri konusunda kullanıcı bilgilendirilir. Arayüz içerisinde kullanıcıya yardım, hata ve uyarı mesajları ile yapılır. Kullanıcıya sunulacak olan yardım, kullanıcının eğitimine göre belirlenir.

2.4.2.3 Üçüncü Kat : Arayüz Sunumu

Arayüz sunumu katı, arayüzün görselliğini ve metinsel bütünlüğünü birarada buldurmaya çalışır. Kullanıcı arayüzünün en üst kademesi olan **arayüz sunumu** katı üç kısımdan oluşur :

- Düzen
- Biçim
- Metin

İyi bir kullanıcı arayüzü oluşturmak için arayüz bir bütün olarak değerlendirilir. Bu yüzden birbirinden bağımsız işlemler üzerinde odaklanılır. Arayüz sunumu katında detaylara çok önem verilir.

2.4.2.3.1 Yedinci Katman : Düzen

Düzen katmanında, çeşitli dizayn kararları alınarak arayüz elemanlarının arayüz üzerinde yerleşim ve düzenine karar verilir. Arayüzün görsel akışına dikkat edilmesi ve arayüz elemanlarının düzenli olmasının sağlanması kullanıcı arayüzünün biçim ve görünümünü olumlu yönde etkiler.

2.4.2.3.2 Sekizinci Katman : Biçim

Bir kullanıcı arayüz dizaynı sırasında, arayüzün biçimini olumsuz etkileyen en önemli faktörler, anlatım biçiminin bozukluğu ve yanlış anlamda kullanılan sözcüklerdir. Biçimdeki bozukluk, arayüzün kullanılabilirliğini ve kullanıcı memnuniyetini olumsuz etkiler. Kullanıcı arayüzünün biçimindeki uygunluk, arayüzün görselliği için önemli bir etkidir.

2.4.2.3.3 Dokuzuncu Katman : Metin

Kullanıcı arayüz içerisindeki metin katmanını etkileyen en önemli faktör kullanılan dildir. Kullanılan dil, akıcı ve anlaşılır olmalıdır. Kullanılan metin, kullanıcı arayüzünün görünümünü ve biçimini doğrudan etkiler.

2.4.2.4 Grafikselle Kullanıcı Arayüz (GUI) Modeli'nin Önemli Noktaları

Grafikselle kullanıcı arayüz (GUI) modeli; esnekliğe, kullanıcı bilinçsizliğine ve kullanılabilirliğe olan zarara göre bölümlere ayrılmıştır. Arayüz modelindeki her bölümün bu üç önemli noktaya verdiği zararlar farklı şekildedir. Arayüz dizaynının çeşitli katmanlardan oluşması, kullanıcı arayüz dizaynının etkinliğini artırır.

2.4.2.4.1 Esneklik

Kullanıcı arayüz modelinin yapı bölümü, uygulamanın program kodu mimarisinde önemli bir yere sahiptir. Uygulamanın mimarisinde ve arayüzün yapısal elemanlarında olabilecek değişikliklere karşı kullanıcı arayüzü esnek bir yapıda olmalıdır. Buda arayüz uygulamasındaki program kodunu yazmadan önce başarılı bir çözüm yöntemidir.

2.4.2.4.2 Kullanıcı Bilinçsizliği

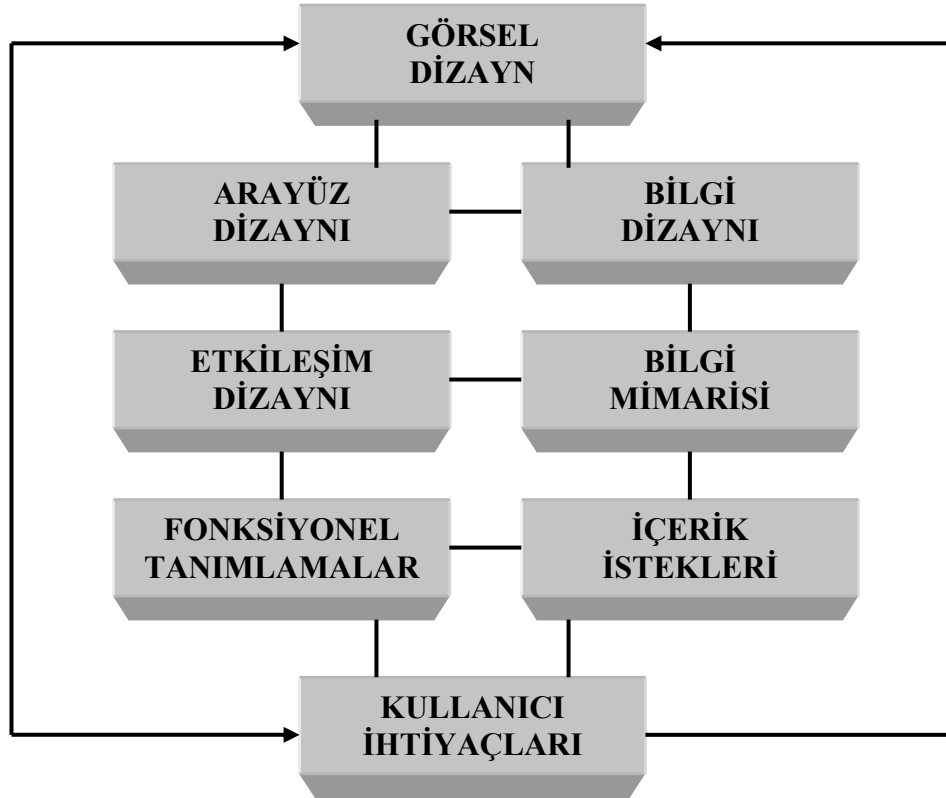
Kullanıcı arayüzünün görselliğini etkileyen en önemli faktörler : Renk, biçim, düzen ve metin'dir. Bu faktörler kullanıcı bilinçsizliği ve kullanılabilirlik üzerinde doğrudan egemen olur.

2.4.2.4.3 Kullanışlılık

Yapısal modeli ve organizasyon modelini içinde bulunduran arayüzün yapı bölümü, kullanıcı arayüzünün kullanılabilirliğine en fazla zarar veren bölümdür. Çünkü modeli yanlış ve kötü oluşturulan bir arayüz kullanıcı tarafından hoş karşılanmaz. Arayüz modelindeki sunum bölümü, en üstte olmasına rağmen kullanılabilirliğe en az zararı verir.

2.4.3 Kullanıcı Arayüz Dizaynının Temel İskeleti ve Mimarisi

Arayüz dizayn kararlarını ve içerik gelişimini göstermek için bir taslak oluşturulur. Kullanıcı arayüz dizaynının temel iskeletini ve mimarisini gösteren yapı aşağıda gösterilmiştir [52]:



Şekil 37 : Kullanıcı Arayüz Dizayn'ının Temel İskeleti ve Mimarisi

2.4.3.1 Görsel Dizayn

Kullanıcı arayüzünü oluşturan metin, şekil ve grafiklerin görselliği arayüzün görsel dizaynının temelini oluşturur.

2.4.3.2 Arayüz Dizaynı

Arayüz elemanlarının dizaynı, kullanıcı ile sistem fonksiyonları arasındaki etkileşimi kolaylaştırır.

2.4.3.3 Bilgi Dizaynı

Sunulan bilginin dizaynı, anlamayı kolaylaştırır.

2.4.3.4 Etkileşim Dizaynı

Arayüz uygulamasının gelişimi, kullanıcı görevlerini kolaylaştırır ve kullanıcı ile arayüz fonksiyonları arasındaki etkileşimin nasıl olacağını gösterir.

2.4.3.5 Bilgi Mimarisi

Bilginin yapısal dizaynı, arayüzün içeriğine girişi kolaylaştırır.

2.4.3.6 Fonksiyonel Tanımlamalar

Fonksiyonların ayrıntılı tanımlamaları, kullanıcı ihtiyaçları ile aynı hedef doğrultusunda olmalıdır.

2.4.3.7 İçerik İstekleri

Arayüzün içeriğinin, kullanıcının gereksinimleri ile aynı doğrultuda olması gerekir.

2.4.3.8 Kullanıcı İhtiyaçları

Kullanıcının ihtiyaçları ve amaçları, arayüzün temelini oluşturur.

2.4.4 Kullanıcı Arayüz Dizaynının Yapısı

İyi bir arayüz oluşturmak ve geliştirmek için, grafiksel kullanıcı arayüzünün yapısını çok iyi bilmek gerekir. Kullanıcı arayüz dizaynı birbiriyle ilişkili üç yapıdan oluşur : Kullanılabilirlik, görsellik ve fonksiyonellik. Son zamanlarda ortaya çıkan bir diğer dördüncü faktör ise : Ulaşılabilirlik'tir. Kullanıcı arayüz dizaynının yapısını oluşturan bu dört faktör ve özellikleri şunlardır [53]:

2.4.4.1 Kullanılabilirlik

Kullanılabilirlik, grafiksel kullanıcı arayüz dizaynının kalbi ve en önemli yapısıdır. Arayüz dizaynındaki elemanların nasıl kullanılması gerektiğini ve nesnelerin kolayca nasıl yönetileceğini belirler. Kullanıcıya yardım etmek ve kullanım kolaylığı sağlamak en temel ilkesidir. Bu yapı, arayüz elemanları kullanıcı tarafından nasıl daha rahat ve kolay kullanılabilir düşüncesini benimser. Hem görsellik hemde fonksiyonellik tarafından etki altındadır.

2.4.4.2 Görsellik

Kullanıcı arayüz dizaynında ilgi çekici, göze hitap eden, insanları sıkmayan, estetik olan ve kullanıcıyı memnun eden nesnelerin kullanımı, görselliğin temelini oluşturur. Kullanıcı arayüzünde görselliğin sağlanması için, görsel efektler ve animasyonlardan yararlanılmalıdır. Arayüz elemanlarının düzen ve hizalarındaki bozukluklar, görselliği bozduğu gibi kötü görünümlere de yol açar.

2.4.4.3 Fonksiyonellik

Arayüz işlevsel olmalıdır. Fonksiyonellik, bütün arayüz elemanlarının çalışır durumda olması durumudur. Arayüzdeki elemanlar nasıl daha faydalı bir şekilde görevlerini yerine getirebilirler konusu, fonksiyonelliğin temelini oluşturur. Bir web sitesinin arayüzünde kullanılan arama motorları, sıkça sorulan sorular ve site haritası gibi bölümler fonksiyonelliği ilgilendiren bölümlerdir.

2.4.4.4 Ulaşılabilirlik

Ulaşılabilirlik, web sitelerinin arayüz dizaynlarında meydana gelen sorunlar sonucu ortaya çıkan bir konudur. Çalışmayan linkler, ulaşılamayan web sitesi adresleri, web sitesinin kullanılabilirliğini negatif yönde etkiler. Arayüz erişilebilir olmalıdır. Ulaşılabilirlik, çalışmayan arayüz elemanlarının kalmamasını sağlar.

2.4.5 Grafikselle Kullanıcı Arayüz Elemanları

Arayüz elemanları, kullanıcı arayüzünü geliştirmek için kullanılırlar. Kullanıcı ile bilgisayar arasında köprü vazifesi görürler. Çok iyi özellikler ile donatılmış olan kullanıcı arayüz elemanları, belirli ve özel görevleri yerine getirmek amacı ile oluşturulurlar. Arayüz elemanları, belirli görevleri kolayca yapmak, arayüze görsellik katmak ve arayüzü kullanıcıya kolay ve açık hale getirmek için kullanılırlar. Kullanıcı ile arayüz arasındaki etkileşimi en iyi şekilde sağlamak amacı ile dizayn edilen bu arayüz elemanları şunlardır [54]:

2.4.5.1 Düğme (Button)

Uygulamaların ve olayların eyleme geçirilmesinde kullanılır.



Şekil 38 : Düğme (Button)

2.4.5.2 İkon (icon)

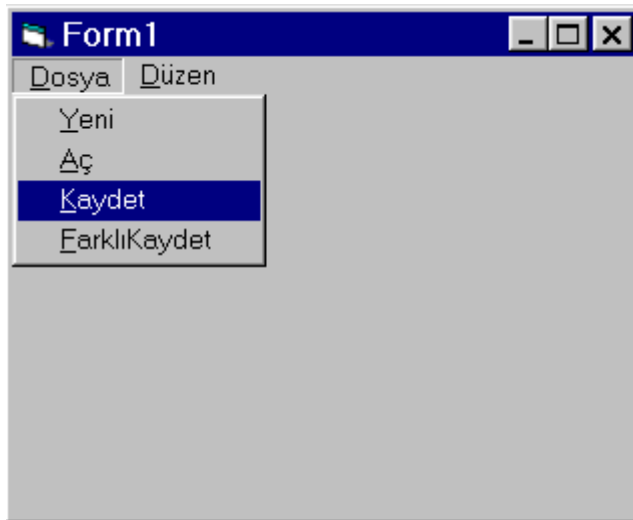
Dosya ve komutları temsil eden, bir nesneyi ya da programı simgeleyen küçük resimler. İkonlar, kullanıcı arayüzünde gerçek nesnelere simgelemek ve arayüzdeki fonksiyonları temsil etmek amacıyla tasarlanmıştır. Bir arayüzdeki bütün dosyalar, klasörler ve programlar ikon ile temsil edilirler. İkonlar orijinal, küçük ve her uygulama için farklı özellikte olmalıdır.



Şekil 39 : İkon (icon)

2.4.5.3 Menü (menu)

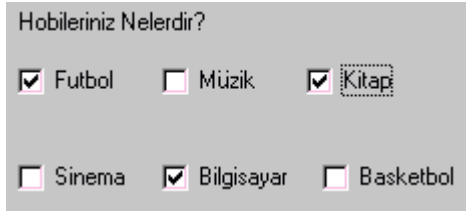
Bazı fonksiyonları ve komutları yerine getirmek amacıyla oluşturulan ve arayüze görsellik ve çekicilik katan, kullanıcıya ise kolaylık sağlayan yapılardır. Menü, kullanıcıya birçok komut ve seçme hakkı sunup, bunların kullanılmasına olanak sağlayan bir grafiksel arayüz elemanıdır.



Şekil 40 : Menü (menu)

2.4.5.4 Denetim Kutusu (Check Box)

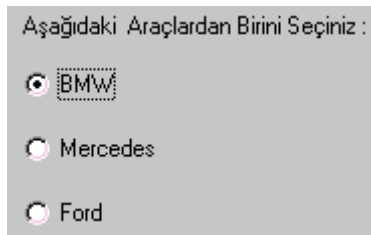
Aynı anda iki farklı şeyi birden seçme ve kontrol etme olanağı sunan bir grafiksel arayüz elemanıdır.



Şekil 41 : Denetim Kutusu (Check Box)

2.4.5.5 Seçenek Düğmesi (OptionButton)

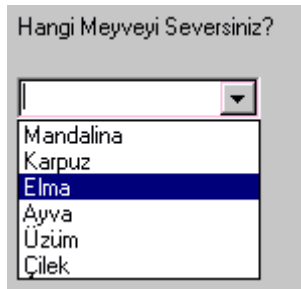
İki farklı şeyden sadece bir tanesini seçme imkanı sağlayan bir grafiksel arayüz elemanıdır.



Şekil 42 : Seçenek Düğmesi (OptionButton)

2.4.5.6 Grup Kutusu (Combo Box)

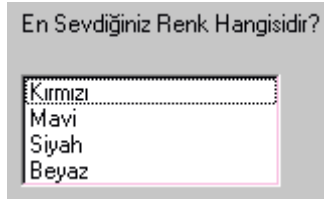
Bir topluluğa ait elemanları bir grup içinde tutmayı sağlar. Grup içindeki elemanlardan sadece bir tanesini seçme ve o seçilen elemanın fonksiyonlarını yerine getirme imkanı sunar.



Şekil 43 : Grup Kutusu (Combo Box)

2.4.5.7 Listeleme Kutusu (List Box)

Kullanıcı, dizayn edilen arayüzden tek bir elemanı ya da birden çok elemanı seçmek istediği zaman kullanılır.



Şekil 44 : Listeleme Kutusu (List Box)

2.4.5.8 Metin Alanı (TextField)

Bir metni görüntülemek ya da okumak için kullanılır.



Şekil 45 : Metin Alanı (TextField)

2.4.5.9 Etiket (Label)

Basit bir metin gösterim yeridir. Her türlü yazı genişliğine, büyüklüğüne ve yazı karakterine sahip metinlerin gösteriminde kullanılır.

Grafiksel Kullanıcı Arayüz (GUI) Elemanları

Şekil 46 : Etiket (Label)

2.4.6 Grafiksel Kullanıcı Arayüz (GUI) Dizayn Teknikleri

Kullanıcı arayüz dizayn teknikleri, etkili bir arayüz dizaynının ve gelişiminin temelini oluştururlar. Bu teknikler, arayüzün kalitesini arttırmayı sağladığı gibi, arayüzün daha önemli ve daha güvenilir olmasına katkıda bulunurlar. Arayüz dizaynını gerçekleştiren kişilere kılavuzluk ederler ve yol gösterirler. Arayüz dizayn teknikleri, arayüzün kullanılması konusunda fazla bilgiye gerek olmadan arayüzde yeterince iş yapabilme olanağı sağlarlar. Arayüzü dizayn eden kişi ya da kişilerin en büyük amacı, mümkün olabilecek en iyi arayüz dizaynını gerçekleştirebilmektir. Fakat bu konuda, amaçlarına ne kadar ulaşabildikleride büyük bir soru işaretidir. Arayüzü dizayn eden kişi ya da kişilerin amaçlarına uygun bir arayüzü oluşturabilmeleri, arayüz dizayn tekniklerine uymakla mümkündür. Oluşturulan kullanıcı arayüzünü geliştirmek için bazı teknikler kullanmak, arayüzü dizayn eden kişi ya da kişilere yardımcı olacaktır. Arayüzü dizayn etmek için kullanılan grafiksel kullanıcı arayüz dizayn teknikleri şunlardır [55] [56]:

2.4.6.1 Kullanıcının Etkinliđi

Bilgisayar, arayüz ve çalışma alanı, kullanıcıya ait olmalıdır. Arayüz, insanların kendilerini çok iyi hissettikleri bir yer olmalıdır. Arayüz dizayn edilirken kullanıcı etkinliđi ön planda tutulmalıdır. Kullanıcı kendisini arayüzü kullanırken rahat hissetmelidir. Aksi halde kullanıcının verimliliđi azalacak ve arayüze karşı bir sođukluk oluşacaktır. Arayüzün kullanıcıyı deđil, kullanıcının arayüzü yönettiđi bir ortam oluşturulmalıdır. Kullanıcı arayüzde zorlanmamalıdır. Arayüzdeki herşey kullanıcı düşünülerek basite indirgenmelidir. Kullanıcının yaptıđı işlerin hepsi arayüzde tutulmalı ve istenildiđi zaman kolayca görülebilmelidir. Arayüz kullanıcının yaptıđı işlerin sonuçlarını otomatik olarak saklamalı, korumalı ve kaydetmelidir. Böylece kullanıcılar yaptıkları işleri görmüş ve üzerine yeni ilaveler yapma imkanı bulmuş olacaklardır. Kullanıcılar gereksiz yere fiziksel ve zihinsel açıdan yorulmamalıdır. Kullanıcı memnuniyetini sağlamak, her zaman gözden kaçırılmaması gereken bir olaydır. Arayüz dizayn edilirken kullanıcı açısından dikkat edilmesi gereken en önemli unsurlardan biri, kullanıcının genç-yaşlı olma durumudur. Genç kullanıcıların arayüzü anlama ve kullanma kapasitesi ile yaşlı kullanıcıların arayüzü anlama ve kullanma kapasitesi birbirinden farklıdır. Bu yüzden arayüz dizayn edilirken kullanıcı kitlesinin yaş grubu göz önünde bulundurulmalıdır.

2.4.6.2 Renk Körlüğü Problemi

Renk ile ilgili en büyük problem, arayüz kullanıcılarından bazılarının renk körü olma durumudur. Ekranda bazı şeylerin dikkatini çekmek için renk kullanımı yapılırsa, renk körü kişiler dizayn edilen bu arayüzde hesaba katılmamalıdır. Dizayn edilen arayüzün gerçek sonuçlarının daha önceden farkedilmesi gerekir. Örneđin; arayüz dizaynında kullanılan farklı amaçlı renklerin farkına varılamaması, renk körlüğü olan kullanıcılarda sorunlara yol açacaktır. Buna karşılık, renk körü olan kullanıcılar düşünülerek renk kullanımının yapılmaması arayüz dizaynını olumsuz yönde etkileyecektir. Gözler kırmızı, yeşil ve mavi renklere hassastır. Renk körü olan kullanıcılar düşünülerek bu renklerin arayüzde kullanılmaması ise birtakım sorunlara yol açar. Çünkü, renkler arayüzde vurgulanmak istenen bilgileri açıklamak için kullanılırlar. Bu yüzden arayüzde anlatılmak istenenler tam olarak vurgulanamaz. Renk körü olan insanlara arayüzde sunulmak istenen şeyler semboller kullanılarak aktarılmalıdır.

2.4.6.3 Arayüzde Tutarlılık

Arayüzde tutarlılık, kullanıcı arayüzünün yapısına ve içeriğine uygun bir modelin oluşturulması ile başlar. Arayüzde tutarlılığı sağlamak, arayüzü dizayn eden kişi ya da kişilerin tecrübelerine bağlıdır. Arayüz tasarımında kararlılık ve tutarlılık, arayüzün ana noktalarının sunulması açısından çok önemlidir. Uygulamada kullanılan veriler, arayüz tasarımının doğruluğunu belirler. Arayüzde tutarlılığın seviyesi iki aşamada incelenmelidir : Birincisi, arayüz elemanlarının tutarlılığıdır. Arayüzü oluşturan arayüz elemanları, arayüzün genelinde aynı genişlik ve büyüklükte olmalıdır. Buda arayüzün genelinde görsel bütünlüğü sağlamış olacaktır. Metin fontları ve genişlikleri arayüzün her yerinde aynı olmalıdır. Çok küçük metin fontlarından kaçınılmalıdır. Aynı fonksiyonun devamı olan farklı ekranlarda, renkler tutarlı olmalıdır. Arayüz elemanlarında tutarlılığın sağlanmaması, kullanıcı arayüz dizaynını olumsuz yönde etkileyecektir. Örneğin; bir düğme (button) işlevini yerine getiriyorsa arayüzdeki bütün düğmeler görevlerini yerine getirmelidir. Eğer çalışmayan, görevini yerine getirmeyen düğme (button) olursa arayüzde tutarsızlık olur. Arayüz elemanlarının farklı biçimlerde ve genişliklerde olması arayüzün geneli düşünüldüğünde tutarsızlığa yol açacaktır. Tutarlılıkta ikinci aşama ise, arayüz'ün dizayn edilmiş amacı ile kullanıcı isteklerinin birbirini tutmasıdır. Kullanıcının istek ve beklentilerine uygun olmayan bir arayüz tutarsızdır ve kötü dizayn edilmiş demektir.

2.4.6.4 Arayüzün Kullanıcı Hassasiyeti

Arayüzün çeşitli yönlerini kullanıcıya göstermek gerekir. Çünkü kullanıcı, arayüzün yeni yerlerini keşfettikçe arayüze ısınacak, böylece arayüze olan güveni artacaktır. Bununla birlikte kullanıcı memnuniyeti sağlanmış olacaktır. Arayüzün değişik yerlerini ve gidişatını yönetmek ve kullanıcıya tanıtmak için görsel işaretler kullanılmalıdır. Kullanıcı, arayüzde yavaş çalışıp zamandan kaybetmemelidir. Az zamanda çok iş yapabilmelidir. Kullanıcı, arayüzde dikkatini hata yapmamak için sarfetmemelidir. Dikkatini arayüzde yapacağı işler ya da aramalar için sarfetmelidir. Arayüz, kullanıcı için hassas olmalıdır. Arayüzün dizayn edilmesinin ilk ve en önemli amacı, kullanıcı olmalıdır.

2.4.6.5 Arayüzde Kullanıcıya Yol Gösterici Olmak

Kullanıcı, arayüz programını kullandığında ve bir işlem yaptığında arayüzde beklediği sonucun geleceğine dair görsel nesnelere kullanılmalıdır. Böylece kullanıcı doğru yolda olup olmadığını ve bilgiye nasıl kolay ve az zamanda ulaşabileceğini anlamış

olur. Arayüzde kullanıcıya bilgilendirici nesnelere sunulmalıdır. Kullanıcı arayüzde yalnız bırakılmamalıdır. Arayüzde gizlilik kullanıcı için ortadan kaldırılmalıdır. Kullanıcı arayüzde neyin nerde olduğunu ve arayüzün nasıl bir yapısı olduğunu bilmelidir. Kullanıcıya arayüzde yol gösterici olunmalıdır. Kullanıcının olayları ve işleri bekleme süresi azaltılmalıdır. İşlemler kısa zamanda ve doğru bir şekilde olmalıdır. Örneğin; kullanıcı, arayüzdeki bir dosyayı kaydederken, bu işlemin ne kadar zaman alacağı konusunda bilgilendirilmelidir. Kullanıcının hata ve problemlerinde, arayüzde kullanıcıya yol göstermek için yardım bölümleri oluşturulmalı ve uyarı mesajları oluşturulmalıdır.

2.4.6.6 Arayüzde Açıklık ve Sadelik İlkesini Benimsemek

Arayüz kullanıcının kafasını karıştırmamalı ve kullanıcı tarafından anlaşılır olmalıdır. Kullanıcı arayüzü, bilgisayar bilgisi hiç olmayan kimselere göre dizayn edilmelidir. Arayüz elemanları, açık ve sade olmalıdır. Kötü ve yetersiz organize edilmiş bir arayüz, karmaşıklığa ve kullanıcının dikkatinin dağılmasına neden olur. Arayüz, anlaşılması açısından basit ve sade olmalıdır. Çoğu arayüz uygulaması genelde kullanıcılara pek açık olmaz. Buda kullanıcının arayüzde çoğu şeyi anlamasını engeller. Bir arayüz uygulamasının açıklığını arttırmanın en etkili yolu, önceden belirlenmiş ve özenle seçilmiş kelimeler kullanmaktır. Özenle seçilmiş bu kelimeler, arayüzün açıklık ve sadelik ilkesini ön plana çıkartacaktır. Kullanıcıların arayüz hakkındaki en belirgin şikayetlerinden biri, arayüzdeki ana terimlerin anlaşılır ve net olmamasıdır. Bu yüzden kullanıcı memnuniyeti sağlanmalı ve arayüz kullanıcının anlayacağı netlikte dizayn edilmelidir.

2.4.6.7 Tekrarlama ve Yineleme'lerden Kaçınmak

Arayüzde tekrarlama ve yineleme özelliği, metinden çok grafiksel elemanlarda olmalıdır. Arayüzün kullandığı metinlerde dönüp dolaşıp hep aynı şeylerden söz etmesi kullanıcıyı sıkar. Böylece, kullanıcı tarafından arayüze karşı bir hoşnutsuzluk oluşmuş olur. Tekrarlanan grafiksel elemanlar rahat ve farkına varılabilir, ayrıntılı bir konuya neden olur. Arayüzde ayrıntıların olması, kullanıcının bazı bilgileri unutmasını engeller. Arayüzde tekrarlılık ile yineleme özelliği, tutarlılık ve yoğunluk sağlar. Tekrarlama ve yineleme özelliği ile arayüz, kullanıcının kolay öğrenme ve hatırlamasına imkan tanır. Ancak, her zaman arayüzde tekrarlama ve yinelemenin olması kullanıcı tarafından hoş karşılanmayabilir. Arayüzde sürekli aynı şeylerin

tekrar edilmesi kullanıcının sıkılmasına yol açar. Arayüzün sürekli kendini yenilemesi ve yeni teknolojilere açık olması gerekir. Çağın gerisinde kalmış bir arayüz, kullanıcılar tarafından sevilmez.

2.4.6.8 Kullanım Kolaylığı

Arayüzün kullanılması kolay olmalıdır. Arayüz, bilgisayar bilgi seviyesi en düşük olan insanlara göre dizayn edilmelidir. Çünkü arayüz dizayn edilirken kullanıcı memnuniyeti ön planda olmalıdır. Kullanıcı, arayüzü kullanırken rahat olmalıdır, zorlanmamalıdır. Arayüzün kullanılabilirlik özelliği üst düzeyde olmalıdır. Eğer kullanıcı, arayüzü kullanırken kendini rahat hissetmezse, bu durum kullanıcının performansını olumsuz yönde etkileyecektir. Arayüz dizayn edilirken ilk düşünülmesi gereken arayüzün kullanım kolaylığının olmasıdır. Çünkü arayüz her ne kadar kullanıcının istek ve beklentileri doğrultusunda oluşturulmuş olsa da, eğer arayüzün kullanım kolaylığı yoksa kullanıcı bu bilgilere asla ulaşamaz. Böyle bir durumda, kullanıcının arayüz hakkındaki düşünceleride olumsuzla döner. Arayüz, kullanıcının karşısına her türlü hatalardan arınmış olarak çıkmalıdır. Kullanıcıya, arayüzde kullanım sağlayan en önemli faktör; arayüzün başka kullanıcılar tarafından test edilmiş olmasıdır. Daha önce başka kullanıcılar tarafından test edilmiş olan bir arayüzün kullanılabilirlik düzeyi, en üst seviyededir.

2.4.6.9 Kullanıcı Profili'ni Belirlemek

Arayüz dizayn edilmeden önce kullanıcı profili mutlaka belirlenmelidir. Çünkü kullanıcı, arayüzün en önemli ve en temel ögesidir. Arayüz dizaynı sırasında kullanıcı profili belirlenirken dikkat edilmesi gereken çeşitli prensipler vardır. *Arayüzü kim kullanacak?* prensibi, arayüz dizaynı sırasında düşünülmesi gereken en önemli unsurdur. Arayüzü kullanacak olan kullanıcının özelliklerini bilip, bu özellikler doğrultusunda arayüzün dizayn edilmesi en mantıklısıdır. Kullanıcı profilini belirleyen ikinci prensip : *Kullanıcının amaçları nelerdir?* prensibidir. Kullanıcının arayüzü hangi amaçla kullandığı bilinmeli ve kullanıcıya göre arayüz dizayn edilmelidir. *Kullanıcının beceri ve tecrübesi var mıdır?* prensibi, kullanıcı profilini belirleyen üçüncü prensiptir. Kullanıcının daha önce bilgisayar kullanıp kullanmadığını bilmek, arayüzü dizayn eden kişi ya da kişilerin önem verdikleri önemli bir konudur. Kullanıcının bilgi seviyesi ve eğitim düzeyi göz önüne alınmalıdır. Çünkü daha önce bilgisayarı birçok kez kullanmış bir kullanıcı ile

bilgisayarı hiç kullanmamış bir kişinin, arayüzü kullanma becerileri aynı olmayacaktır. Kullanıcı profilinde kullanılan önemli ve son prensip : *Kullanıcının ihtiyaç ve istekleri nelerdir?* prensibidir. Kullanıcının beklentileri, arayüzün genel yapısının belirler. Kullanıcının ihtiyaç ve istekleri ile arayüzün içeriği aynı olmalıdır. Kullanıcı profilini belirleyen bu dört prensip, aynı zamanda arayüz dizaynının temelinde de önemli bir yere sahiptirler. Arayüz dizaynında kullanıcı profilini ikiye bölen çeşitli nedenler vardır. Örneğin; kullanıcının genç-yaşlı olması, bilgisayar konusunda tecrübeli-acemi olması ya da eğitim seviyesinin yüksek-düşük olması gibi. Bu kullanıcı farklılıkları, arayüz dizayn aşamasında iki farklı düşüncenin ortaya çıkmasına yol açar.

2.4.6.10 Ekran Yönetimi

Arayüzde bir ekrandan diğerine geçiş zor ise, bu kullanıcının engellenmesine ve hoşnutsuzluğuna yol açacaktır. Belli bir süre sonra kullanıcı, arayüzü bırakma noktasına kadar gelecektir. Ekranlar arasındaki akış sağlanmalı, iletişim ve etkileşim kurulmalıdır. Buda kullanıcı memnuniyetini getirecektir. Arayüz ekranı, farklı kullanıcıların farklı işlerini yapabilecek esnekliğe sahip olmalıdır. Ortalama olarak insan beyni, 2000 ile 3000 arası kelimeyi anımsayabilir. Ayrıca, 50.000'den daha fazla kelimeyi de tanıyabilir. Bu yüzden arayüz ekranı, fazlaca metin ve kelime yoğunluğu içinde olmamalıdır. Bunların dışında, arayüz ekranı oluşturulurken herşeyi arayüzün birinci ekranına koymaktan kaçınılması gerekir. Çünkü kullanıcı aradıklarını arayüzün ilk ekranında bulursa, arayüzün diğer ekranlarına bakmaz. Böylece arayüzün diğer ekranlarında anlatılanlar arka planda kalmış olur. Buda arayüzün kalitesini ve bütünlük ilkesini zedeler.

2.4.6.11 Ekran Organizasyonu

Batı toplumlarında yazılar, soldan sağa ve yukarıdan aşağıya biçiminde okunur. Genel olarak ekran dizayn edilirken, soldan sağa ve yukarıdan aşağıya olarak organize edilir. Bunun dışında bir ekran dizaynı bu tip toplumlar için bir anlam karmaşıklığına yol açacaktır. Örneğin; Araplar yazıları sağdan sola doğru okuduklarından onlar için soldan sağa doğru dizayn edilmiş bir arayüz ekranının kullanışlılığı yoktur. Bu nedenle arayüz dizayn edilirken ihtiyaçların belirlenmesi aşamasında kullanıcıların hangi toplumdan oldukları muhakkak bilinmeli ve arayüz ekranı ona göre oluşturulmalıdır.

2.4.6.12 Mesajları Uygun Bir Şekilde İfade Etmek

Arayüzde, kullanıcı için bilginin en önemli kaynaklarından biri kullanılan mesajlardır. Mesajlar, kullanıcılar için açıklayıcı, bilgilendirici ve yol gösterici olmalıdır. Eğer mesajlar kötü ve yetersiz bir şekilde ifade edilirse, o zaman kullanıcılarda arayüzü kötü ve yetersiz olarak algılayacaklardır. Mesajlar kullanıcıyı uyarıcı nitelikte olmalıdır. Yanlış kelimeler kullanmak ya da yanlış içeriği olan farklı mesajlar oluşturmak, kullanıcının kafasının karışmasına ve arayüze olan inancının azalmasına sebep olacaktır. Arayüzde mesajlar kullanıcıya yol gösterecek ve yardım edecek şekilde oluşturulmalıdır. Mesajlarda kullanılan kelimeler doğru seçilmeli ve arayüzün içeriğine uygun olmalıdır. Mesajlarda kısaltmalar kullanılmamalıdır. Bu durum, kullanıcıda anlam karmaşıklığına yol açacaktır. Mesajlar oluşturulurken anlamlı kelimeler kullanmak, mesajlar ile anlatılmak ve vurgulanmak istenen şeylerin amacına ulaşmasını sağlayacaktır. Mesajlar oluşturulurken kullanıcı, arayüz uygulamasını nasıl kullanacak düşüncesi ön planda olmalıdır. Mesajlar kullanıcıyı yormamak için tutarlı olmalıdır.

2.4.6.13 Arayüz Elemanlarını Doğru Kullanmak

Doğru görevler için doğru arayüz elemanları kullanılmalıdır. Arayüz elemanlarını doğru kullanmak, arayüz uygulamasındaki tutarlılığı arttırmak için yardımcı olacaktır. Arayüzün kullanımının kolay olması, arayüz elemanlarının doğru yerde ve doğru şekilde kullanılması ile mümkündür. Kullanıcının arayüzü anlaması ve arayüzü nasıl kullanacağını bilmesi, arayüz elemanlarının doğru şekilde kullanılmasına bağlıdır. Her arayüz elemanının farklı bir özelliği vardır. Arayüz elemanları, farklı görevleri yerine getirmek amacı ile oluşturulurlar. Arayüz elemanlarının işlevleri birbirine karıştırılmamalıdır. Örneğin; seçenek düğmesi (option button) yerine denetim kutusu (check box) kullanılması gibi. Kullanıcı arayüz standartlarına uygun olarak oluşturulan arayüz elemanları doğru kullanılmış olur. Arayüz elemanlarının doğru kullanılması arayüzün kullanıcılar tarafından benimsenmesini sağlar.

2.4.6.14 Renk Kullanımına Dikkat Etmek

Renk kullanımı, arayüzün görselliğine direkt etki eden bir unsurdur. Renk kullanımı, arayüze uygun olmalıdır. Arayüz uygulamasında vurgulanmak istenen yerlerin gösterilmesinde renklerden yardım alınır. Arayüzün farklı ekranlarında kötü renk kullanımı, hem arayüzün görselliğini bozar hemde kullanıcının gözünü yorar. Arayüzün arka plan rengide, arayüz tasarımında dikkat edilmesi gereken önemli bir

konudur. Renk arayüz uygulamalarında tutarlı kullanılmalıdır. Arayüzün bir ekranında renk kullanımı iyiyken, diğer ekranında yetersiz renk kullanımı olmamalıdır. Bu arayüzün kötü dizayn edildiğini gösterir.

2.4.6.15 Tezatlık Kuralını Benimsemek

Arayüzde karşıtlık ile tezat, dikkat ve ilgi çekmek için kullanılır. Arayüz nesnelерinin sivil, renk ve büyüklüklerindeki tezatlıklar kullanıcının arayüz üzerindeki dikkatinin dağılmasını önler. Arayüz uygulamasında renk kullanılırken, arayüz ekranının okunabilir olmasına dikkat etmek gerekir. Arayüzde birbirinden zıt özelliklerin kullanılması kullanıcının dikkatinin çekilmesini sağlar. Tezatlık kuralının uygulanması ile ilgili en iyi örnek : Açık renkli arayüz arka planında koyu renkli metin kullanılması ya da koyu renkli arayüz arka planında açık renkli metin kullanılması gösterilebilir. Örneğin; beyaz renkli arka planda siyah renkli metin kullanılması gibi. Ancak tezatlık kullanımı her zaman yararlı sonuçlar vermez. Örneğin; beyaz renkli arayüz arka planında mavi renkli metnin okunması oldukça kolaydır. Fakat, kırmızı renkli arayüz arka planında mavi renkli metnin okunması çok zordur.

2.4.6.16 Metin Fontlarının Uygunluğunu Sağlamak

Arayüzde, metin fontları birbirine uygun biçimde kullanılmalıdır. Arayüz üzerindeki her yerde metin fontlarının genişlikleri standart olmalıdır. Aksi halde bu durum arayüzün görselliğini olumsuz yönde etkileyecektir. Metin fontları farklı türde, farklı genişlikte ve farklı sivilde kullanılmamalıdır. Görülemeyecek kadar küçük genişlikte metin fontlarının kullanımı arayüz ekranı üzerinde okumayı zorlaştırır. Farklı tür, genişlik ve sivilde metin fontlarının kullanılmamasının ilk amacı, kullanıcının arayüz ekranında okumasını kolaylaştırmayı sağlamaktır. Font kullanımı, arayüzün bütün ekranlarında aynı tutarlılığı sağlamalıdır. Bir arayüz ekranında iki ya da üç tip font'un kullanılması, beş ya da altı tip font kullanımından daha etkili ve daha güzel bir görünüm sağlar.

2.4.6.17 Arayüz Elemanlarının Düzen ve Hizalarına Önem Vermek

Arayüzde sıraya ve hizaya sokulmuş nesnelер gözü yormaz, kullanıcıya kolaylık sağlar ve kullanıcının dikkatini dağıtmaz. Bütün arayüz elemanları bir hizada bulunmalıdır. Arayüzü oluşturan elemanların düzen ve hizalarına dikkat edilmelidir. Aksi halde bu durum düzen ve görünüşte bozukluğa ve kötü görünüme yol açacaktır.

Arayüz üzerindeki arayüz elemanları, belirli bir model dahilinde ve bir düzen içerisinde yerleştirilmelidirler. Arayüzde simetrinin eksik olması, arayüz elemanlarının kullanımını zorlaştırır. İnsan beyni düzeni ve sıralamayı anladığı için, arayüzde arayüz elemanları mantıksal bir yapıda sıralanmalıdırlar. Arayüz elemanlarının, arayüz üzerindeki görünümüleri arayüzün görselliğini etkiler. Görselliği ilgi çekici ve etkili olan arayüz elemanları, iyi bir organizasyon dahilinde arayüz ekranına yerleştirilmelidirler. Yerleri ve hizaları bozuk olan arayüz elemanları kullanıcının arayüz üzerindeki fikirlerini olumsuz yönde etkileyecektir. Arayüz elemanlarının düzen ve hizalarının sağlanmasında arayüz elemanlarının arayüz üzerinde buldukları yönler önemlidir.

2.4.6.18 Kalabalık Arayüz Ekranı Yaratmaktan Kaçınmak

Arayüz ekranı üzerinde gereksiz nesne, grafik, resim ve metin kullanımından kaçınılmalıdır. Bu durum arayüz ekranının kalabalık olmasına yol açacaktır. Kalabalık bir arayüz ekranı, kullanıcının kafasının karışmasına, istediği şeyleri bulamamasına ve dikkatinin dağılmasına neden olacaktır. Bir arayüz ekranının çok gereksiz bir şekilde kalabalık olması, anlamayı ve kullanmayı zorlaştırır. Arayüz ekranı sade ve anlaşılır olmalıdır. Ayrıca, arayüz ekranı kullanıcının gözünü yoracak şekilde oluşturulmamalıdır. Bununla birlikte, yapılan deneysel sonuçlara göre genelde arayüz ekran çözünürlüğü %40'ı aşmamalıdır.

2.4.6.19 Klavyeden (Keyboard) Kısa Yollar Sunmak

Arayüz dizayn işlemi sırasında, kullanıcının veriyi kolayca bilgisayardan alması, kopyalaması, değiştirmesi ve silmesi işlemlerini yapması için arayüzde kısa yol tuşları kullanılmalıdır. Kısa yol tuşları, arayüzün iyi ve eksiksiz bir şekilde dizayn edilmesi ve kullanıcılara arayüzde kolaylıklar sağlamak için kullanılır. Kullanıcılar, her zaman arayüzdeki bilgiye daha çabuk ve daha kolay ulaşmak isterler. Klavye kısayollarında bu imkanı kullanıcıya sundukları için, arayüz dizayn edilirken bu klavye kısa yollarının kullanılması gerekir. Örneğin; kullanıcı, arayüzden bir bilgiyi kopyalarken, kopyalama düğmesinin (button) yerine “Ctrl+C” kısayol tuşunu kullanması gibi. Arayüzün klavyeden kısa yollar sunması, kullanıcıların arayüzde kısa zamanda çok işler yapması için faydalıdır. Klavye kısa yolları kullanıcılar için zamandan tasarruf sağlar. Sonuç olarak, kısa yol tuşlarının kullanılmasının amacı arayüzde kullanıcılara kullanım kolaylığı sağlamak ve karmaşıklığa yol açmamaktır.

2.4.6.20 Dalgınlıklardan Kaçınmak

Arayüz dizaynı dalgın olmayı içinde barındırmayacak kadar sorumluluk gerektiren önemli bir iştir. Arayüzü dizayn eden kişi ya da kişilerin farkında olmadan yaptıkları hatalar, arayüzün sürekli problem çıkarmasına neden olacaktır. Bu olayda kullanıcı tarafından pek hoş karşılanmayacaktır. Arayüzü dizayn eden kişi ya da kişilerin, dalgınlık sonunuca arayüzde yaptıkları hatalar, kullanıcıların arayüzden sıkılmasına neden olur. Örneğin; dalgınlık sonucu arayüzün bazı ekranlarında arka plan rengi olarak kırmızı kullanılırken, bazılarında ise mavi kullanılması hem arayüzün görsel bütünlüğünü bozacak hemde kullanıcının arayüzden soğumasına neden olacaktır. Arayüzü dizayn eden kişi ya da kişilerin çok dikkat etmeleri ve önemli bir sorumluluğun içinde olduklarının farkında olmaları gerekir. Aksi halde hem kullanıcı açısından hemde arayüzü dizayn eden kişi ya da kişiler açısından zaman ve para kaybı olacaktır.

2.4.6.21 Uyarı ve Hata Mesajları Kullanmak

İyi bir grafiksel kullanıcı arayüzü için uyarı ve hata mesajlarına ihtiyaç vardır. Uyarı ve hata mesajları arayüzün gelişimine katkıda bulunurlar. Yapılan hataların geriye dönülmez olması ve uyarı mesajları içermemesi kullanıcıyı zor durumda bırakır. Oysa arayüz dizayn edilirken kullanıcı hatalarının göz önünde bulundurulması gerekir. Dizayn edilen arayüzün kullanıcının veri girmesine, arayüzden veri almasına yardım etmemesi ve arayüzün bu olay ile ilgili uyarı mesajları içermemesi arayüzün en büyük eksikliği olarak görülebilir. Kullanıcının yaptığı hatalarda arayüz tarafından uyarılması, kullanıcının arayüze olan güvenini artırır. Uyarı ve hata mesajları ile, kullanıcı yaptığı hatalardan kurtulur ve arayüzde istediği bilgiye ulaşmış olur. Bilgilendirici hata mesajları, arayüzde kullanıcının neyi yanlış yaptığını ve arayüzde yapacağı işlemlerde nasıl davranması gerektiğini gösterir. Bu durum, kullanıcının yaptığı hatalardan sonra uyarılacağını bilme şansı ile arayüzü kullanırken rahat olmasını sağlar. Mesajlarda kullanıcının anlayabileceği kelimeler kullanılmalıdır. Anlamı belli olmayan, kullanıcıyı yanıltan mesajlar kullanılmamalıdır.

2.4.6.22 Bilgiye Geri Dönüşüm Sağlamak

Kullanıcı, arayüzü kullanırken arayüzde ilerlemelerin olmaması kullanıcının yeni bilgilere ulaşmasını engelleyecektir. Bu durumda kullanıcı, arayüzü kapatır ve terkeder. Kullanıcının, arayüzde daha önce kullandığı bilgilere yeni birşeyler ekleme ya da silme ihtimali olabilir. Bu yüzden, arayüzde mutlaka bilgiye geri dönüşüm

olmalıdır. Kullanıcı, arayüzde yaptığı her işleme tekrar geri dönebilme şansına sahip olmalıdır. Arayüz elemanları kullanıldığında, arayüz elemanlarının çalışmaması ya da görevlerini yerine getirmemesi, arayüzün geriye dönüşüm imkanına sahip olmamasına neden olur. Arayüz, yapılan hareketin telafisini kullanıcıya vermelidir. Kullanıcı bir önceki hareketine geri dönebilmelidir. Arayüzde yapılan işlemlerde gecikmeler olmamalıdır. Arayüzde gecikmelerin olması, kullanıcının bilgiye geri dönüşümünü olumsuz yönde etkileyecektir.

2.4.6.23 Ses, Resim, Grafik ve Animasyon Kullanımı

Ses, resim, grafik ve animasyon elemanları arayüzün görünümünü, özelliklerini ve yapısını etkileyen faktörlerdir. İlk bakışta kullanıcının hoşuna giden şeyler gibi gözüksekselerde zaman zaman kullanıcı memnuniyetsizliğine yol açabilecek durumda olabilirler. Örneğin; büyük çaplı resim, grafik ve animasyonların kullanılması arayüzün geç açılmasına yol açar. Buda kullanıcı için zaman kaybına sebep olacaktır. Genel olarak, arayüzün başlangıcında ses efektlerinin kullanılmaması gerekir. Çünkü arayüzün başlangıcında kullanılan ses, kullanıcının dikkatinin dağılmasını ve kullanıcının arayüzden başka bir tarafa yönelmesini sağlar. Arayüzde sesin kullanılmasında dikkat edilmesi gereken bir başka unsur ise kulakları duymayan kullanıcıların olmasıdır. Arayüzde ses kullanıldığı zaman, kulakları duymayan kullanıcılar, arayüzde ses ile verilmek istenen bilgilere ulaşamayacaklardır. Bu yüzden ses efektleri ile anlatılmak istenen şeylerin, metinler kullanılarak anlatılması en doğru yol olur.

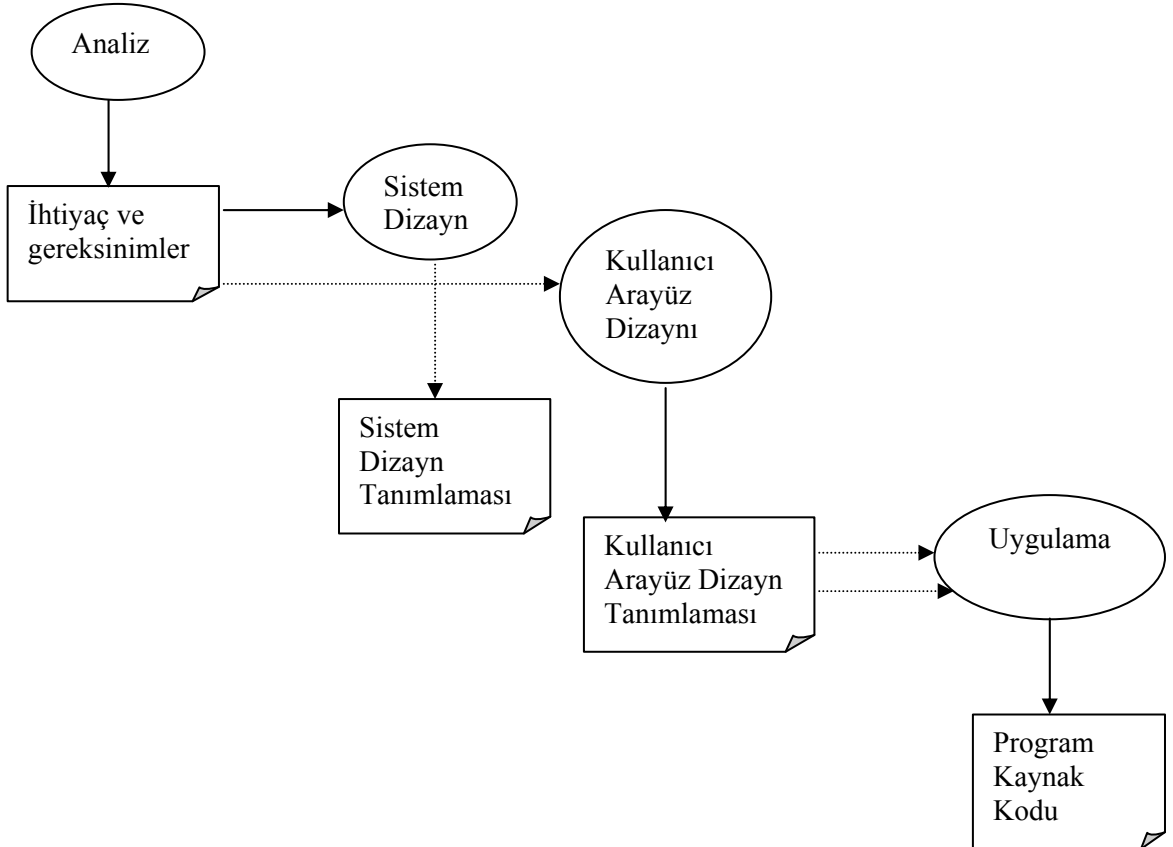
2.4.6.24 Eş Anlamlı Kelimeler & Benzerlikler'den Kaçınmak

“Eş anlamlı kelimeler & benzerlikler'den kaçınmak” yöntemi, kullanılan arayüz elemanının bir başka benzerinin olmaması ilkesine dayanır. Aynı fonksiyonları içeren farklı arayüz elemanları karmaşıklığa ve belirsizliğe neden olur. Yine arayüz elemanları üzerinde yazılan kelimelerin eş anlamlı olması hatalara yol açar. Örneğin; farklı amaçlar için oluşturulmuş iki tane düğme (button) üzerinde eş anlamlı kelimeler kullanılmamalıdır. Arayüz'ün farklı ekranlarında aynı anlamı temsil eden farklı kelimelerin kullanılması, arayüzde sorunlara neden olacaktır. Örneğin; bir arayüz uygulamasında vurgulanmak istenen “eşya” kelimesinin arayüzün bir ekranında “madde”, diğer ekranında “ürün” ve bir başka ekranında “parça” olarak adlandırılması arayüzde tutarsızlığa yol açar. Arayüzdeki bu tutarsızlık, kargaşa ve engellemelere neden olacaktır. Üç kelimedede aynı şeyi temsil etmesine rağmen, aynı

anlamı taşıyan farklı kelimelerin kullanılması arayüzün kullanıcı için açık olmadığını gösterir. Böyle bir durumda kullanıcının kafası karışır ve ne yapacağını bilemez.

2.4.7 Grafikselle Kullanıcı Arayüz (GUI) Dizayn Prensipleri

Kullanıcı arayüz dizayn işlemlerini ve bu dizayn işlemlerinin oluşum süreçlerini göstermek için waterfall (şelale) modeli kullanılır. Waterfall (şelale) modeli, grafikselle kullanıcı arayüz'ünün dizayn ve gelişim aşamalarını göstermek için kullanılan bir model türüdür. Waterfall (şelale) modeli'nde ilk olarak sistemin ihtiyaçları belirlenir ve analiz edilir. İkinci olarak sistem mimarisinin ve yazılımının dizaynı yapılır. Daha sonra sistem oluşturulur ve kullanıcı arayüzünü oluşturan fonksiyonların kontrolü ve arayüz yazılım kodunun testi gibi işlemleri içeren birim testi uygulanır. Son olarak sistemin entegrasyonu yapılarak, yazılım kodunun kontrolü ve sistemi oluşturan fonksiyonların doğruluğunun onaylanması için gerekli olan sistem testi uygulanır. Aşağıdaki şekil, kullanıcı arayüz dizayn işlemlerini ve bu dizayn işlemlerinin oluşum süreçlerini belirten waterfall (şelale) modelini göstermektedir [57]:



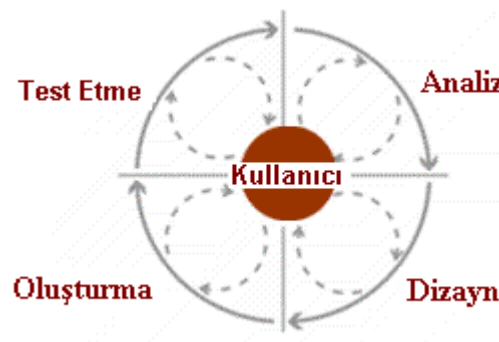
Şekil 47 : Waterfall (Şelale) Modeli

Arayüz gelişiminin en önemli evrelerinden biri, dizayn'dır. Dizayn aşaması, kullanıcı arayüzü'nün fonksiyonelliğini test etmek açısından önemlidir. Dizayn evresi arayüz platformunun kurulumundan, düzenlenmesinden ve görünümünden sorumludur. Arayüz dizaynını değerlendirmek kullanıcının bilgisine ve tecrübesine bağlıdır. İyi bir grafiksel arayüz dizaynı, kullanıcı açısından zamanı en aza indirir. Ayrıca acemi kullanıcılar için çok faydalar ve kolaylıklar sağlar. Dizayn evresinden önce arayüzün fiziksel görünümünü düzenlemek, grafiksel kullanıcı arayüzü'nün en birinci ve en kolay adımıdır. Dizayn evresinde oluşan problemleri çözmek, karmaşık sonuçların oluşumunu önlemek için kontrol ve test etme metodları kullanılır. Dizayn evresinin ana amacı, grafiksel kullanıcı arayüzünü geliştirmek, arayüzün görselliğini arttırmak ve kullanıcıya daha güzel bir arayüz sunabilmektir.

Sürekli tekrar eden arayüz dizayn işlemlerinin odağında kullanıcı vardır. Kullanıcı, arayüz dizaynı oluşturulmadan önce test etme özelliği ile ön plana çıkar. Grafiksel kullanıcı arayüzü dizayn edilirken sürekli tekrar eden dört evre vardır. Bu evreler şunlardır :

- Analiz (İhtiyaçların Belirlenmesi)
- Dizayn
- Oluşturma
- Test Etme

Kullanıcı arayüz dizayn işlemleri, döngüsel bir yapı içerisinde sürekli tekrar halindedir. Grafiksel kullanıcı arayüz dizaynı, kullanıcının istek ve beklentilerine göre oluşturulduğundan arayüz dizayn işlemlerinin merkezinde kullanıcı vardır. Aşağıdaki şekil kullanıcının, arayüz dizaynının ve kullanılabilirlik prensibinin merkezinde olduğunu ve arayüz dizayn işlemlerinin sürekli tekrar ettiğini göstermektedir :



Şekil 48 : Sürekli Tekrar Halindeki Dizayn İşlemleri

Bir arayüz dizaynı yapılmadan önce bazı evrelerden geçmesi gerekir. Bu evreler kullanıcıya yararlı, kullanımı kolay, hatasız bir arayüz meydana getirebilmek için mutlaka olması gereken işlemlerdir. Arayüz dizaynında ilk olarak, arayüz elemanları belirlenir. Daha sonra arayüzün gerçekleştirilmesi düşünülen fonksiyonları ve işlevleri saptanır. Bunlardan sonra arayüz ekranının düzeni ve fiziksel platformu meydana getirilir. Kullanıcı arayüz uygulaması oluşturulur. Son olarak, arayüzün kullanılabilirliğinin uygun olup olmadığına bakmak için kullanılabilirlik testi yapılır. Her yönüyle eksiksiz bir arayüz dizaynının yaratılması için yapılması gereken dizayn aşamaları ve tanımlamaları şöyledir [58] [59]:

Aşama	Tanımlama
İhtiyaçlar	Arayüz uygulamasının ihtiyaç ve gereksinimlerinin belirlenmesi
Yapısal Dizayn	Uygulama içinde belirlenen işin modelinin oluşturulması
Mantıksal Dizayn	Uygulamanın nasıl çalışacağını genel terimleri ile dizayn etmek
Fiziksel Dizayn	Uygulamanın nasıl çalışacağını genel terimleri ile oluşturmak
Oluşturma	Arayüz uygulamasının oluşturulması
Kullanılabilirlik Testi	Kullanıcı arayüzünün kullanılabilirliğinin test edilmesi

Tablo 4 : Kullanıcı Arayüz Dizayn Aşamaları ve Tanımlamaları

2.4.7.1 İhtiyaçlar Aşaması

Grafiksel kullanıcı arayüz (GUI) elemanlarını belirlemek için sistemin fonksiyonları tanımlanmalıdır. Sistem fonksiyonları, sistemin amacına göre oluşturulurlar. Kullanıcı arayüz dizaynı, sistemin içeriğine uygun olmalıdır. Arayüz dizayn işlemine başlamadan önce, uygulamanın ihtiyaç ve gereksinimleri belirlenmelidir. Başarılı bir ihtiyaçlar aşaması oluşturmak için şu adımlara dikkat edilmesi gerekir :

- Uygulamanın bakış açısını belirlemek.
- Uygulamadaki problemlerin belirlenmesi.
- Belirlenen ihtiyaç ve gereksinimleri incelemek.
- Uygulamayı bölümlere ayırmak.

2.4.7.1.1 Analiz

Kullanıcı arayüzü dizayn edilmeden önce iyi bir analiz yapılması gerekir. Ayrıntılı bir analiz, iki adımda incelenir : Birinci adımda, kullanıcı arayüz sistemini oluşturacak

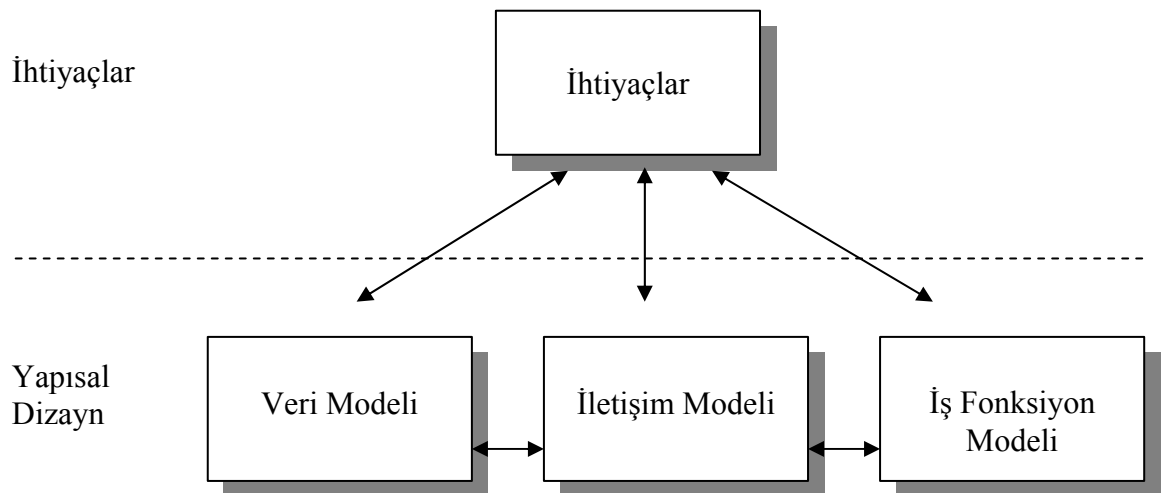
olan arayüz elemanlarının analizi yapılarak, arayüz elemanları gözden geçirilir. Hangi kullanıcı arayüz elemanlarının kullanılacağına karar verilir. Daha sonra arayüz elemanlarının hangi amaçla kullanılacağına analizi yapılır. İkinci adımda, arayüzü oluşturan sistemin ana fonksiyonları için analiz yapılır. Arayüz fonksiyonlarının görevlerini iyi analiz etmek gerekir. Çünkü fonksiyonlar, arayüzün işleyişini oluştururlar.

2.4.7.1.2 Arayüz Elemanlarının Dizaynı ve Oluşturulması

Bu aşamada, sistem için önemli olan arayüz elemanları dizayn edilir ve oluşturulur. Arayüze daha sonradan ilave edilecek arayüz elemanlarının sorun yaratıp yaratmayacağına karar verilir.

2.4.7.2 Yapısal Dizayn

Yapısal dizayn aşamasında arayüz uygulaması içinde belirlenen işin modeli oluşturulur. Yapısal dizayn üç bölümden oluşur : *Veri modeli*, veri kümelerini ve veriler arasındaki ilişkileri gösterir. *İş fonksiyon modeli*, uygulamanın iş kümelerini ve fonksiyon elemanlarını belirler. *İletişim modeli*, fonksiyon elemanları ile veri kümeleri arasındaki etkileşimi gösterir. İletişim modeli'nin bir diğer adı da, veri akış diyagramı'dır.



Şekil 49 : Yapısal Dizayn

2.4.7.3 Mantıksal Dizayn

Mantıksal dizayn aşamasında, uygulamanın nasıl çalışacağı genel terimleri ile dizayn edilir. Mantıksal dizayn aşaması, yapısal dizayn aşamasında modeli oluşturulan uygulamaların detayları ile ilgilenir. Arayüzün başlangıcı mantıksal dizayn aşamasında oluşturulur. Mantıksal dizayn aşamasında, uygulamada hangi olayların olacağı belirlenir. Daha sonra bu olay ve işlemler sistem fonksiyonlarına dönüşürler. Kullanıcı arayüzünün neyi yapabildiğini neyi yapamayacağı konusunda kararlar alınır. Uygulamaya uygun donanım biçimi belirlenir. İstek ve ihtiyaçların kullanıcı arayüzünde yapılabileceği kararlar alınır.

2.4.7.4 Fiziksel Dizayn

Fiziksel dizayn aşamasında, arayüzün fiziksel platformu belirlenir. Mantıksal dizaynın fiziksel platformda nasıl oluşturulacağı konusuna çözüm yolları aranır. Uygulamanın nasıl çalışacağı genel terimleri ile fiziksel dizayn aşamasında oluşturulur. Mantıksal dizaynın kesin görünümü oluşturulduktan sonra fiziksel dizayna geçilir.

2.4.7.5 Oluşturma

Arayüz dizaynının oluşum aşaması; bütün arayüz elemanlarının, nesnelere ve fonksiyonlarının görevlerini yerine getirmesi ve birbirleriyle bir etkileşim içerisinde olup, bir bütünlüğün sağlanmasından sorumludur. Bunun içinde kullanıcı arayüzünü oluşturan kişi ya da kişilerin arayüz idaresinden çok iyi anlaması ve kontrolü sürekli elinde bulundurması gerekir. Bir kullanıcı arayüzünü oluşturmak için iki şeye çok önem vermek gerekir. Birincisi, grafiksel kullanıcı arayüzünü oluşturmak için esaslı bir bilgiye ve büyük bir tecrübeye gereksinim vardır. İkincisi ise, arayüz dizayn mimarisini çok iyi anlamış olmak gerekir. Oluşturma aşamasında, arayüz uygulamasının meydana getirilmesi işlemi yapılır. Uygulama programı, oluşturma aşamasında inşa edilir. Uygulamanın oluşturulması demek dizayn işleminin bittiği anlamına gelmez. Oluşturma evresi, arayüz elemanları ile program kodu arasındaki bağdan sorumludur. Arayüzün problem çıkaran bazı bölümleri olabilir ya da oluşturulan arayüzün bazı kısımlarını değiştirmek gerekebilir. Bu yüzden arayüz test edilmeden oluşturulmuş sayılmaz. Oluşturma aşamasından sonra kullanılabilirlik testinde arayüze uygulanması gerekir. Arayüz işte o zaman daha iyi sonuçlar verecektir.

2.4.7.5.1 Programlama Dili

Bu aşamada, kullanıcı arayüz sisteminde kullanılacak olan programlama diline karar verilir. Bu seçilen programlama dili, platform bağımsızlığı için olanak tanınmalıdır. Bir bilgisayar programında dizayn edilen bir arayüz, farklı türdeki bilgisayarlarda farklı görünümlem vermemeli ve arayüz görünüm problemleri ortaya çıkarmamalıdır. Yazılımdaki karmaşıklığa çare bulmak, arayüz dizaynının sağlıklı bir şekilde olmasına zemin hazırlayacaktır. Sistemi oluşturacak olan program kodunun analiz ve dizaynını yapacak olan programcının, daha önce çok fazla programlama tecrübesinin olmasına dikkat edilmelidir. Program kodunu yazmaya başlamadan önce arayüzü dizayn eden kişi ya da kişilerin programın hangi amaçla tasarlandığını çok iyi bilmeleri gerekir.

2.4.7.6 Kullanışlılık Testi

Kullanışlılık testi, kullanıcı arayüz dizaynının geçerli olması için kullanılan bir tekniktir. Kullanışlılık testi aşaması, kullanıcı arayüzünün kullanılabilirliğini test edilmesi amacı ile yapılır. Arayüzün, hatalarından ve problemlerinden arınmasını sağlar. Arayüzün kullanılabilirliğini etkili bir hale getirmek için kullanılan son derece faydalı bir yöntemdir. Grafikselle kullanıcı arayüzünün gelişimini sağlamak için gerekli olan en önemli adımlardan biridir. Kullanışlılık testi, bir arayüz dizaynında hangi arayüz elemanının ne işe yaradığını ve nerde kullanılırsa daha faydalı olabileceğini söyler. Bu yöntem, arayüz elemanlarının doğru kullanılıp kullanılmadığını test eder. Arayüz elemanları arasındaki kullanım ve içerik karmaşıklığını önlemeye çalışır. Kullanışlılık testi, arayüz elemanları nasıl ve hangi amaçla kullanılırsa en doğru olur sorusuna cevap arar. Kullanıcılar üzerinde sürekli testler yapar. Böylece, arayüz hatalarını ve problemlerini bularak en doğruya ulaşmaya çalışır. Yazılım şirketlerinin ve çeşitli firmaların iş hayatındaki verimliliğini ve olgunluğunu artırır. Kullanışlılık testi kısaca, arayüzü hatalardan arındırma ve geçerli kılma işlemidir.

Kullanıcı arayüzünde test etme, hatalardan korunmak, kullanıcıya kolaylık sağlamak, arayüzü doğrulamak, geliştirmek ve oluşturmak için yapılan bir aşamadır. Arayüzün görünümlerini arttırmak, arayüz dizayn işleminin maliyetini azaltmak, zaman kaybını önlemek, arayüz elemanlarının çalışma durumunu kontrol etmek ve arayüz programının olası hatalarını bulmak için test işlemi zorunludur. Arayüzde “doğruluk” kavramı, ancak test işlemi ile sağlanabilir. Test işleminin en önemli yararlarından biri,

arayüzdeki fonksiyonların görevlerini yerine getirip getirmediğini görmek ve bu fonksiyonların kullanıcı beklentilerine ne kadar cevap verdiğini belirlemektir.

Kullanıcıyı yormayan, dikkati dağıtmayan grafiksel kullanıcı arayüzü; en iyi dizayn edilmiş arayüzdür. İyi bir arayüz dizaynı oluşturmak için kullanılan bir takım prensipler vardır. Yalnız bu prensipler hiçbir zaman körü körüne uyulan mutlak kurallar değildir. Bütün çaba sadece arayüz ekranını daha düzgün bir hale sokmak içindir. Arayüz dizaynında kullanılan bu prensipler arayüzün fonksiyonelliğini, performansını ve kullanılabilirliğini artırır. Kullanıcı arayüz dizayn prensipleri, arayüzün yapısını ve estetik görünümünü oluşturmak için geliştirilir. Grafiksel kullanıcı arayüzü dizayn edilirken göz önünde bulundurulması gereken ve arayüzü dizayn eden kişilere ya da kurumlara yol gösterme özelliği taşıyan bu prensipler şunlardır [60]:

1. Arayüz tasarımında kullanıcılara ve onların isteklerine odaklanılmalıdır. Teknoloji bundan sonra gelmelidir.
2. Arayüz dizaynını yapmadan önce geliştirilecek olan arayüz tasarımının fonksiyonları düşünülmelidir. Arayüz dizaynı için düşünülen fonksiyonlar için bir model oluşturulmalıdır.
3. Arayüz tasarımı kullanıcı isteklerine göre oluşturulmalıdır.
4. Arayüz kullanıcıların anlayabileceği dilde ve kelimelerden dizayn edilmelidir. Kullanılan terimler kullanıcılar için açık ve net olmalıdır.
5. Kullanıcıların istekleri birbirine karıştırılmamalı, arayüz kullanılırken kullanıcıya ekstra problemler ve ekstra yük yüklenmemelidir.
6. GUI'de tutarlılık ve aynı ilkelere uyma söz konusu olmalıdır.
7. Arayüz tasarımında yazımsal ve grafiksel belirsizliklerden arınılmalıdır.
8. Arayüz tasarımındaki riskler en aza indirgenmelidir.
9. Görüntülenecek arayüz dizaynı çok dikkatlice yapılmalıdır. Bilgi kullanıcıya ulaştırılmalıdır. Arayüz ekranı kullanıcıya ait olmalı ve kullanıcı için oluşturulmalıdır.
10. Arayüz dizaynı büyük bir heveslilik ve isteklilik ile kullanıcı gereksinimlerini gidermek amacı ile oluşturulmalıdır.
11. Dizayn edilmiş arayüz bazı kullanıcılar üzerinde denenmeli, daha sonra uygunluğuna karar verilmelidir. Arayüz kullanıcıları üzerinde yapılan test sonuçları arayüzü geliştirenler için iyi bir tecrübe olacaktır. Test, problemleri

gidermek ve kullanıcı arayüzünün son haline karar vermek için sürekli yapılmalıdır.

2.4.8 Grafikselle Kullanıcı Arayüzü (GUI) Hataları

Bu tez çalışmasında MATLAB ve MathCAD yazılım programlarının grafikselle kullanıcı arayüz hataları ele alınıp, incelenecektir. Grafikselle kullanıcı arayüzü'nde (GUI) çok sıkça yapılan ve kullanıcının hoşuna gitmeyen bu hataları dört grupta toplayabiliriz. Bu hatalar ve içerikleri şunlardır [60]:

1. Grafikselle kullanıcı arayüz (GUI) elemanlarında yapılan hatalar
2. Düzen ve görünüşteki hatalar
3. Metinsel hatalar
4. Etkileşim hataları

2.4.8.1 Grafikselle Kullanıcı Arayüz Elemanlarında Yapılan Hatalar

1. Menülerde yapılan hatalar
 - 1.1. Aynı amaçlı olan ve aynı fonksiyonları içeren menüleri oluşturmak
 - 1.2. Farklı menülerde aynı alt menüleri kullanmak
 - 1.3. Hem menü hem de alt menü isminin aynı olması
2. Menülere klavyeden atanan kısayol tuşlarında yapılan hatalar
3. Uygulama pencerelerinde yapılan hatalar
4. Gereksiz yere çerçeve (frame) kullanılması
5. Denetim kutusu (check box) ve seçenek düğmesi'nin (optionbutton) işlevlerini birbirine karıştırmak. Denetim kutusu'nun seçenek düğmesi gibi kullanılması
6. Arayüz elemanlarının fonksiyonlarının yanlış ve farklı kullanılması (Örneğin; aynı anda iki farklı şeyi birden seçmek için denetim kutusu yerine seçenek düğmesi'nin kullanılması)
7. Metin alanları'nı (textfields) yanlış kullanma
8. Sekme (tab) kullanımında yapılan hatalar

2.4.8.2 Kullanıcı Arayüzünün Düzen ve Görünüşünde Yapılan Hatalar

1. Farklı amaçlı düğme (button) türlerini birbirine karıştırmak
2. Arayüz elemanlarını iç içe sokarak arayüzün düzenini bozmak
3. GUI elemanlarının düzen bozuklukları
4. Etiket'lerin (label) dizilme hataları. Etiketlerin farklı hizalarda bulunmalarının görünüşü bozması ve düzen bozukluğu yaratması
5. Yazı tipinde yapılan hatalar. Aynı arayüz içerisinde farklı yazı tiplerinin olması (Örneğin; Times New Roman yazı tipi ile Arial yazı tipinin aynı anda kullanılması)
6. Yazı tiplerinin okunamayacak derecede ufak olması
7. Yazı büyüklüklerinin farklı olmasının arayüzün görünüşünü bozması
8. Farklı metin renklerinin kullanılması sonucu arayüzdeki görünümün bozulması
9. Arayüzdeki arka plan (background) renginin, görünümü bozması ve okumayı zorlaştırması
10. Çerçeve'nin (frame) çerçeve içerisinde olması
11. Grafik dizaynında yapılan hatalar

2.4.8.3 Metinsel Hatalar

1. Uygun ve açık olmayan terimlerin kullanılması
2. Farklı terimlerin aynı kavramlar için kullanılması, aynı terimlerin farklı kavramlar için kullanılması (eşanlamlı kelimeler)
3. Kullanılan dillerin farklı olması (Örneğin; yazının bir bölümü Türkçe olurken diğer bölümünün İngilizce olması)
4. Menüdeki isimlerin aynı anlamlı olması
5. Farklı pencerelerde aynı başlığı kullanmak
6. Pencere başlıklarının farklı olması

7. Etiket'ten (label) sonra (:) işaretinin kullanılmaması
8. "...” işaretinin yanlış kullanılması
9. Tooltip'lerin (Düğme'nin ya da ikon'un üzerine gelindiği zaman düğmenin ya da ikon'un ne işe yaradığını gösteren açıklayıcı bilgi) kullanılmaması

2.4.8.4 Etkileşim Hataları

1. Detaylarla kullanıcıyı yormak
2. Arayüzün yapısının karmaşık olması
3. Bilginin çok kötü ve fakirce sunulması
4. Kullanıcılara arayüzdeki ifade ve işlemleri açıklayıcı bilginin verilmemesi
5. Veri sonuçlarının liste biçiminde sunulmasının yarattığı belirsizlik
6. Arayüzde gereksiz yol ve yöntemlerin bulunması
7. Birbirine benzemeyen kullanıcı arayüzlerinin aynı fonksiyonları içermesi
8. Arayüz dizaynının kullanıcıların kolaylığı için yapılmamış olması
9. Arayüzde kullanılan uyarı mesajlarında yapılan hatalar
 - 9.1. Mesajların aynı anlamlı olması
 - 9.2. Mesajlarda sembollerin yanlış kullanılması. Örneğin; uyarı mesajında “!” işareti yerine “?” işaretinin kullanılması
 - 9.3. Mesajların kullanıcıyı yanıltması ve yanlış yönlendirmesi
 - 9.4. Mesajdaki seçimlerin açık ve net olmaması
 - 9.5. Mesajın kullanıcıya çok fazla seçenek sunmaması
 - 9.6. Mesajda “OK” ve “Cancel” düğmelerin aynı şeyi yapması
10. Diyalog Kutusu'nda (Dialog Box) derin bir hiyerarşinin olması

Sonuç olarak tezin bu bölümünde; MATLAB, MathCAD ve Grafiksel Kullanıcı Arayüzü (Graphical User Interface - GUI) ile ilgili olarak şu konular ele alınmış ve incelenmiştir :

Tezde MATLAB programı ile ilgili genel olarak şu konulara yer verilmiştir :

- MATLAB'in ne işe yaradığı
- MATLAB'in yapısı
- MATLAB'in nasıl bir program olduğu
- MATLAB'in elemanları ve bileşenleri
- MATLAB'in karakteristik özellikleri
- MATLAB'in kullanım alanları
- MATLAB'te programlama kavramı
- MATLAB'te GUI oluşturma
- MATLAB anket araştırma sonuçları

Tezde MathCAD programı ile ilgili genel olarak şu konulara yer verilmiştir :

- MathCAD programının içeriği
- MathCAD programının önemi
- MathCAD programının genel özellikleri
- MathCAD'in kullanım alanları
- MathCAD programının temel yapısı
- MathCAD'te programlama geliştirmek
- MathCAD programının tercih edilme sebepleri
- MathCAD kullanıcısı olmanın avantajları
- MathCAD anket araştırma sonuçları

Tezde grafiksel kullanıcı arayüzü (GUI) hakkında genel olarak şu konulara yer verilmiştir :

- GUI hakkında genel bilgi
- GUI dizaynının amacı
- Kullanıcı arayüz dizaynının yapısı
- GUI dizaynının önemi
- GUI elemanları
- Kullanıcı arayüz dizaynının temel iskeleti ve mimarisi

- GUI hataları
- GUI dizayn prensipleri
- GUI dizayn teknikleri

3. METODOLOJİ

Tezin bu bölümünde, hesaplama programlarından MATLAB ve MathCAD'i değerlendirmek, programların özelliklerini görmek ve kalitelerinin ölçümü için kullanılan, kaliteyi etkileyen faktörler ile kalite ölçüm kriterlerini içeren kalite ölçüm kontrol listesine ve MATLAB ile MathCAD programlarındaki grafiksel kullanıcı arayüz hata türlerini göstermek için oluşturulan GUI hataları ölçüm kontrol listesine yer verilmiştir. Bu listeler, literatür bilgileri eşliğinde oluşturulmuş ve tanımlanmıştır. Bu tez kapsamında MATLAB ve MathCAD programları, adı geçen kalite ölçüm kontrol listeleri eşliğinde içerik analizi yöntemiyle incelenmiştir. Araştırmacı, ilgili listeleri literatür bilgileri eşliğinde tanımlanan açıklamalar ekseninde bireysel olarak değerlendirmiştir. Daha sonra bu değerlendirmeler, tez danışmanı nezaretinde kontrol ve revize edilmiştir.

3.1. Kalite Ölçüm Kontrol Listesi

Bir programın özelliklerini daha iyi anlamak için "ölçüm" kriteri kullanılır. Oluşturulan ürünün ya da sistemin kalitesini değerlendirmek, ürünün nitelikli olup olmadığını göstermek için yararlı olacaktır. Bir yazılım programının temel ölçüm elemanı "kalitedir". Teknik ölçümler, bir yazılım programının dizayn edilmesi ve oluşturulması işlemlerinde yazılım mühendislerine yardım ederler. Yüksek kaliteli yazılım programları oluşturmak için teknik ölçüm kriterleri kullanılır.

Bir programda kalite, yazılımın isteklerinin ve beklentilerinin temelini oluşturur. Kalite açıklamasını en iyi anlatan üç tanım şudur :

- Kalite, bir yazılım programını geliştirme standartıdır.
- Kalite, bir yazılım programının fonksiyonel gereksinimlerinin temelini oluşturan etkendir.
- Kalite, bir yazılım programının performans gereksinimlerinin özünü oluşturan faktördür.

Bir programdaki kalite ölçüm kontrol listesini etkileyen iki önemli etken vardır. Bu etkenler şunlardır [61]:

1. Kaliteyi Etkileyen Faktörler
2. Kalite Ölçüm Kriterleri

3.1.1 Kaliteyi Etkileyen Faktörler

Bir arayüzde yazılım programının kalitesini direkt ya da dolaylı yoldan etkileyen çeşitli faktörler vardır. Bu faktörler şunlardır :

1. Uygunluk : Programın kendi açıklamalarını ve kullanıcının istek ve beklentilerini yerine getirmesi.
2. Güvenilirlik : Programın daha önceden tasarlanan fonksiyonları yerine getirmesi.
3. Yeterlilik : Oluşturulan yazılım kodunun ve uygulamaların program tarafından yerine getirilmesi.
4. Doğruluk : Yazılıma ya da veriye giriş için yetkilendirilen kişilerin kontrol edilmesi.
5. Kullanışlılık : Programın öğrenilmesi, çalıştırılması, veri girişinin (input) hazırlanması ve veri çıkışının (output) yorumlanması işlemlerinin yapılması.
6. Tutulabilirlik : Bir programdaki hataları belirlemek ve hataları elinde tutma gayreti.
7. Esneklik : Bir programda istenildiği zaman değişiklikler yapılabilmesi.
8. Test Edilebilirlik : Bir programın fonksiyonlarını yerine getirip getirmediğini görmek için test edilmesi.
9. Taşınabilirlik : Programı bir yazılım ya da donanım sisteminden diğerine transfer etmek için uğraşmak.
10. Tekrar Kullanılabilirlik : Bir programın belirli bölümlerinin ya da fonksiyonlarının diğer uygulamalarda tekrar kullanılması.
11. Birleştiricilik : Bir sistemi diğerine birleştirme gayreti içinde olunması.

3.1.2 Kalite Ölçüm Kriterleri

Kalite faktörlerini geliştirmek için ölçüm kriterleri kullanılır. Kullanılan kalite ölçüm kriterleri şunlardır :

1. Denetleme : Arayüz dizayn standartlarının kolaylıkla uygulanması için kontrol edilmesi.
2. Kesinlik : Bilgisayar programında işlemlerin ve kontrolün açık olması.
3. İletişim Ortaklığı : Standart arayüzlerin, protokollerin ve bant genişliğinin kullanım derecesi.
4. Bütünlük : Programın, fonksiyonlardan kaç tanesini başarıyla yerine getirdiğinin değerlendirilmesi.
5. Karmaşıklık : Arayüzdeki uygulamaların ve işlemlerin anlaşılabilirliğinin ölçülmesi.

6. Özlülük : Programın kod satırlarındaki terimlerin anlaşılır olması.
7. Tutarlılık : Yazılım geliştirme projesinin her bölümünde birbirine benzer dizayn ve dokümantasyon tekniklerinin kullanımı.
8. Veri Ortaklığı : Programın her bölümünde standart veri yapılarının ve veri türlerinin kullanımı.
9. Hata Toleransı : Programda hata ile karşılaşıldığında programın zarara uğraması durumu.
10. Uygulamayı Çalıştırma Verimliliği : Programın çalışma zamanındaki performansı.
11. Genişletilebilirlik : Bir programın mimari, veri ya da yapısal dizayn olarak büyütülmesi özelliğinin olması.
12. Genellik : Program elemanlarındaki uygulamaların genişliği.
13. Donanım Bağımsızlığı : Programın donanımının farklı yazılımları kullanabilme özelliğinin olması.
14. Donanım Araçları : Program monitörünün kendi uygulama ve tanımlamalarındaki hataları ortaya çıkarması.
15. Modülerlik : Program elemanlarının fonksiyonel bağımsızlığı.
16. Çalışma Durumu : Programın uygulamalarının kolay olması.
17. Güvenlik : Bilgisayarın program ve veriyi kontrol etme ya da koruma özelliğinin olması.
18. Dokümantasyonunun Olması : Programın kendi kaynak kodunu anlatan anlamlı dokümantasyonunun olması.
19. Basitlik : Programın kullanıcı için kolayca anlaşılır olması.
20. Sistem Bağımsızlığı : Programın standart olmayan programlama dili özelliklerinin, işletim sistemi karakteristiğinin ve diğer çevresel baskıların bağımsız olması.
21. İzlenebilirlik : Programın dizayn sunumunun ya da ana program elemanlarının, istek ve ihtiyaçları karşılayabiliyor olması.
22. Kullanıcıyı Eğitme : Yazılımın yeni kullanıcılara programı kullanmaları konusunda yardım etmesi.

Aşağıdaki tablo, MATLAB ve MathCAD programlarının kalitelerinin ölçümü için kullanılan ve kaliteyi etkileyen faktörler ile kalite ölçüm kriterlerinden oluşan kalite ölçüm kontrol listesini göstermektedir :

Kalite Ölçüm Kriterleri Kaliteyi Etkileyen Faktörler	Kalite Ölçüm Kriterleri										
	Uygunluk	Güvenilirlik	Yeterlilik	Doğruluk	Tutulabilirlik	Esneklik	Test Edilebilirlik	Taşınabilirlik	Tekrar Kullanılabilirlik	Birleştiricilik	Kullanışlılık
Denetleme				X			X				
Kesinlik		X									
İletişim Ortaklığı										X	
Bütünlük	X										
Karmaşıklık		X				X	X				
Özlülük			X		X	X					
Tutarlılık	X	X			X	X					
Veri Ortaklığı										X	
Hata Toleransı		X									
Uygulamayı Çalıştırma Verimliliği			X								
Genişletilebilirlik						X					
Genellik						X		X	X	X	
Donanım Bağımsızlığı								X	X		
Donanım Araçları				X	X		X				
Modülerlik		X			X	X	X	X	X	X	
Çalışma Durumu			X								X
Güvenlik				X							
Dokümantasyonunun Olması					X	X	X	X	X		
Basitlik		X			X	X	X				
Sistem Bağımsızlığı								X	X		
İzlenebilirlik	X										
Kullanıcıyı Eğitme											X

Tablo 5 : Kalite Ölçüm Kontrol Listesi

3.2. GUI Hataları Ölçüm Kontrol Listesi

GUI hataları ölçüm kontrol listesi, MATLAB ile MathCAD programlarındaki grafiksel kullanıcı arayüz hata türlerini göstermek için oluşturulmuştur. Bu ölçüm listesi, MATLAB ve MathCAD programlarında hangi GUI hatalarının yapıldığının görülmesi açısından faydalı bir tablodur. Aşağıdaki tablo, MATLAB ve MathCAD programlarının grafiksel kullanıcı arayüz hatalarını göstermek için oluşturulan GUI hataları ölçüm kontrol listesini göstermektedir :

		MATLAB	MathCAD
Grafiksel Kullanıcı Arayüz (GUI) Hata Türleri	1. Grafiksel Kullanıcı Arayüz (GUI) Elemanlarında Yapılan Hatalar	1. Menülerde yapılan hatalar	
		2. Kısayol tuşlarında yapılan hatalar	
		3. Denetim kutusu'nun seçenek düğmesi gibi kullanılması	
		4. Arayüz elemanlarının fonksiyonlarının yanlış ve farklı kullanılması	
		5. Sekme (tab) kullanımında yapılan hatalar	
	2. Düzen ve Görünüşteki Hatalar	1. GUI elemanlarının düzen bozuklukları	
		2. Arayüz elemanlarının iç içe sokulması	
		3. Çerçeve'nin (frame) çerçeve içerisinde olması	
		4. Etiket'lerin (label) dizilme hataları	
		5. Yazı tipinde yapılan hatalar	
		6. Yazı tiplerinin ufak olması	
		7. Metin renklerinin farklı olması	
		8. Arayüz arka plan renginin, görünümü bozması	
	3. Metinsel Hatalar	1. Uygun ve açık olmayan terimlerin kullanılması	
		2. Pencere başlıklarının farklı olması	
		3. Farklı terimlerin aynı kavramlar için, aynı terimlerin farklı kavramlar için kullanılması	
		4. İki farklı pencerede aynı başlığı kullanmak	
	4. Etkileşim Hataları	1. Arayüzün yapısının karmaşık olması	
		2. Bilginin çok kötü ve fakirce sunulması	
		3. Veri sonuçlarının liste biçiminde sunulmasının yarattığı belirsizlik	
		4. Diyalog kutusu'nda derin bir hiyerarşinin olması	
		5. Kullanıcılara ifade ve işlemleri açıklayıcı bilginin verilmemesi	
		6. Uyarı mesajlarında yapılan hatalar	

Tablo 6 : GUI Hataları Ölçüm Kontrol Listesi

4. SONUÇLAR

Bu bölümde, tez çalışması ile ilgili ortaya çıkan sonuçlar altı bölüme ayrılarak incelenmiştir :

1. MATLAB kalite ölçüm kontrol listesi sonuçları
2. MathCAD kalite ölçüm kontrol listesi sonuçları
3. MATLAB & MathCAD GUI hataları ölçüm kontrol listesi sonuçları
4. MATLAB programındaki GUI hataları & hataların doğruları
5. MathCAD programındaki GUI hataları & hataların doğruları
6. Genel değerlendirme

Tezin bu bölümünde ilk olarak, MATLAB programının kalitesini ölçmek için kullanılan, MATLAB kalite ölçüm kontrol listesinin sonuçlarına ve bu tablo sonucunda ortaya çıkan sonuçlar sonunda bu program hakkındaki yorumlara yer verilmiştir. Daha sonra, MathCAD programında kaliteyi değerlendirmek için oluşturulan MathCAD kalite ölçüm kontrol listesinin sonuçlarına yer verilmiştir. Bunların dışında tezin bu bölümünde, MATLAB ve MathCAD programlarında hangi grafiksel kullanıcı arayüz (GUI) hata türlerinin olduğunu göstermek için oluşturulan MATLAB & MathCAD GUI hataları ölçüm kontrol listesi sonuçlarına yer verilmiştir. Bu bölümün en sonunda ise; MATLAB ile MathCAD programlarında bulunan grafiksel kullanıcı arayüz (GUI) hatalarının incelenmesine ve bu hataların doğrularının nasıl olması gerektiğine yer verilmiştir.

4.1. MATLAB Kalite Ölçüm Kontrol Listesi Sonuçları

MATLAB kalite ölçüm kontrol listesi, MATLAB programının kalitesinin değerlendirilip, yorumlanması için kullanılan ve kullanıcıların MATLAB programı hakkında bilgi sahibi olmalarını sağlayan bir tablodur. Kalite ölçüm kontrol listesi doldurulurken literatürde sunulan bilgiler, yapılan tanımlamalar dahilinde işlemler yapılmıştır. Araştırmacının bireysel olarak tamamlamış olduğu değerlendirme formları, tez danışmanı nezaretinde yeniden değerlendirilmiş ve kontrol edilmiştir. Bu bağlamda, MATLAB programının kalitesinin ölçümü için kullanılan kalite ölçüm kontrol listesine göre ortaya çıkan sonuçlar şunlardır :

Kalite Ölçüm Kriterleri Kaliteyi Etkileyen Faktörler	Kalite Ölçüm Kriterleri										
	Uygunluk	Güvenilirlik	Yeterlilik	Doğruluk	Tutulabilirlik	Esneklik	Test Edilebilirlik	Taşınabilirlik	Tekrar Kullanılabilirlik	Birleştiricilik	Kullanışlılık
Denetleme	X		X			X					
Kesinlik		X		X							X
İletişim Ortaklığı		X		X			X			X	
Bütünlük	X	X		X							
Karmaşıklık		X				X	X				
Özlülük			X		X	X					
Tutarlılık	X	X			X	X					
Veri Ortaklığı									X	X	
Hata Toleransı		X				X					
Uygulamayı Çalıştırma Verimliliği			X								
Genişletilebilirlik						X					
Genellik	X		X			X		X	X	X	
Donanım Bağımsızlığı	X			X				X	X		
Donanım Araçları	X	X		X	X		X				
Modülerlik		X			X	X	X	X	X	X	
Çalışma Durumu	X	X	X								
Güvenlik		X		X							X
Dokümantasyonunun Olması					X	X	X	X	X		
Basitlik		X			X	X	X				
Sistem Bağımsızlığı	X			X				X	X		
İzlenebilirlik	X	X	X								
Kullanıcıyı Eğitme		X	X	X							X

Tablo 7 : MATLAB Kalite Ölçüm Kontrol Listesi

MATLAB kalite ölçüm kontrol listesi, esas olan kalite ölçüm kontrol listesinin olması gereken sonuçlarından, benzer yanlarının olmasının yanında farklı sonuçlar da vermektedir. Bu tablo MATLAB'in, kalitesi çok üst düzey bir program olduğunu göstermektedir. MATLAB kalite ölçüm kontrol listesinin, olması gereken kalite ölçüm kontrol listesinden farklı olarak kullanıcıyı memnun edecek yönde pozitif bir çok değişik özellikleri vardır. Sonuç olarak, MATLAB kalite ölçüm kontrol listesine göre; MATLAB programı, son derece güvenilir, yeterliliği üst düzeyde olan,

kullanışlılık açısından inanılmaz kolaylıklar sağlayan, uygulamaları çalıştırma verimliliği bakımından son derece başarılı, kendi kullanıcılarını eğitme konusunda çeşitli özelliklere sahip, kullanıcı dostu bir yazılımdır.

4.2. MathCAD Kalite Ölçüm Kontrol Listesi Sonuçları

MathCAD programının kalitesini ölçmek ve programın çeşitli özelliklerini değerlendirmek için kullanılan kalite ölçüm kontrol listesine göre ortaya çıkan sonuçlar şunlardır :

Kaliteyi Etkileyen Faktörler	Kalite Ölçüm Kriterleri										
	Uygunluk	Güvenilirlik	Yeterlilik	Doğruluk	Tutulabilirlik	Esneklik	Test Edilebilirlik	Taşınabilirlik	Tekrar Kullanılabilirlik	Birleştiricilik	Kullanışlılık
Denetleme		X	X			X					
Kesinlik		X		X							X
İletişim Ortaklığı	X	X		X			X			X	
Bütünlük		X	X								
Karmaşıklık		X				X	X				
Özlülük	X		X			X					
Tutarlılık	X	X	X		X	X					
Veri Ortaklığı			X						X	X	
Hata Toleransı		X				X					
Uygulamayı Çalıştırma Verimliliği			X								
Genişletilebilirlik	X					X					
Genellik		X	X			X		X	X	X	
Donanım Bağımsızlığı	X			X				X	X		
Donanım Araçları	X	X	X		X		X				
Modülerlik		X			X	X	X	X	X	X	
Çalışma Durumu	X	X	X			X					
Güvenlik		X	X	X							X
Dokümantasyonunun Olması					X	X	X	X	X		
Basitlik	X				X	X	X				
Sistem Bağımsızlığı			X	X				X	X		
İzlenebilirlik		X	X								
Kullanıcıyı Eğitme	X	X	X	X							X

Tablo 8 : MathCAD Kalite Ölçüm Kontrol Listesi

MathCAD kalite ölçüm kontrol listesi sonuçlarına göre : MathCAD programı, kaliteyi ön planda tutan, gelişim standartı yüksek bir yazılımdır. MathCAD kalite ölçüm kontrol listesi, standart kalite ölçüm kontrol listesi ile kıyaslandığında; kullanıcı açısından faydalı sonuçlar vermekte ve MathCAD'in kaliteyi temel prensip olarak ilke edindiğini göstermektedir. Tablo, olması gereken kalite ölçüm kontrol listesi sonuçları ile benzer ve ortak özellikler taşımaktadır. MathCAD kalite ölçüm kontrol listesine göre, MathCAD genel olarak; üst düzey bir çalışma performansına sahip, kullanıcıya hata toleransı sağlayan, güçlü bir yapısı olan, güvenlik konusunda yeterli verimliliğe sahip olan, karmaşıklığı az, fazla kafa karıştırmayan bir içeriği olan, kullanımı basit bir yazılımdır.

4.3. MATLAB & MathCAD GUI Hataları Ölçüm Kontrol Listesi Sonuçları

MATLAB & MathCAD GUI hataları ölçüm kontrol listesi, MATLAB ve MathCAD programlarında dört grafiksel kullanıcı arayüz (GUI) hata türlerinden hangilerinin bu programlarda olduğunu göstermek ve incelemek için oluşturulmuştur. Bu tablonun oluşturulmasındaki bir diğer amaç; grafiksel kullanıcı arayüz (GUI) hata türleri içerisinde en fazla hangilerinin MATLAB ve MathCAD programlarında bulunduğunu göstermektir. MATLAB ve MathCAD programlarında hangi GUI hatalarının yapıldığını incelemek ve bu programlardaki grafiksel kullanıcı arayüz hata türlerini göstermek için oluşturulan MATLAB & MathCAD GUI hataları ölçüm kontrol listesine göre şu sonuçlar ortaya çıkmıştır :

		MATLAB	MathCAD	
Grafiksel Kullanıcı Arayüz (GUI) Hata Türleri	1. Grafiksel Kullanıcı Arayüz (GUI) Elemanlarında Yapılan Hatalar	1. Menülerde yapılan hatalar	√	√
		2. Kısayol tuşlarında yapılan hatalar	√	√
		3. Denetim kutusu'nun seçenek düğmesi gibi kullanılması	√	
		4. Arayüz elemanlarının fonksiyonlarının yanlış ve farklı kullanılması	√	√
		5. Sekme (tab) kullanımında yapılan hatalar		√
	2. Düzen ve Görünüşteki Hatalar	1. GUI elemanlarının düzen bozuklukları	√	√
		2. Arayüz elemanlarının iç içe sokulması	√	√
		3. Çerçeve'nin (frame) çerçeve içerisinde olması	√	
		4. Etiket'lerin (label) dizilme hataları	√	√
		5. Yazı tipinde yapılan hatalar	√	
		6. Yazı tiplerinin ufak olması	√	√
		7. Metin renklerinin farklı olması	√	
		8. Arayüz arka plan renginin, görünümü bozması	√	√
	3. Metinsel Hatalar	1. Uygun ve açık olmayan terimlerin kullanılması	√	
		2. Pencere başlıklarının farklı olması	√	√
		3. Farklı terimlerin aynı kavramlar için, aynı terimlerin farklı kavramlar için kullanılması	√	
		4. İki farklı pencerede aynı başlığı kullanmak		√
	4. Etkileşim Hataları	1. Arayüzün yapısının karmaşık olması	√	
		2. Bilginin çok kötü ve fakirce sunulması	√	√
		3. Veri sonuçlarının liste biçiminde sunulmasının yarattığı belirsizlik	√	√
4. Diyalog kutusu'nda derin bir hiyerarşinin olması			√	
5. Kullanıcılara ifade ve işlemleri açıklayıcı bilginin verilmemesi		√	√	
6. Uyarı mesajlarında yapılan hatalar		√	√	

Tablo 9 : MATLAB & MathCAD GUI Hataları Ölçüm Kontrol Listesi

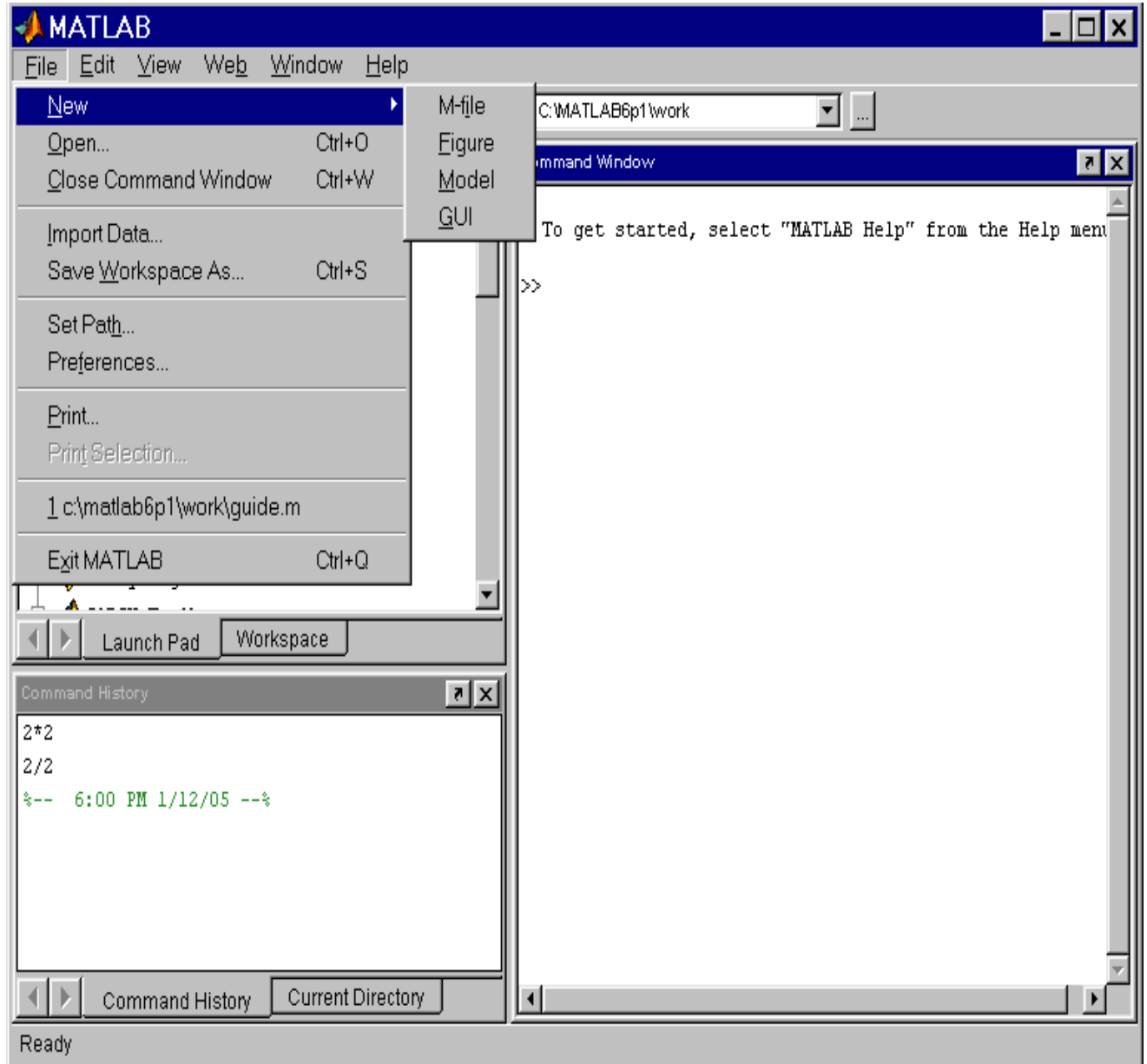
MATLAB & MathCAD GUI hataları ölçüm kontrol listesinden ortaya çıkan sonuçlara göre şu bilgiler elde edilmiştir : MATLAB programında, dört grafiksel kullanıcı arayüz hata türünün hepsi bulunmaktadır. MATLAB'te en çok, düzen ve görünüm hataları görülmüştür. MATLAB programı içerisinde, dört arayüz hata türünden en az metinsel hatalara rastlanmıştır. GUI hataları ölçüm kontrol listesinden çıkan en önemli sonuçlardan biri; MathCAD programının da aynen MATLAB programı gibi, grafiksel kullanıcı arayüz hata türlerinde benzerlik göstermesidir. MathCAD programında da, MATLAB'te olduğu gibi dört kullanıcı arayüz hata türünün hepsine rastlanmıştır. MathCAD'te arayüz hata türleri içerisinde en fazla düzen ve görünüm hatalarının olduğu görülmüştür. MathCAD programında, dört arayüz hata türünden en az metinsel hatalara rastlanmıştır.

4.4. MATLAB'teki GUI Hataları & Hataların Doğruları

Morgan Kaufmann Yayınevinden çıkan, Jeff Johnson'ın 2000 yılında yazmış olduğu "*Grafiksel Kullanıcı Arayüz Hataları (GUI Bloopers)*" adlı kitap örnek alınarak, MATLAB programındaki GUI hataları incelenmiştir. Bu kitaptaki grafiksel kullanıcı arayüz (GUI) hataları ile ilgili elde edilen bilgiler ışığında, MATLAB programında ortaya çıkan GUI hataları ve bu hataların doğruları şu şekildedir [60]:

4.4.1 MATLAB'te GUI Elemanlarında Yapılan Hatalar (GUI Component Bloopers)

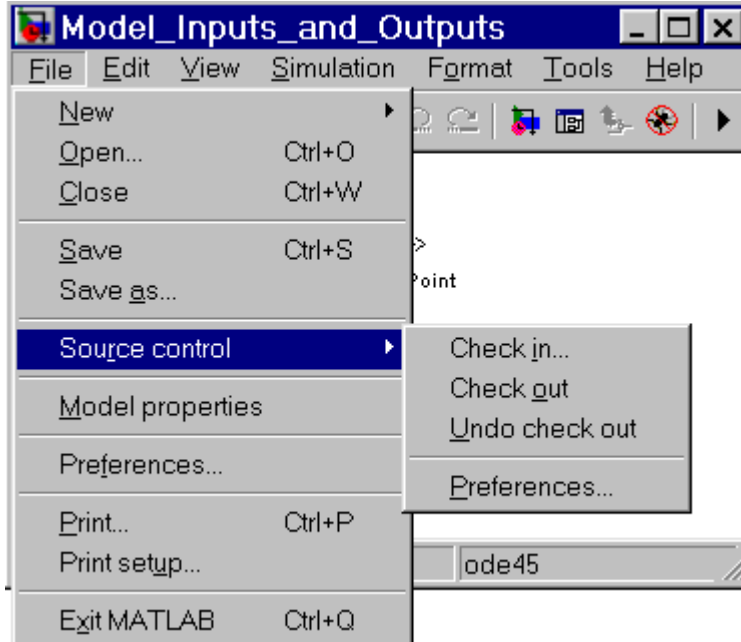
HATA 1 : Bir menünün açıldığı zaman diğer menülerin üstüne taşması, hataya sebep olmuştur. Menülerin dengeli olmaması, kullanıcılar tarafından karışıklığa neden olmaktadır.



Şekil 50 : MATLAB GUI Hatası-1

DOĞRU 1 : Menüler açıldığı zaman diğer menülerin üzerine taşmamalı ve karışıklığa sebep olmamalıdır. Arayüzde menüler dengeli olmalıdır.

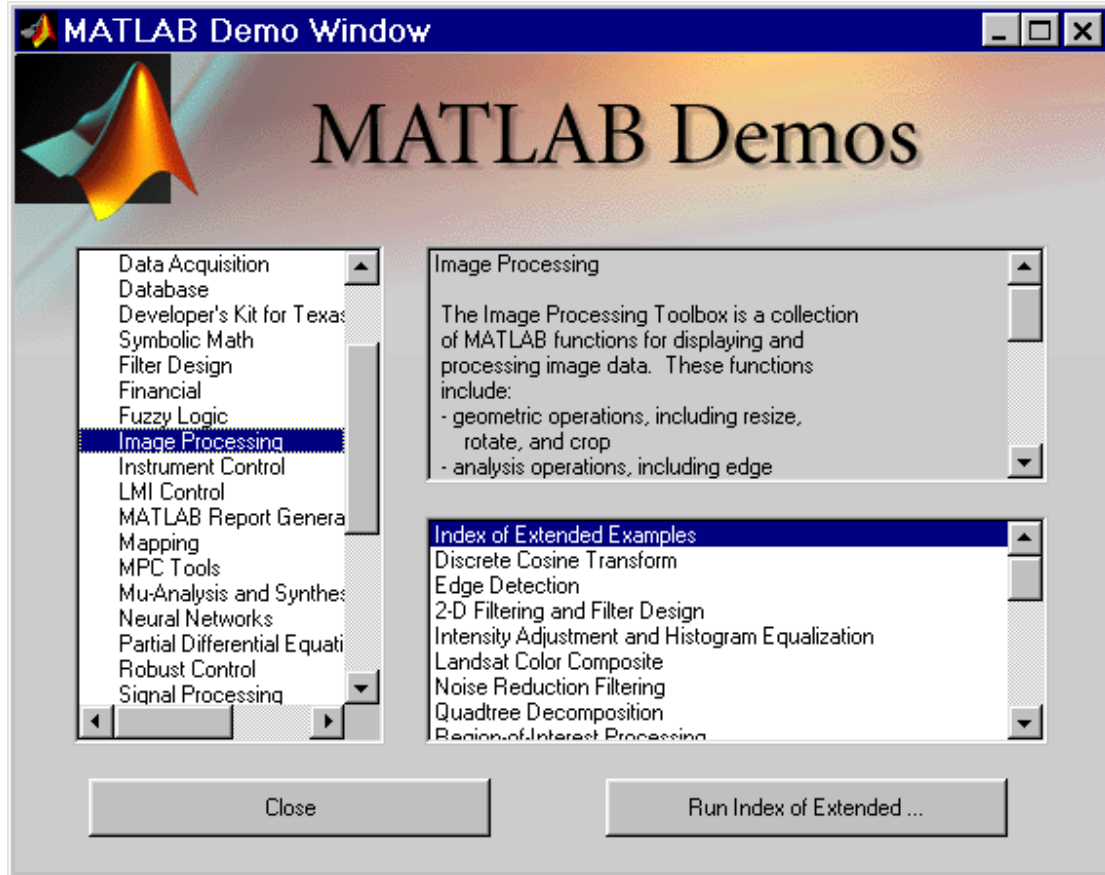
HATA 2 : Bir menü içerisinde aynı isimli iki alt menünün (*Preferences*) olması, hataya sebep olmuştur. Bu durum, kullanıcılar açısından belirsizliğe neden olmaktadır.



Şekil 51 : MATLAB GUI Hatası-2

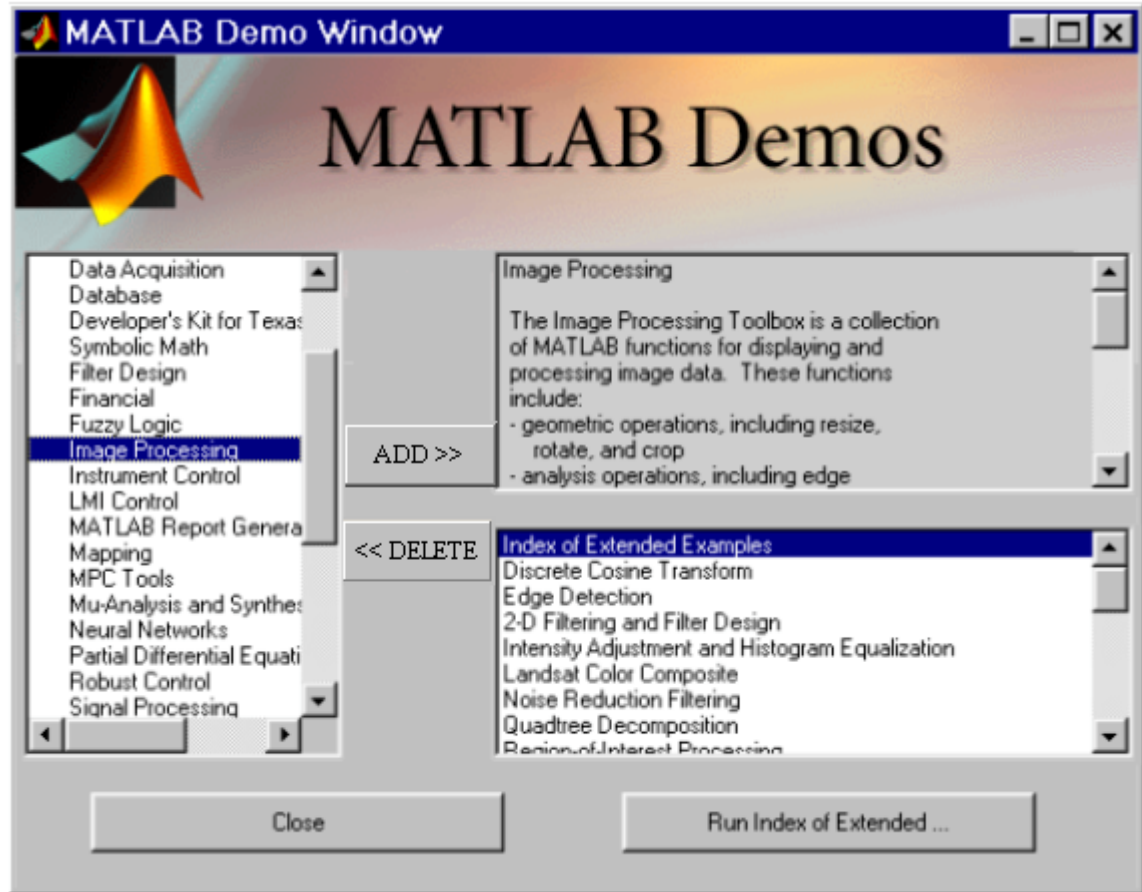
DOĞRU 2 : Bir menüde, aynı isimli iki alt menü olamaz. Bunu önlemek için, menülere farklı isimler verilmesi gerekmektedir.

HATA 3 : Bir listeden diğereine eleman eklemek veya silmek için, iki liste kutusu (listbox) arasında düğme (button) olmaması hataya sebep olmuştur. Soldaki liste kutusundan seçilen bir eleman ile ilgili bilgilerin, sağdaki liste kutusunda görünmesini sağlamak için ne yapılması gerektiğinin bilinmemesi, kullanıcı açısından anlam karışıklığına yol açmaktadır.



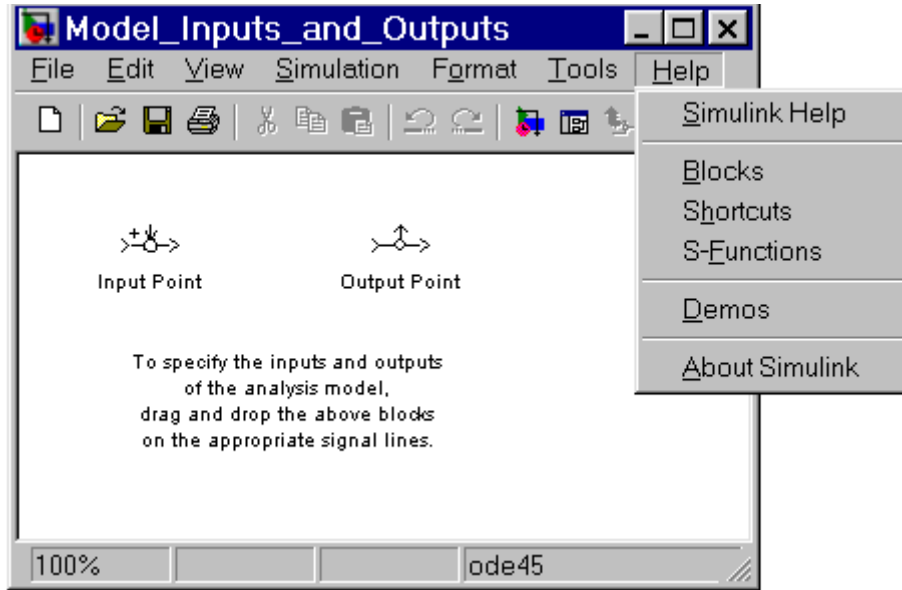
Şekil 52 : MATLAB GUI Hatası-3

DOĞRU 3 : Bir listeden diğerine eleman eklemek veya silmek için, iki liste kutusu (listbox) arasında düğme (button) olmak zorundadır. Soldaki liste kutusundan seçilen bir eleman ile ilgili bilginin, sağdaki liste kutusu'nda görünmesini sağlamak için iki liste kutusu arasında düğme (button) olması gerekir. Arayüzde iki liste kutusu arasında eleman eklemek ya da silmek için, “ADD” ve “DELETE” düğmelerini (button) kullanabiliriz.



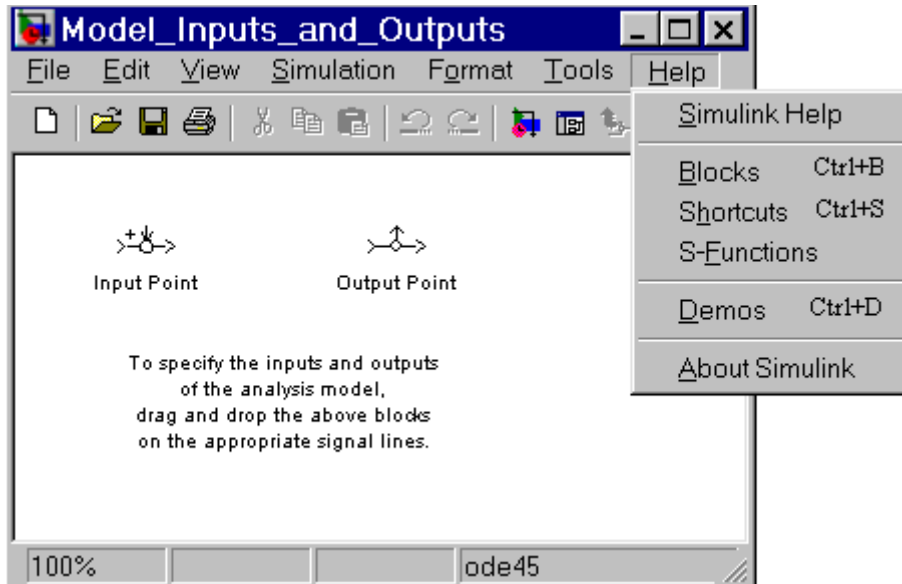
Şekil 53 : MATLAB GUI Hatası-3'ün Doğrusu

HATA 4 : Menülerde, klavye kısayol tuşlarının (shortcuts) kullanılmaması hataya sebep olmuştur.



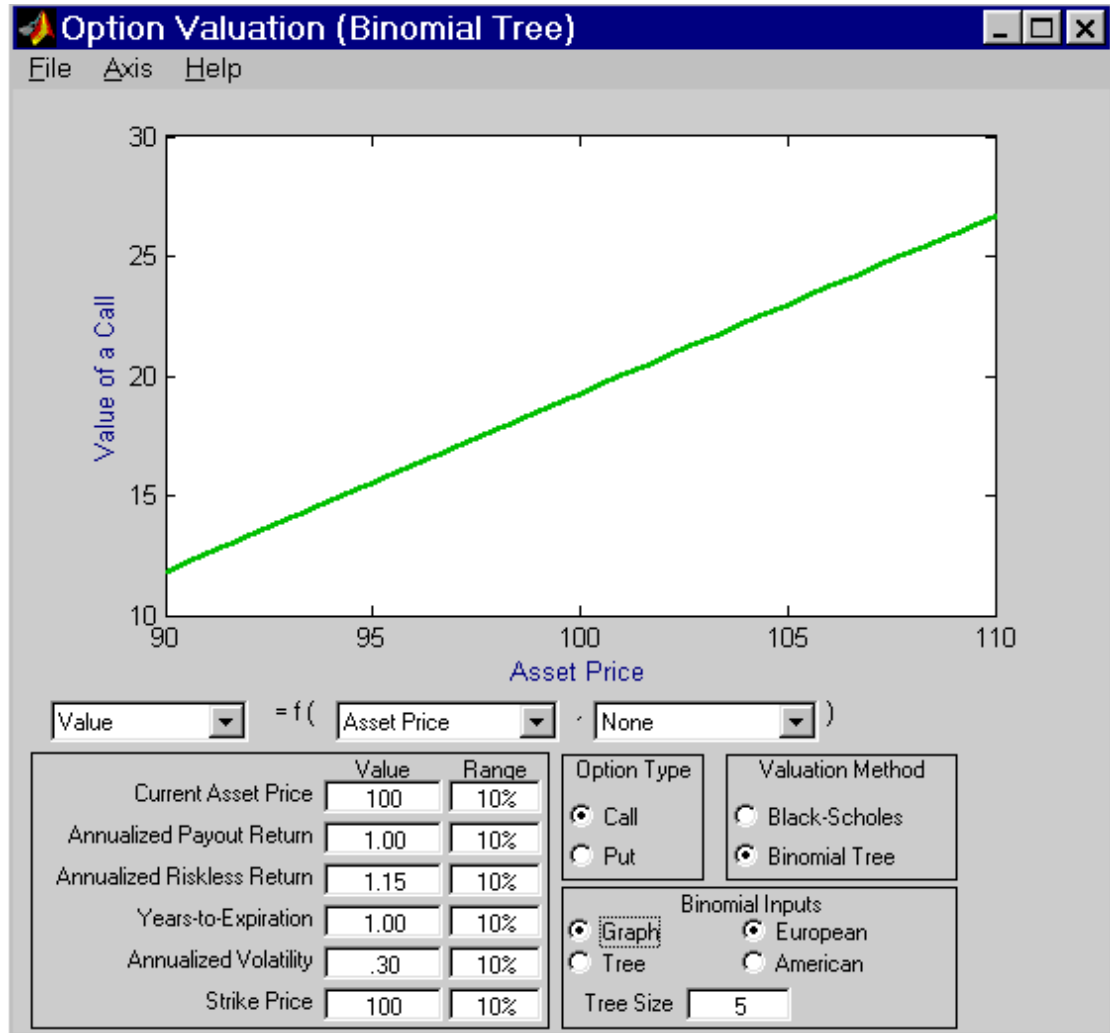
Şekil 54 : MATLAB GUI Hatası-4

DOĞRU 4 : Menülerde, kullanıcılara kolaylık sağlaması açısından, klavye kısayol tuşlarının kullanılması gerekmektedir. Bu durum, zaman bakımından ve arayüzün basit bir kullanımının olmasının sağlanması açısından faydalı olacaktır.



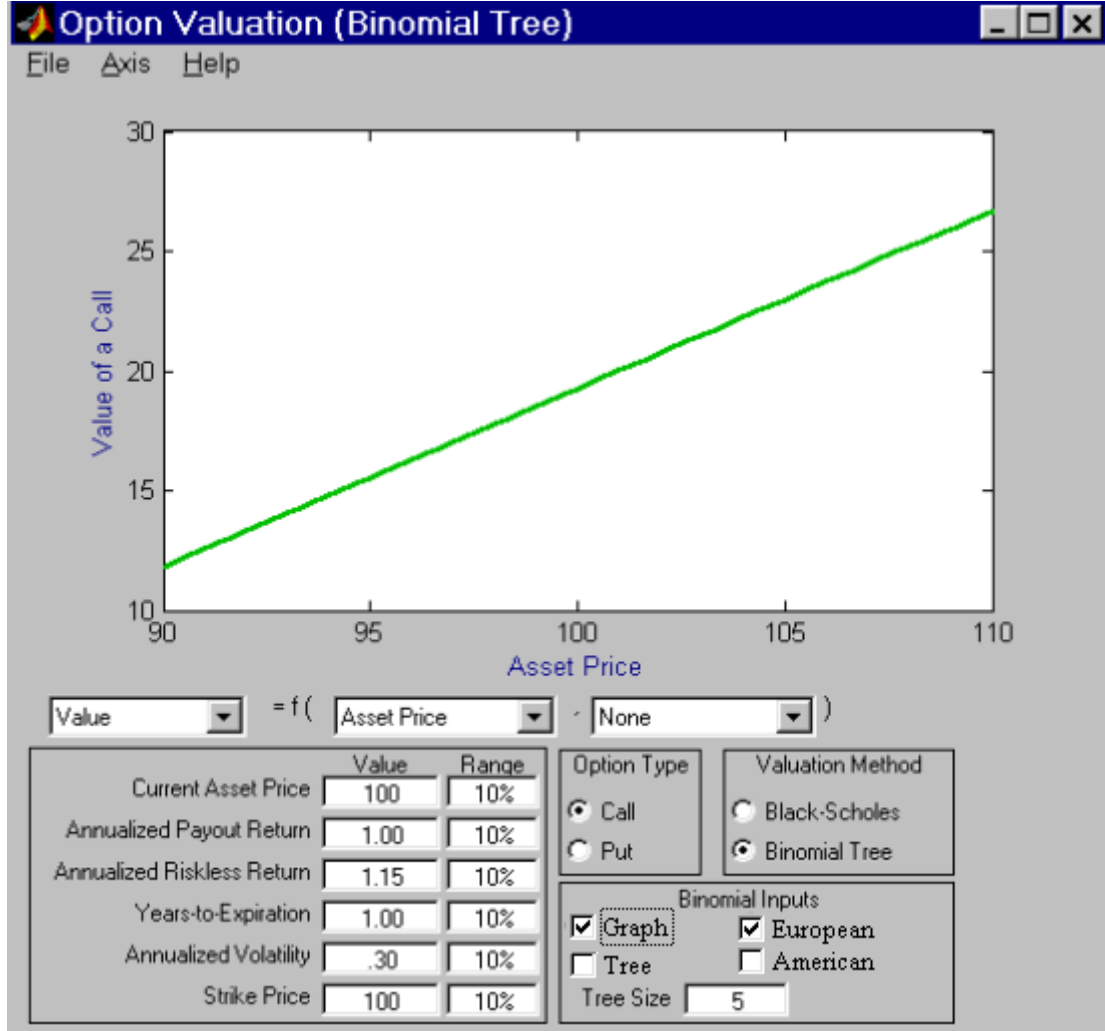
Şekil 55 : MATLAB GUI Hatası-4'ün Doğrusu

HATA 5 : Hatanın temel nedeni; arayüz elemanlarının işlevlerinin birbirine karıştırılmasıdır. “*Binomial Inputs*” bölümünde, birden fazla seçim yapmak için denetim kutusu (checkbox) kullanılması gerekirken, seçenek düğmesi (optionbutton) kullanılması, hataya sebep olmuştur.



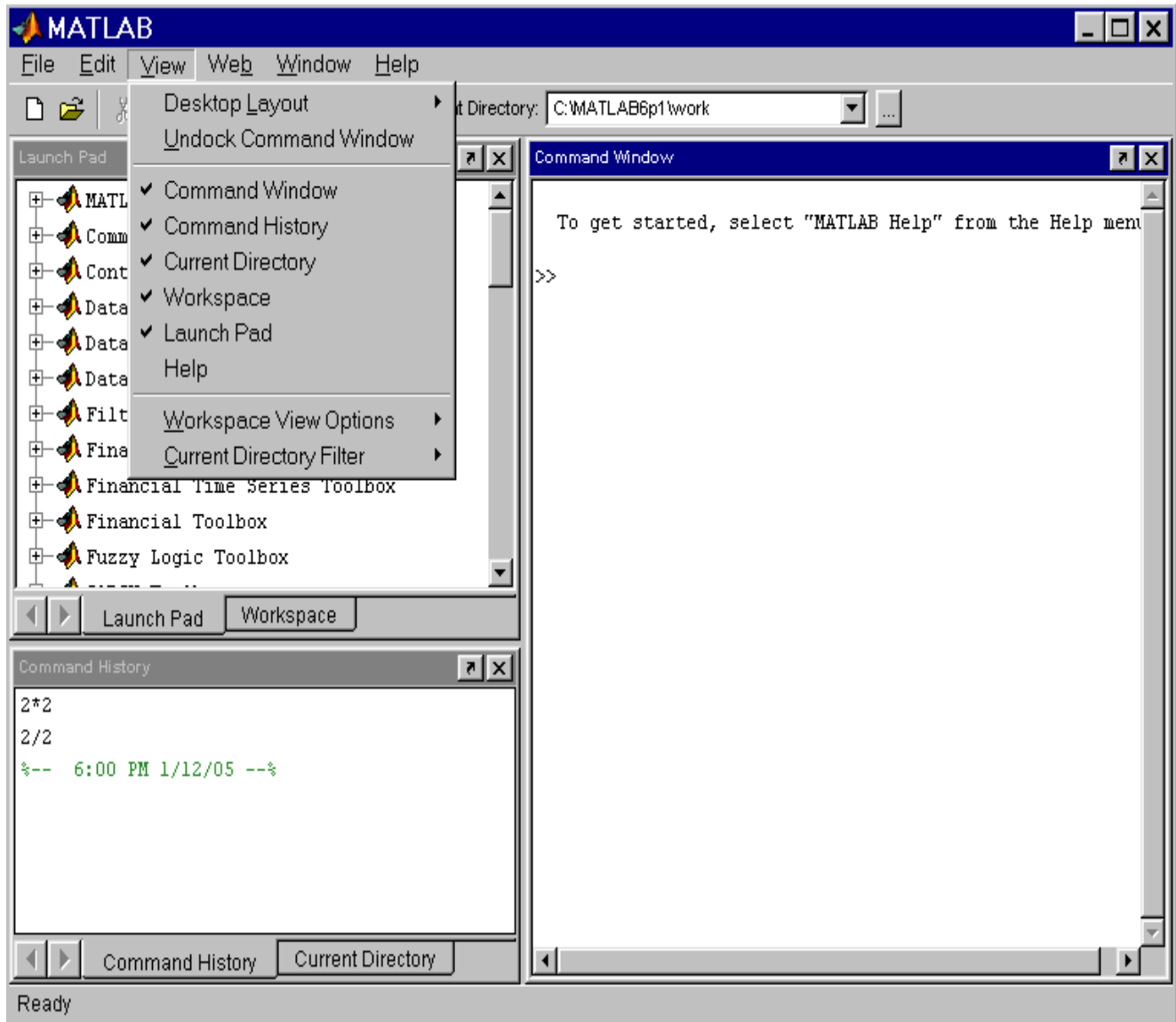
Şekil 56 : MATLAB GUI Hatası-5

DOĞRU 5 : Arayüzde “*Binomial Inputs*” bölümünde, seçenek düğmesi (optionbutton) yerine, denetim kutusu (checkbox) kullanılmalıdır. Çünkü birden fazla seçim ancak denetim kutusu (checkbox) ile yapılabilir.



Şekil 57 : MATLAB GUI Hatası-5'in Doğrusu

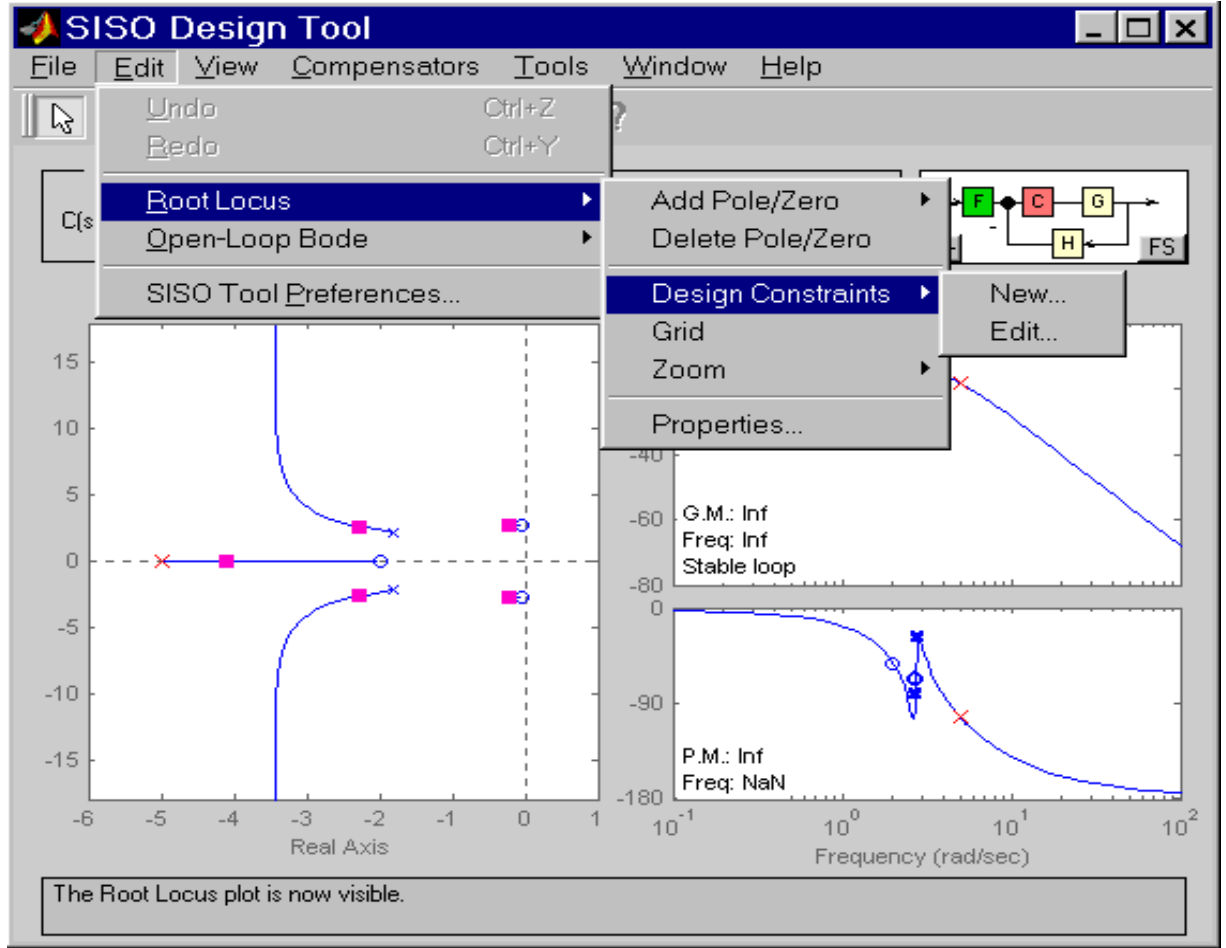
HATA 6 : Alt menü ismi ile ana menü isminin (*Help*) aynı olması, hataya sebep olmuştur. Bu durum, kullanıcı açısından belirsizliğe yol açmaktadır.



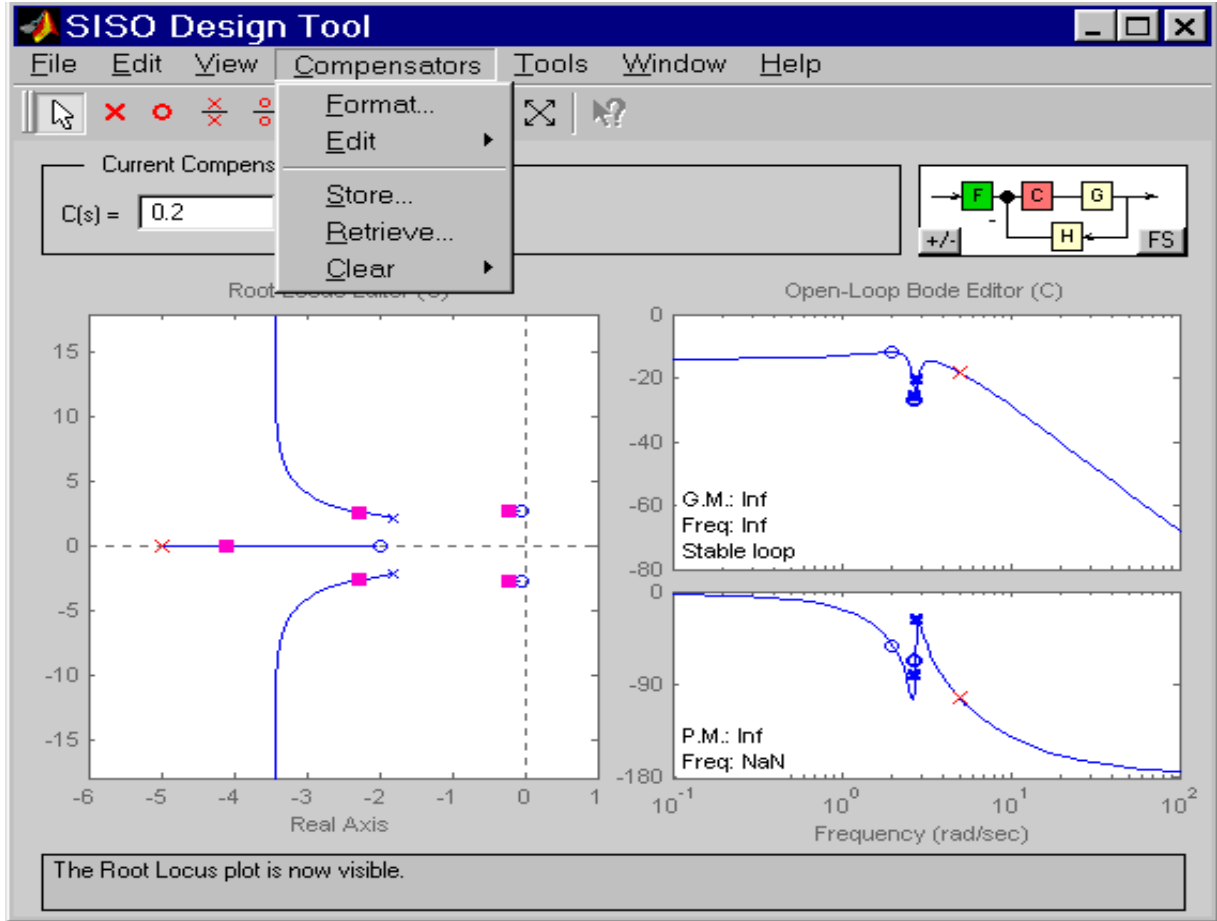
Şekil 58 : MATLAB GUI Hatası-6

DOĞRU 6 : Alt menü ismi ile ana menü ismi aynı olmamalıdır. Menüler arasındaki bu karışıklığı önlemek için, menülere farklı isimler verilmelidir.

HATA 7 : Aynı arayüzün farklı menülerindeki alt menü isimlerinin (*Edit*) aynı olması, hataya sebep olmuştur. Bu durum, kullanıcının kafasının karışmasına neden olmaktadır.



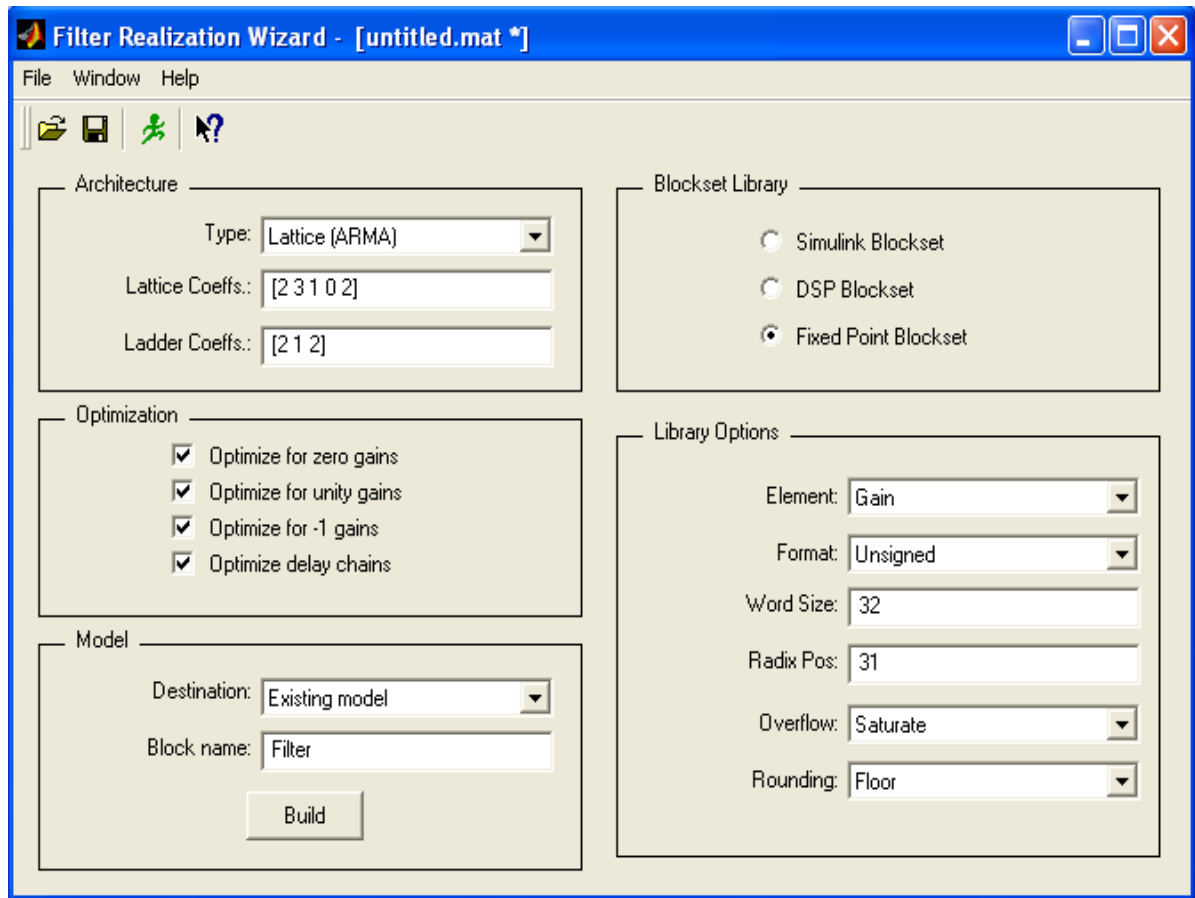
Şekil 59 : MATLAB GUI Hatası-7'nin Birinci Arayüzü



Şekil 60 : MATLAB GUI Hatası-7'nin İkinci Arayüzü

DOĞRU 7 : Bir arayüzdeki iki farklı menüde, aynı isimli iki alt menünün olmaması gerekir. Menülerdeki alt menülerin, farklı isimli olmaları gerekmektedir.

HATA 8 : Arayüzdeki “*Blockset Library*” bölümünde, seçenek düğmesi’nin (option button) yanlış kullanılması, hataya sebep olmuştur.



Şekil 61 : MATLAB GUI Hatası-8

DOĞRU 8 : “*Blockset Library*” bölümünde; seçenek düğmesi’nin (option button), çerçeve (frame) içerisinde kullanılmaması gerekir. “*Blockset Library*” bölümü, etiket (label) içerisinde aşağıdaki gibi gösterilmelidir :

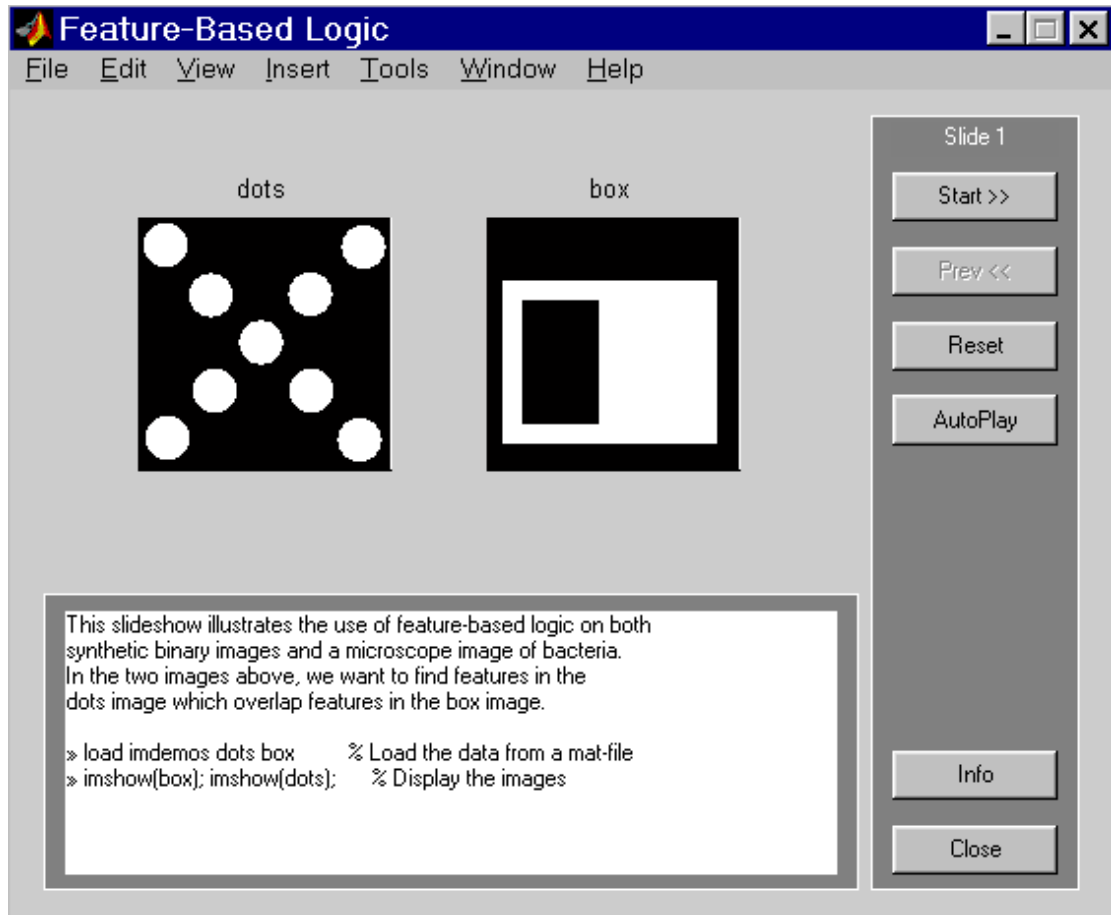
Blockset Library :

- Simulink Blockset
- DSP Blockset
- Fixed Point Blockset

Şekil 62 : MATLAB GUI Hatası-8’in Doğrusu

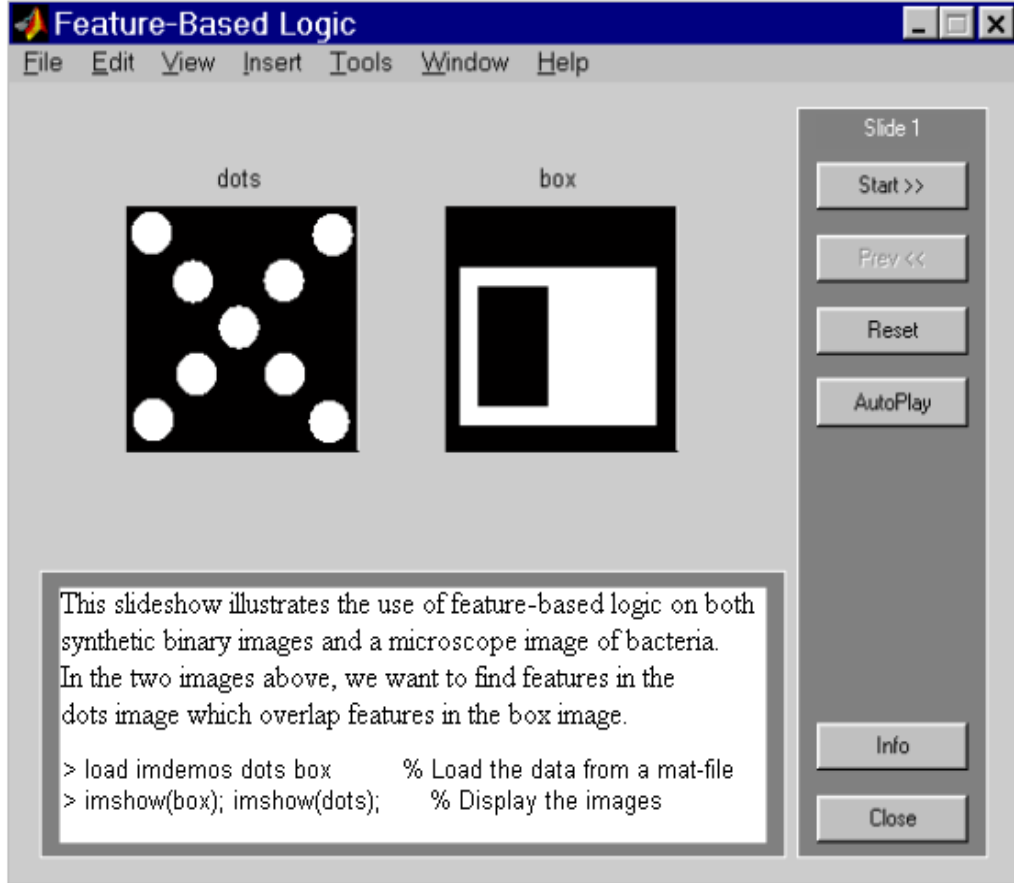
4.4.2 MATLAB'teki Düzen ve Görünüm Hataları (Layout & Appearance Bloopers)

HATA 9 : Ana sorun : Arayüzde, çok ufak kullanılan yazı tipi boyutunun olmasıdır. Metnin yazı tipi boyutundan kaynaklanan problemin olması, hataya neden olmuştur. Arayüzde yazı tipi boyutunun 8 olması, kullanıcıyı rahatsız etmektedir. Oldukça küçük yazı tipi boyutunun kullanımı, görünüşü bozmakta ve okumayı zorlaştırmaktadır.



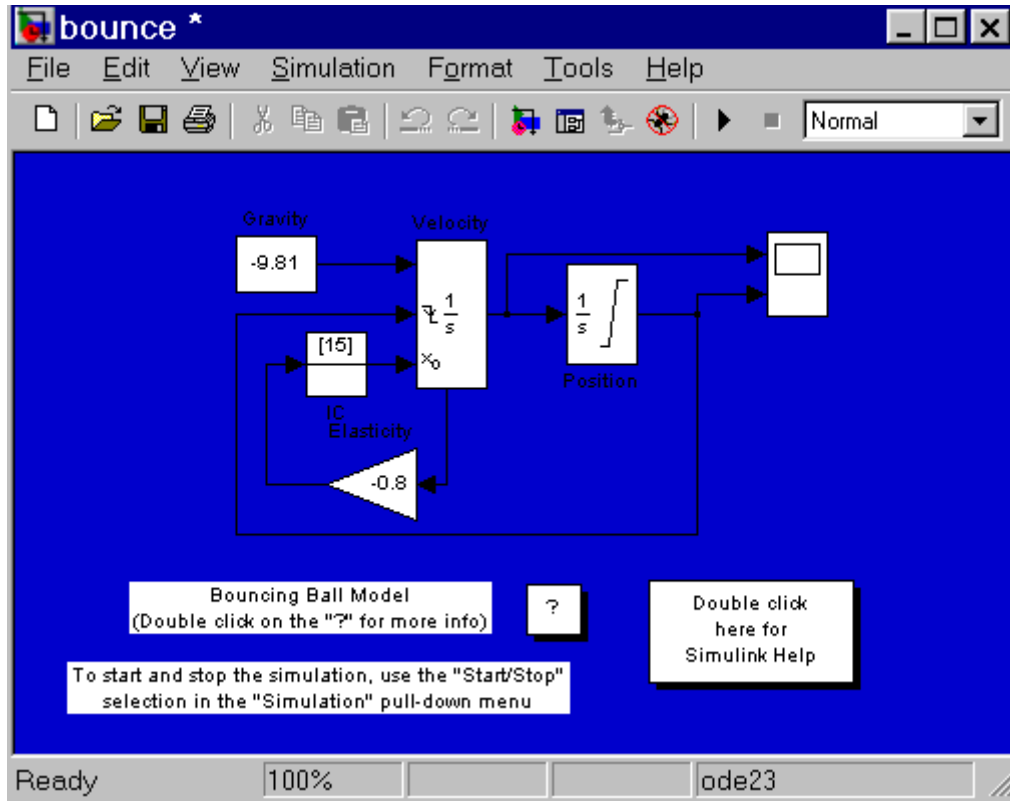
Şekil 63 : MATLAB GUI Hatası-9

DOĞRU 9 : Minimum metin yazı tipi boyutu, 10 olmalıdır. Standart olan, tavsiye edilen ve kullanıcı için daha iyi olacak olan metin yazı tipi boyutu, 12'dir. Arayüzde kullanılan metin yazı tipi boyutları standart ve sabit olmalıdır.



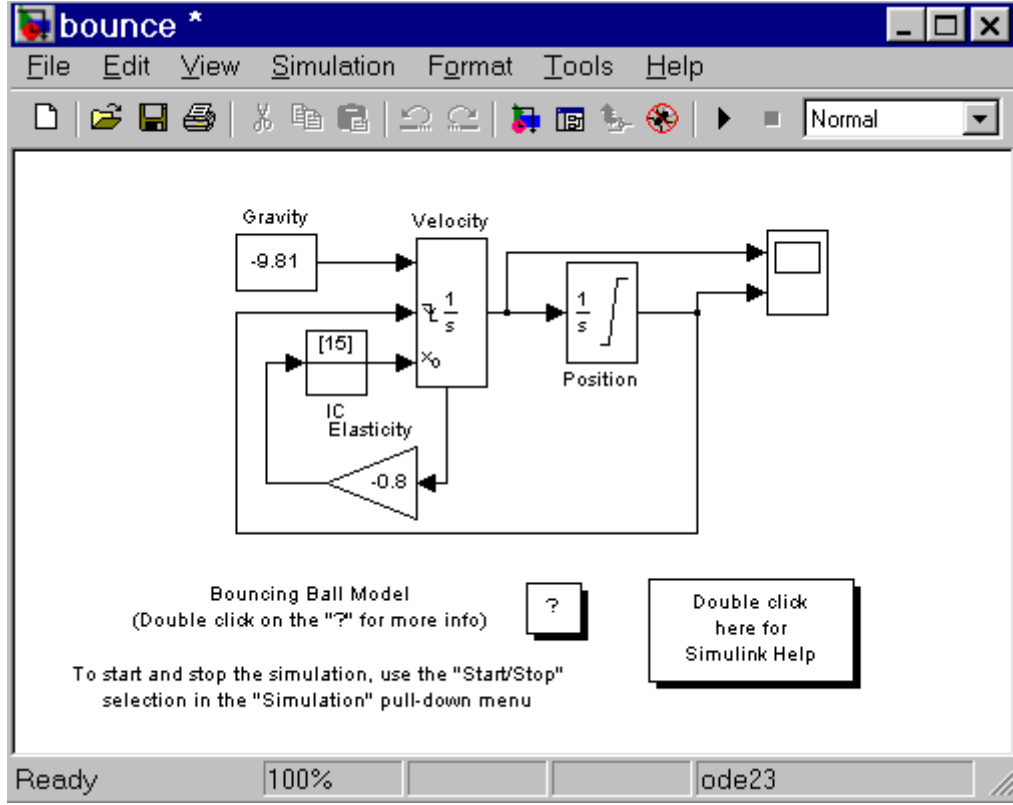
Şekil 64 : MATLAB GUI Hatası-9'un Doğrusu

HATA 10 : Hatanın ana sebebi; kötü bir arayüz arka plan renginin seçilmiş olmasıdır. Arayüz arka plan renginin koyu olması, okunabilirliği olumsuz yönde etkilemektedir.



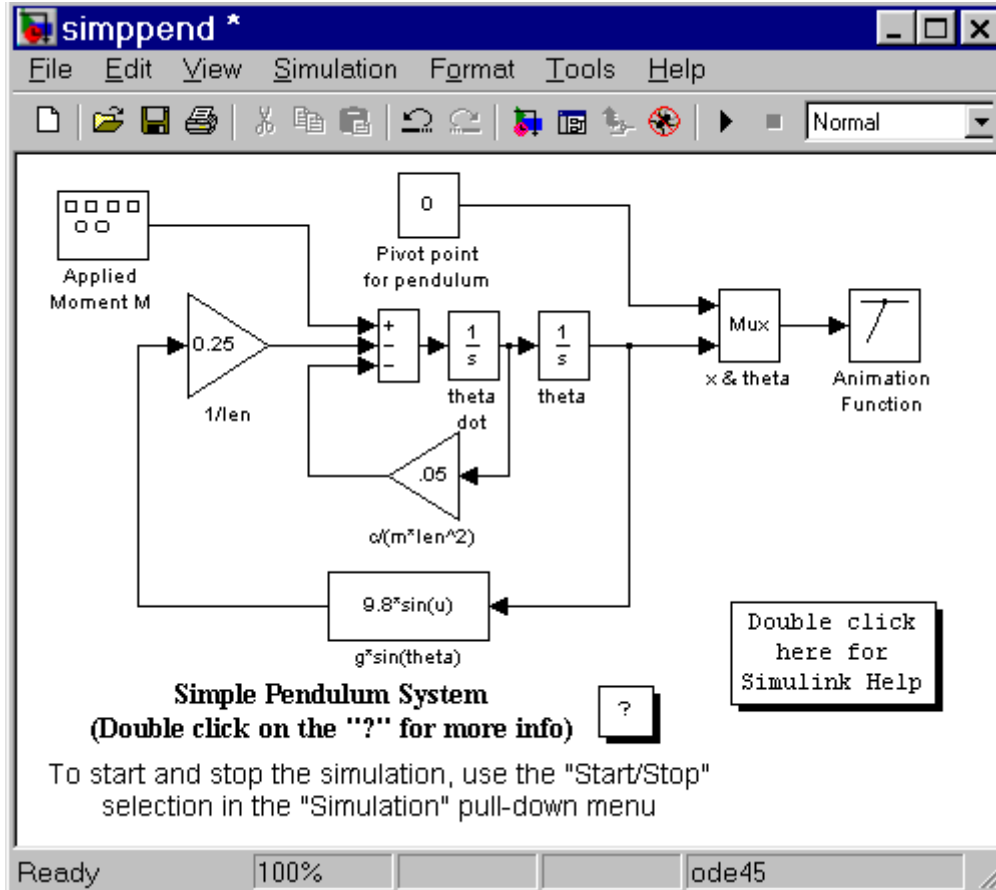
Şekil 65 : MATLAB GUI Hatası-10

DOĞRU 10 : Arayüz arka plan rengi, metin rengi ile çakışmamalıdır ve kullanıcı açısından okunabilirliği zorlaştırmamalıdır. Arayüz arka planı için doğal, sade ve gözü rahatsız etmeyecek renkler kullanılmalıdır. Örneğin; gri ve beyaz gibi. Arka plan rengi olarak beyaz seçilerek, rahat okunabilirlik sağlanmış olur.



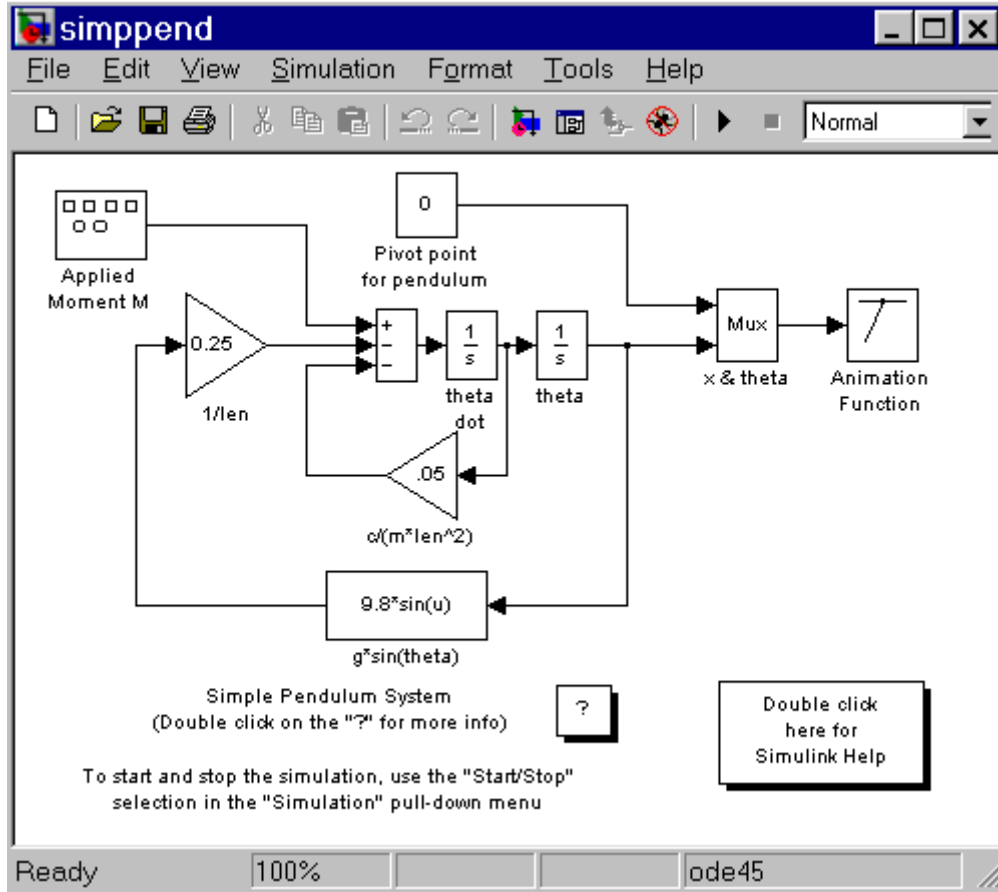
Şekil 66 : MATLAB GUI Hatası-10'un Doğrusu

HATA 11 : Hatanın ana sebebi : Tutarsız, uygun olmayan yazı tiplerinin kullanılmasıdır. Farklı yazı tiplerinin kullanılması, arayüzün görünümünü bozmaktadır. Yazı tipleri, birbirine karıştırılmamalıdır. Bir arayüzde üç farklı yazı tipi kullanılmaz. Arayüzde “Times New Roman”, “Arial” ve “Courier New” gibi farklı yazı tiplerinin olması görünüşü bozmakta ve okumayı zorlaştırmaktadır.



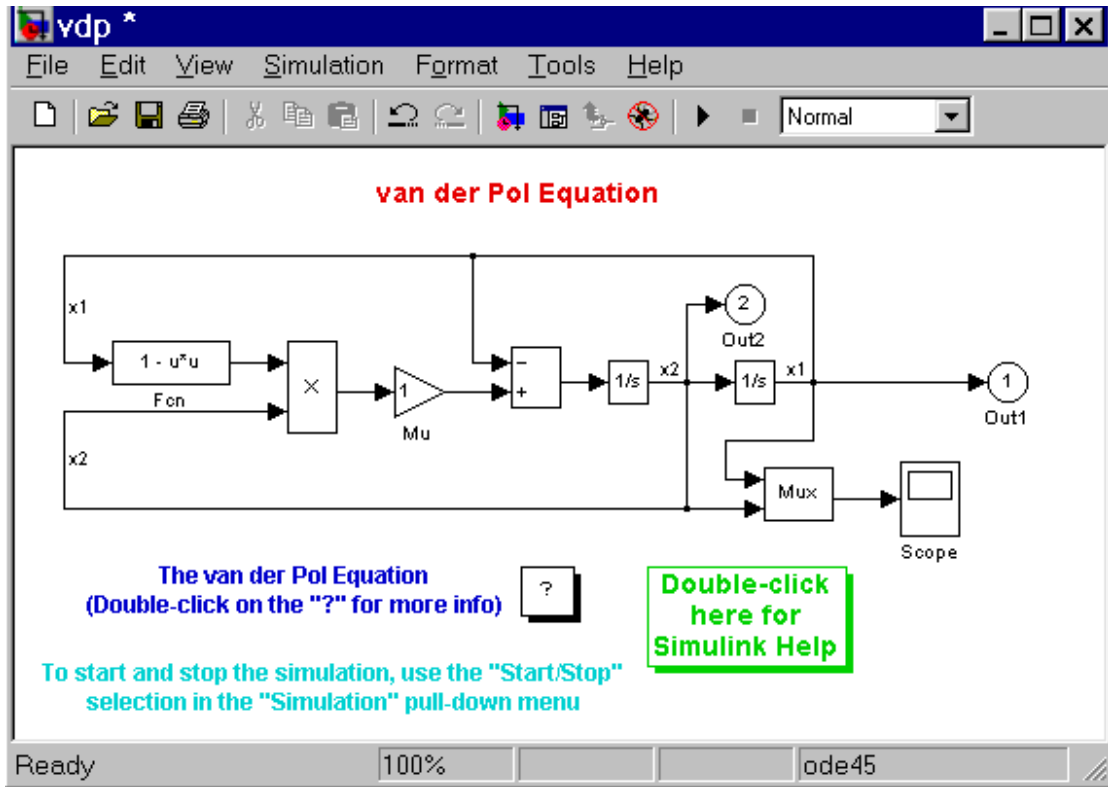
Şekil 67 : MATLAB GUI Hatası-11

DOĞRU 11 : Arayüzdeki yazı tipleri, tutarlı ve birbirine uygun bir biçimde kullanılmalıdır. Arayüzde standart bir yazı tipi belirlenmeli ve yazı tipi seçiminde, kullanıcının metni rahat okuyabilmesi durumu göz önünde bulundurulmalıdır. Aşağıdaki arayüzün yazı tipi “Times New Roman” seçilerek, arayüzde sabit bir yazı tipinin olması sağlanmıştır. Buda arayüzün bütünlük ilkesinin sağlanmasına ve arayüzün görünümünün düzelmesine yol açmıştır.



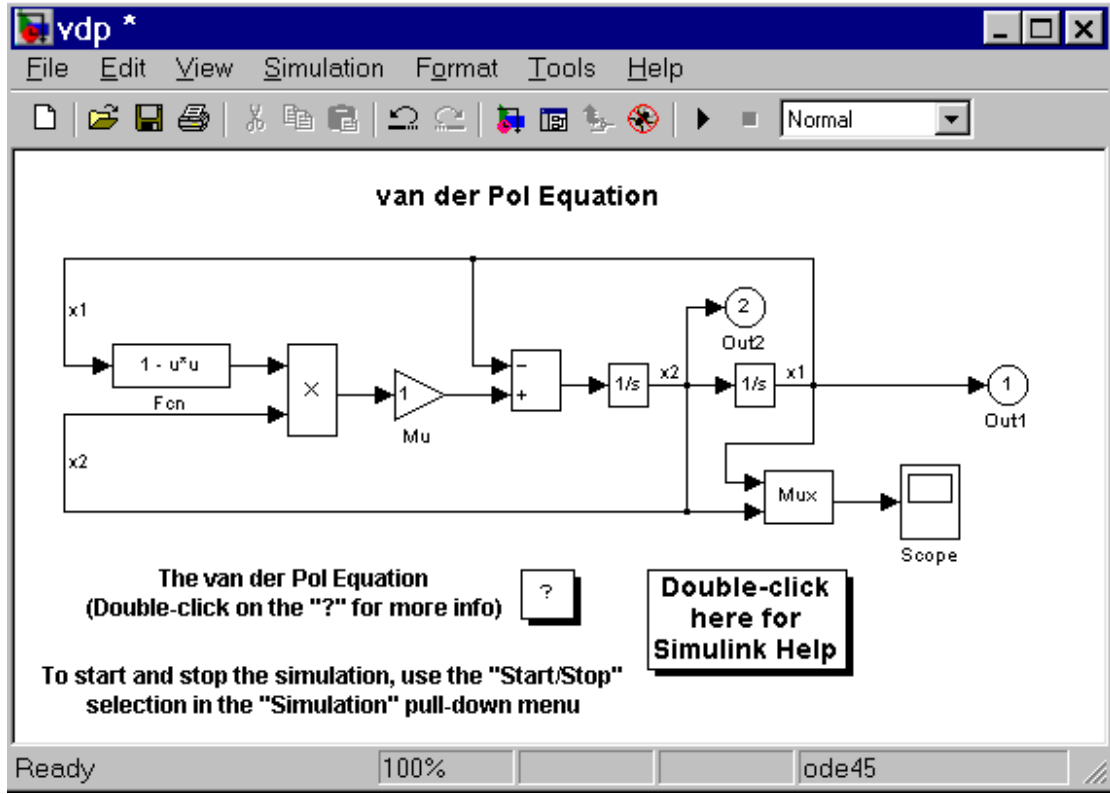
Şekil 68 : MATLAB GUI Hatası-11'in Doğrusu

HATA 12 : Arayüzde farklı metin renklerinin kullanılması, kötü bir görünüme ve gözün yorulmasına sebep olmaktadır. Bir arayüzde, dört farklı metin rengi kullanılamaz. Farklı metin renklerinin kullanılması, kullanıcının ilgisinin dağılmasına neden olacaktır.



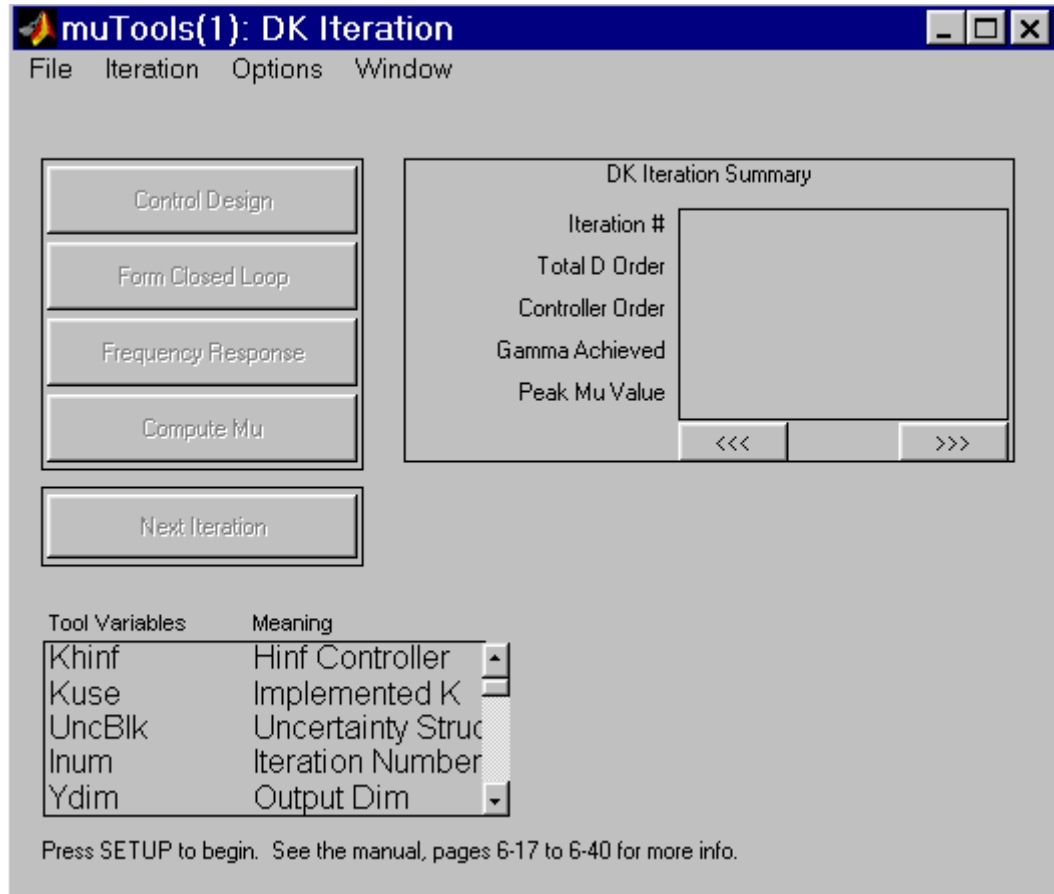
Şekil 69 : MATLAB GUI Hatası-12

DOĞRU 12 : Arayüzdeki bütünlüğü ve görselliği sağlamak için, tek bir metin rengi seçilmelidir. Tek bir rengin seçilmesi, dikkatin dağılmasını engelleyecektir. Aşağıdaki arayüzde metin rengi siyah seçilerek, gözün rahatsız edilmesi önlenmiş ve arayüzdeki bütünlük sağlanmıştır.



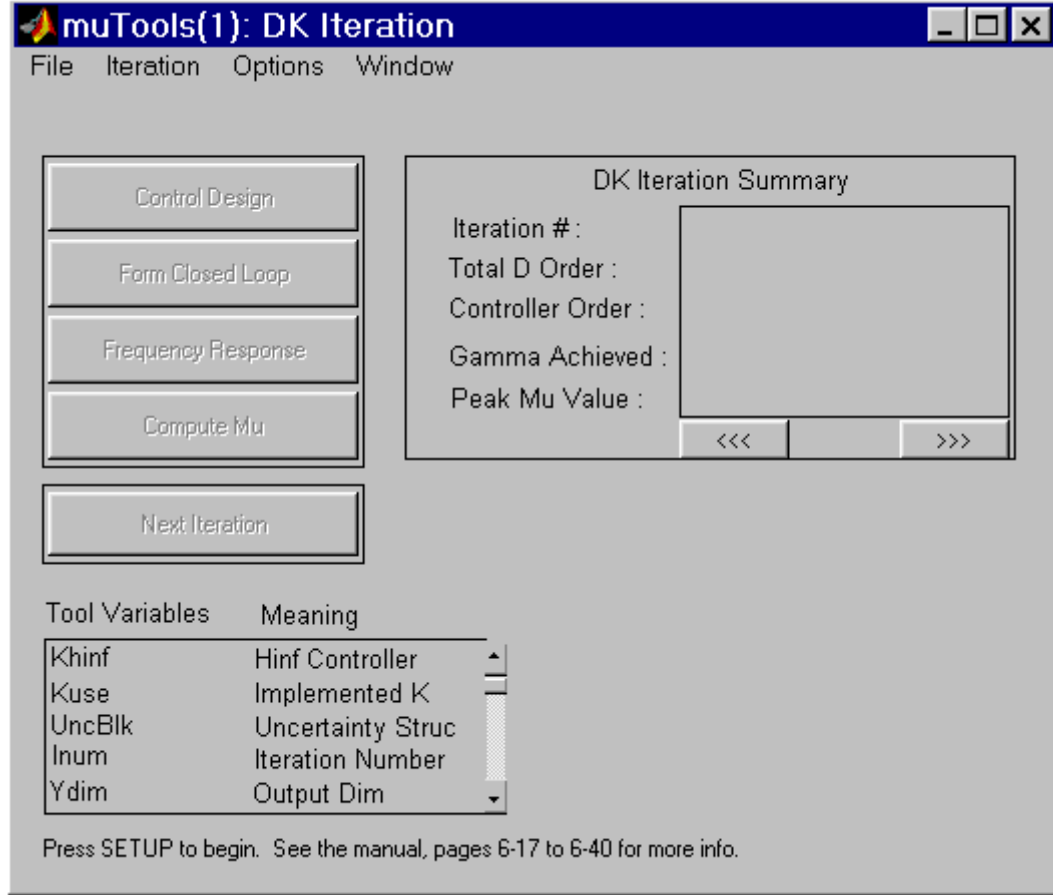
Şekil 70 : MATLAB GUI Hatası-12'nin Doğrusu

HATA 13 : Ana sorun : Yazı tipinde yapılan hatalardır. Aynı arayüz içerisinde, arayüz elemanları üzerindeki metinlerde farklı yazı tiplerinin olması, hataya neden olmuştur. “*Tool Variables & Meaning*” bölümünde, yazı tipi ve boyutu “*Arial-12*” iken; “*DK Iteration Summary*” çerçevesinin (frame) olduğu yerde, yazı tipi ve boyutunun “*Times New Roman-10*” olması yazı tipi farklılıklarının oluşmasına ve görünümün bozulmasına neden olmuştur.



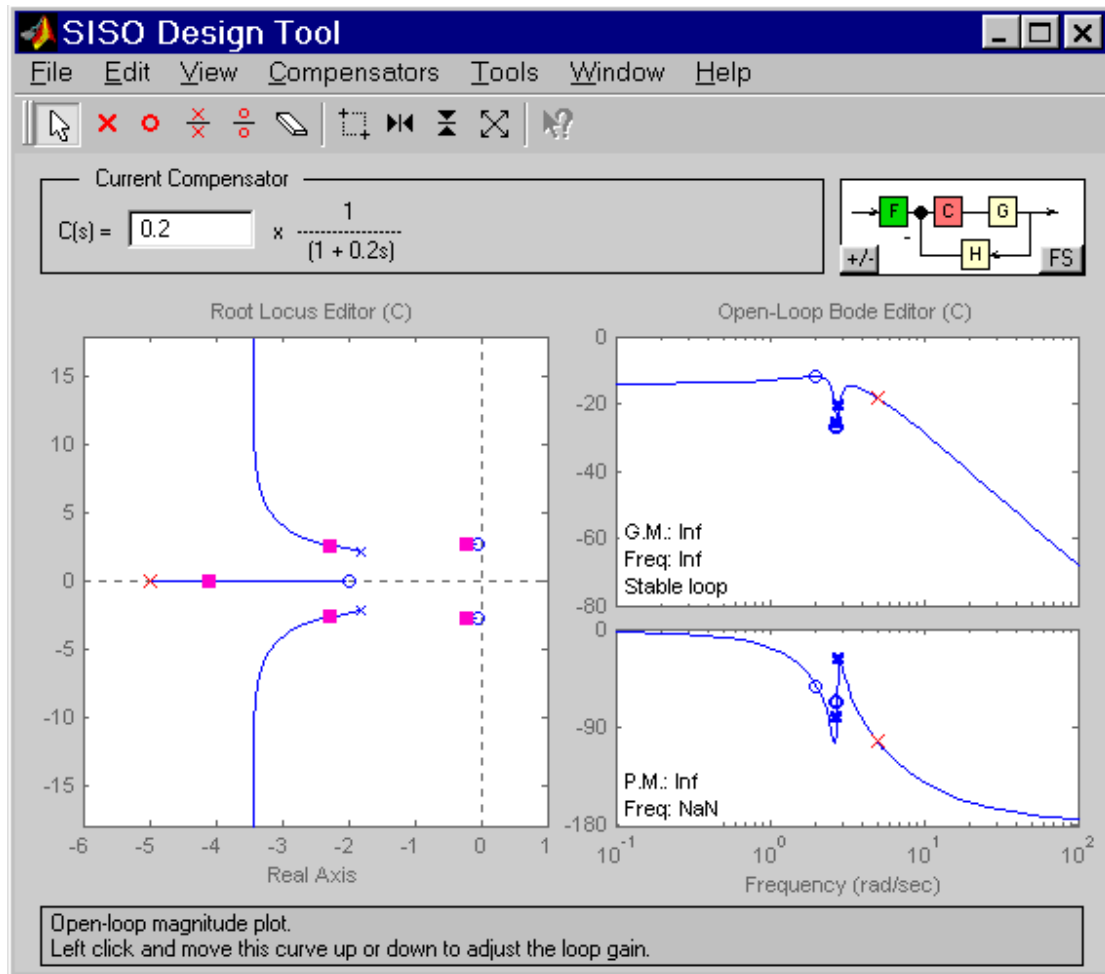
Şekil 71 : MATLAB GUI Hatası-13

DOĞRU 13 : Arayüzün bütününde, yazı tipi ve yazı tipi boyutunun aynı olması gerekmektedir. Böylece arayüzde, bütünlük ve görünüm ilkesi bozulmamış olacaktır. Aşağıdaki şekilde, “*DK Iteration Summary*” ve “*Tool Variables & Meaning*” bölümlerinin her ikisinde de yazı tipi ve boyutunun “*Arial-10*” olması, arayüzde düzenin ve görünümün tek bir yapıda olmasını sağlamıştır.



Şekil 72 : MATLAB GUI Hatası-13'ün Doğrusu

HATA 14 : “*Current Compensator*” bölümünün, çerçeve (frame) içerisinde olması hataya sebep olmuştur. Çünkü “*Current Compensator*” bölümünün etiket olması, kullanıcı açısından daha bilgilendirici olacaktır.



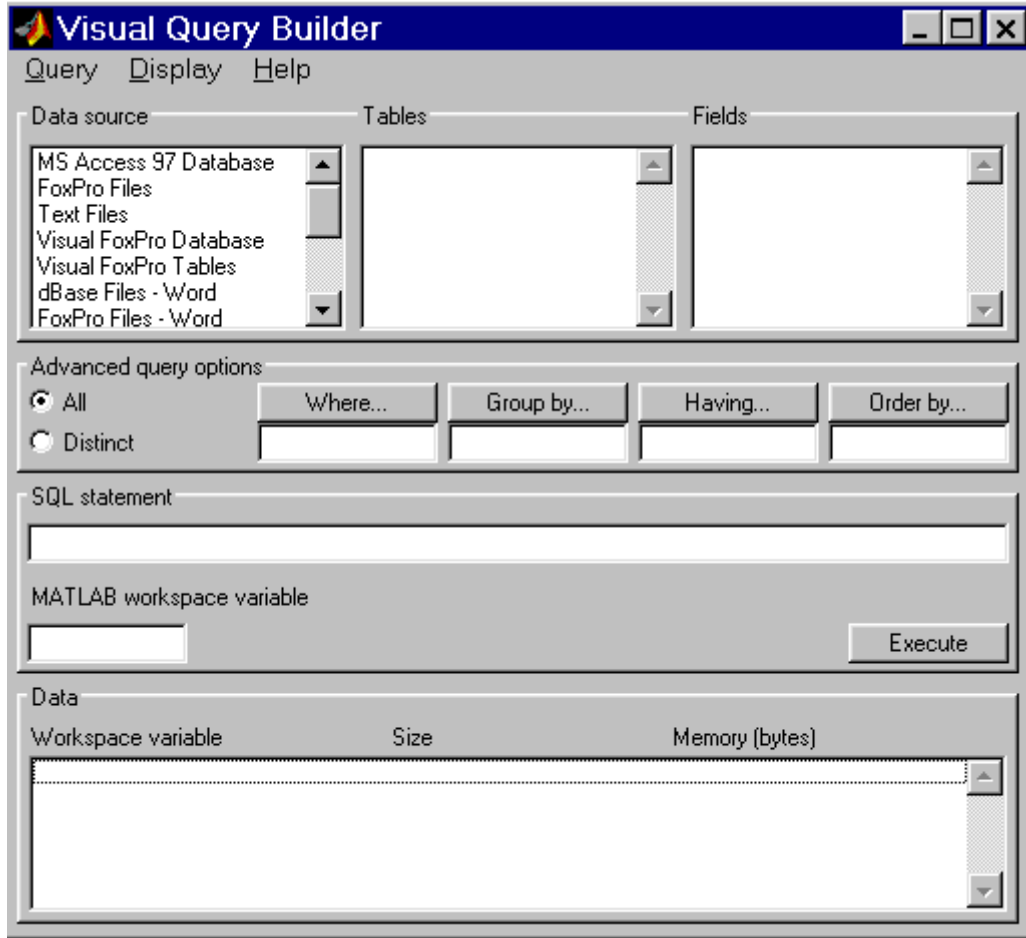
Şekil 73 : MATLAB GUI Hatası-14

DOĞRU 14 : “*Current Compensator*” bölümünün, çerçeve (frame) içerisinde olması gereksizdir. Kaldırılması gerekir. Çerçevenin yerine, etiket (label) olması gerekmektedir. Hatanın doğru şekli aşağıdaki gibi olmalıdır :

Current Compensator - C(s) : x $\frac{1}{(1 + 0.2s)}$

Şekil 74 : MATLAB GUI Hatası-14’ün Doğrusu

HATA 15 : “*SQL statement*” bölümünde, etiket (label) yerine çerçeve’nin (frame) kullanılmış olması hataya sebep olmuştur.

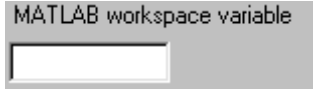


Şekil 75 : MATLAB GUI Hatası-15

DOĞRU 15 : “*SQL statement*” bölümünde, çerçeve (frame) yerine, etiket (label) olmalıydı. Çerçeve (frame) gereksizdir ve kaldırılması gerekir. Çerçeve (frame) yerine etiket (label) kullanılmalıdır. “*SQL statement*” bölümünün, doğru kullanım şekli aşağıdaki gibi olmalıdır :

SQL statement :

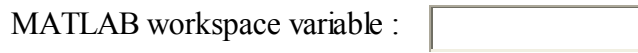
Şekil 76 : MATLAB GUI Hatası-15’in Doğrusu



Şekil 77 : MATLAB GUI Hatası-16

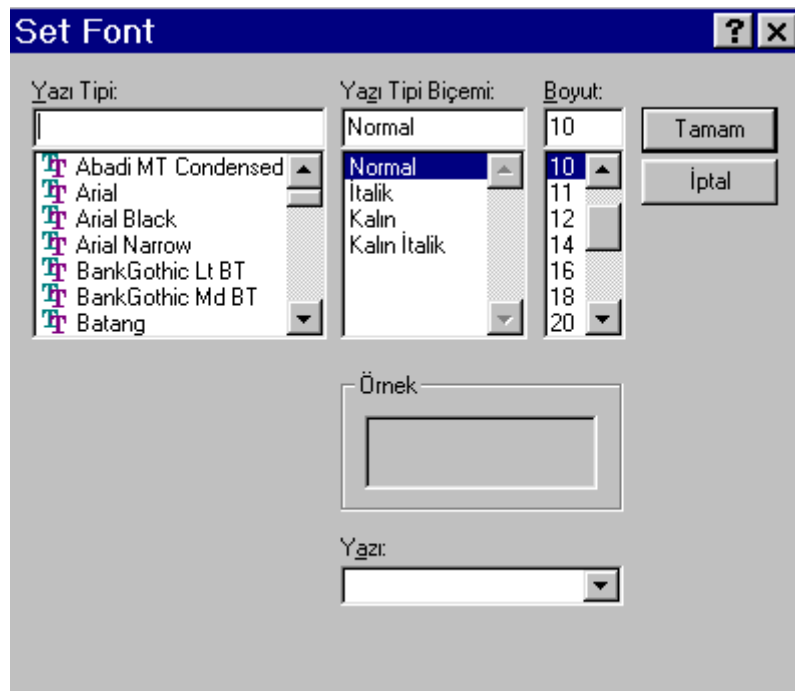
HATA 16 : Yukarıdaki şekilde ana sorun; etiket'in (label) metin alanı (textfield) ile birlikte kullanımının hatalı olmasıdır. Etiket ile metin alanının kullanım şekli, standart değildir.

DOĞRU 16 : Etiket (label) ve metin alanı'nın (textfield) standart kullanım şekli şöyle olmalıdır : Etiket solda, metin alanı ise sağda bulunmalıdır. Yapının nasıl olması gerektiği, aşağıda gösterilmiştir :



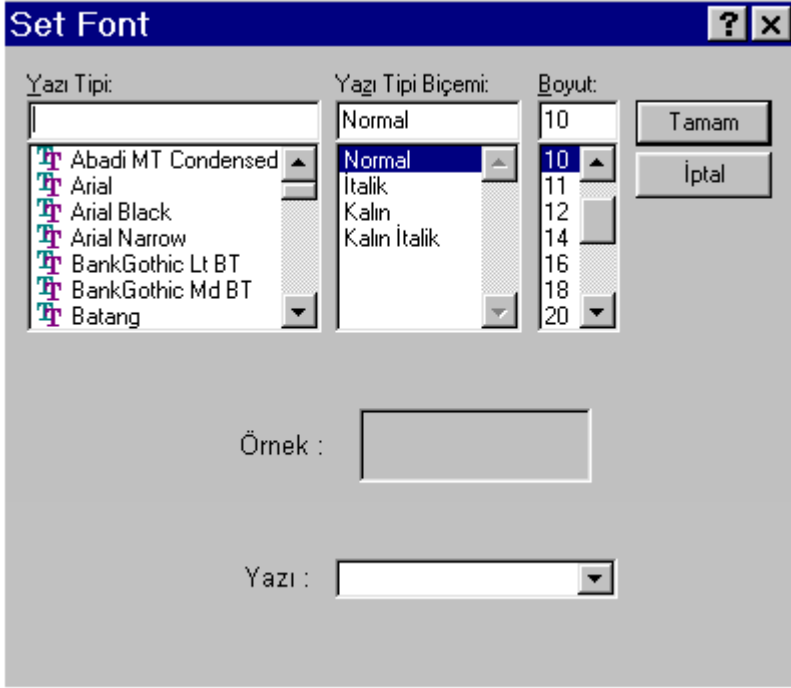
Şekil 78 : MATLAB GUI Hatası-16'nın Doğrusu

HATA 17 : Hata, “*Örnek*” yazısının olduğu alandır. “*Örnek*” alanı, seçilen yazı tipinin, yazı tipi biçiminin ve yazı boyutunun nasıl görüneceğini gösteren yerdir. Bu durum yazıların, “*Örnek*” çerçevesi (frame) içinde sıkıştırılmasına yani hapsedilmesine ve arayüzde belirsizliğe yol açmaktadır. Ayrıca, “*Yazı*” etiketi (label) ile grup kutusu'nun (combo box) birlikte kullanılmasında hata yapılmıştır.



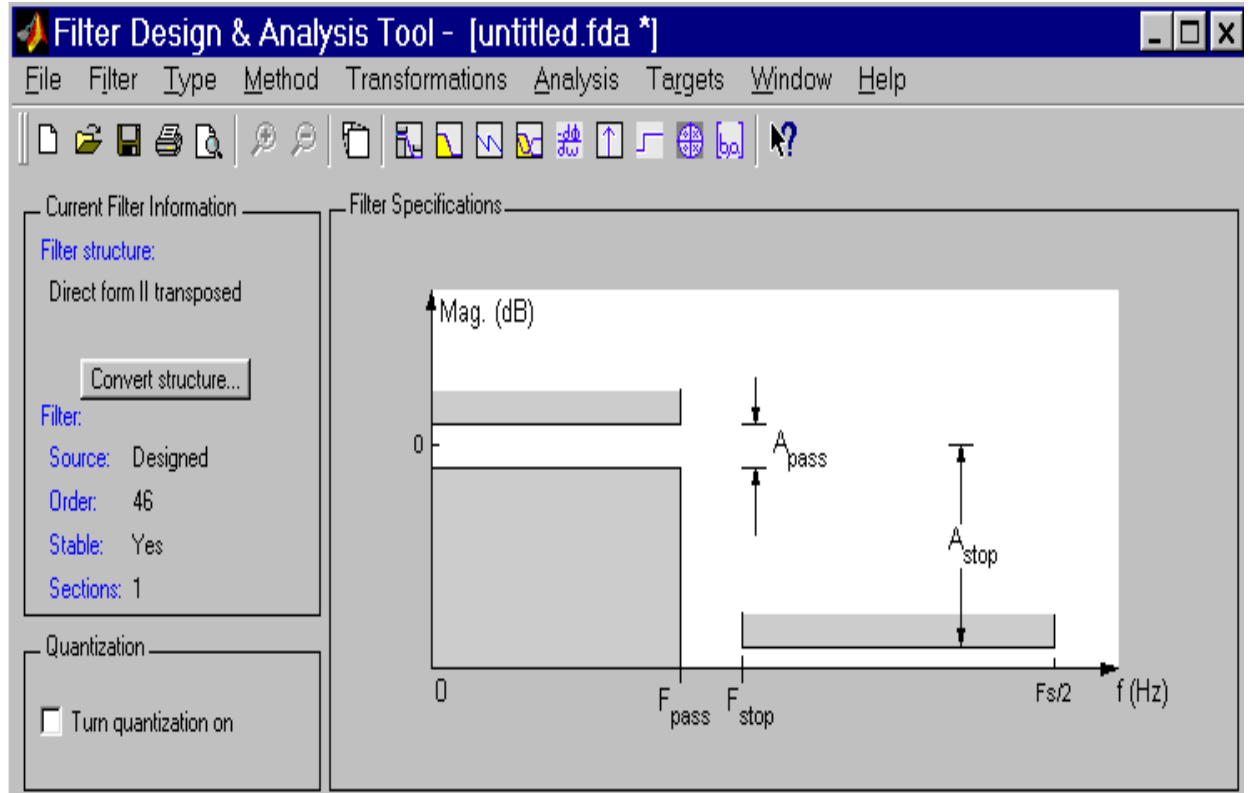
Şekil 79 : MATLAB GUI Hatası-17

DOĞRU 17 : “*Örnek*” çerçevesi (frame) gereksizdir. Açık ve anlaşılır bir yapı olması için “*Örnek*” çerçevesi’nin kaldırılması gerekmektedir. Çerçeve yerine, etiket (label) kullanılmalıdır. Ayrıca, grup kutusu (combo box) ile etiket’in (label) birlikte kullanımı yanlıştır. Bu iki GUI elemanının kullanımlarının standart ve doğru şekli şöyle olmalıdır : Etiket (label) solda, grup kutusu (combo box) ise sağ tarafta bulunmalıdır. “*Örnek*” çerçevesi yerine label olmasının ve “*Yazı*” etiketinin grup kutusu (combo box) ile birlikte kullanımının doğru şekli aşağıdaki gibi olmalıdır :



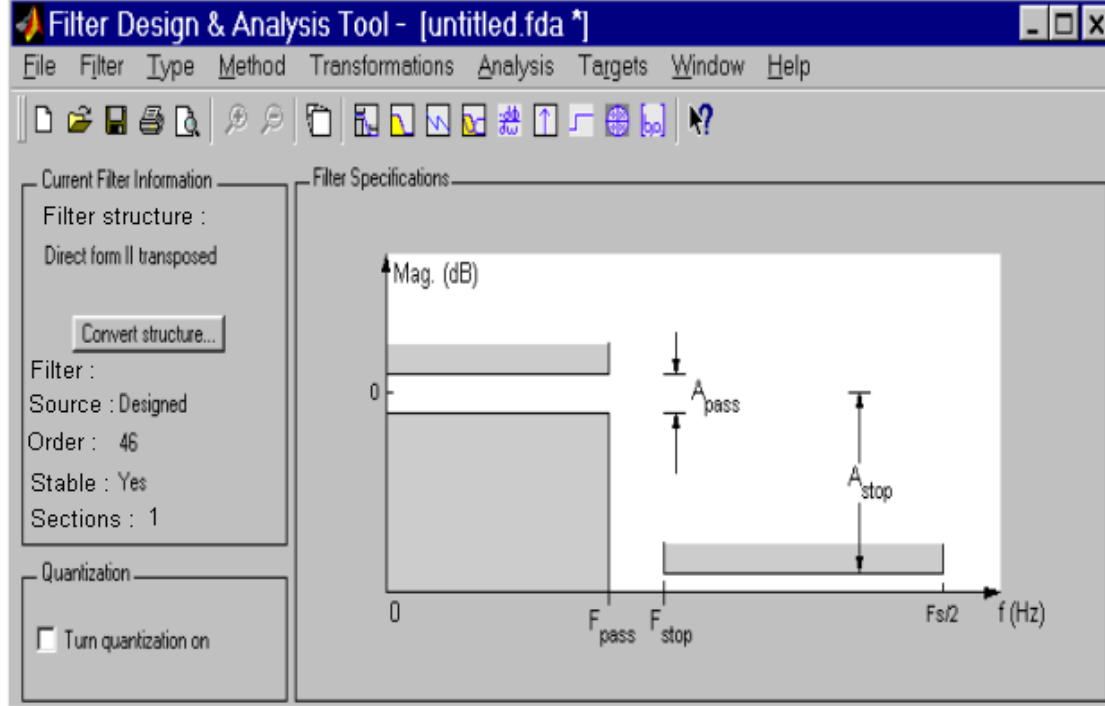
Şekil 80 : MATLAB GUI Hatası-17’nin Doğrusu

HATA 18 : “Current Filter Information” bölümündeki “Filter structure”, “Filter”, “Source”, “Order”, “Stable” ve “Sections” etiketlerindeki (label) yanlış renk seçimi, hataya sebep olmuştur.



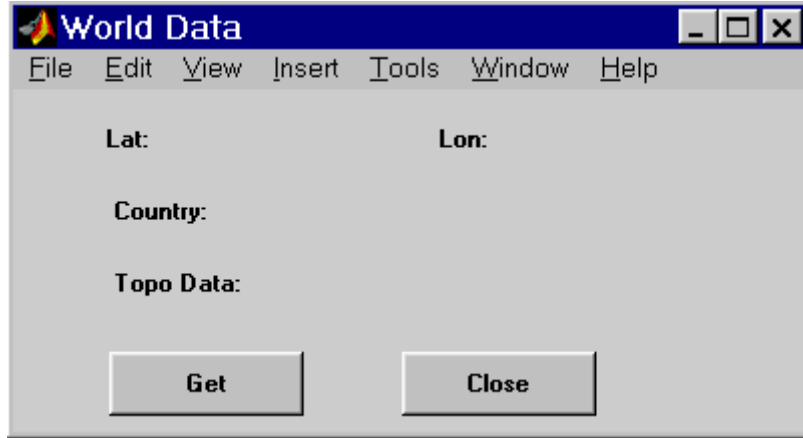
Şekil 81 : MATLAB GUI Hatası-18

DOĞRU 18 : “Current Filter Information” bölümündeki “Filter structure”, “Filter”, “Source”, “Order”, “Stable” ve “Sections” etiketlerindeki (label) metin rengi, mavi olmamalıdır. Çünkü, mavi metin rengi rahat okumayı ve görünümü zorlaştırmaktadır. Bu etiketlerdeki metin rengi, arayüze uygunluğu açısından siyah olmalıdır.



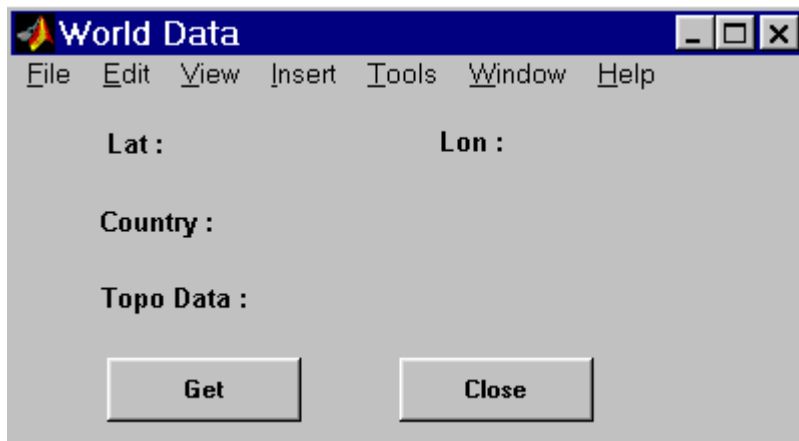
Şekil 82 : MATLAB GUI Hatası-18'in Doğrusu

HATA 19 : Etiketlerin (label) aynı sırada olmamaları ve görüntü düzensizliklerinin bulunması, hataya sebep olmuştur. Bu durum, arayüzün düzen ve görünüşünde bozukluğa ve düzensizliğe yol açtığı gibi, kötü bir görünümde neden olmaktadır.



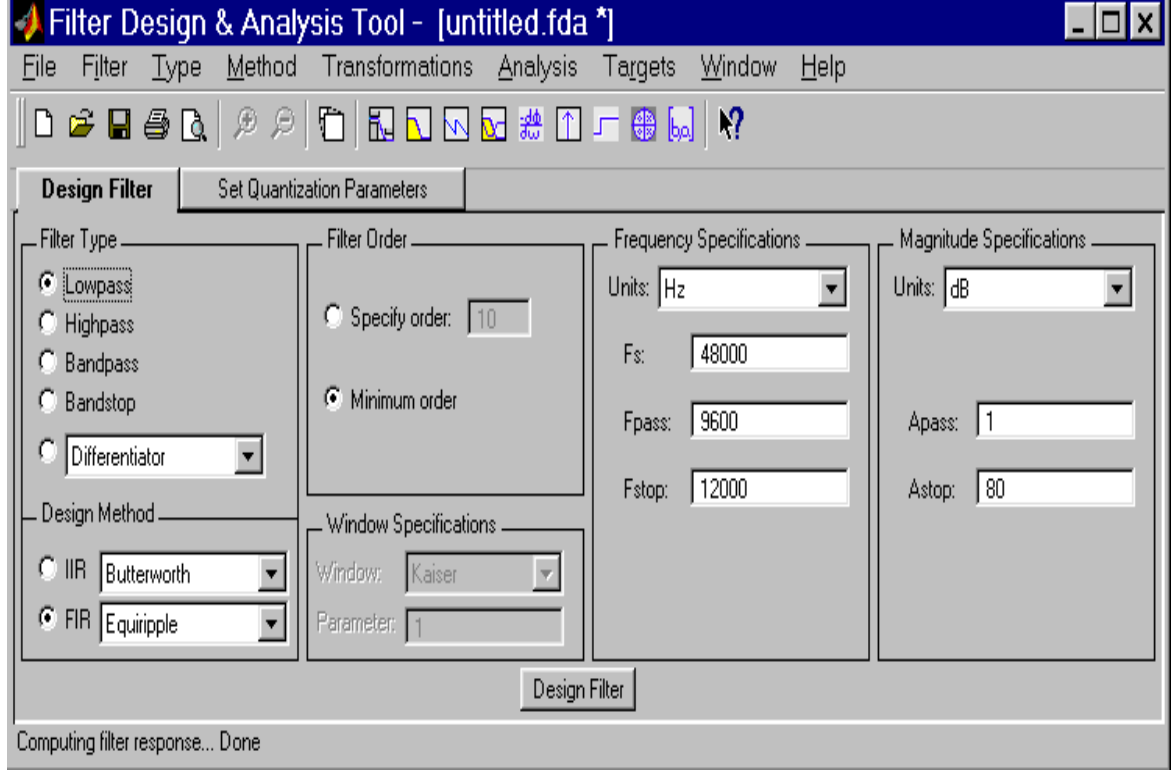
Şekil 83 : MATLAB GUI Hatası-19

DOĞRU 19 : Etiketler (label) aynı hizada olmalıdırlar. Etiketler'in düzenleri ve hizaları keyfe göre olmamalıdır. Etiketler arayüzde belirli bir düzen içerisinde sıralanmalıdırlar.



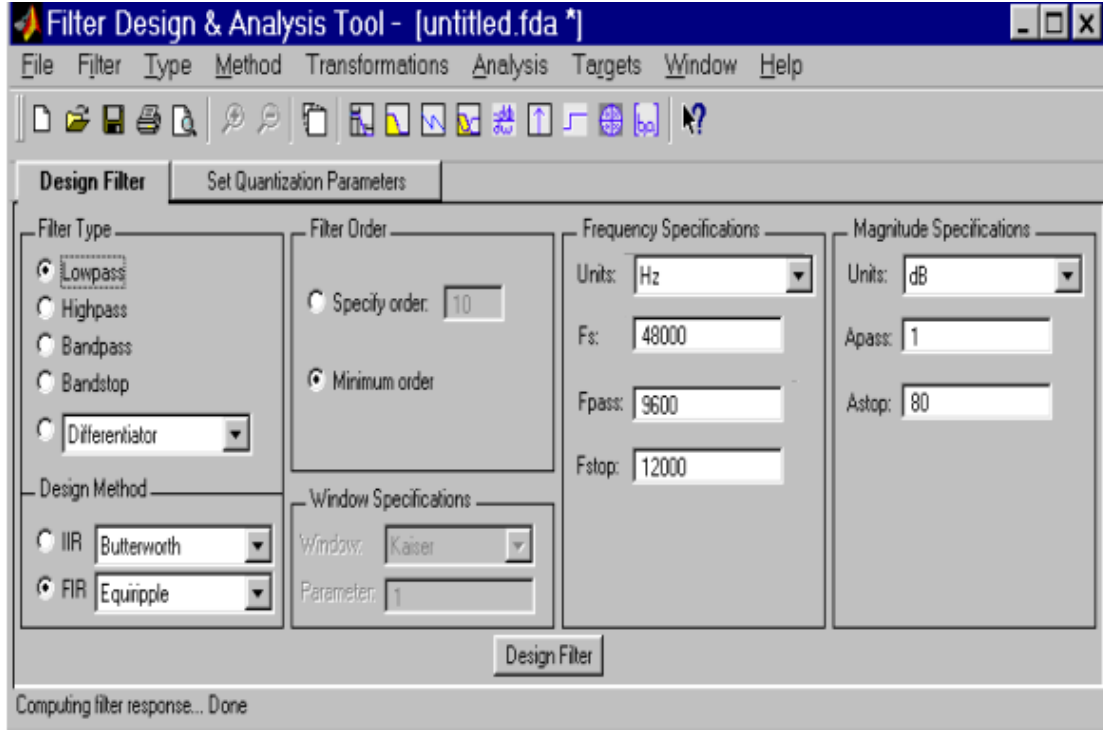
Şekil 84 : MATLAB GUI Hatası-19'un Doğrusu

HATA 20 : “*Frequency Specifications*” ve “*Magnitude Specifications*” bölümlerinde etiket’lerin (label) aynı hizada olmamaları, hataya sebep olmuştur. Bu durum, arayüzde düzen ve görünümün bozulmasına yol açmaktadır.



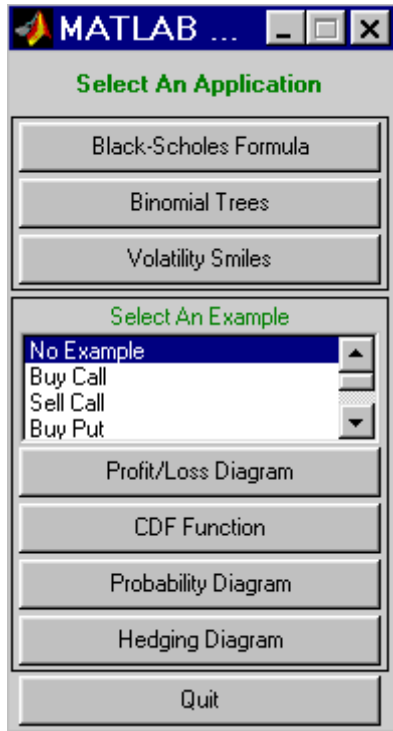
Şekil 85 : MATLAB GUI Hatası-20

DOĞRU 20 : Etiket'lerin (label) arayüzdeki sıralanmaları ve düzenleri tutarlı, düzenli ve göze hoş gelecek şekilde olmalıdır. Etiketlerin bir şekilde sıralanmaları (sola dayalı ya da sağa dayalı) için karar verilmelidir.



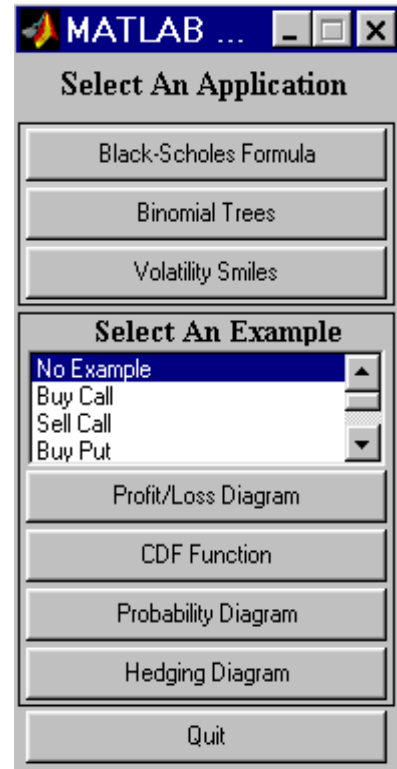
Şekil 86 : MATLAB GUI Hatası-20'nin Doğrusu

HATA 21 : İnsan gözü, kırmızı ve mavi renkle birlikte yeşil renge karşı hassastır. Bu yüzden, metin rengi olarak yeşil renk kullanılmamalıdır. Yeşil renk kullanımı, okumayı zorlaştırdığı gibi görünümü de etkilemektedir. Ayrıca, yeşil renk kullanımı renk körü kişileri de olumsuz yönde etkileyecektir.



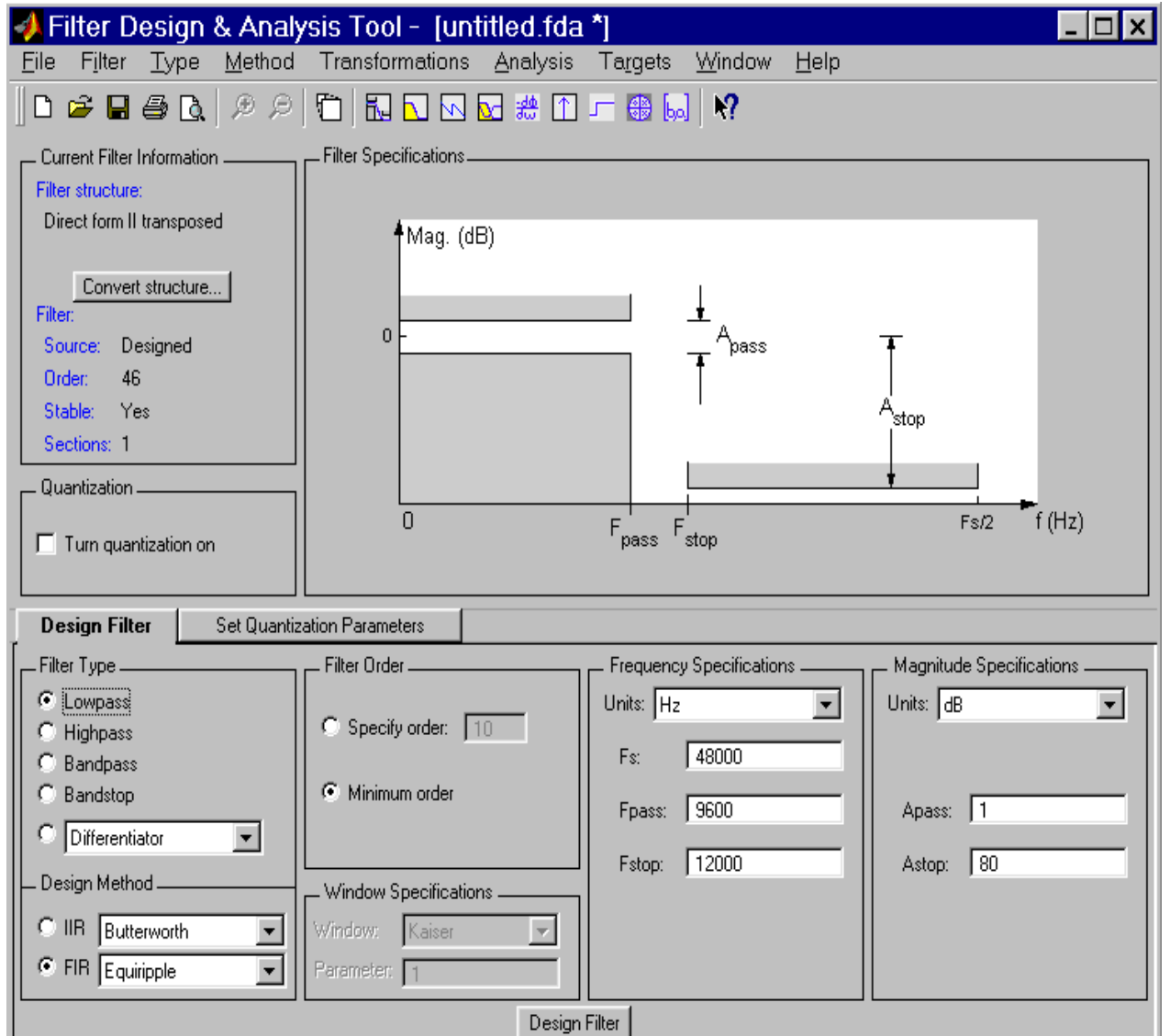
Şekil 87 : MATLAB GUI Hatası-21

DOĞRU 21 : Arayüzde yeşil renk yerine, siyah renk tercih edilmelidir. Eğer metin, kullanıcıların dikkatini çekmek için özellikle vurgulanmak isteniyorsa; yazı tipi sitili, koyu (bold) yapılabilir.



Şekil 88 : MATLAB GUI Hatası-21'in Doğrusu

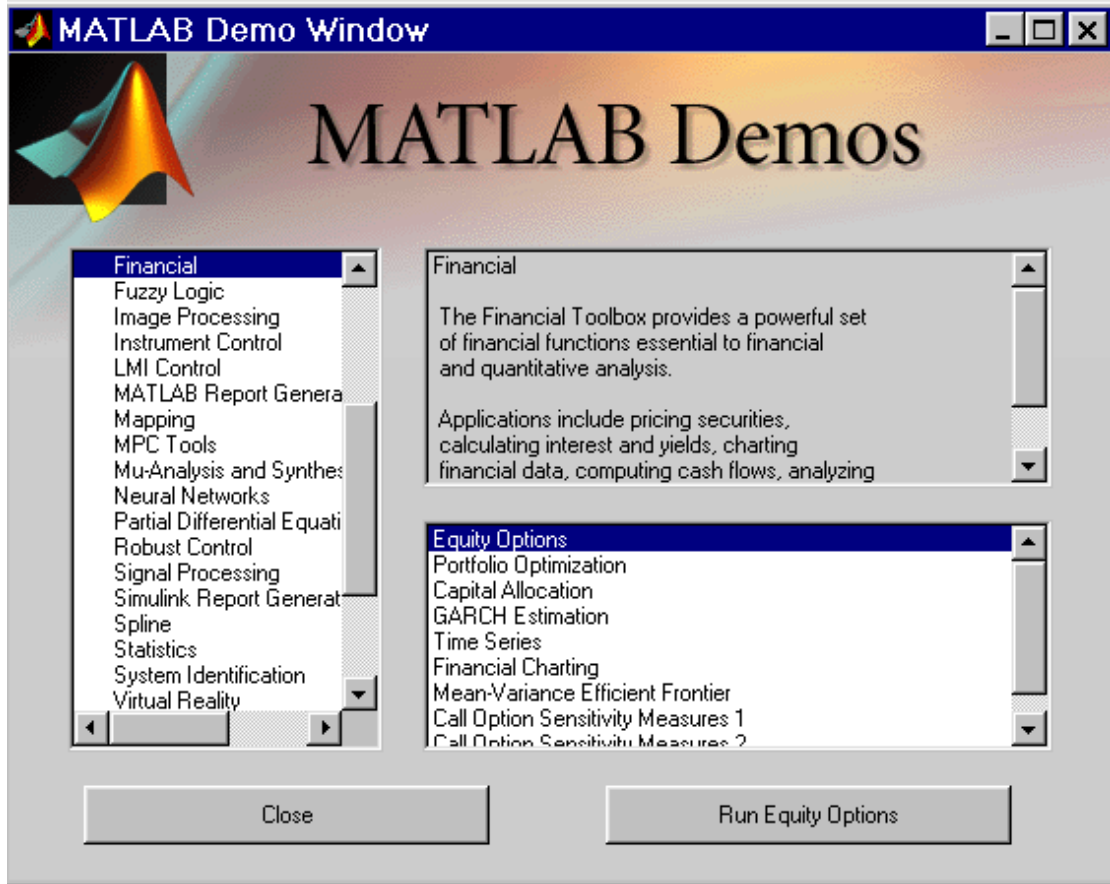
HATA 22 : Hatanın ana sebebi : Arayüz elemanlarının iç içe sokularak arayüzün düzeninin bozulması ve karmaşık bir yapının oluşmasıdır. Bu durum, hem kullanıcıların gözünü yoracak hem de kullanıcıların arayüzü anlamasını zorlaştıracaktır. Ayrıca, arayüzün çok kalabalık olması kullanıcıların hangi arayüz elemanını nerde ve nasıl kullanacaklarını bilememelerine yol açmaktadır. Buda, arayüzün kullanımını zorlaştırmaktadır.



Şekil 89 : MATLAB GUI Hatası-22

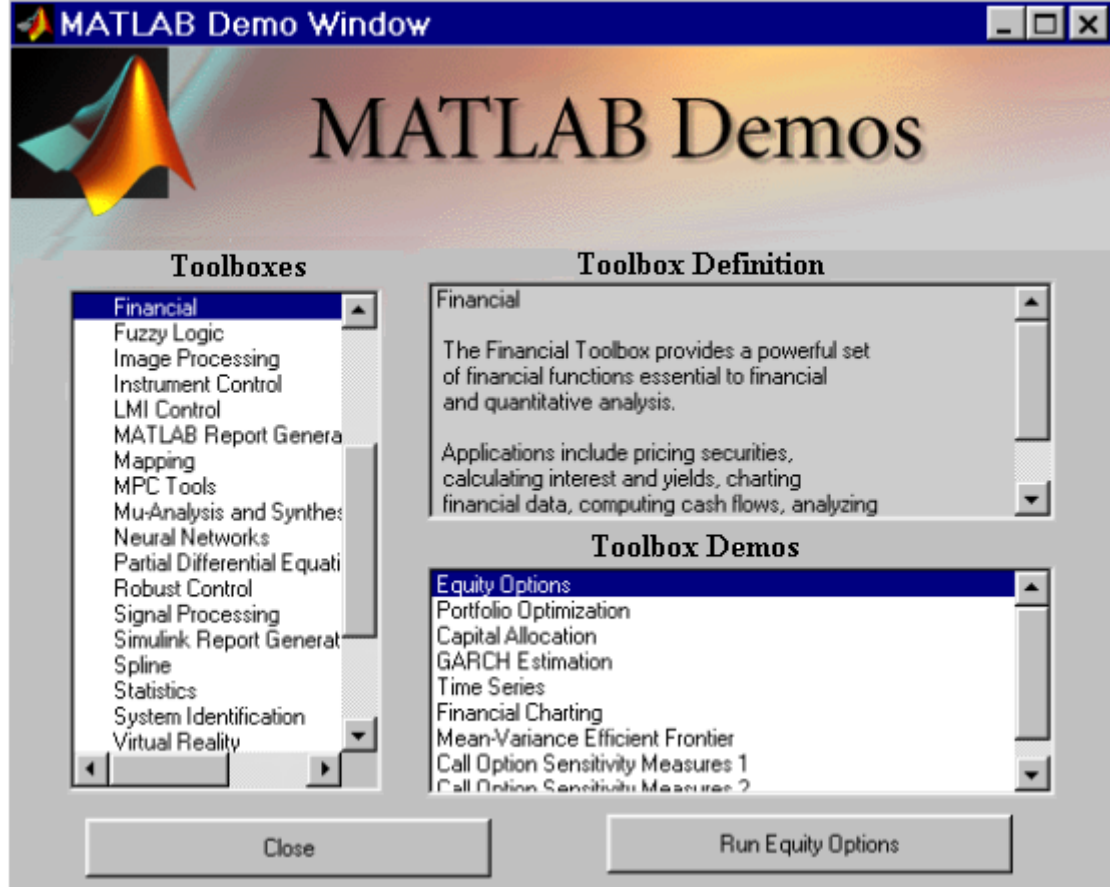
DOĞRU 22 : Kalabalık bir arayüz ekranı, kullanıcının kafasının karışmasına, istediği şeyleri bulamamasına ve dikkatinin dağılmasına neden olacağından; arayüz açık, sade ve anlaşılır bir yapıda olmalıdır. Arayüz, karışık ve yoğun bir bilgi yığını içerisinde olmamalıdır. Bir arayüz ekranının çok gereksiz şekilde kalabalık olması, kullanmayı zorlaştırır.

HATA 23 : Metin alanı'nda (text area) başlık kullanılmaması, hataya sebep olmuştur. Çünkü böyle bir durumda kullanıcı ne yapacağını bilemez ve metin alanı içindeki bilgilerin ne işe yaradığını anlayamaz. Bu durum, arayüzün kullanımını zorlaştıracaktır.



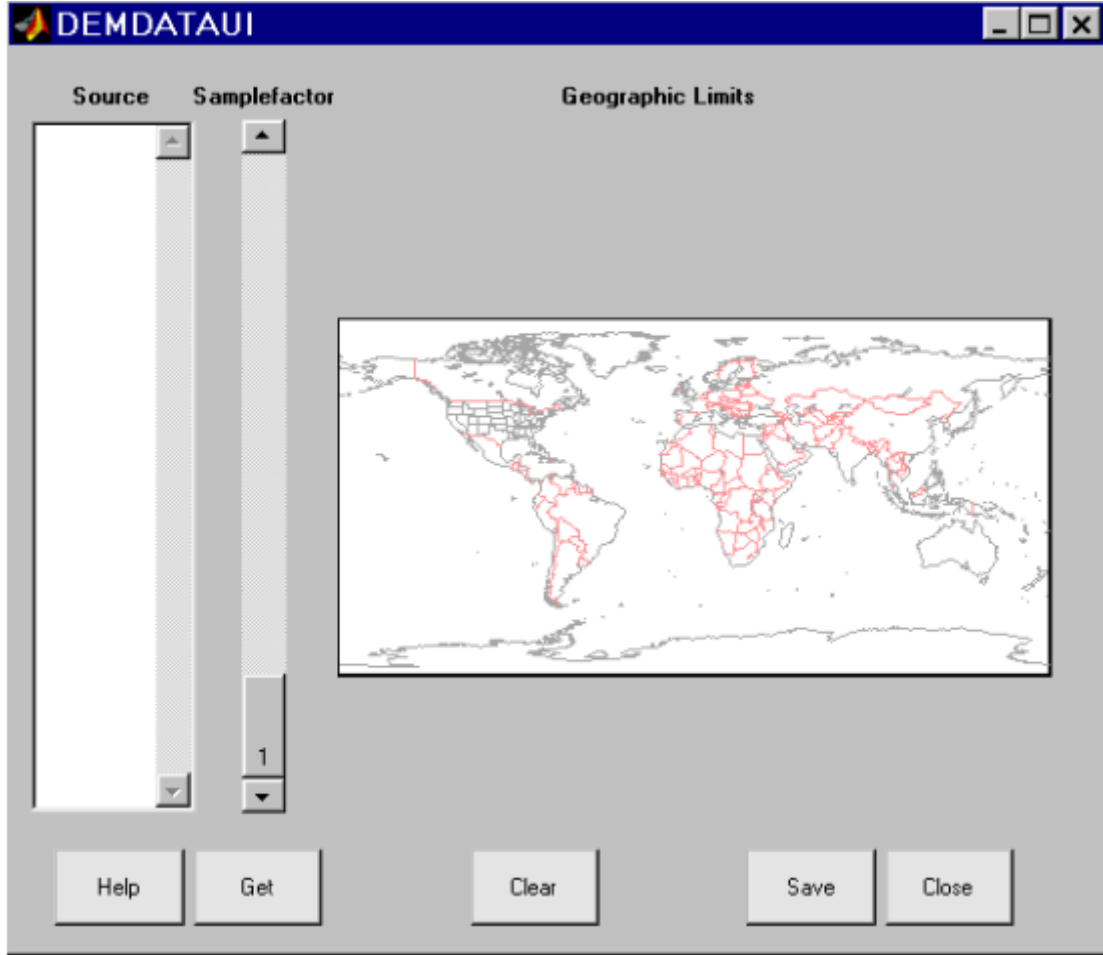
Şekil 90 : MATLAB GUI Hatası-23

DOĞRU 23 : Metin alanı'nda (text area), mutlaka başlık kullanılmalıdır. Böylece metin alanı'nın amacının ne olduğu ve niçin kullanıldığı anlaşılabilir olur. Metin alanı'nda başlık kullanılması, kullanıcının ne yapması gerektiği konusunda, kullanıcıya yardımcı olur. Ayrıca metin alanı'ndaki başlık, metin alanı'nın kullanılma konusu hakkında kullanıcıyı bilgilendirir.



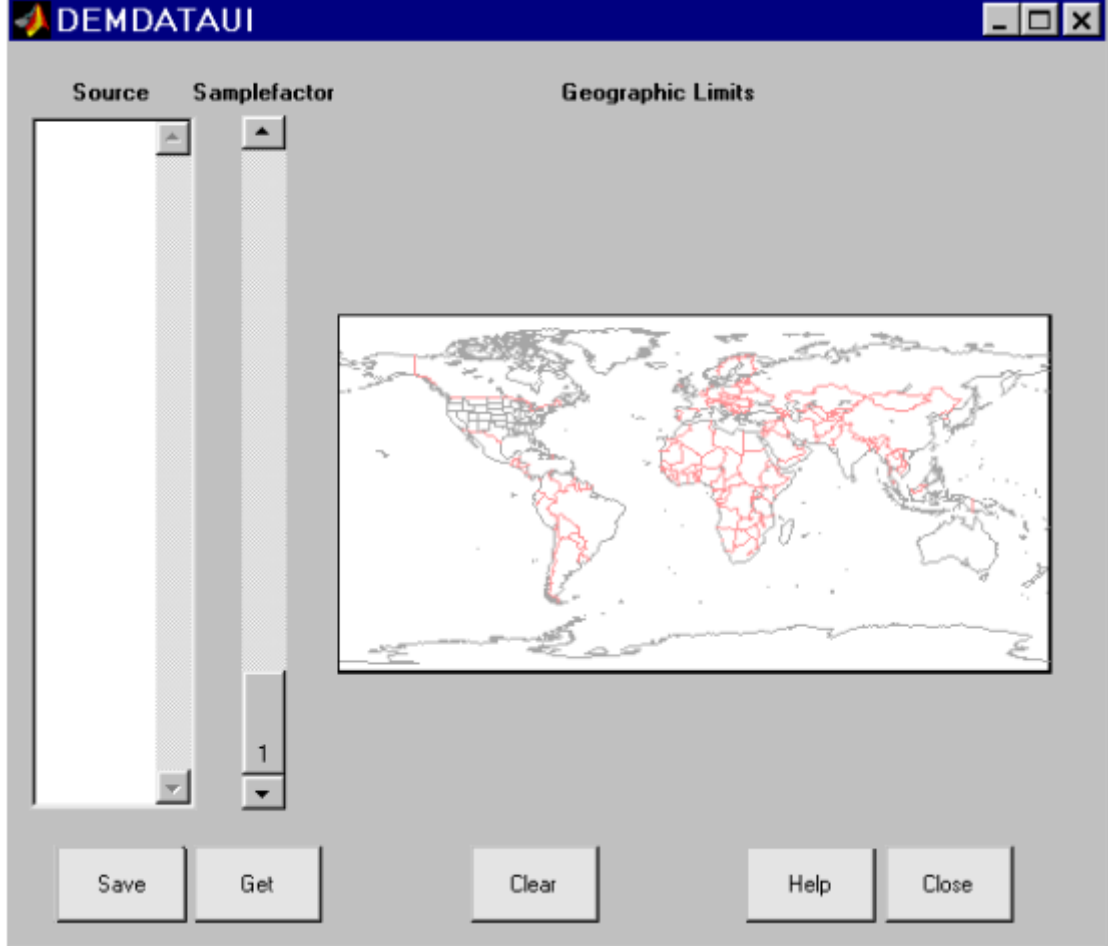
Şekil 91 : MATLAB GUI Hatası-23'ün Doğrusu

HATA 24 : Diyalog kutusunda (dialog box), “OK”, “Cancel”, “Close”, “Help” ve “Apply” düğmelerinin (button) diğer düğmelerden ayrı olmamaları ve diğer düğmelerle iç içe bulunmaları, hataya neden olmuştur.



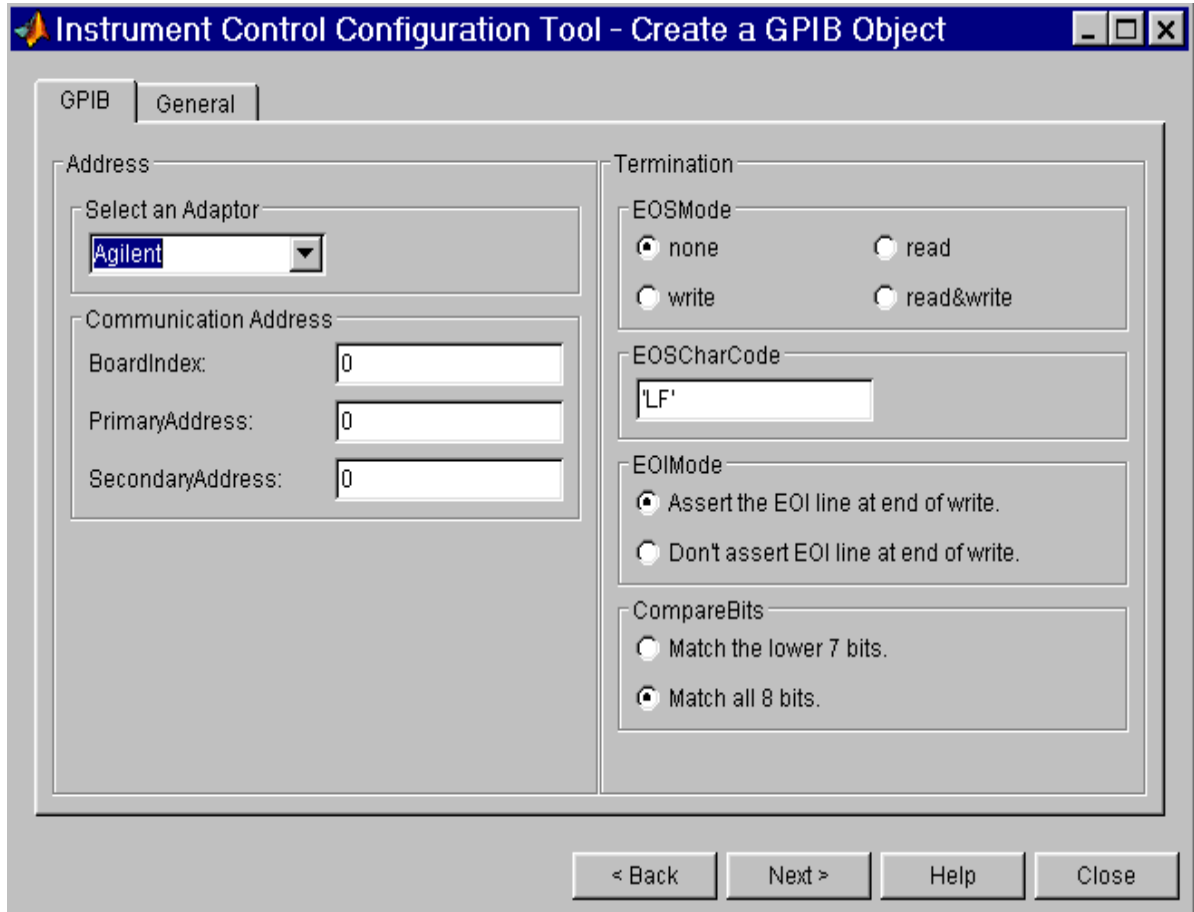
Şekil 92 : MATLAB GUI Hatası-24

DOĞRU 24 : “OK”, “Cancel”, “Close”, “Help” ve “Apply” düğmeleri (button), arayüzde kullanıcılar tarafından çok kullanıldığı ve daha fazla ihtiyaç duyulduğu için; kendi aralarında birarada, diğer düğmelerden ise ayrı olmalıdırlar. Bu yüzden aşağıdaki diyalog kutusunda (dialog box), “Help” ve “Close” düğmeleri (button) bitişik ve yan yana olmalıdır.



Şekil 93 : MATLAB GUI Hatası-24’ün Doğrusu

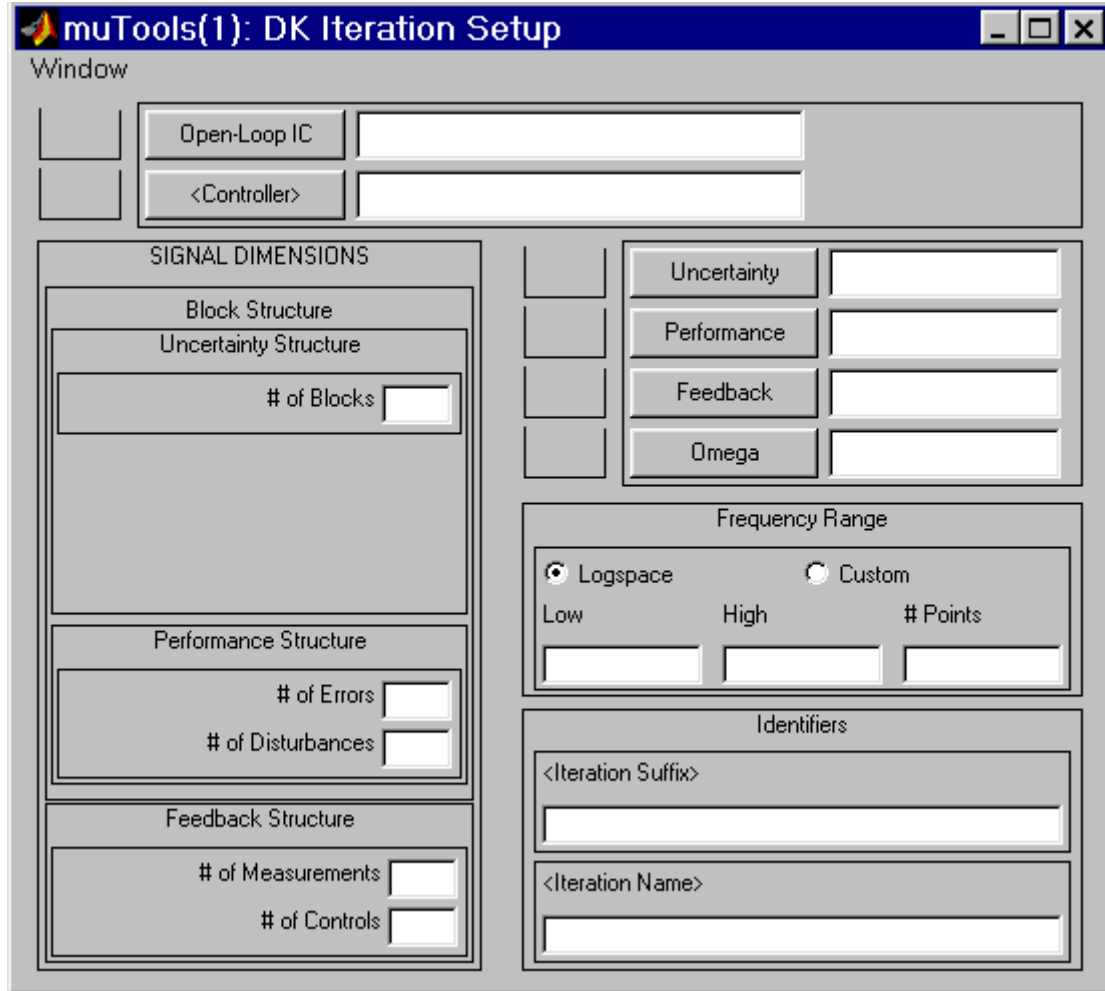
HATA 25 : Çerçeve (frame) içerisinde çerçeve kullanılması, hataya yol açmıştır. Herşeyin bir çerçeve etrafında toplanmış olması, görünümü bozmaktadır. Arayüzde birinci çerçeve “*EOSMode*”, ikinci çerçeve “*EOSCharCode*”, üçüncü çerçeve “*EOIMode*” ve dördüncü çerçeve ise “*CompareBits*” tir. “*Termination*” çerçevesi ise, bu dört çerçeveye de içine almıştır. İç içe çerçevelerin olması, kötü bir görünüme neden olduğu gibi, arayüzün anlaşılmasını ve kullanılmasını da zorlaştırmaktadır.



Şekil 94 : MATLAB GUI Hatası-25

DOĞRU 25 : Arayüzdeki dört çerçeveyi (farme) içine alan “*Termination*” çerçevesi, gereksizdir. Bu yüzden kaldırılması gerekir. Çerçeve yerine etiket (label) kullanılması, görünümün düzelmesine yardım edecektir.

HATA 26 : Gereksiz yere çok fazla sayıda çerçeve (frame) kullanılması, arayüzde görünümü bozduğu gibi, karışıklığa da yol açmaktadır. Çerçevelerin iç içe olması, görünüm bozukluğuna yol açmaktadır. Arayüz elemanlarını iç içe sokmak, arayüz düzeninin bozulmasına sebep olmaktadır. Ayrıca etiketlerin (label) dizilme ve düzenlerindeki hatalar, bir diğer sorundur.

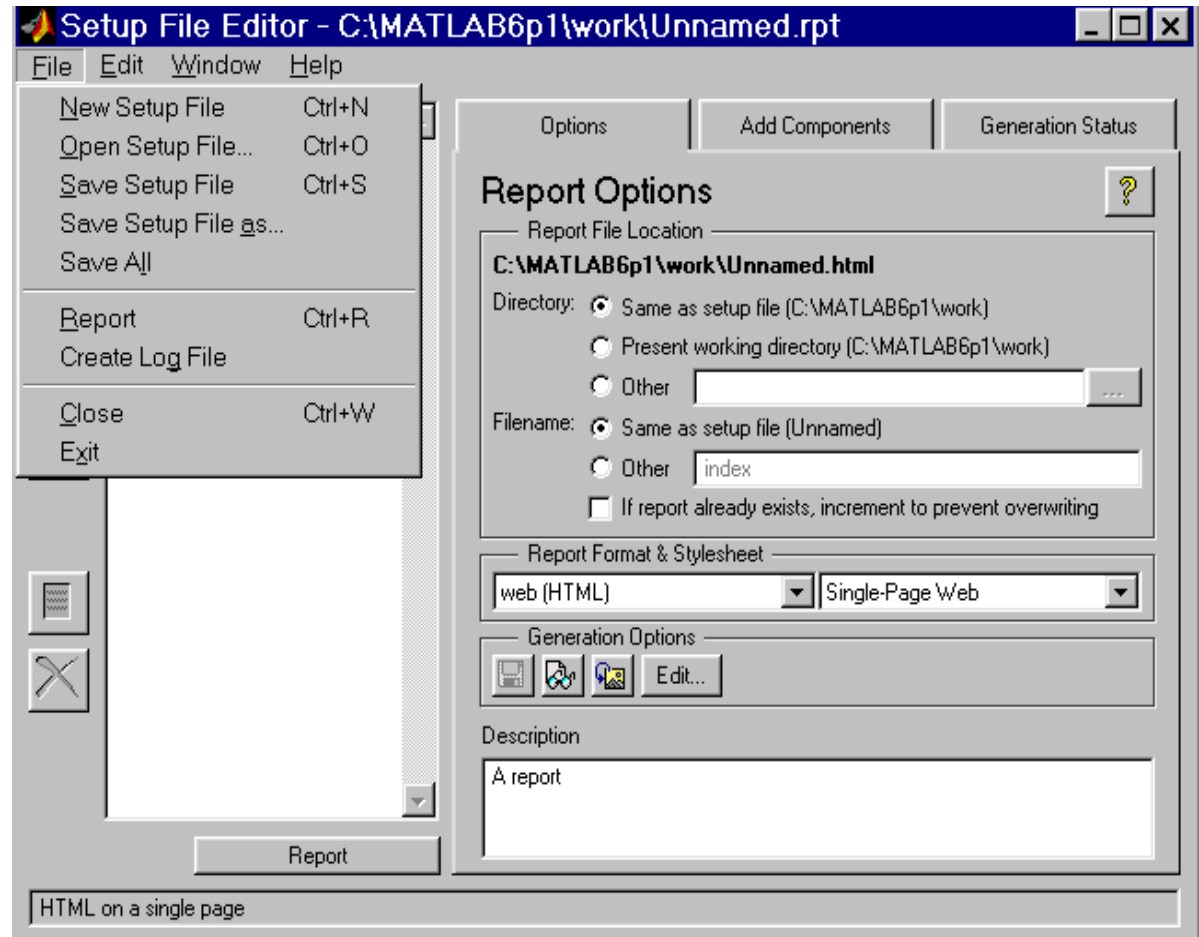


Şekil 95 : MATLAB GUI Hatası-26

DOĞRU 26 : Arayüz elemanlarının belirli bir düzen ve plan içerisinde arayüzde sıralanmaları gerekmektedir. Gereksiz çerçevelerden (frame) kurtulmak, arayüzün görünümünün düzelmesine yardımcı olacaktır. Çerçeveleri iç içe sokmaktan kaçınmak gerekir. Ayrıca, etiketlerin (label) belirli bir yöne göre sıralanmaları ve etiketlerden sonra (:) işaretinin kullanılması gerekmektedir.

4.4.3 MATLAB'teki Metinsel Hatalar (Textual Bloopers)

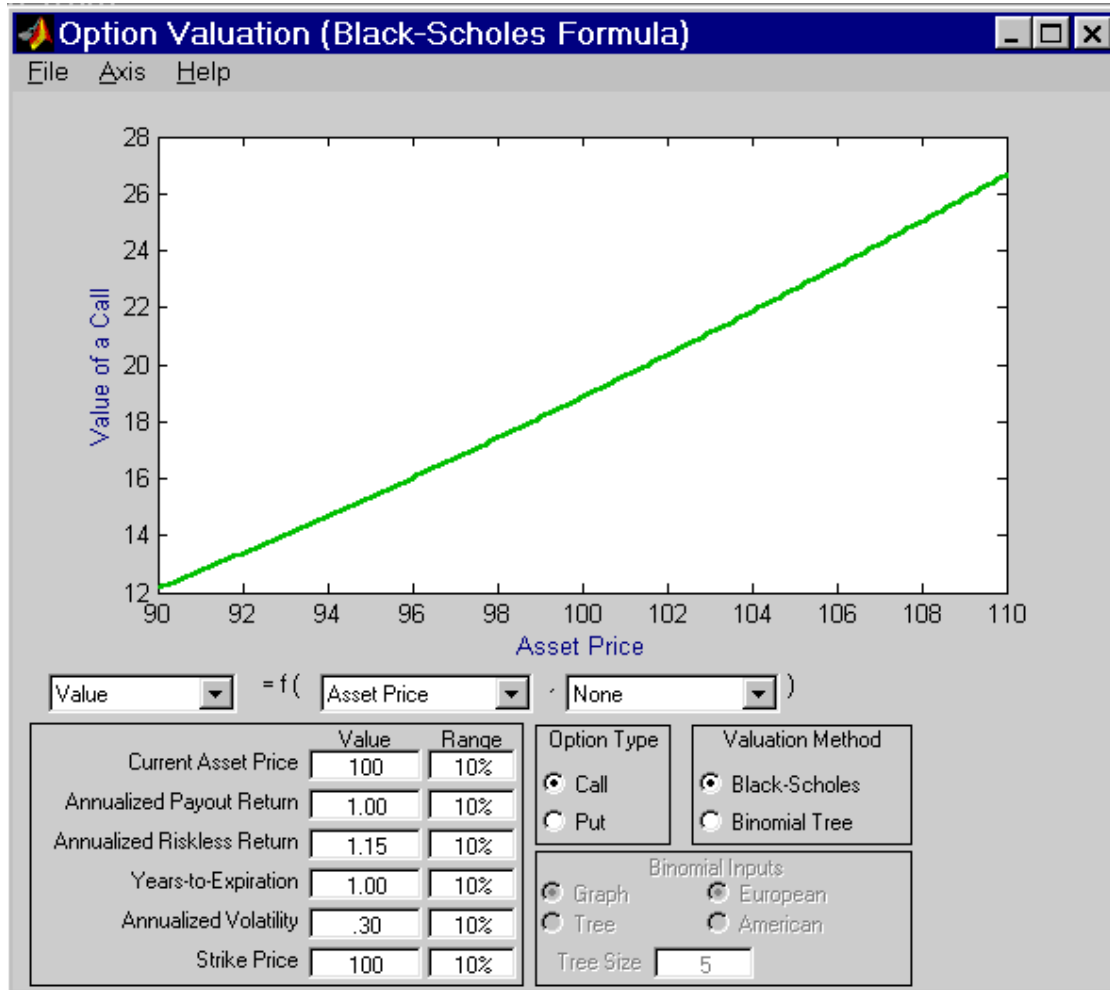
HATA 27 : Aynı kavram için farklı terimlerin kullanılması, hataya sebep olmuştur. Tutarsız terminoloji, hatanın ana sebebidir. “File” menüsü içindeki alt menüler, “Close” ve “Exit” benzer anlam içerdiklerinden, burda metinsel bir hata yapılmıştır.



Şekil 96 : MATLAB GUI Hatası-27

DOĞRU 27 : Benzer ve anlam karışıklığına yol açacak, kullanıcıların kafasını karıştıracak terimlerden kaçınmak gerekir. Arayüzde, aynı anlamı taşıyan terimler kullanılmamalıdır.

HATA 28 : Etiket'ten (label) sonra (:) işaretinin kullanılmaması, hataya yol açmıştır. Bununla birlikte arayüzdeki bir başka hata : Etiket'lerin düzensiz sıralanması ve hizaya dikkat edilmemesi görünümde bozukluğa yol açmıştır.



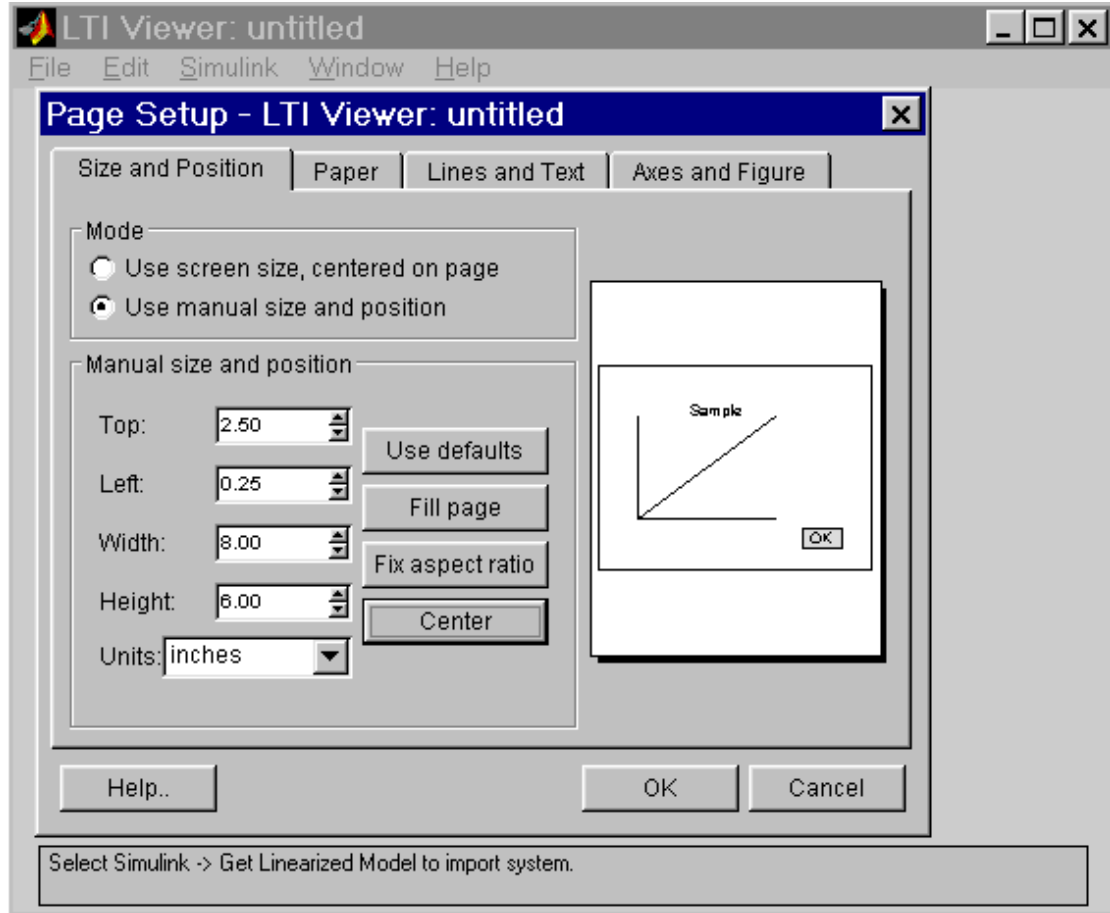
Şekil 97 : MATLAB GUI Hatası-28

DOĞRU 28 : Her etiket'ten (label) sonra, (:) işareti kullanılmalıdır. Ayrıca, etiketler belirli bir düzen içerisinde olmalıdırlar. (:) işaretinin kullanılma şekli ile etiketlerin düzen ve hizaları aşağıdaki gibi olmalıdır :

	Value	Range
Current Asset Price :	100	10%
Annualized Payout Return :	1.00	10%
Annualized Riskless Return :	1.15	10%
Years-to-Expiration :	1.00	10%
Annualized Volatility :	.30	10%
Strike Price :	100	10%

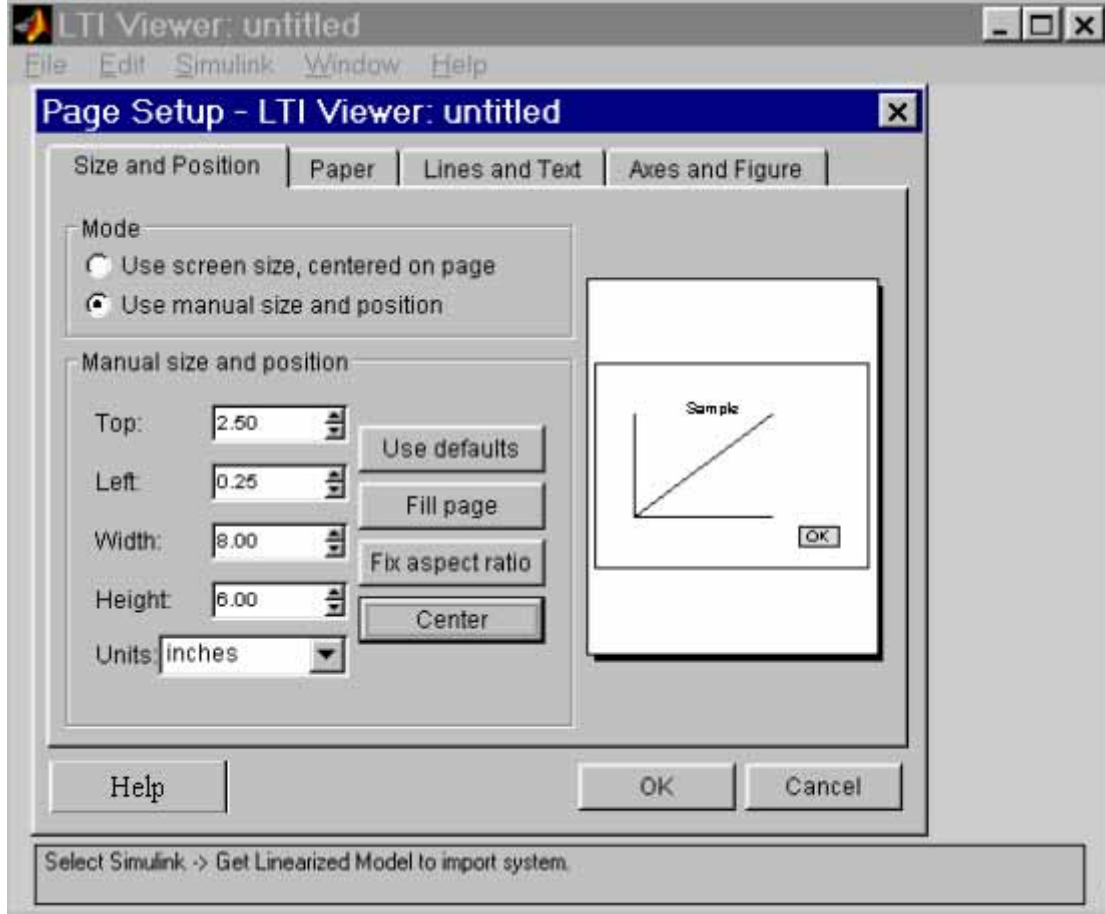
Şekil 98 : MATLAB GUI Hatası-28'in Doğrusu

HATA 29 : “*Help*” düğmesinde (button) “...” yorum satırının yanlış kullanılması, hataya sebep olmuştur. “...” işareti, yeni bir pencerenin açılması suretiyle kullanıcıya daha fazla bilgi vermek için kullanılır. “*Help*” düğmesi, zaten yardım anlamını içermektedir. Yani, “*Help*” düğmesine basan bir insan yeni bir pencerenin açılacağını ve o konu hakkında kullanıcıya yardım edileceğini bilir. Bundan dolayı, “*Help*” düğmesinde “...” işareti gereksiz yere kullanılmıştır.



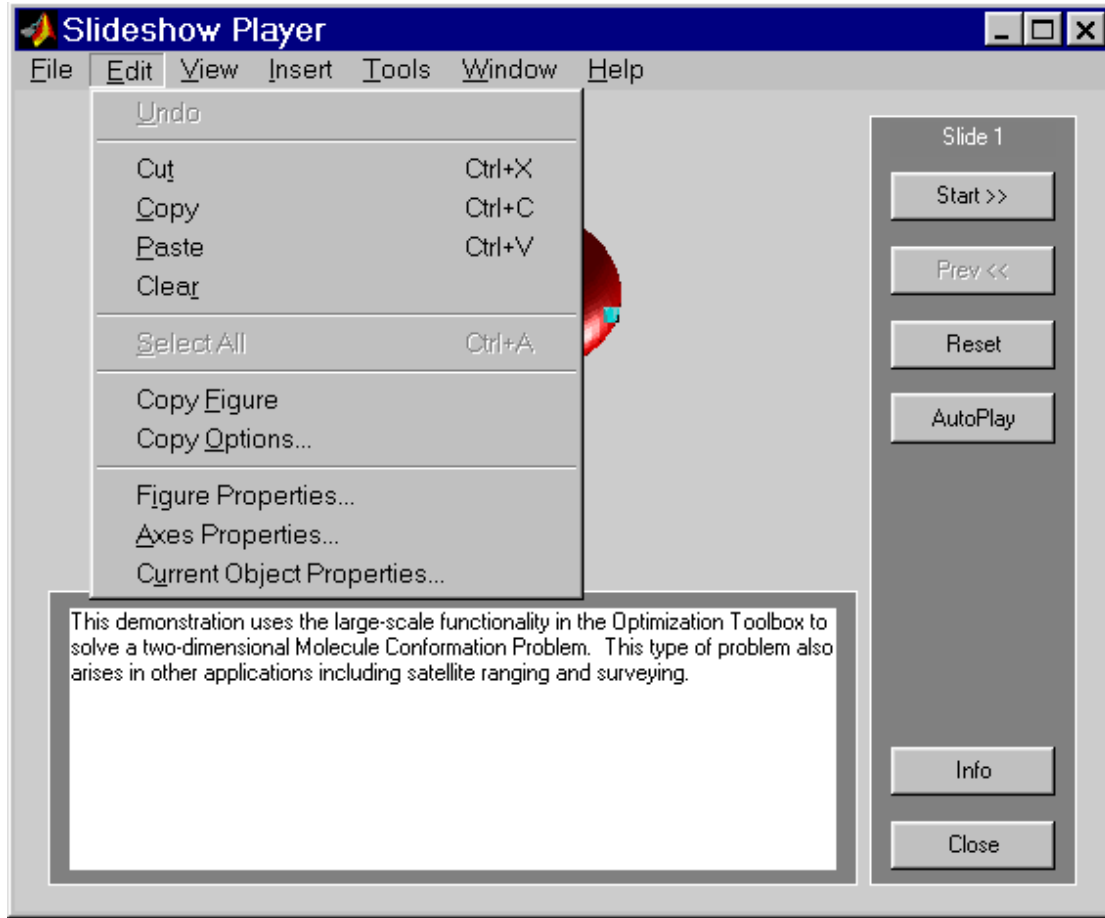
Şekil 99 : MATLAB GUI Hatası-29

DOĞRU 29 : “...” işareti, eylemin devam ettiğini ve bu eylem ile ilgili ayrıntılı bilgileri yeni pencereler kullanarak görme olanağı sağlamak için kullanılır. “Help” düğmesine (button) basılınca kullanıcıya yardım etmek için zaten yeni pencerede yeni bir bilgi sunulacaktır. Bu yüzden, “Help” düğmesinde, “...” işaretinin kullanılması gereksizdir ve kaldırılması gerekir.



Şekil 100 : MATLAB GUI Hatası-29’un Doğrusu

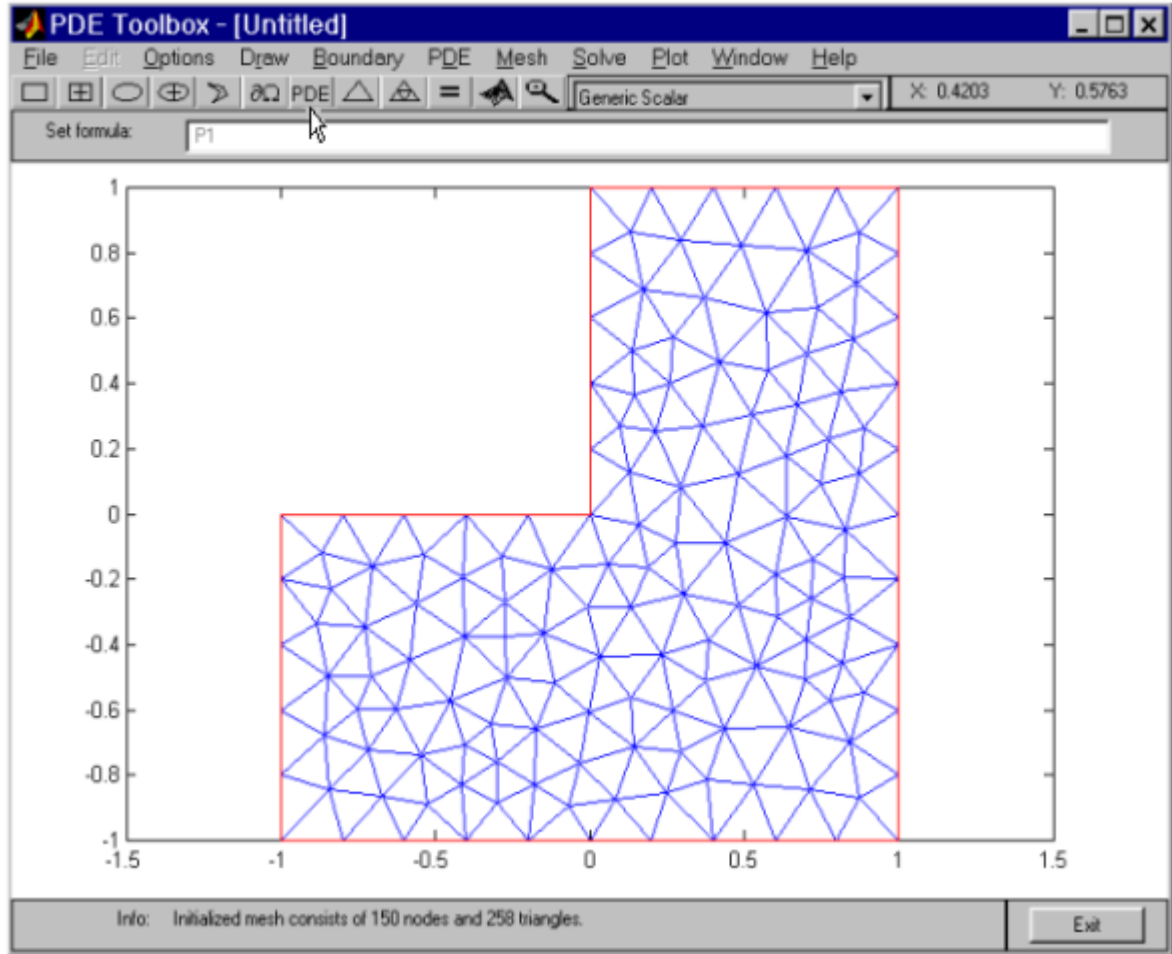
HATA 30 : Bir menü içerisinde aynı anlamlı iki alt menünün kullanılması, hataya sebep olmuştur. “Copy” ve “Copy Figure” kelimeleri farklı yazılışlı olmalarına rağmen, yaptıkları işler ve anlamları aynıdır.



Şekil 101 : MATLAB GUI Hatası-30

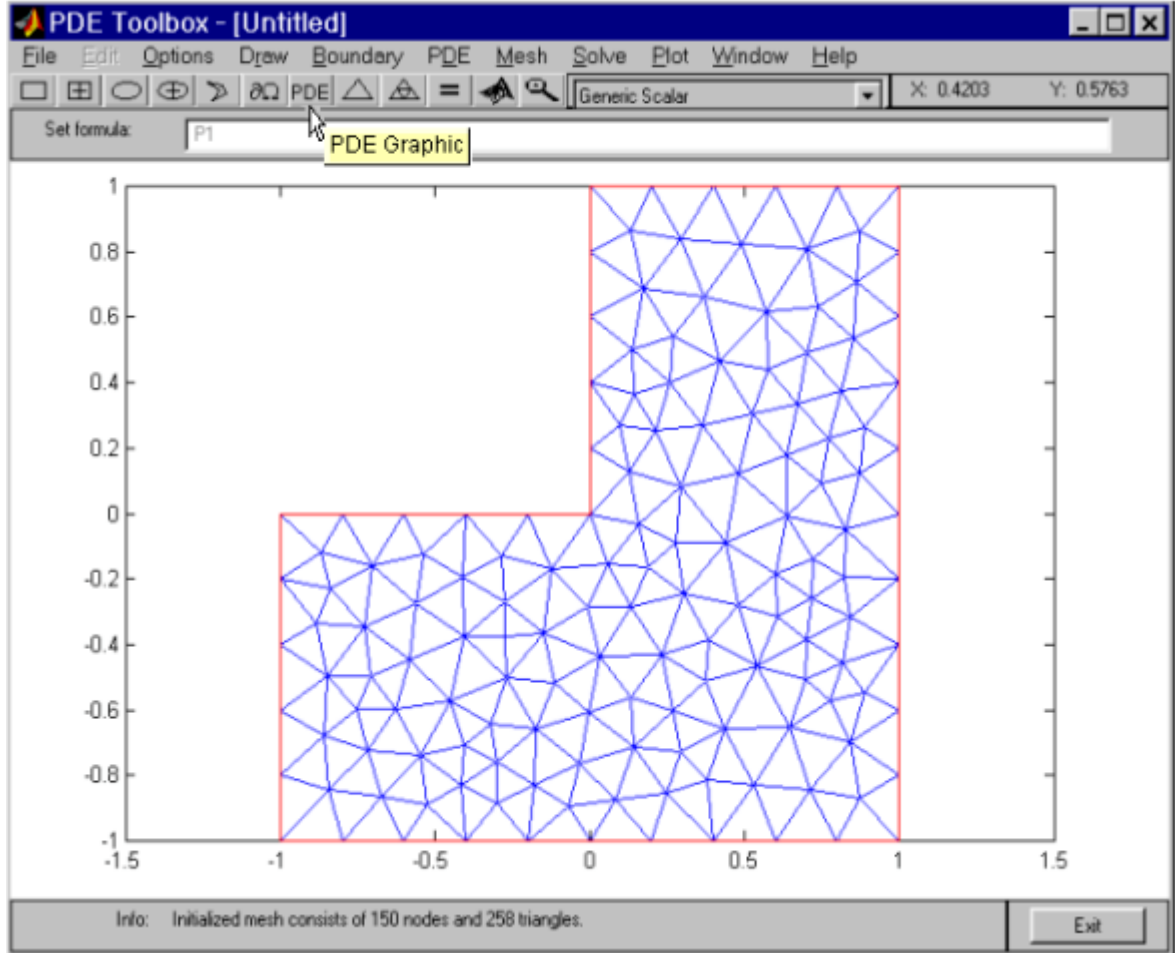
DOĞRU 30 : Bir menü içerisindeki alt menüler farklı isimli ve farklı anlamlı olmalıdır. Menü isimlerinde benzer terimlerden kaçınmak gerekir. Alt menülerin herbiri farklı işlevler içermelidir. Böylece, kullanıcının kafasının karışması önlenmiş olacaktır.

HATA 31 : İkon'ların (icons) üzerine mouse (fare) ile gelindiğinde *tool tip*'lerin (düğme ya da ikonların üzerine gelindiği zaman düğmenin ya da ikon'un ne işe yaradığını gösteren açıklayıcı bilgi) görünmemesi, hataya sebep olmuştur. İkonlar için açıklayıcı bilginin (*tool tip*) kullanılmaması sonucu, kullanıcı hangi ikonun ne işe yaradığını anlayamayacaktır. Bu durum, arayüzün karışık bir yapıda olmasına ve kullanıcıların arayüzü kullanırken zorlanmalarına sebep olacaktır.



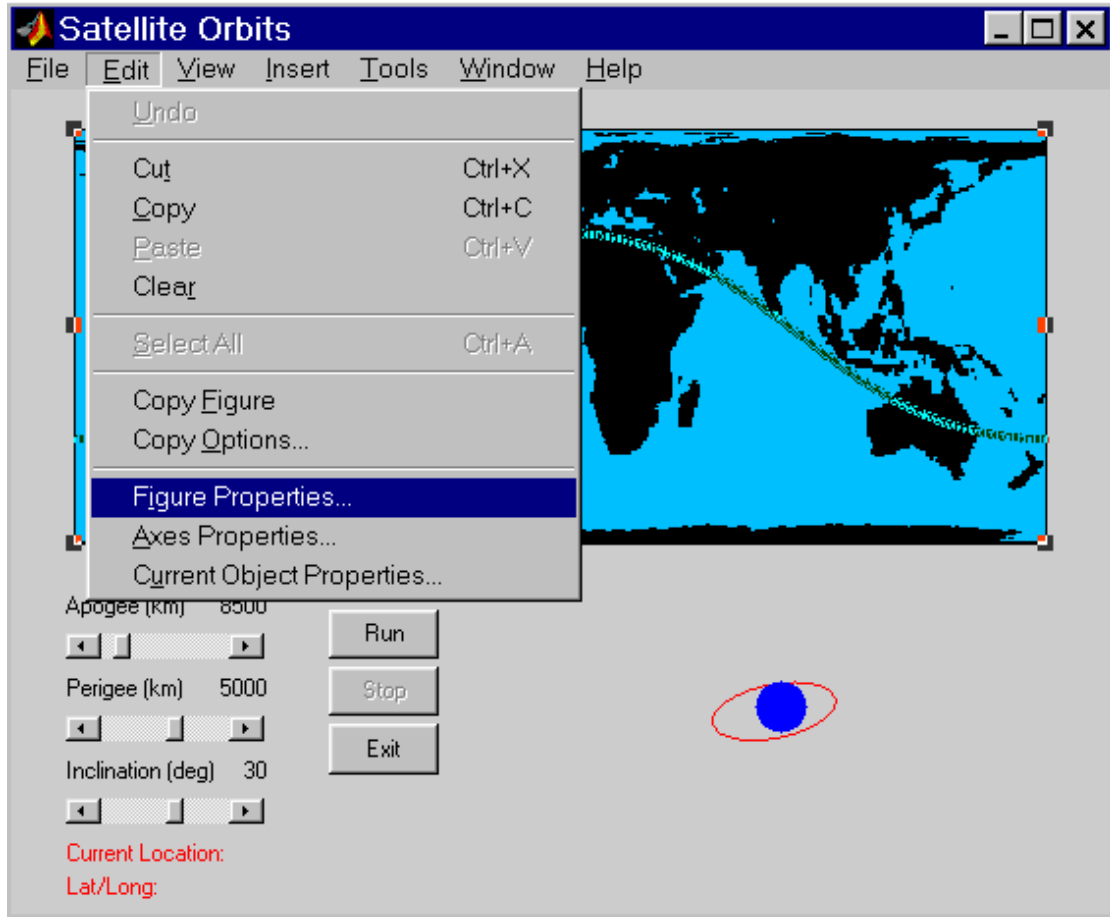
Şekil 102 : MATLAB GUI Hatası-31

DOĞRU 31 : İkonların (icons) üzerine gelindiğinde, *tool tip* (açıklayıcı bilgi) kullanılmalıdır. Kullanıcıya ikon hakkında bilgi verilmelidir. Böylece kullanıcı, arayüzü kullanırken hangi ikonun ne işe yaradığını anlamış olacaktır. Bu durum, arayüzün kolay kullanılmasını sağlayacaktır.

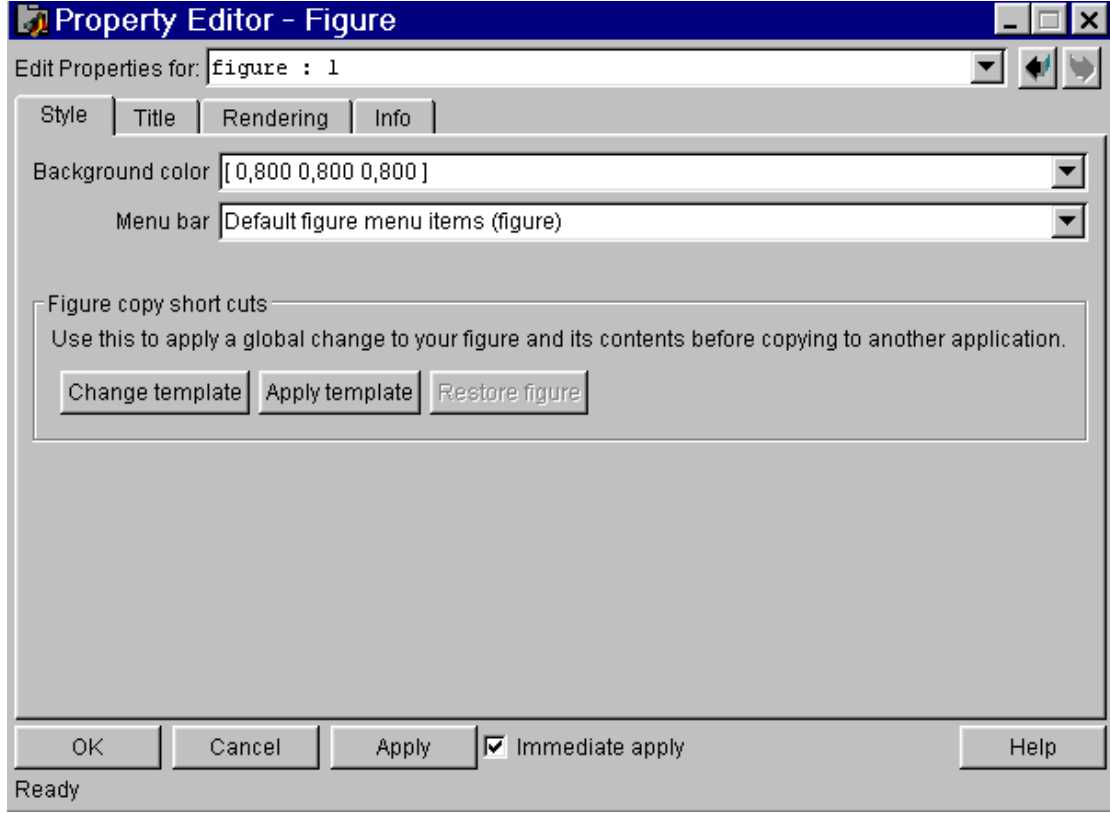


Şekil 103 : MATLAB GUI Hatası-31'in Doğrusu

HATA 32 : Pencere başlığının farklı olması, hatanın oluşmasına sebep olmuştur. Birinci arayüzde, “Edit” menüsündeki “Figure Properties...” alt menüsüne tıklanıldığında yeni açılan pencerenin başlığının “Property Editor - Figure” olması karışıklığa yol açmıştır. Böyle bir durumda, kullanıcı yeni açılan sayfanın hangi sayfanın devamı olduğunu bilemeyecektir.



Şekil 104 : MATLAB GUI Hatası-32'nin Birinci Arayüzü

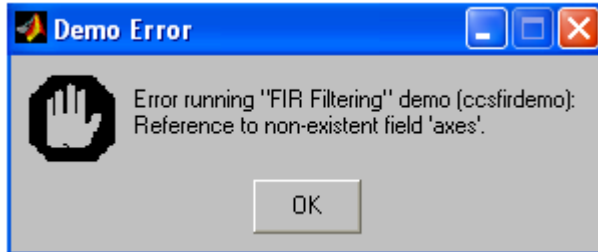


Şekil 105 : MATLAB GUI Hatası-32'nin İkinci Arayüzü

DOĞRU 32 : Menü adı ile açılan pencere başlığı aynı olmak zorundadır. Yani birinci arayüzde, “*Edit*” menüsündeki “*Figure Properties...*” alt menüsüne tıklanıldığında yeni açılan pencerenin başlığının “*Figure Properties*” olması gerekmektedir.

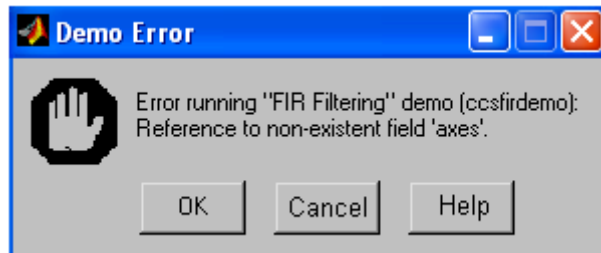
4.4.4 MATLAB'teki Etkileşim Hataları (Interaction Bloopers)

HATA 33 : Verilen hata mesajında, “OK” düğmesinin (button) yanlış amaçta kullanılması hataya sebep olmuştur. Bu durum, kullanıcıyı yanıltmaktadır. “OK” düğmesi “Cancel” düğmesi gibi kullanılmıştır.



Şekil 106 : MATLAB GUI Hatası-33

DOĞRU 33 : Hata mesajında “OK”, “Cancel” ve “Help” düğmelerinin olması gerekmektedir. “OK” düğmesi, “Cancel” gibi kullanılmamalıdır.



Şekil 107 : MATLAB GUI Hatası-33'ün Doğrusu

HATA 34 : Ana sorun : Verilen hata mesajında kullanıcıya yardım edilmemesidir. Yaptığı yanlışın ne olduğu konusunda kullanıcıya yardım edilmemesi ve kullanıcının bilgilendirilmemesi hataya neden olmuştur.



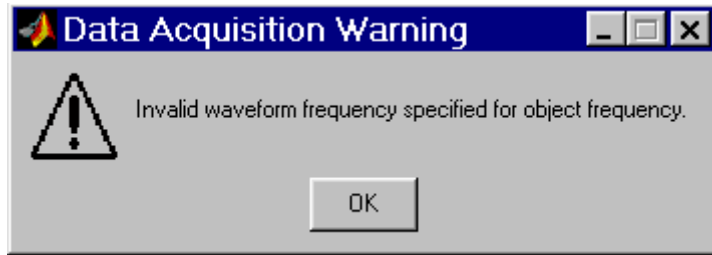
Şekil 108 : MATLAB GUI Hatası-34

DOĞRU 34 : Verilen mesajda hatanın ortadan kaldırılması için “*Help*” (yardım) düğmesinin (button) de mesajda olması gerekmektedir. Böylece mesaj, kullanıcının ne yapması gerektiğini söylemiş olup, kullanıcıyı uyaracaktır.



Şekil 109 : MATLAB GUI Hatası-34’ün Doğrusu

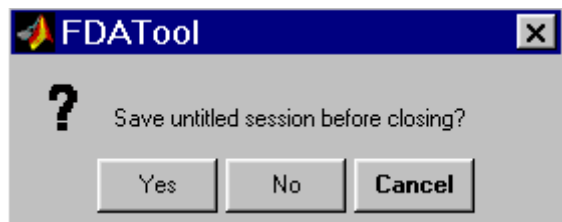
HATA 35 : Sorun kötüye giderken, “*OK*” düğmesine (button) basmak, hataya neden olmaktadır. Mesajda kullanıcıya sunulan seçimin, problemin çözümü için yardımcı olmaması, bir etkileşim hatasına sebep olmuştur.



Şekil 110 : MATLAB GUI Hatası-35

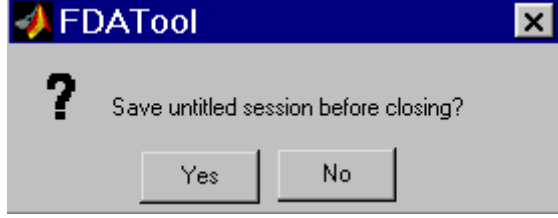
DOĞRU 35 : Mesajın kullanıcıya çok daha fazla seçenek sunması ve yardım etmesi gerekir. Bu durum, kullanıcının hatanın neden olduğunu anlamasına ve hata karşısında ne yapması gerektiğini bilmesine yardımcı olacaktır.

HATA 36 : Mesajda gereksiz düğmelerin (button) kullanılması, hataya neden olmuştur. Mesajda, “*No*” düğmesi ile “*Cancel*” düğmesi aynı işi yapmaktadır. Kullanıcının, “*No*” ya da “*Cancel*” düğmelerinden herhangi birine basması; eylemin iptal olacağı, işlemin yapılmak istenmemesi ve yapılan değişikliklerin yok sayılması anlamlarına gelmektedir. Bu yüzden iki düğmeden birinin kullanılması gereksizdir.



Şekil 111 : MATLAB GUI Hatası-36

DOĞRU 36 : Mesajda, aynı işlevi yapan “No” ve “Cancel” düğmelerinden (button) bir tanesinin hiçbir şey yapmadığı ve gereksiz olduğu durumunu önlemek gerekmektedir. Bundan dolayı, “Cancel” düğmesinin kaldırılması gerekir.



Şekil 112 : MATLAB GUI Hatası-36'nın Doğrusu

HATA 37 : Mesajda, “OK” düğmesinin (button), “Cancel” düğmesi gibi kullanılması hataya neden olmuştur.



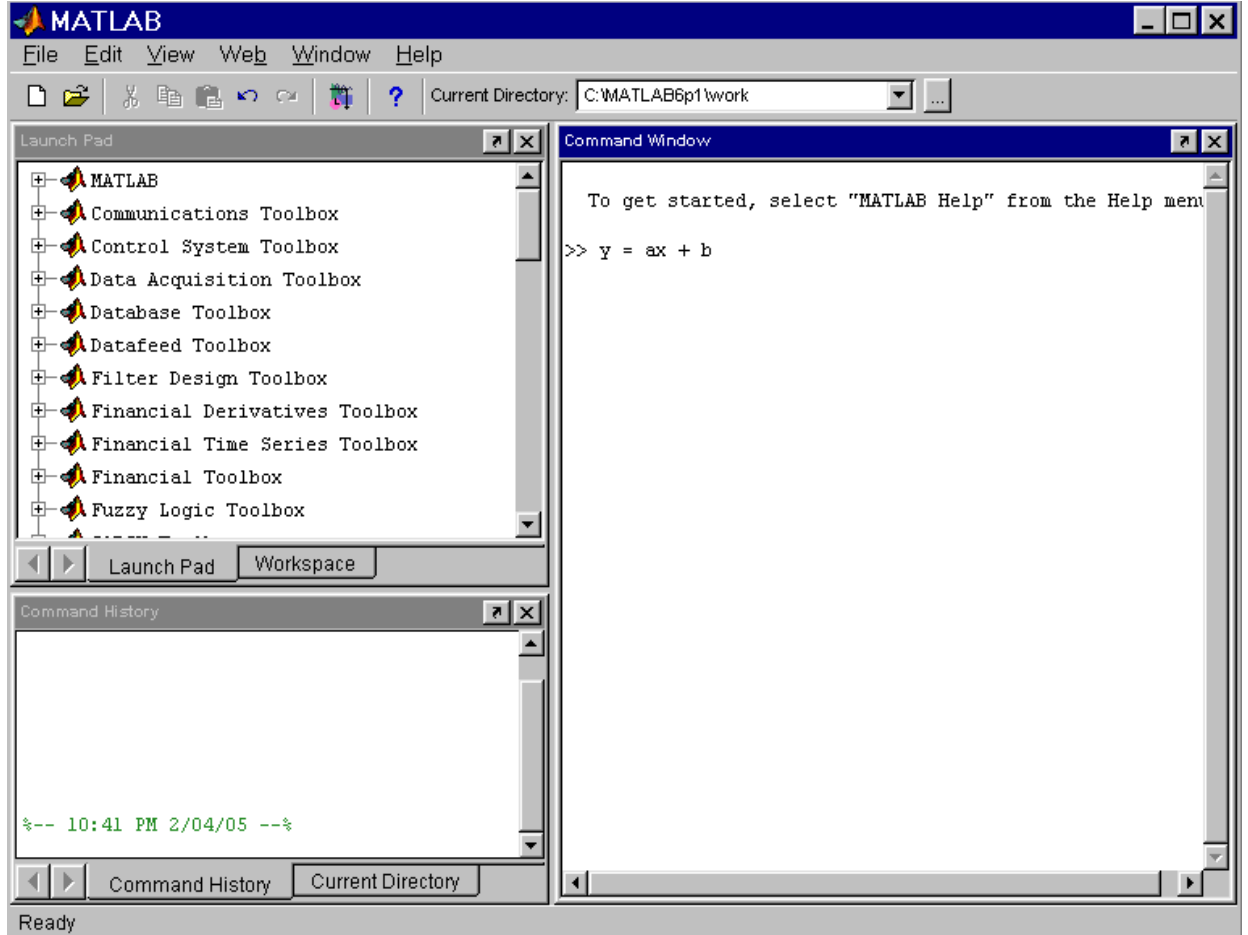
Şekil 113 : MATLAB GUI Hatası-37

DOĞRU 37 : Mesajdaki hatanın giderilmesi için, “OK” düğmesi yerine, “Cancel” düğmesi kullanılmalıdır.



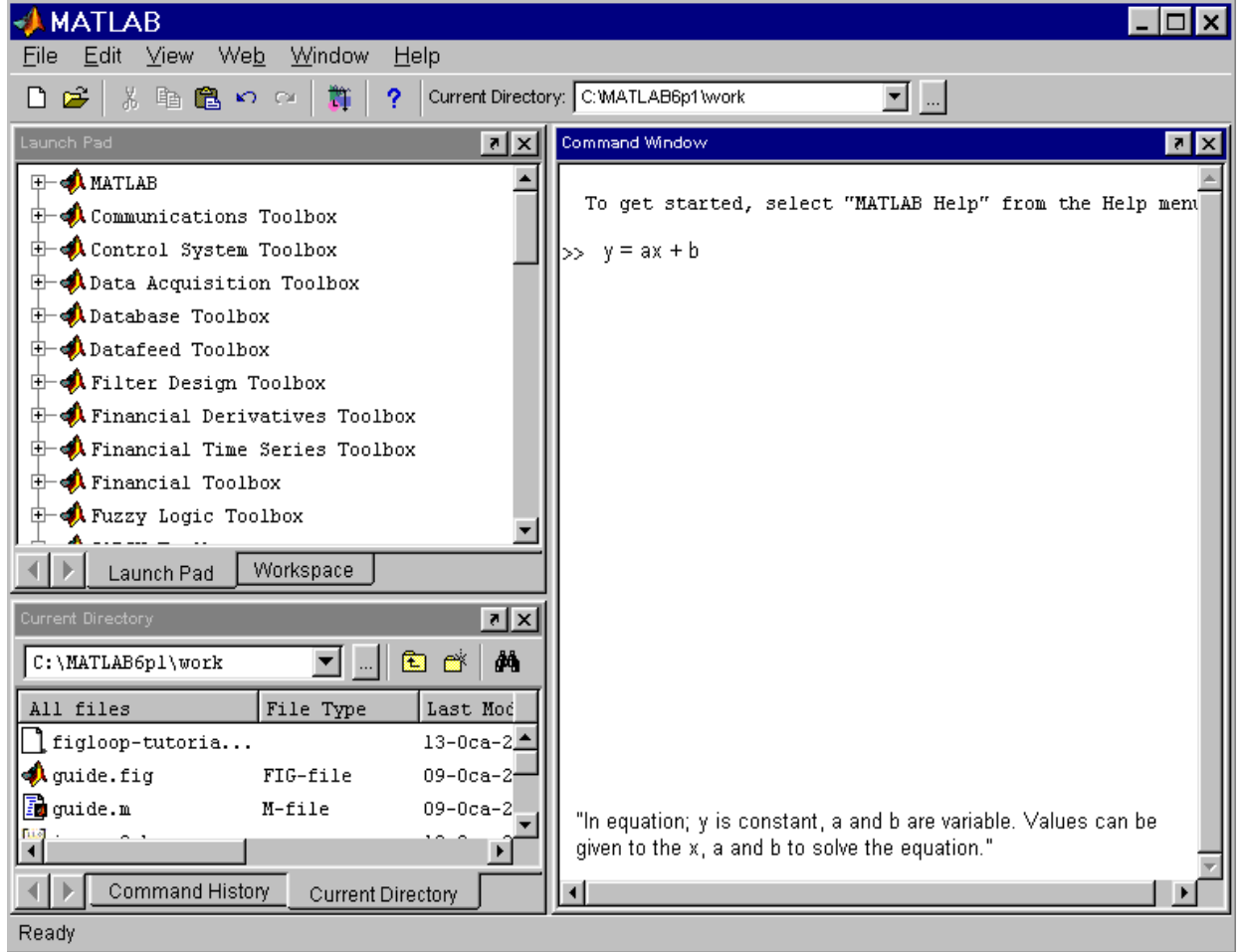
Şekil 114 : MATLAB GUI Hatası-37'nin Doğrusu

HATA 38 : “ $y = ax + b$ ” denkleminin ne anlama geldiğinin ve amacının ne olduğunun açıklanmaması, denklemin kullanıcı tarafından anlaşılmasına neden olacaktır. Kullanıcıya denklem hakkında açıklayıcı bilgi verilmemesi, sorun yaratmaktadır.



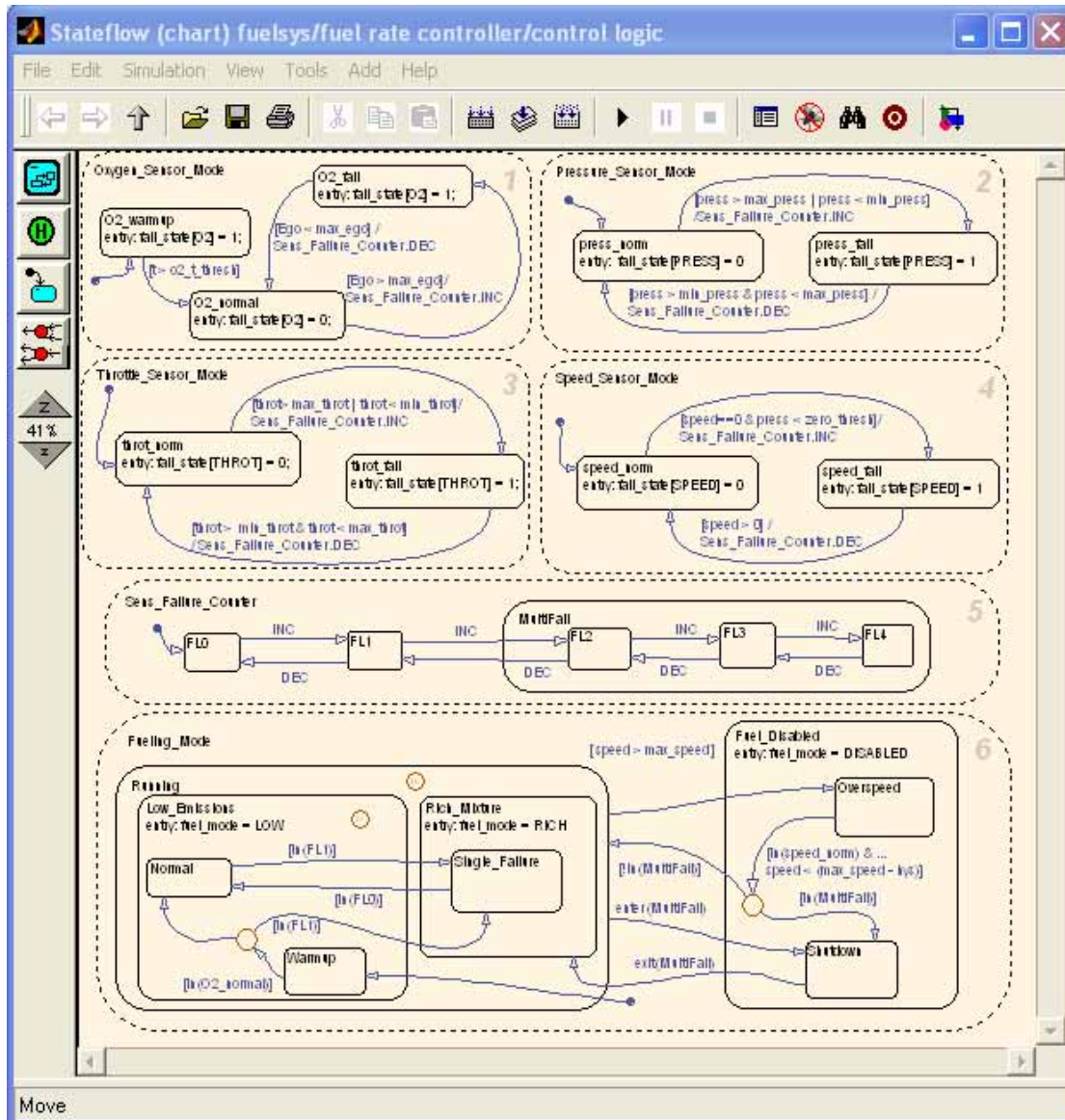
Şekil 115 : MATLAB GUI Hatası-38

DOĞRU 38 : “ $y = ax + b$ ” denkleminin kullanıcı tarafından anlaşılması ve amacının ne olduğunun bilinmesi için, pencerenin altında kullanıcı için denklem ile ilgili uyarıcı bir bilgi verilmelidir. Kullanıcı ile arayüz arasındaki iletişimin sağlanması için denklemi ya da herhangi bir işlemi açıklayan bir mesaj yazılmalıdır.



Şekil 116 : MATLAB GUI Hatası-38'in Doğrusu

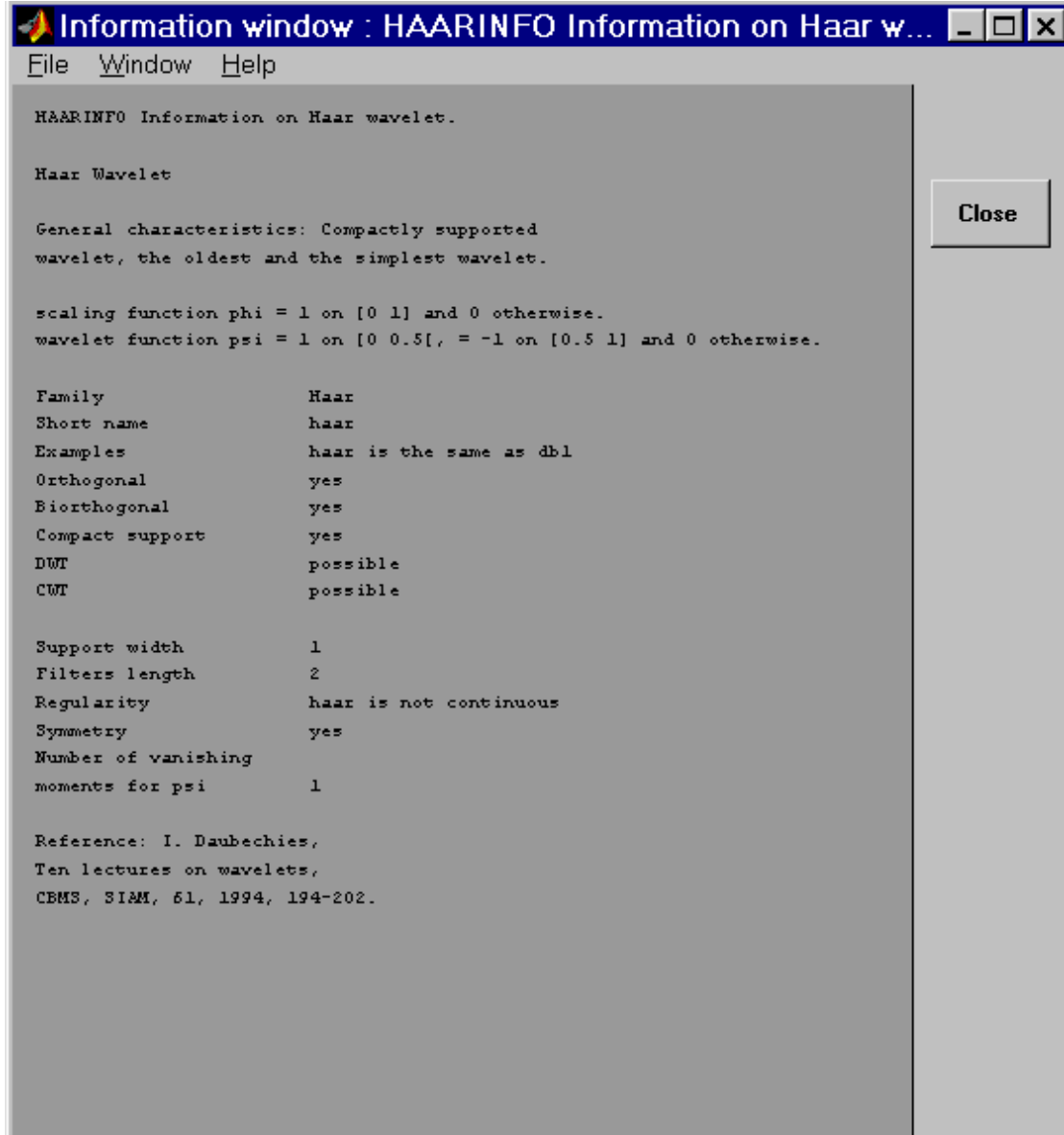
HATA 39 : Kullanıcının rahatlığı düşünülmeden, arayüzün kullanıcı açısından uygun ve kolay olarak dizayn edilmemesi hataya sebep olmuştur. Arayüzü dizayn eden kişinin kendi keyfine göre arayüzü oluşturması, kullanıcı açısından belirsizliğe ve anlam karışıklığına neden olmuştur. Arayüzün karmaşık bir yapı içerisinde olması, arayüzün kullanıcı için oluşturulması düşüncesine ters düşmektedir. Ayrıca, metin yazı tipi boyutlarının küçük ve anlaşılmasız olması okumayı zorlaştırmaktadır.



Şekil 117 : MATLAB GUI Hatası-39

DOĞRU 39 : Arayüz kullanıcı, bir programcı gibi düşünülmemelidir. Arayüz dizayn edilirken, kullanıcının rahatlığı ve kullanım kolaylığı ön planda tutulmalıdır. Arayüz, kullanıcı için kolay ve anlaşılır bir yapıda olmalıdır. Arayüzün metin yazı tipi boyutları, okunaklı ve anlaşılır olmalıdır.

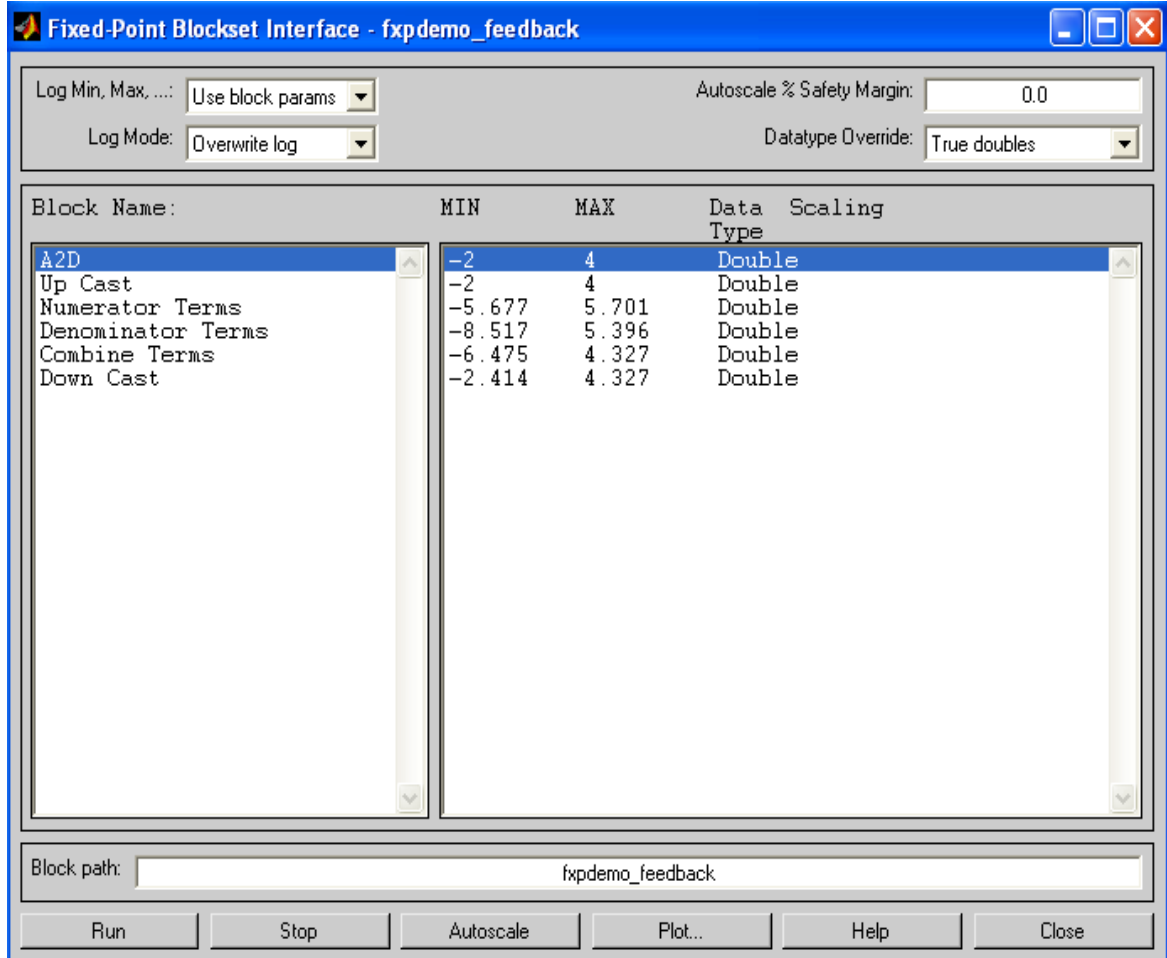
HATA 40 : Açıklayıcı ve bilgi verici bir pencere olmasına rağmen; açıklayıcı metinlerin yazı karakterlerinin küçük olması, kullanıcı ile arayüz arasındaki etkileşimi bozmaktadır. Kullanıcıya aktarılmak istenen bilgiler, iletilenmemiştir. Kullanıcının, arayüzde yazılanları okuması oldukça zordur. Buda, arayüzün kullanıcıyı bilgilendirme ilkesine zarar vermektedir.



Şekil 118 : MATLAB GUI Hatası-40

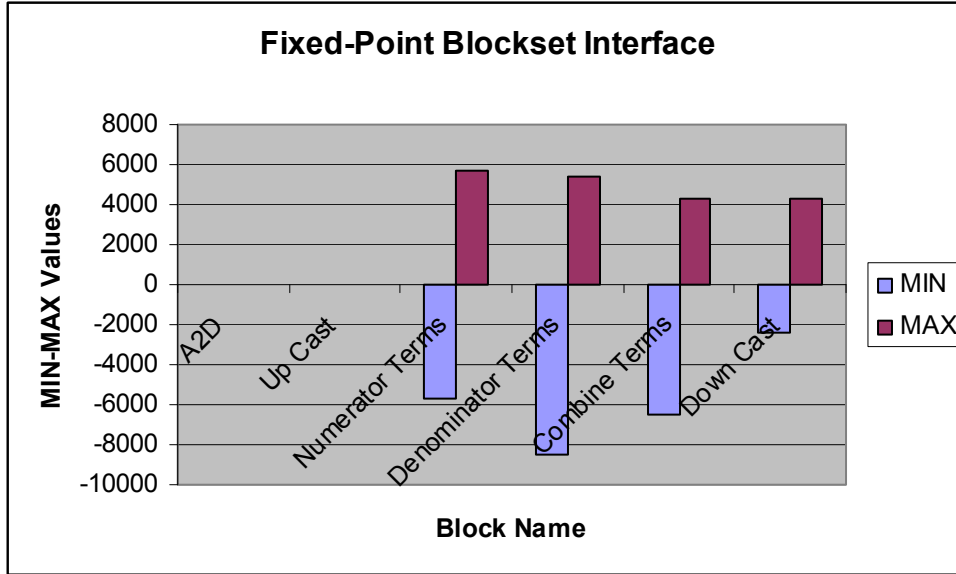
DOĞRU 40 : Arayüzün okunabilirliği ve görünümü açısından, metin yazı karakterlerinin büyük olması gerekmektedir. Yazılar, kullanıcının görüş alanı içerisinde ya da merkezinde olmalıdır. Metinlerin yazı karakterlerinin türleri veya genişlikleri görünüşü ve okumayı zorlaştırmamalıdır. Arayüz, kullanıcının rahatlığı düşünülerek oluşturulmalıdır.

HATA 41 : “*Fixed-Point Blockset Interface*” bölümünde veri sonuçlarının liste şeklinde açıklanması, arayüzün anlaşılmasını olumsuz yönde etkilemiş, belirsizliğe sebep olmuştur. Bu durum, kullanıcı ile arayüz arasındaki etkileşimi olumsuz yönde etkilemektedir.



Şekil 119 : MATLAB GUI Hatası-41

DOĞRU 41 : Arayüzdeki veri sonuçlarını liste biçiminde açıklamak yerine, grafiklerle göstermek; kullanıcıların arayüzü daha rahat anlamaları açısından gereklidir. Veri sonuçları, grafiklerle gösterilirse kullanıcının anlaması daha kolay olacaktır.



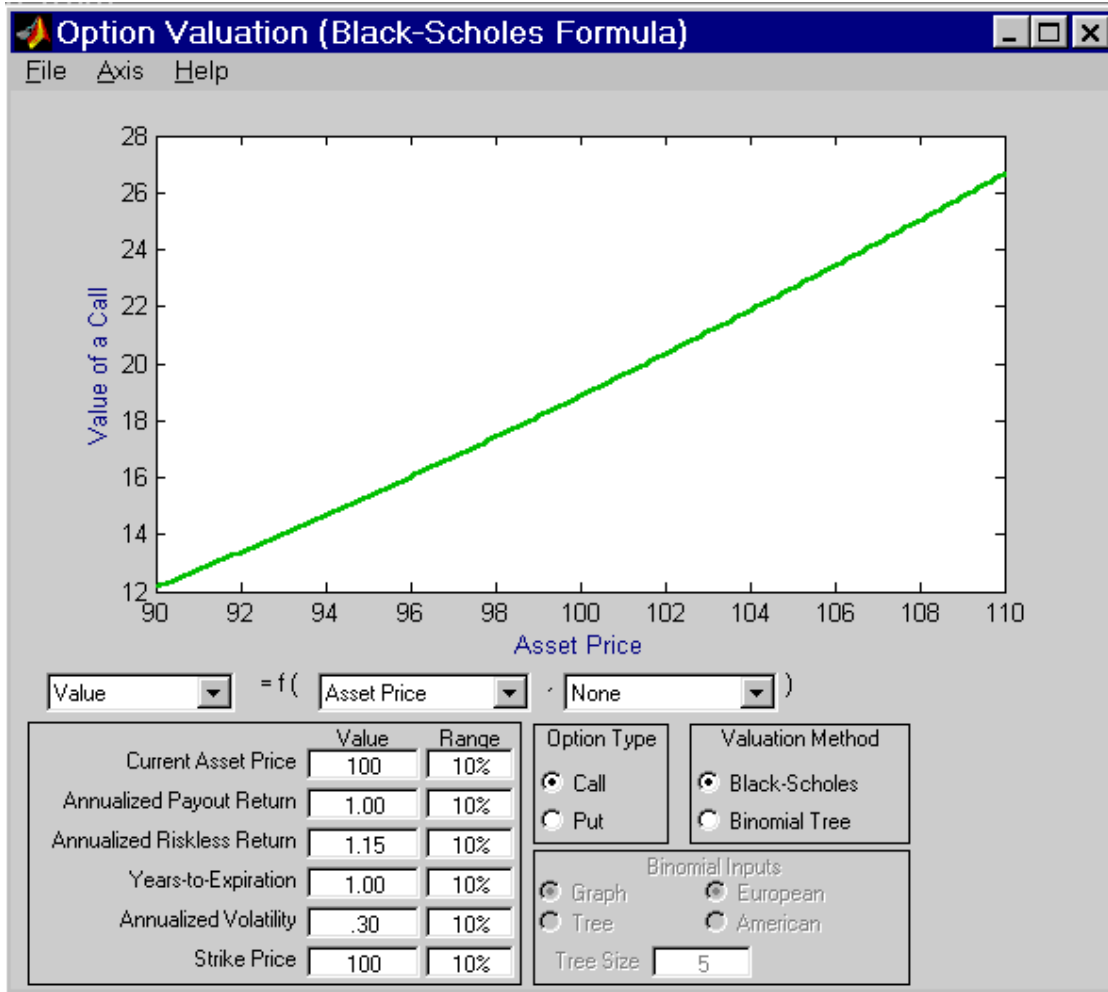
Şekil 120 : MATLAB GUI Hatası-41'in Doğrusu

HATA 42 :

$$\text{Value} = f(\text{Asset Price}, \text{None})$$

Şekil 121 : MATLAB GUI Hatası-42

Aşağıdaki arayüzden alınan bu örnekte, arayüzde bir formül kullanılmıştır. Bilginin çok kötü sunulması sonucu ortaya çıkan bir hata vardır. Çünkü formül açık ve anlaşılır değildir. Aynı zamanda, kullanıcı için de bu durum net ve bilgilendirici değildir. Arayüz dizaynı, kullanıcının kolaylığı düşünülerek yapılmamıştır.



Şekil 122 : MATLAB GUI Hatası-42'nin Arayüzü

DOĞRU 42 : Value = f (x, y) şeklindeki bir formül arayüzde daha belirgin bir şekilde gösterilmesi gerekirdi.

Örneğin ; formülün gösterimi

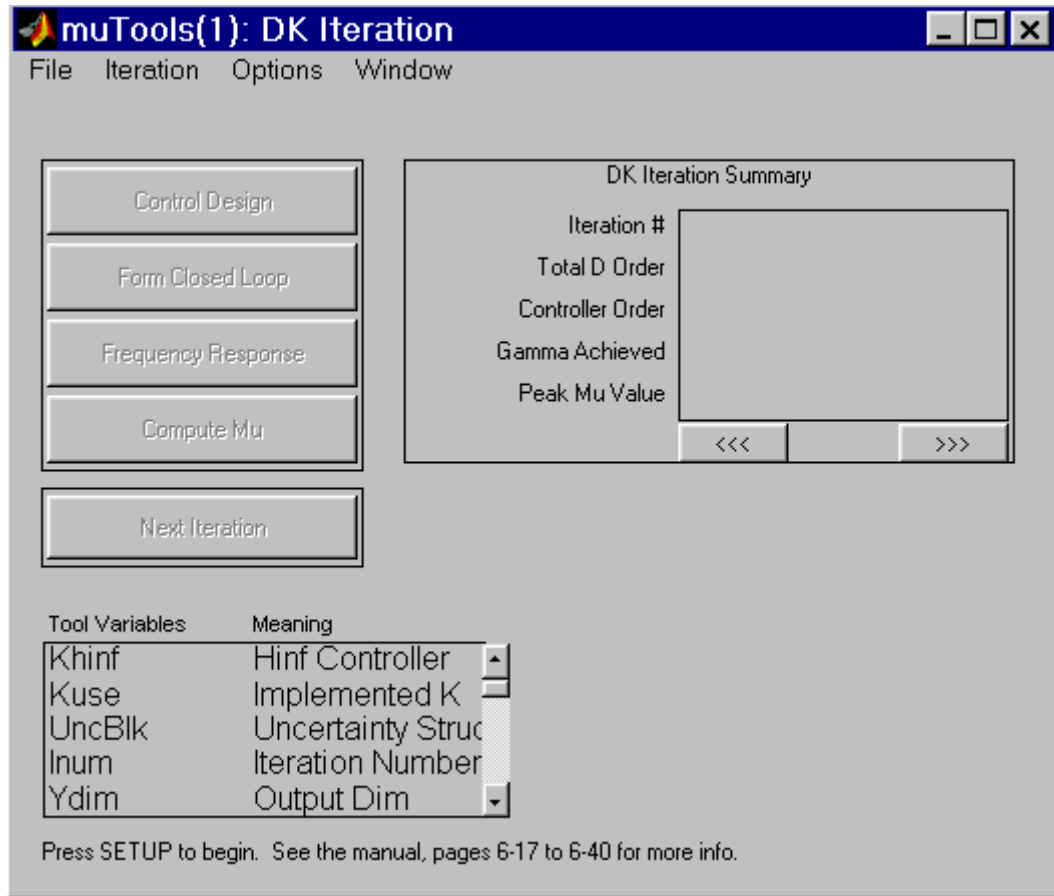
x :

y :

Şekil 123 : MATLAB GUI Hatası-42'nin Doğrusu

şeklinde olabilirdi. Bu şekilde bir kullanımın, kullanıcı açısından anlaşılması ve kullanılması daha kolay olacaktır.

HATA 43 : Hatanın nedeni : Arayüzde, kullanıcıya yardım edecek yol ve yöntemlerin bulunmamasıdır. “DK Iteration Summary” çerçevesinin (frame) olduğu yerde, <<< ve >>> düğmelerinin (button) ne iş yaptığının kullanıcılar tarafından anlaşılabilmesi ve işlevlerinin ne olduğu konusunda kullanıcılara bilginin verilmeyişi, karışıklığa neden olmaktadır.



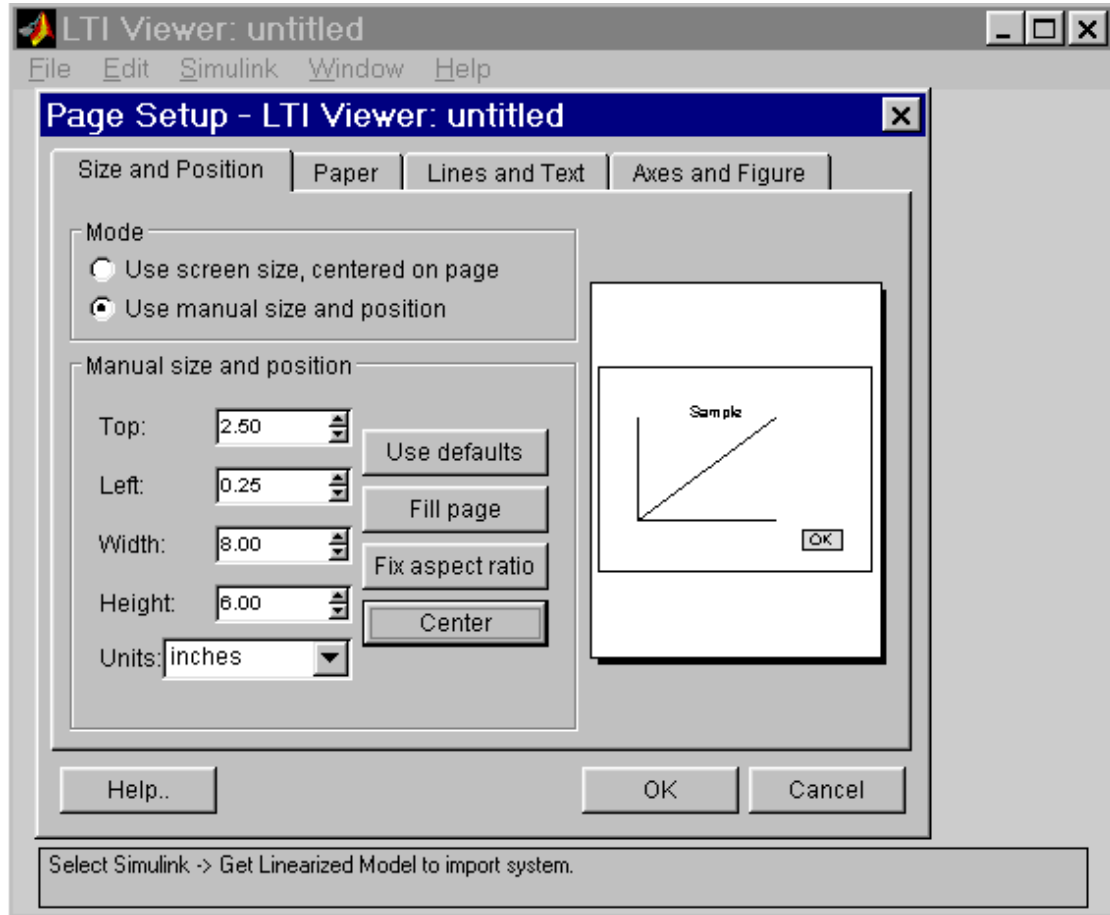
Şekil 124 : MATLAB GUI Hatası-43

DOĞRU 43 : Düğmelerin (button) üstüne ya da yanına, kullanıcılara yardımcı olmak amacıyla ile açıklayıcı bilgi yazılması gerekmektedir. Örneğin; düğmelerden, soldakine geri anlamına gelen “Back”, sağdakine ise ileri anlamına gelen “Next” yazılması, kullanıcılar için bilgilendirici olacaktır. Böylece, kullanıcılar hangi düğmenin ne işe yaradığını anlamış olacaklardır.



Şekil 125 : MATLAB GUI Hatası-43’ün Doğrusu

HATA 44 : Arayüzde, düğmelerin (button) ve denetim kutularının (checkbox) aynı işi yapıyor olması, kullanıcılar açısından anlam karışıklığına yol açmaktadır. Solda denetim kutuları olmasına rağmen, sayfanın genişlik (size) ve yeri (position) düğmeleri ile sağlanmaktadır. Bu durum, arayüzde belirsizliğe yol açmaktadır.



Şekil 126 : MATLAB GUI Hatası-44

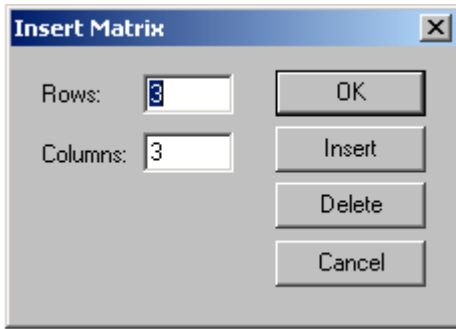
DOĞRU 44 : Arayüzde, düğmeler (button) ile her türlü iş kolaylıkla yapılabildiği için denetim kutularına (checkbox) gerek yoktur. Yani, arayüzde denetim kutuları olmasada olur. Bu yüzden, arayüzde kullanıcılar açısından belirsizliğe yol açacak her türlü durumdan kaçınmak gerekir.

4.5. MathCAD'teki GUI Hataları & Hataların Doğruları

MathCAD programındaki GUI hataları, aynen MATLAB programında olduğu gibi; Jeff Johnson tarafından 2000 yılında yazılmış olan ve Morgan Kaufmann Yayınevinden çıkan “*Grafiksel Kullanıcı Arayüz Hataları (GUI Bloopers)*” adlı kitap örnek alınarak incelenmiştir. Bu kitaptan elde edilen grafiksel kullanıcı arayüz (GUI) hataları hakkındaki bilgiler doğrultusunda, MathCAD programında ortaya çıkan GUI hataları ve bu hataların doğruları şu şekildedir [60]:

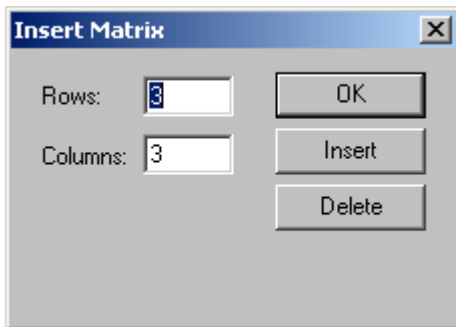
4.5.1 MathCAD'te GUI Elemanlarında Yapılan Hatalar (GUI Component Bloopers)

HATA 1 : Arayüzde fazladan düğme (button) kullanılması, hataya sebep olmuştur. “Cancel” düğmesinin kullanılması gereksizdir. Pencerenin “Close” düğmesi ile “Cancel” düğmesi aynı işi yapmaktadır. Bu düğmelerden her ikisinde tıklayınca pencere kapanmaktadır.



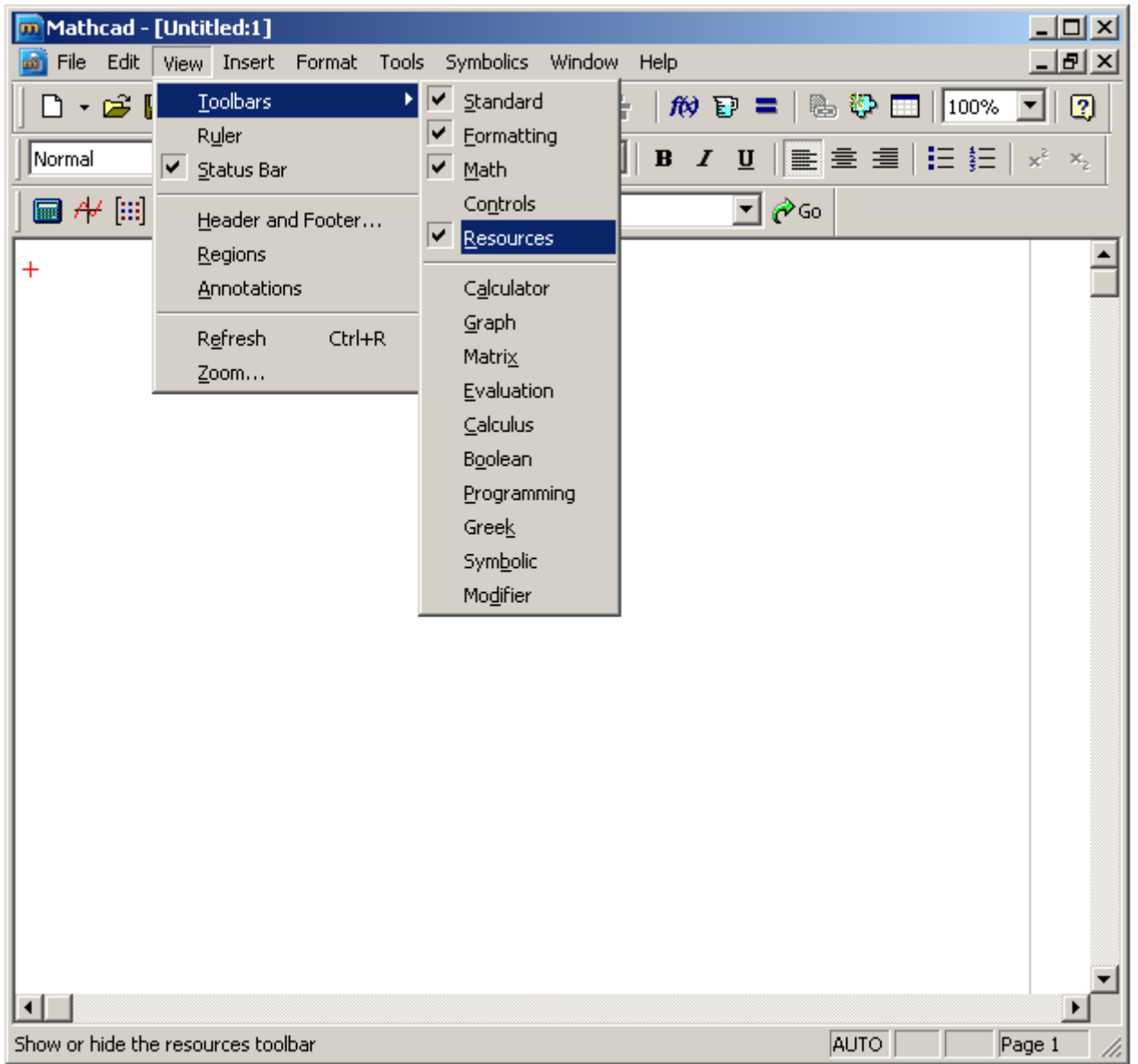
Şekil 127 : MathCAD GUI Hatası-1

DOĞRU 1 : Arayüzde gereksiz yere düğme (button) kullanılmamalıdır. Bu yüzden, “Cancel” düğmesinin arayüzden kaldırılması gerekmektedir.



Şekil 128 : MathCAD GUI Hatası-1'in Doğrusu

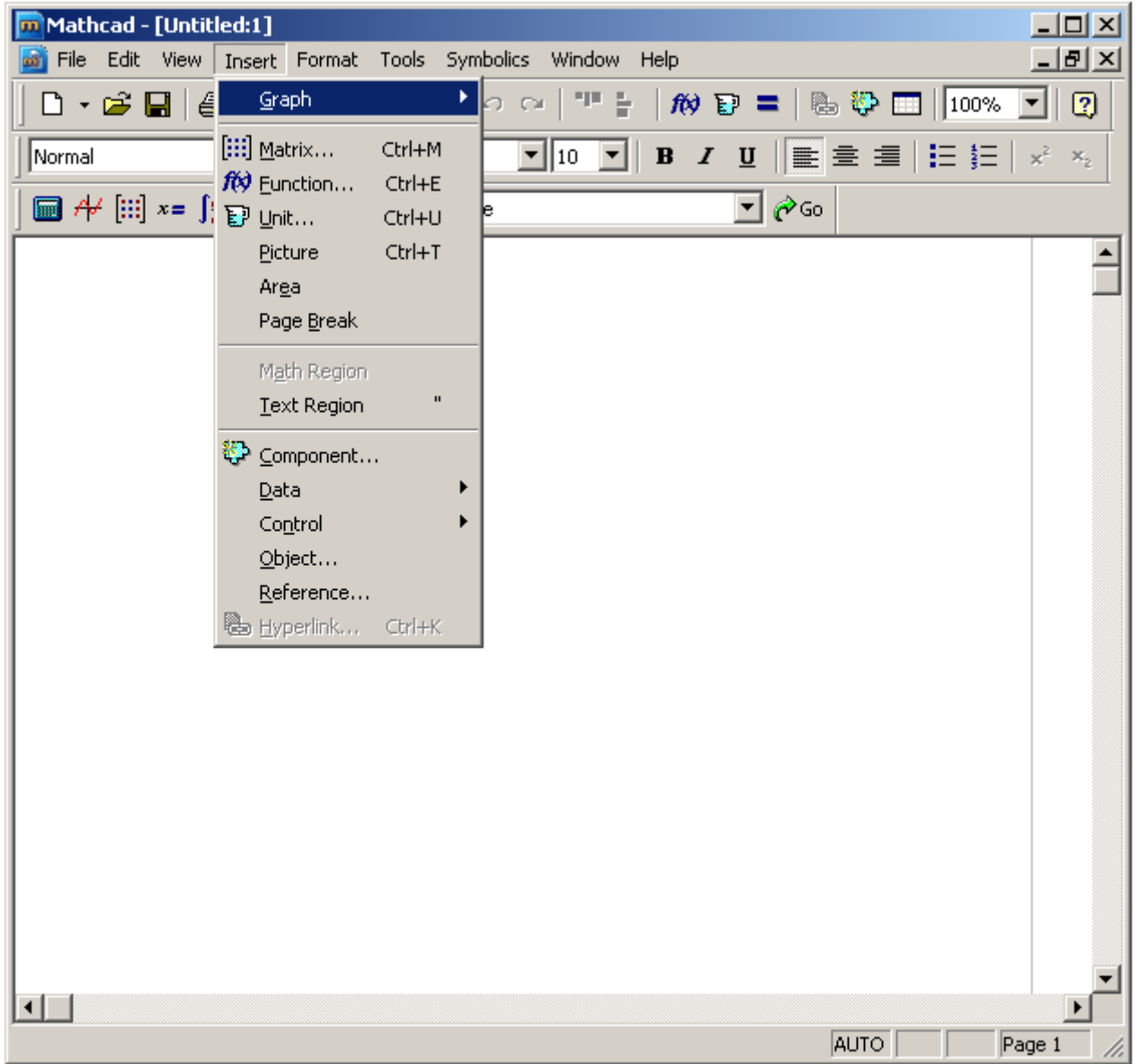
HATA 2 : Bir menünün açılan alt menüleri ile birlikte diğer menülerin üzerine taşması, hataya sebep olmuştur.



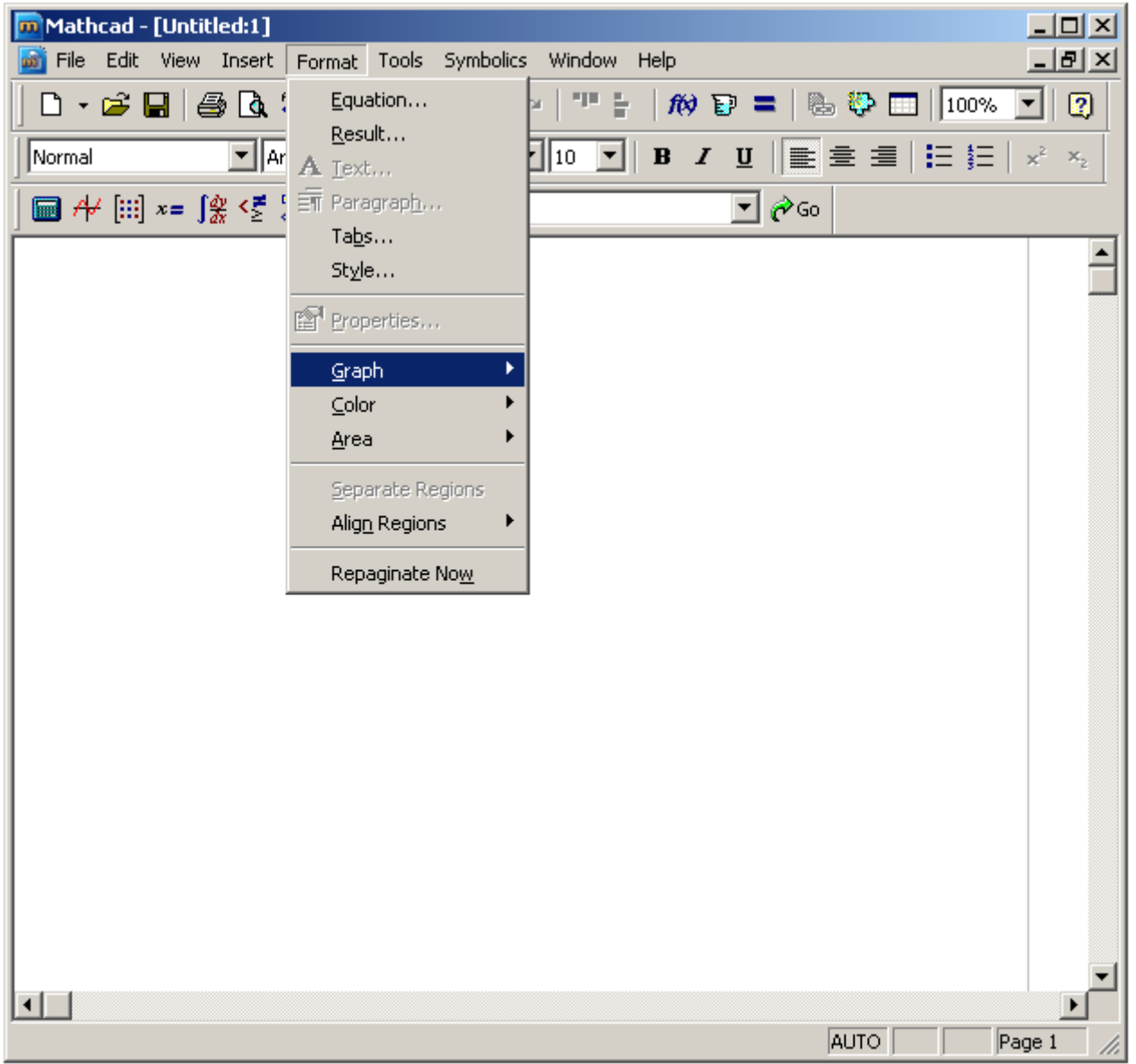
Şekil 129 : MathCAD GUI Hatası-2

DOĞRU 2 : Bir ana menü diğer alt menüleri ile birlikte, diğer menülerin üstüne taşmayacak şekilde dengeli ve tutarlı olmalıdır.

HATA 3 : İki farklı menüde aynı isimli iki alt menünün bulunması, hataya sebep olmuştur. Hem “*Insert*” menüsünde “*Graph*” isimli alt menünün olması, hem de “*Format*” menüsünde “*Graph*” isimli bir alt menünün olması, karışıklığa yol açmaktadır.



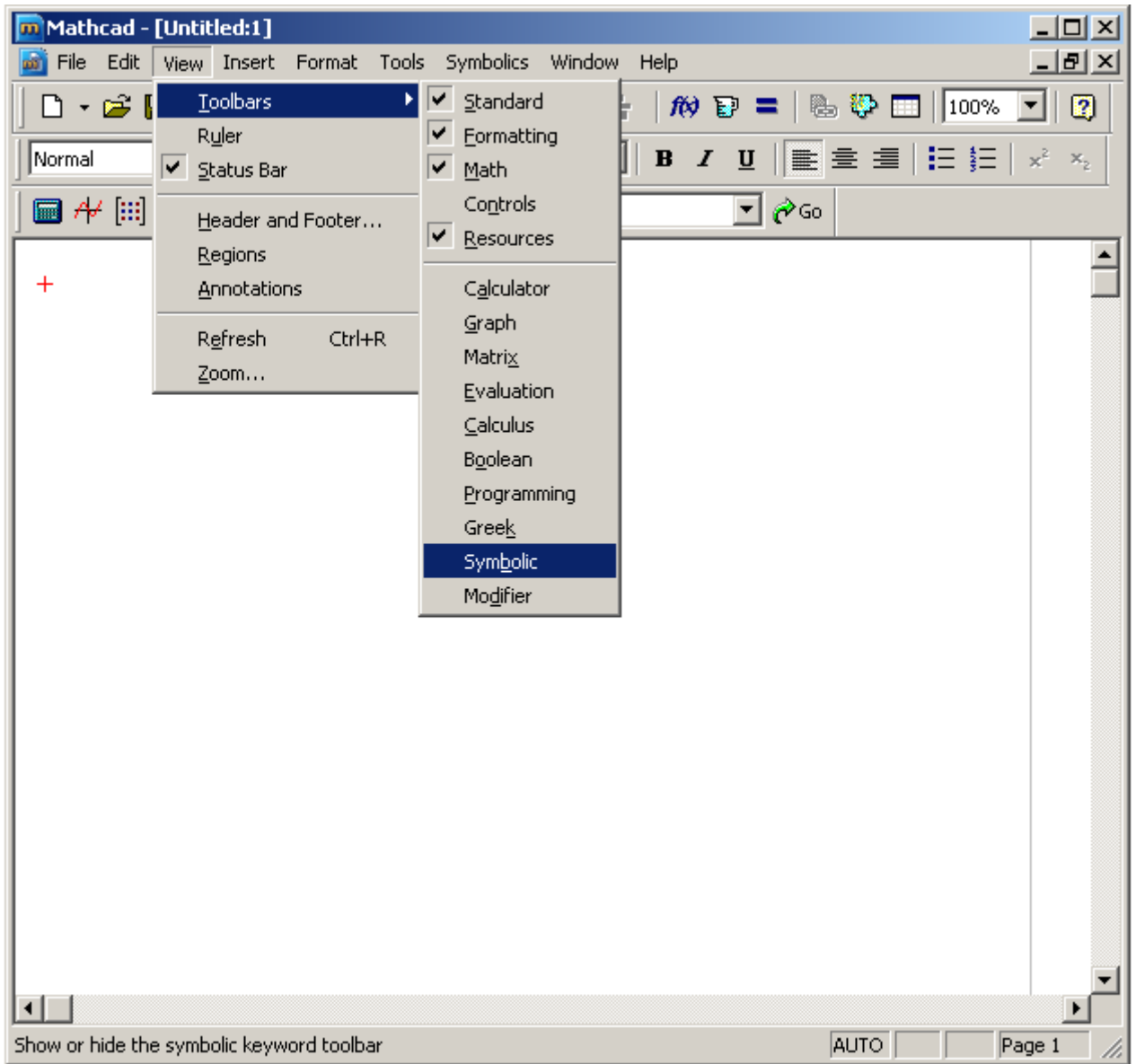
Şekil 130 : MathCAD GUI Hatası-3’ün Birinci Arayüzü



Şekil 131 : MathCAD GUI Hatası-3'ün İkinci Arayüzü

DOĞRU 3 : İki farklı menüde aynı isimli iki alt menü bulunmamalıdır. Arayüzde menüler karışıklığa yol açmamak için, farklı isimlerde olmalıdırlar.

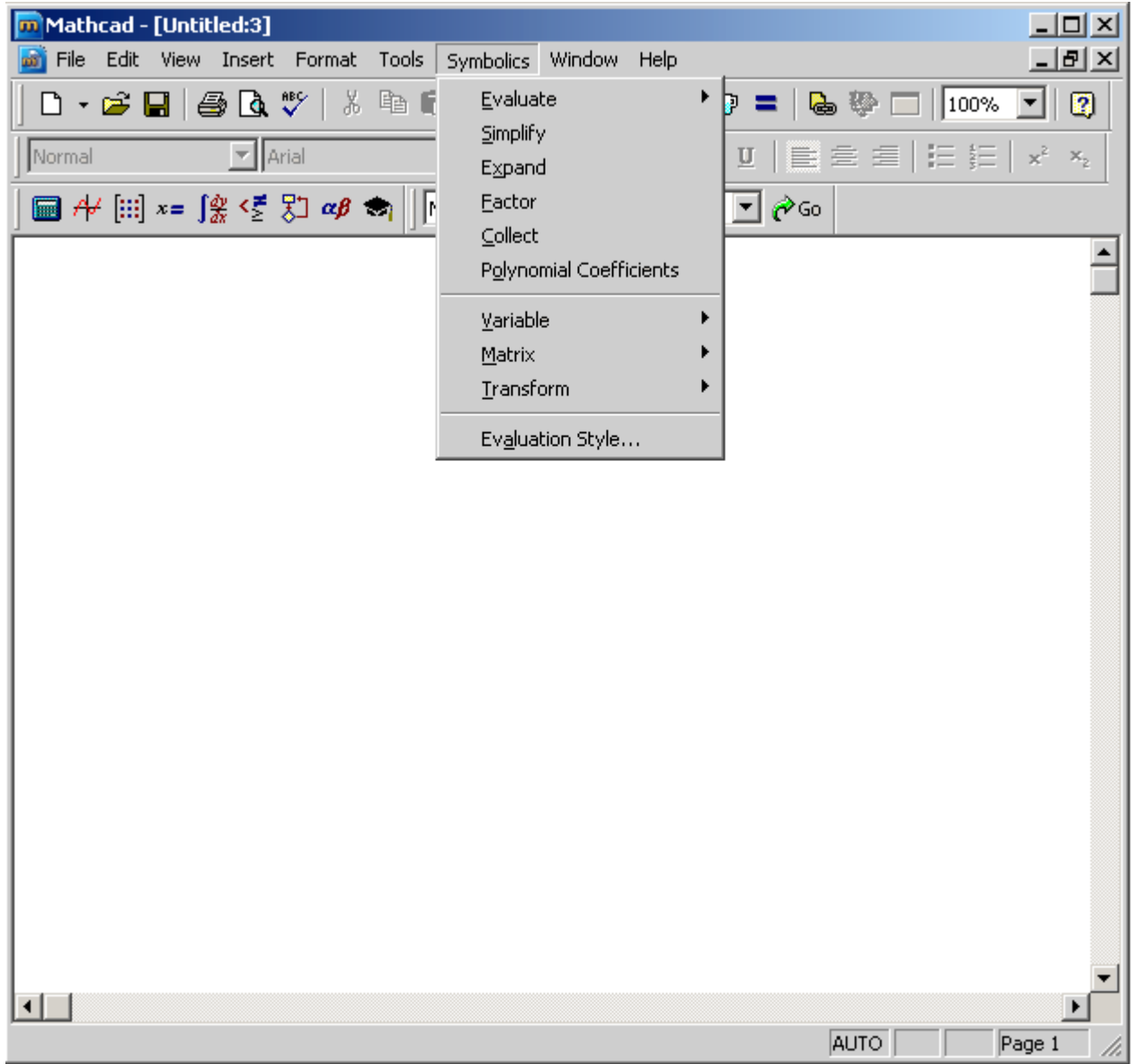
HATA 4 : Alt menü ismi ile ana menü isminin (*Symbolics*) aynı olması, hataya sebep olmuştur. Bu durum, kullanıcılar açısından karışıklığa yol açmaktadır.



Şekil 132 : MathCAD GUI Hatası-4

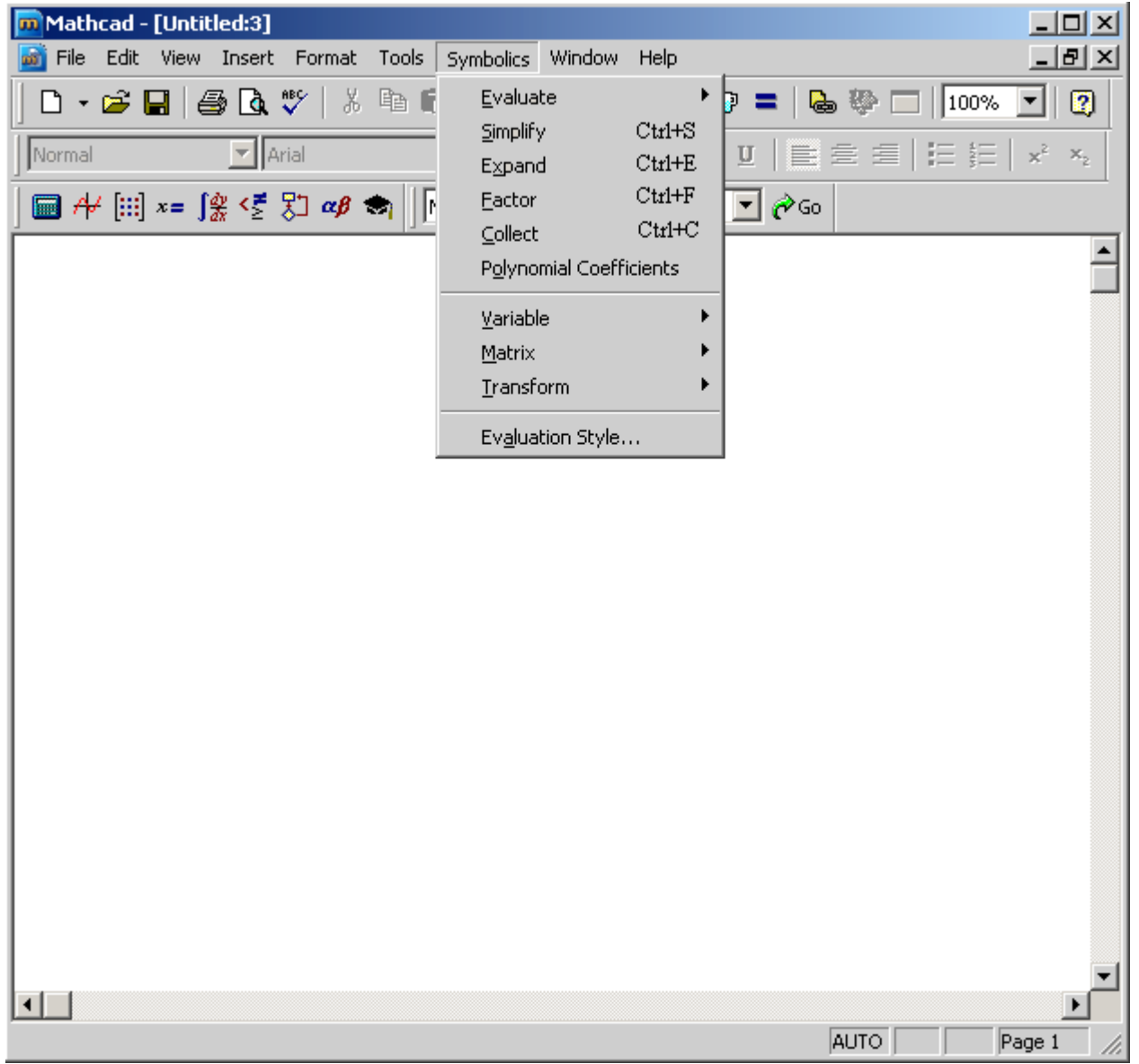
DOĞRU 4 : Arayüzde menüler arasındaki karışıklığı önlemek için, alt menü ismi ile ana menü ismi aynı olmamalıdır. Her bir menüye farklı isimler verilmelidir.

HATA 5 : Arayüzdeki menülerde klavye kısayol tuşlarının (shortcuts) kullanılmaması, hataya sebep olmuştur.



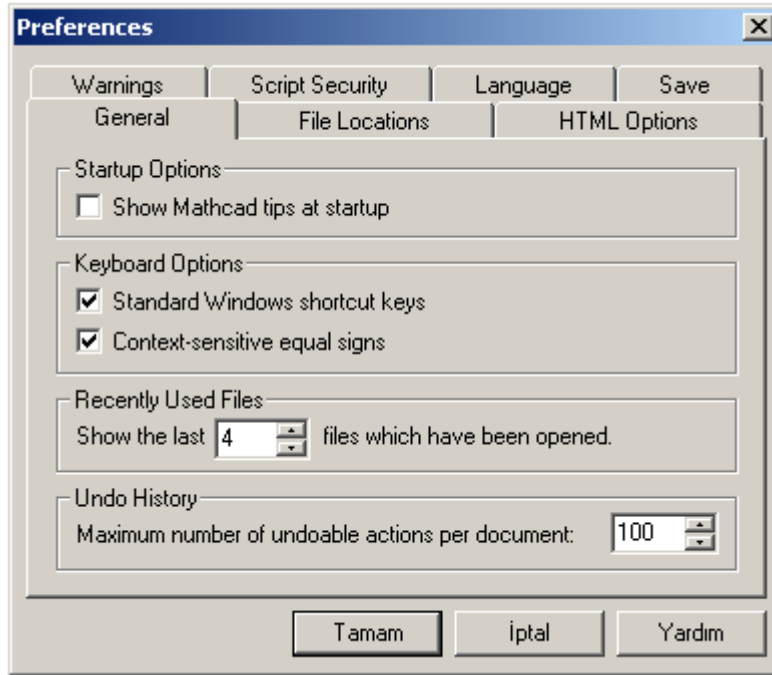
Şekil 133 : MathCAD GUI Hatası-5

DOĞRU 5 : Arayüzün kullanıcılar tarafından daha rahat kullanılması için, menülerde klavye kısayol tuşlarının (shortcuts) kullanılması gerekmektedir.

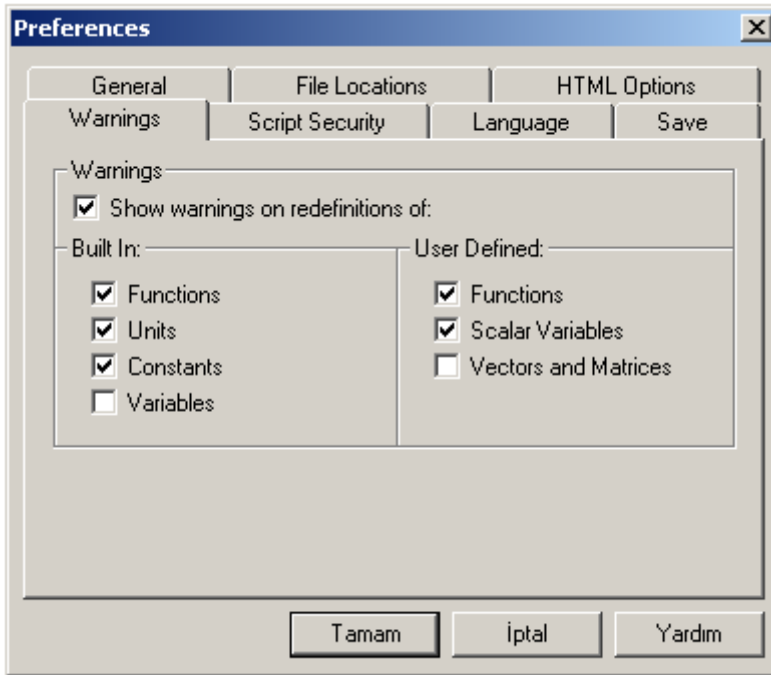


Şekil 134 : MathCAD GUI Hatası-5'in Doğrusu

HATA 6 : Arayüzde, iki satırda sekme (tab) vardır. Birinci satırdaki sekme'lerden herhangi birine bastığımızda, sekme ikinci satırda gözükmemektedir. Buda, karışıklığa neden olmaktadır. Sekme'ye basıldığı zaman, sekme birden gözden kaybolmaktadır.



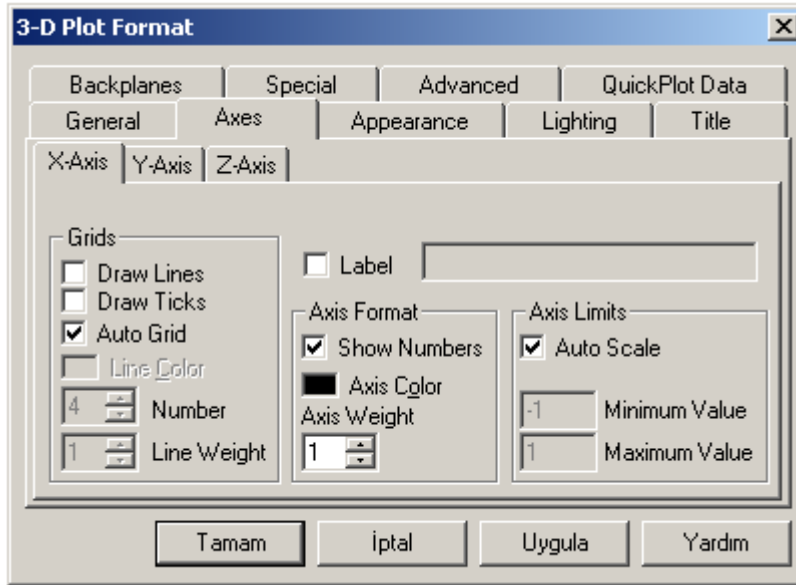
Şekil 135 :MathCAD GUI Hatası-6'nın Birinci Arayüzü



Şekil 136 : MathCAD GUI Hatası-6'nın İkinci Arayüzü

DOĞRU 6 : Sekme'ye (tab) bastığımızda; basılan sekme, ilgili satırda gözükmemelidir. Örneğin; birinci satırdaki sekme'ye bastığımızda, basılı olan sekme birinci satırda gözükmemelidir, ikinci satırda gözükmemelidir.

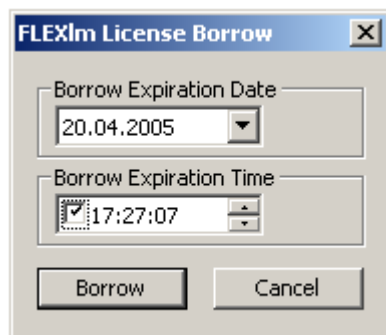
HATA 7 : Arayüzde çok fazla sayıda sekme'nin (tab) kullanılması, hataya sebep olmuştur. Bu durum, arayüzün kullanıcılar tarafından kullanılmasını zorlaştırmaktadır.



Şekil 137 : MathCAD GUI Hatası-7

DOĞRU 7 : Fazla sayıda sekme (tab) kullanılmamalıdır. Arayüzde kullanılan sekme (tab) sayısı, kullanıcıların kafasının karışmaması ve kullanıcıların arayüzü kullanırken zorlanmamaları için, az sayıda olmalıdır.

HATA 8 : Arayüzde gün, ay, yıl ve saat kısımlarının ayrı ayrı bölümlere ayrılmamış olması, hataya sebep olmuştur. Ayrıca, etiket (label) yerine çerçeve (frame) kullanılması, arayüzdeki bir başka hata sebebidir.



Şekil 138 : MathCAD GUI Hatası-8

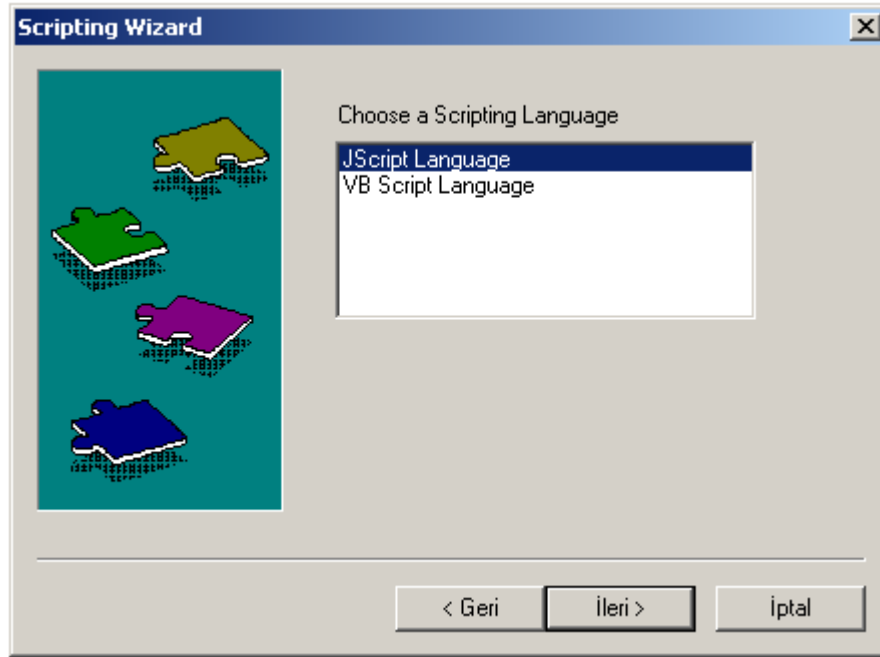
DOĞRU 8 : Gün, ay, yıl ve saat kısımları, arayüzde bölümlere ve parçalara ayrılmalıdır. Böylece, kullanıcılar açısından arayüzün kullanımı kolaylaştırılmış olacaktır. Ayrıca, arayüzde çerçeve yerine etiket kullanılmalıdır. Arayüzde gün, ay, yıl ve saat kısımlarının nasıl olması gerektiği aşağıda gösterilmiştir :

Borrow Expiration Date : . .

Borrow Expiration Time : : :

Şekil 139 : MathCAD GUI Hatası-8'in Doğrusu

HATA 9 : Arayüzde seçenek düğmesi (option button) kullanılması gerekirken, liste kutusu (list box) kullanılması hataya sebep olmuştur.



Şekil 140 : MathCAD GUI Hatası-9

DOĞRU 9 : Arayüzde kullanıcılar açısından kullanılması daha kolay olan arayüz elemanları tercih edilmelidir. Arayüzde liste kutusu (list box) yerine, seçenek düğmesi (option button) kullanılmalıdır.

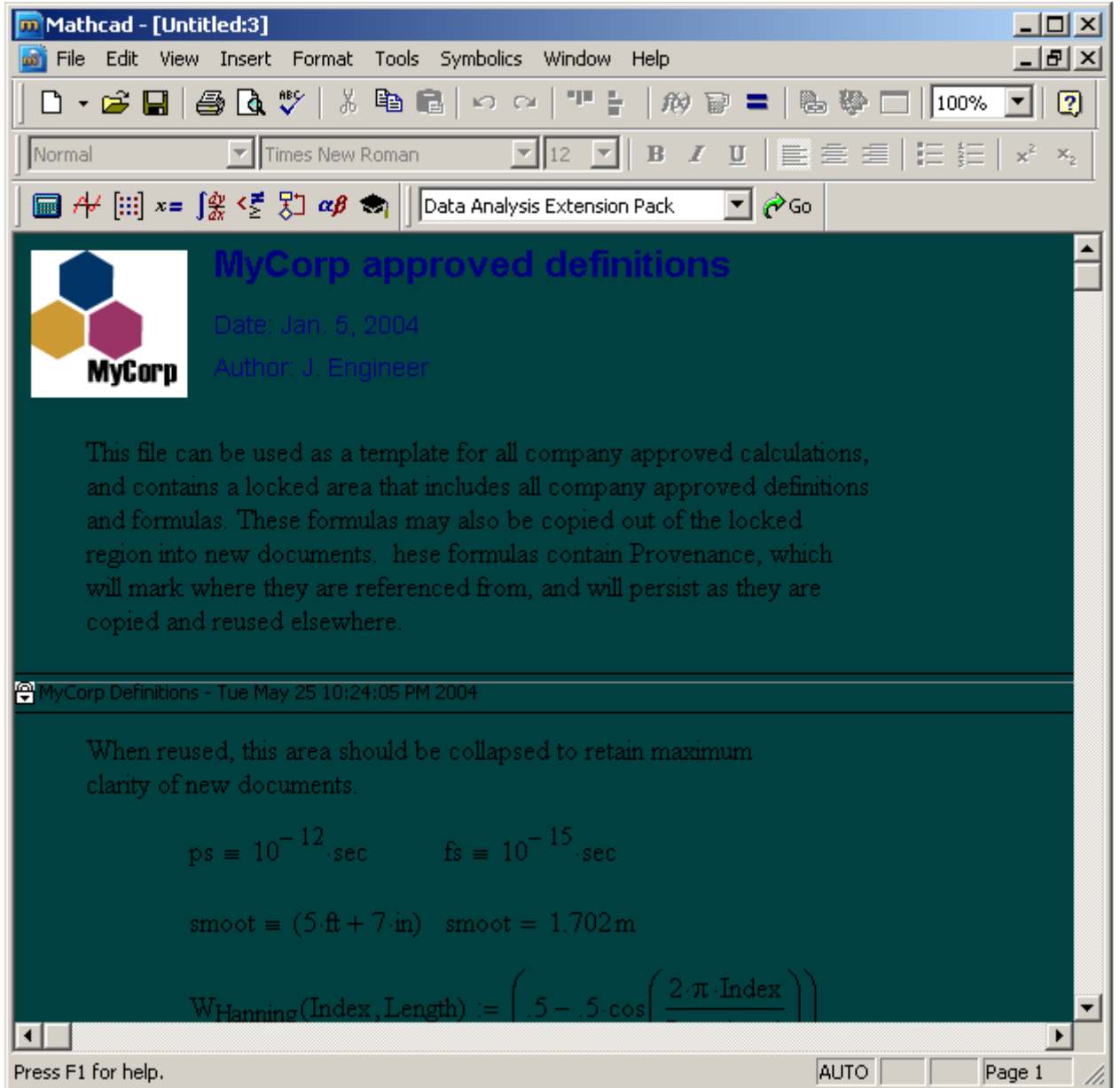
Choose a Scripting Language :

- JScript Language
- VB Script Language

Şekil 141 : MathCAD GUI Hatası-9'un Doğrusu

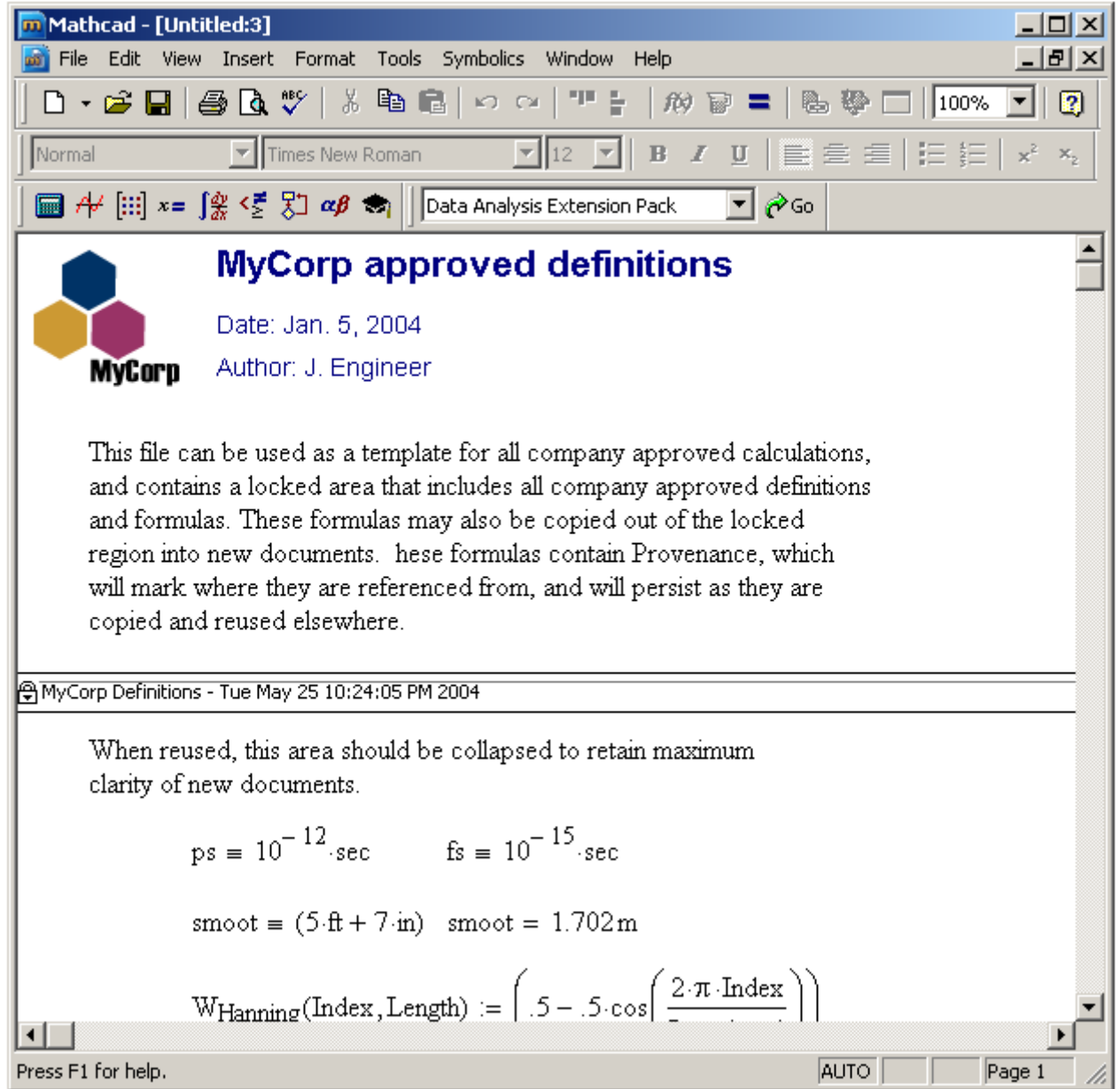
4.5.2 MathCAD'teki Düzen ve Görünüm Hataları (Layout & Appearance Bloopers)

HATA 10 : Arayüzde arka plan (background) renginin, koyu bir renk seçilmesi hataya sebep olmuştur. Bu durum, arayüzde yazılanları görmeyi zorlaştırmaktadır. Ayrıca, arayüz arka planının koyu renk olması, arayüzün görünümünü de bozmaktadır.



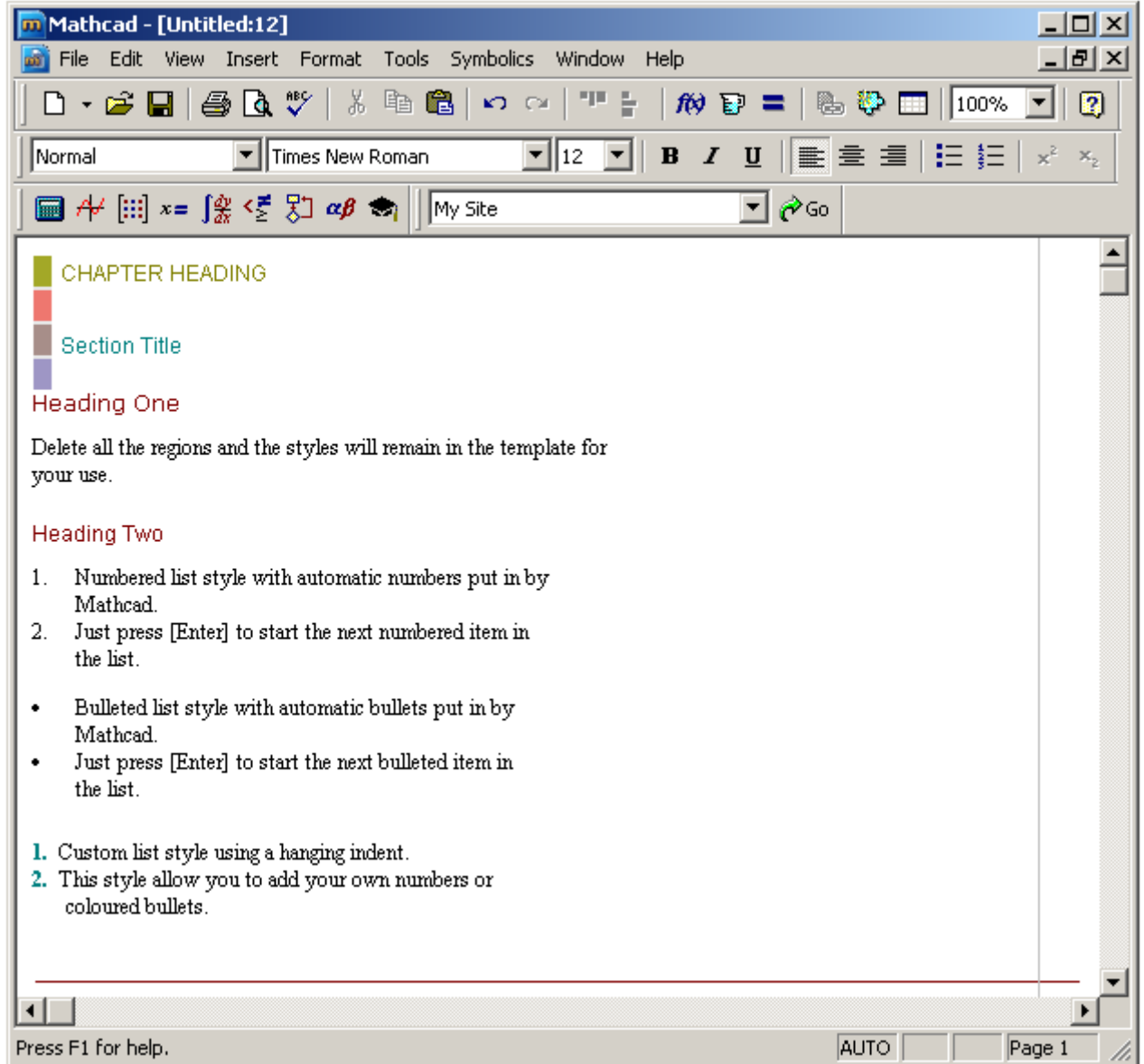
Şekil 142 : MathCAD GUI Hatası-10

DOĞRU 10 : Arayüzde arka plan (background) rengi, okumayı zorlaştırmamalıdır. Görünümü bozan arayüz arka plan renklerinden kaçınmak gerekmektedir. Arayüzde arka plan renginde, koyu renkler değil, beyaz gibi açık ve sade renkler tercih edilmelidir.



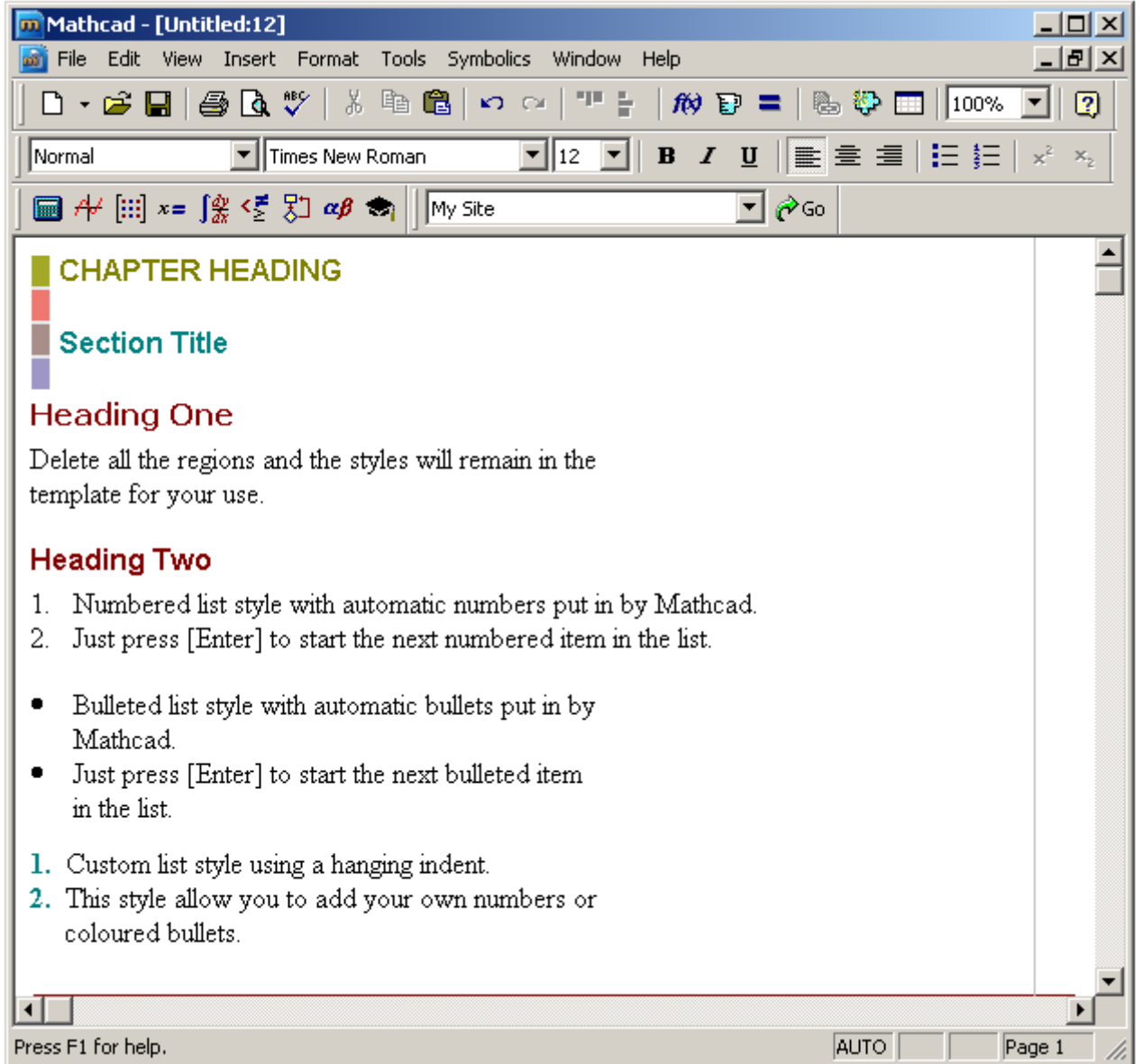
Şekil 143 : MathCAD GUI Hatası-10'un Doğrusu

HATA 11 : Metin yazı tipi boyutunun ufak olması, hataya sebep olmuştur. Oldukça küçük yazı tipi boyutlarının kullanılması, arayüzün görünümünü bozmaktadır. Ayrıca, kullanıcıların metni okumasını zorlaştırmaktadır.



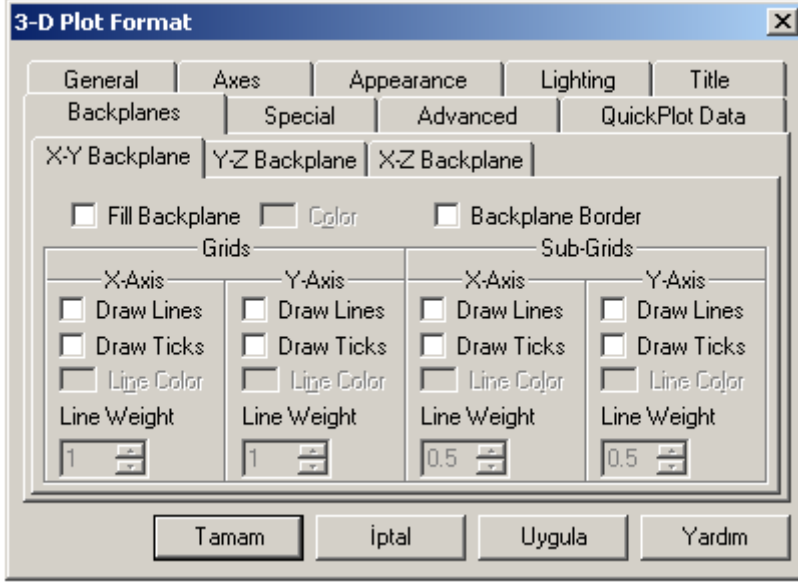
Şekil 144 : MathCAD GUI Hatası-11

DOĞRU 11 : Arayüzde metin yazı tipi boyutu, standart olmalıdır. Yazı tipi boyutu, kullanıcıyı rahatsız etmeyecek, gözü yormayacak ve okumayı zorlaştırmayacak bir şekilde olmalıdır. Yazı tipi boyutu, minimum 10 olmalıdır. Ancak, daha iyi olması için tavsiye edilen yazı tipi boyutu 12'dir.



Şekil 145 : MathCAD GUI Hatası-11'in Doğrusu

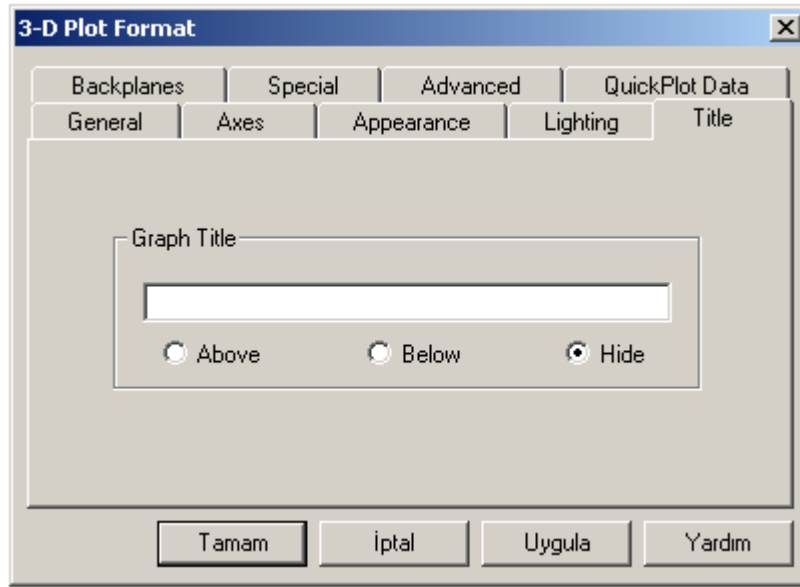
HATA 12 : Arayüzün çok yoğun ve karmaşık bir yapıda olması ile çok fazla sayıda arayüz elemanının iç içe sokularak birarada kullanılması, arayüzün düzeninin bozulmasına sebep olmuştur. Bu durum, arayüzü kullanırken kullanıcı açısından zorluklara yol açacaktır.



Şekil 146 : MathCAD GUI Hatası-12

DOĞRU 12 : Arayüz açık, sade ve anlaşılır bir yapıda olmalıdır. Arayüz içerisinde, çok fazla arayüz elemanı bulundurulmamalıdır. Arayüz elemanları, arayüzün görünümünün bozulmaması için, belirli bir düzen içerisinde olmalıdırlar.

HATA 13 : GUI elemanlarının yanlış kullanılması, hataya sebep olmuştur. Seçenek düğmelerinin (optionbutton), kullanım şekli hatalıdır. Seçenek düğmelerinin, çerçeve (frame) içerisinde kullanılması, bir başka hata sebebidir. Ayrıca, “*Graph Title*” bölümünün çerçeve içerisinde kullanılması gereksizdir.



Şekil 147 : MathCAD GUI Hatası-13

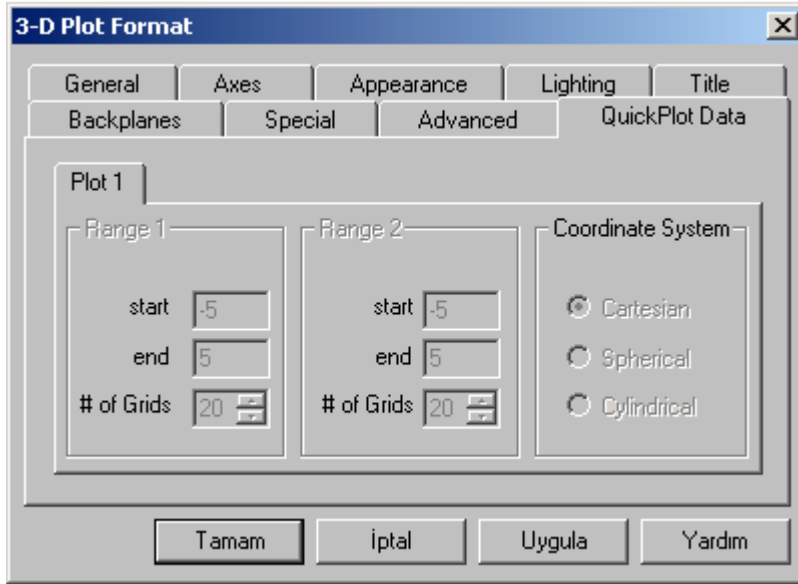
DOĞRU 13 : Arayüzdeki “*Graph Title*” bölümünde; çerçeve (frame) yerine, etiket (label) kullanılması gerekmektedir. Etiket ile metin alanının (textfield) kullanım şekli; etiket solda, metin alanı ise sağda şeklinde olmalıdır. Ayrıca, seçenek düğmeleri (optionbutton) çerçeve içerisinde kullanılamazlar. Bu yüzden, seçenek düğmelerinin “*Graph Title*” etiketi altındaki gösterimleri aşağıdaki gibi olmalıdır :

Style Name :

- Above
- Below
- Hide

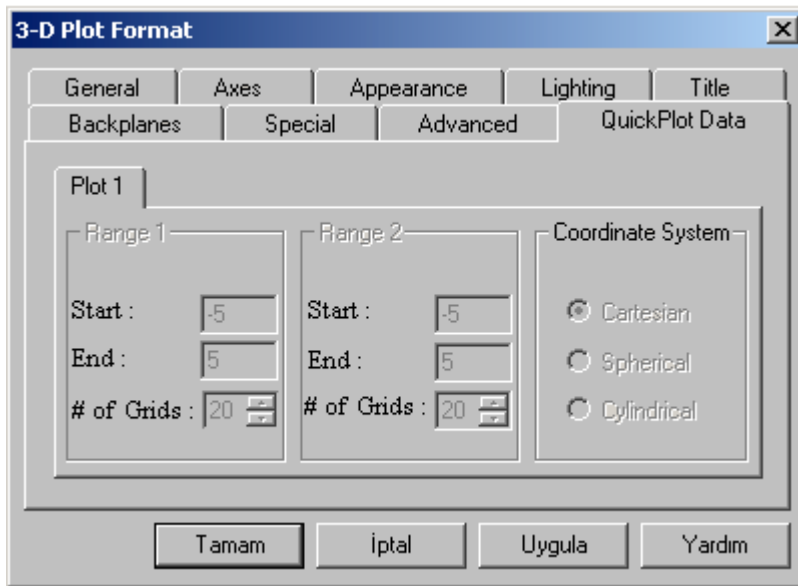
Şekil 148 : MathCAD GUI Hatası-13’ün Doğrusu

HATA 14 : Etiket'lerin (label) hizalarındaki düzensizlik, arayüzün görünümünde bozukluğa ve hataya sebep olmuştur. Ayrıca etiketlerden sonra (:) işaretinin kullanılmaması, bir başka hata sebebidir.



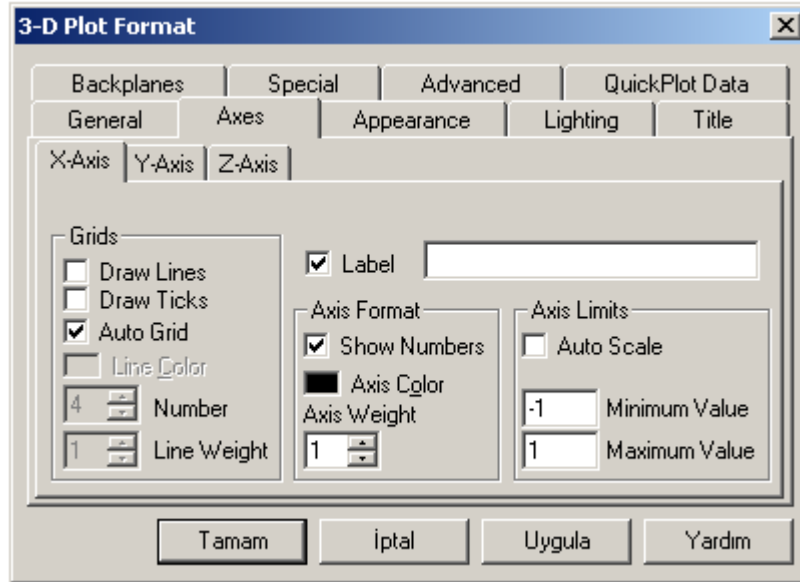
Şekil 149 : MathCAD GUI Hatası-14

DOĞRU 14 : Arayüzün görünümünün bozulmaması için, etiketler (label) belirli bir düzen içerisinde olmalıdırlar. Ayrıca, etiketlerden sonra mutlaka (:) işareti kullanılmalıdır.



Şekil 150 : MathCAD GUI Hatası-14'ün Doğrusu

HATA 15 : Ana sorun : Arayüz elemanlarının birbirleriyle olan kullanım şekillerinde yapılan hatalardır. Bu tür hatalar, arayüzün düzen ve görünümünü bozmaktadır. Arayüzde, etiket (label) ile metin alanının (textfield) kullanım şeklinin yanlış olması, hataya sebep olmuştur. Bununla birlikte, etiket ile değer değiştirme düğmesinin de kullanım şekli yanlıştır.



Şekil 151 : MathCAD GUI Hatası-15

DOĞRU 15 : Arayüz elemanlarının birbirleriyle olan kullanım şekilleri, arayüzün düzen ve görünümünü bozmamalıdır. Arayüzde, etiket (label) ile metin alanının (textfield) kullanım şekli; etiket solda, metin alanı sağda şeklinde olmalıdır. Ayrıca, etiket ile değer değiştirme düğmesinin kullanım şekli de; etiket solda, değer değiştirme düğmesi sağda şeklinde olmalıdır.

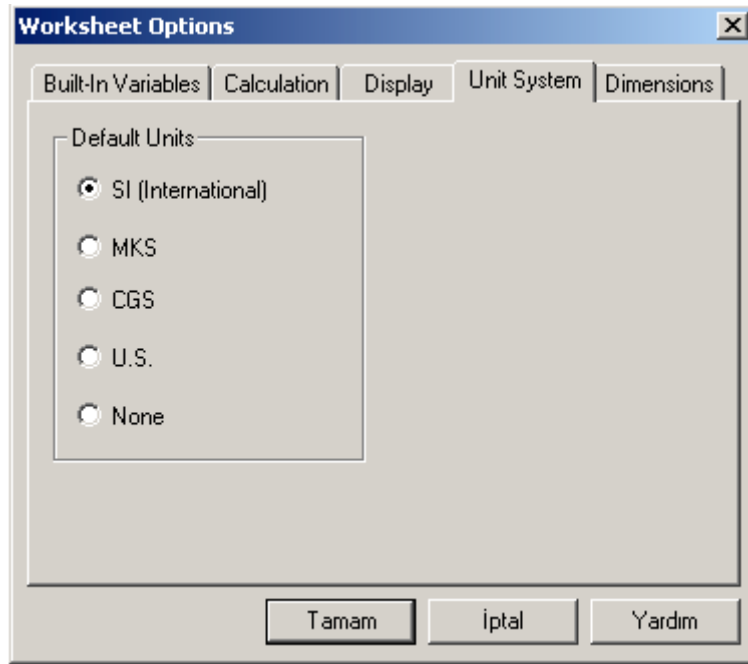
Axis Weight :

Minimum Value :

Maximum Value :

Şekil 152 : MathCAD GUI Hatası-15'in Doğrusu

HATA 16 : Ana sorun : Yanlış arayüz elemanının kullanılmasıdır. Etiket (label) yerine çerçeve kullanılması, arayüzde görünümü bozmaktadır. Arayüzde “*Default Units*” bölümünün, çerçeve (frame) içerisinde kullanılması, hataya sebep olmuştur.



Şekil 153 : MathCAD GUI Hatası-16

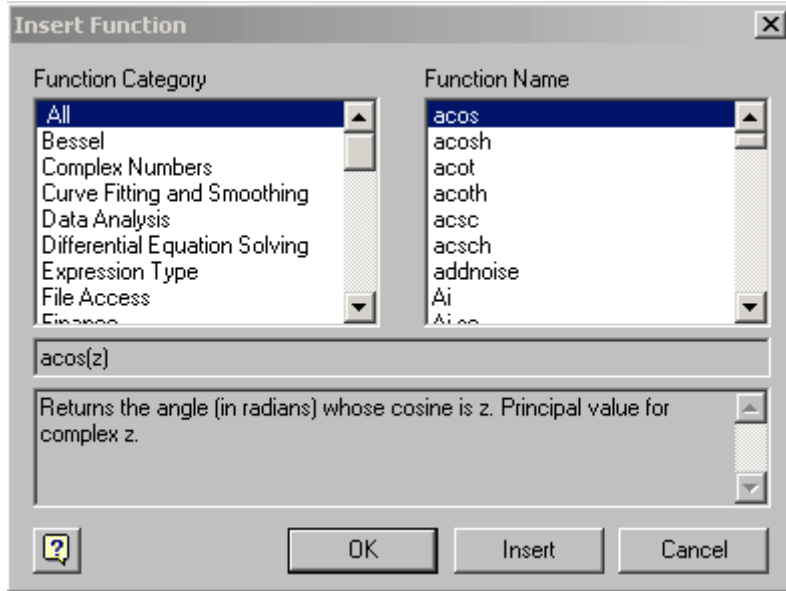
DOĞRU 16 : Çerçeve (frame) yerine, etiket (label) kullanılması gerekir. “*Default Units*” bölümündeki çerçevenin (frame) kaldırılması gerekmektedir. Çerçeve yerine, etiket (label) kullanılmalıdır.

Default Units :

- SI (International)
- MKS
- CGS
- U.S.
- None

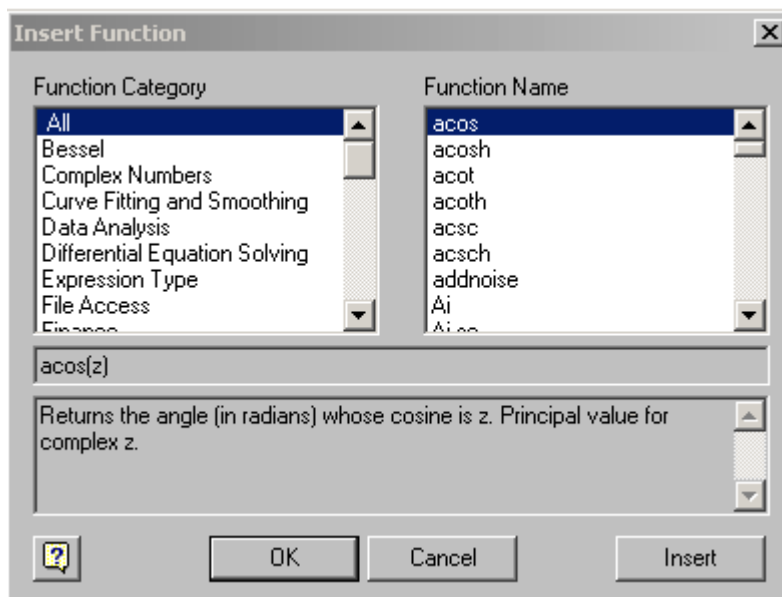
Şekil 154 : MathCAD GUI Hatası-16'nın Doğrusu

HATA 17 : Diyalog Kutusu (Dialog Box) içerisinde, “OK”, “Cancel”, “Close”, “Help” ve “Apply” düğmelerinin (button) birbirlerinden ayrı olmaları, birarada olmamaları hataya sebep olmuştur.



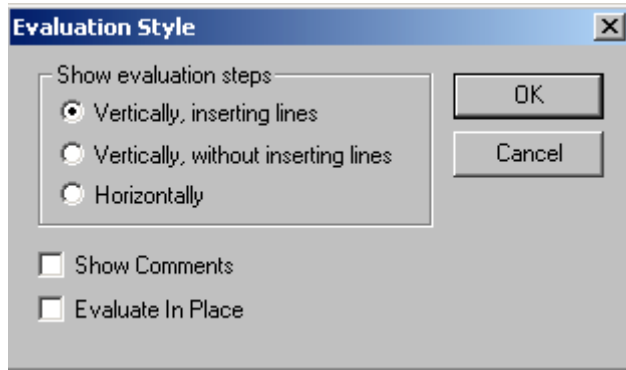
Şekil 155 : MathCAD GUI Hatası-17

DOĞRU 17 : Diyalog Kutusu (Dialog Box) içerisinde, “OK”, “Cancel”, “Close”, “Help” ve “Apply” düğmeleri (button) kullanıcılar tarafından çok fazla kullanıldığı için, bu düğmeler diğer düğmelerden ayrı olmalıdırlar. Bu düğmelerin göz önünde bulunmaları için, kendi aralarında bitişik, diğer düğmelerden ise ayrı olmaları gerekmektedir.



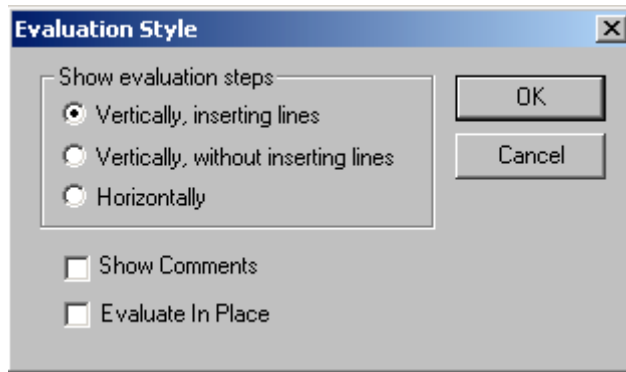
Şekil 156 : MathCAD GUI Hatası-17'nin Doğrusu

HATA 18 : Grafiksel kullanıcı arayüz (GUI) elemanlarının düzen bozuklukları, hataya sebep olmuştur. Arayüzde kullanılan seçenek düğmesi (option button) ile denetim kutusunun (check box) aynı hizada olmamaları, düzen ve görünüm bozukluğuna yol açmıştır.



Şekil 157 : MathCAD GUI Hatası-18

DOĞRU 18 : Arayüzde seçenek düğmesi (option button) ve denetim kutusu (check box) birarada kullanılıyorsa, aynı hizada olmalıdırlar. Hiza ve sıralama keyfe göre olmamalıdır. Arayüzdeki GUI elemanları kötü bir görünümü önlemek için, tutarlı, düzenli ve göze hoş gelecek bir şekilde sıralanmalıdırlar.



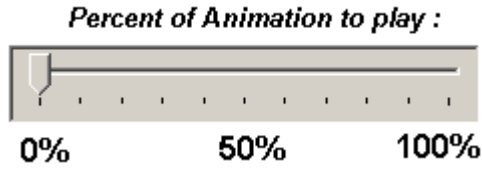
Şekil 158 : MathCAD GUI Hatası-18'nin Doğrusu

HATA 19 : Ana sorun : Kaydırma çubuğu'nun (slider) kullanımı ile ilgili yapılan hatadır. Arayüzde, kaydırma çubuğu yanlış kullanılmıştır. Kaydırma çubuğu'nun ne iş yaptığı ve hangi amaçla kullanıldığı belli değildir. Ayrıca, kaydırma çubuğu ile ilgili kullanıcıya bilgi verilmemektedir.



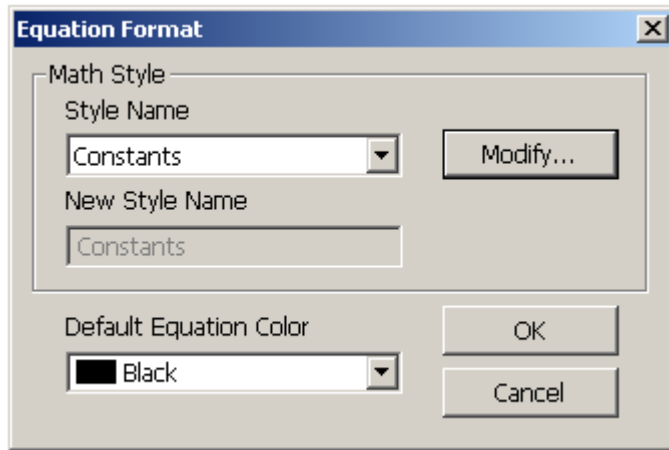
Şekil 159 : MathCAD GUI Hatası-19

DOĞRU 19 : Kaydırma çubuğu'nun (slider) üstünde, kaydırma çubuğu'nun ne işe yaradığını anlatan açıklayıcı bir bilgi verilmelidir. Ayrıca, kaydırma çubuğu'nun alt tarafında ise, kaydırma çubuğu'nun yüzdesi kullanıcıya gösterilmelidir. Bu durum, işin ne kadarının yapıldığını göstermesi açısından kullanıcıya yardımcı olacaktır.



Şekil 160 : MathCAD GUI Hatası-19'un Doğrusu

HATA 20 : Arayüz elemanlarından etiket (label) ile denetim kutusu'nun (checkbox) yanlış kullanılması hataya sebep olmuştur. Bu durum, arayüzün düzen ve görünümünün bozulmasına yol açmaktadır.



Şekil 161 : MathCAD GUI Hatası-20

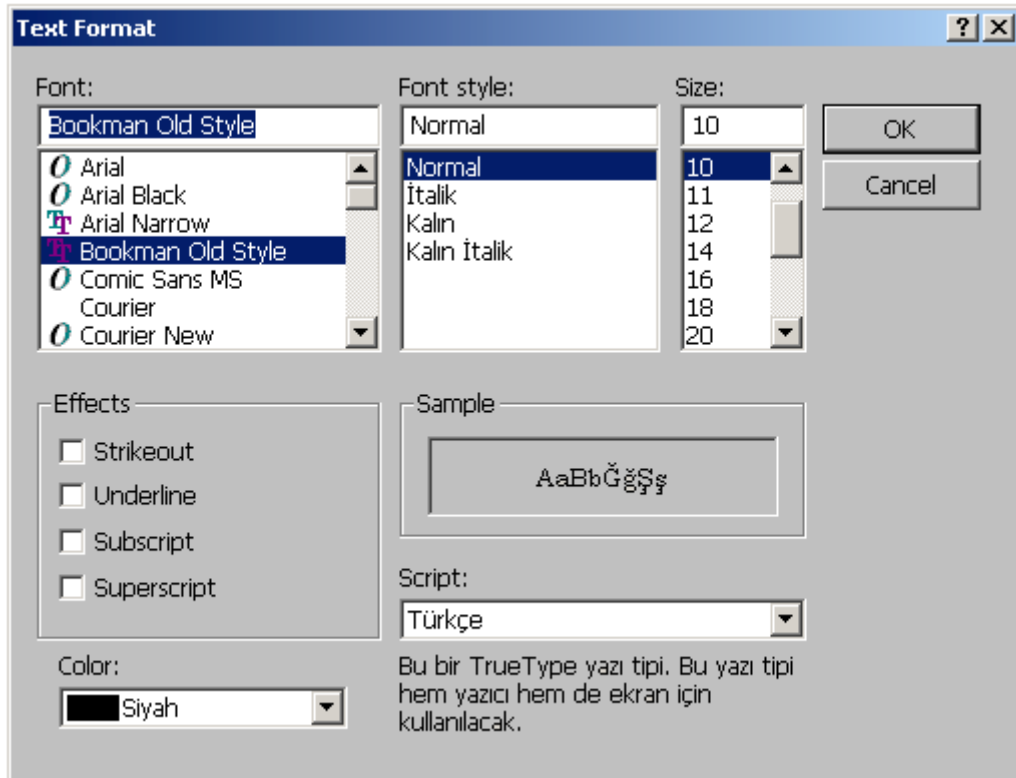
DOĞRU 20 : Arayüzde etiket (label), denetim kutusu (checkbox) ile birlikte kullanılırken; hiza ve düzen olarak etiket solda, denetim kutusu ise sağda olmalıdır. Etiket ve denetim kutusu'nun kullanım şekli aşağıdaki gibi olmalıdır :

Style Name :

Default Equation Color :

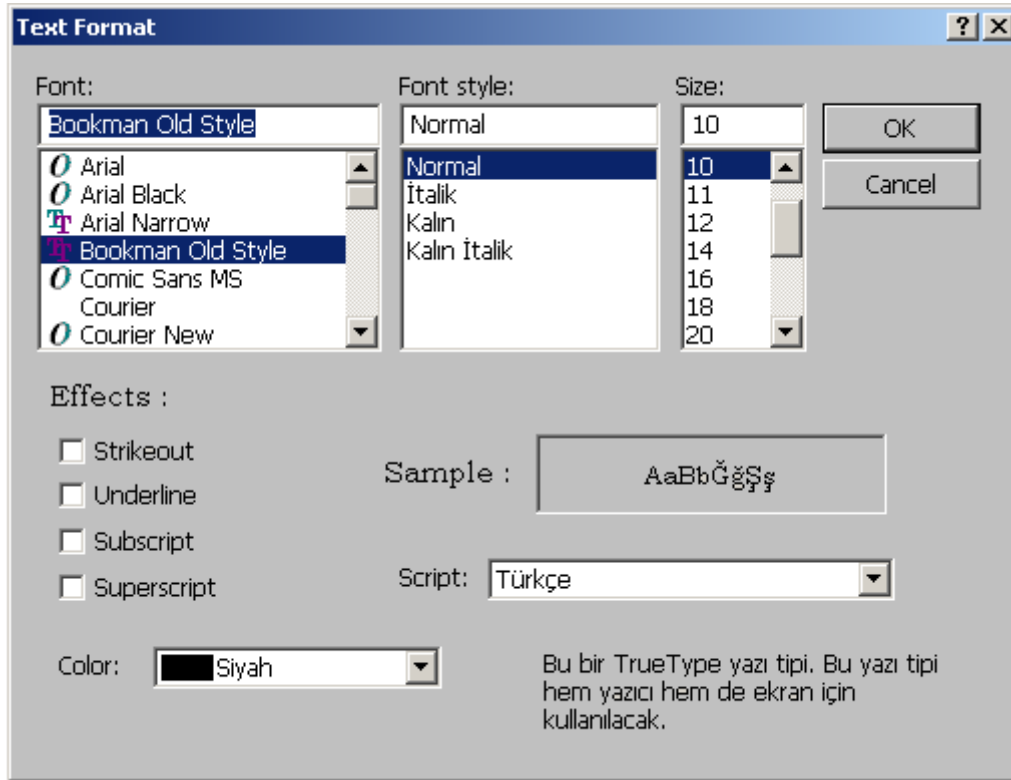
Şekil 162 : MathCAD GUI Hatası-20'nin Doğrusu

HATA 21 : Hatanın ana nedeni : “*Sample*” yazısının olduğu yerdedir. “*Sample*” alanı, metin ile ilgili seçilen her türlü özelliğin (yazı tipi, yazı tipi biçimi ve yazı boyutu) nasıl görüneceğini gösteren yerdir. Bu durum, yazıların “*Sample*” çerçevesi (frame) içinde bağlı kalmasına sebep olmakta ve arayüzde anlaşmazlığa yol açmaktadır. Ayrıca, etiket’lerin (label), grup kutu’ları (combo box) ile kullanım şekli yanlıştır. Bunların dışında, denetim kutularının (checkbox) içinde bulunduğu “*Effects*” bölümünün çerçeve içerisinde olması, bir başka hata nedenidir.



Şekil 163 : MathCAD GUI Hatası-21

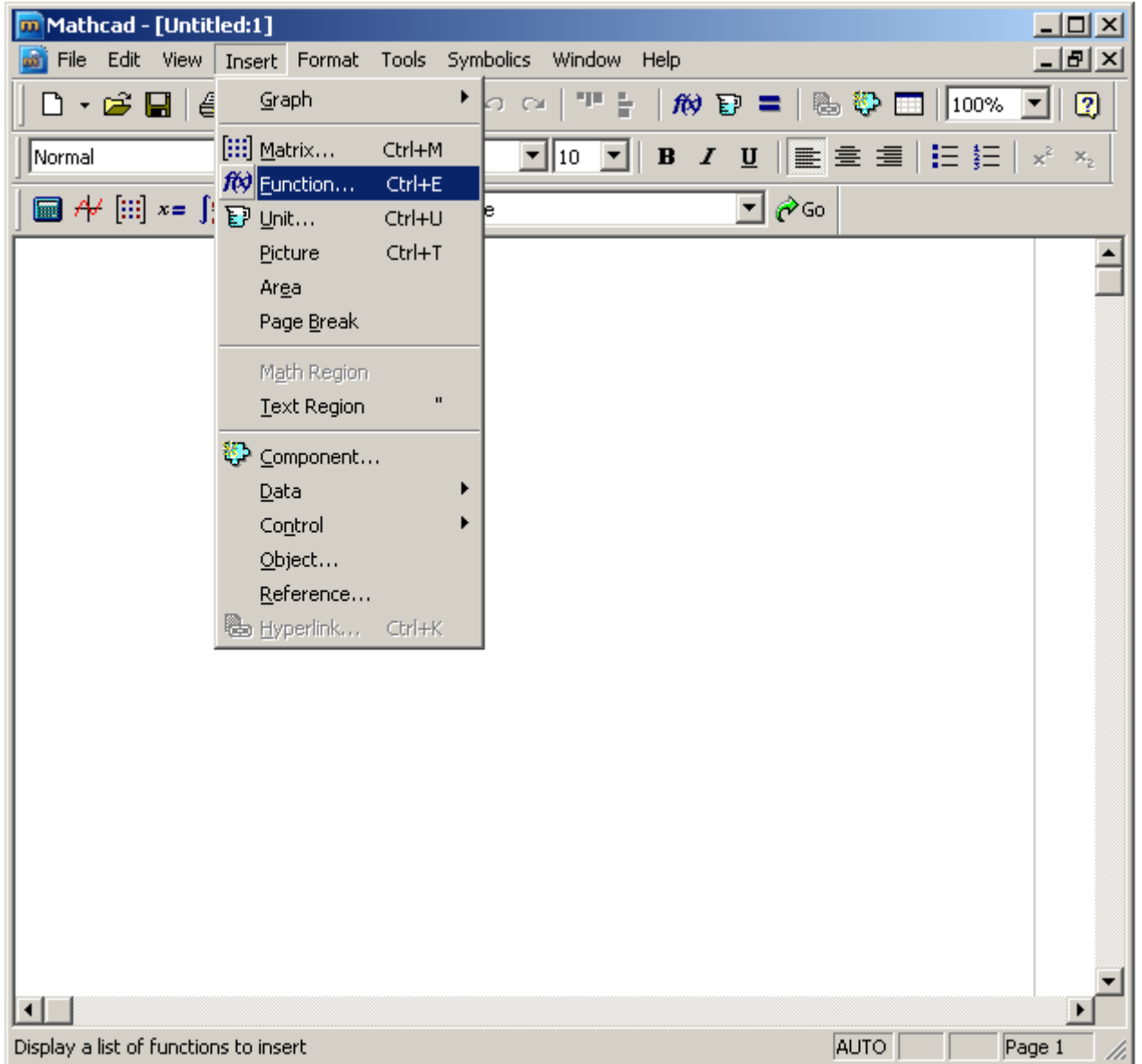
DOĞRU 21 : Arayüzde, “*Sample*” çerçevesi (frame) gereksizdir ve kaldırılması gerekmektedir. Çerçeve yerine etiket (label) kullanılmalıdır. Ayrıca, arayüzde grup kutusu (combo box) ile etiket’in (label) kullanımı ile ilgili olarak iki tane yanlış vardır. Bu iki GUI elemanının kullanımının standart ve doğru şekli şöyle olmalıdır : Etiket solda, grup kutusu ise sağ tarafta bulunmalıdır. Bunların dışında, denetim kutularının (checkbox) içinde bulunduğu “*Effects*” alanı çerçeve değil, etiket olmalıdır.



Şekil 164 : MathCAD GUI Hatası-21’in Doğrusu

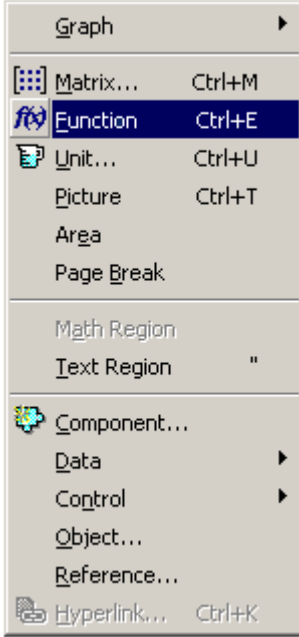
4.5.3 MathCAD'teki Metinsel Hatalar (Textual Bloopers)

HATA 22 : Arayüzde “...” işaretinin kullanımından kaynaklanan bir hata vardır. “*Insert*” ana menüsünün alt menüsü olan “*Function*” komutundan sonra “...” yorum satırının kullanılması, hataya sebep olmuştur. “...” işareti gereksiz yere kullanılmıştır.



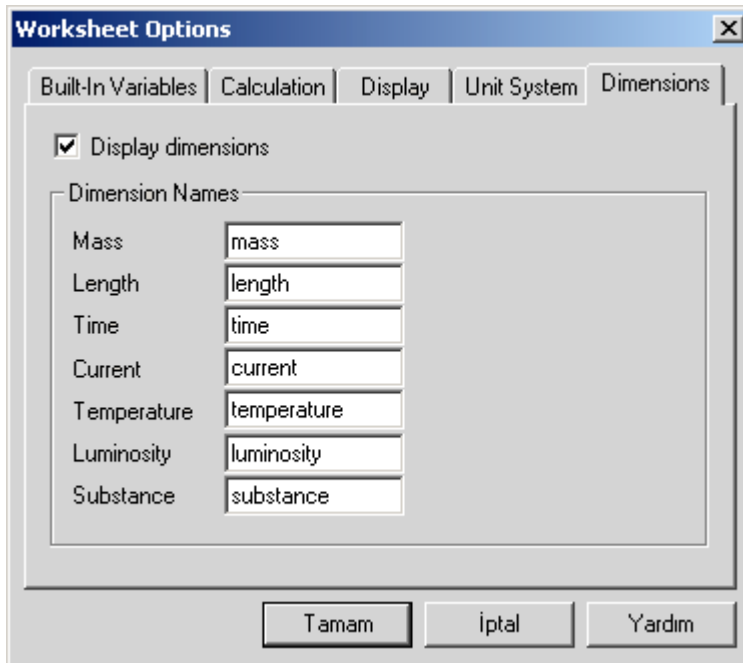
Şekil 165 : MathCAD GUI Hatası-22

DOĞRU 22 : Kullanıcılar, “*Function*” menüsüne tıklayınca adından da anlaşılacağı üzere MathCAD ile ilgili fonksiyonları kullanabileceğini bilmektedirler. Yani, fonksiyonların kullanımını ilgilendiren yeni bir pencerenin açılacağı bellidir. Yeni bir pencere açılacakken “...” işareti kullanılmamalıdır. Bu yüzden, “*Function*” menüsünden sonra “...” işareti gereksizdir ve kaldırılması gerekir.



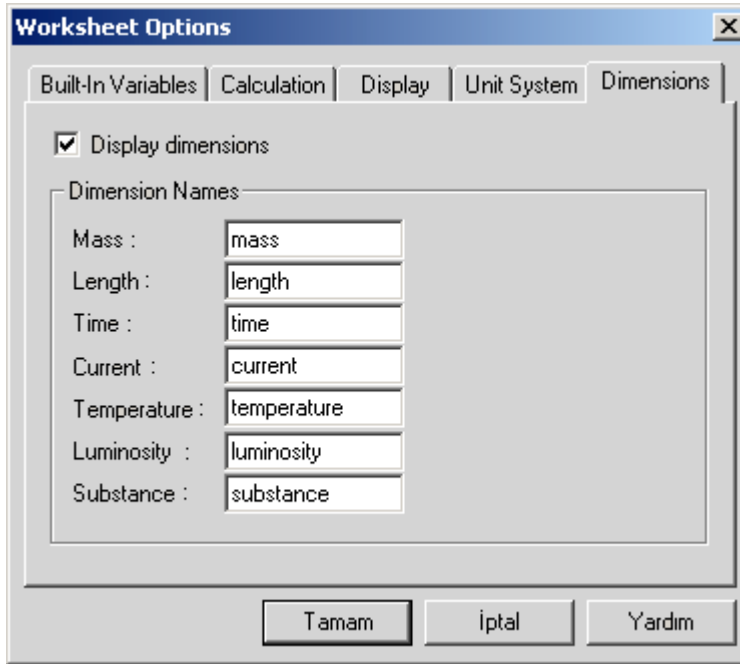
Şekil 166 : MathCAD GUI Hatası-22'nin Doğrusu

HATA 23 : Arayüzde etiketlerden (label) sonra (:) işaretinin kullanılmaması, hataya sebep olmuştur.



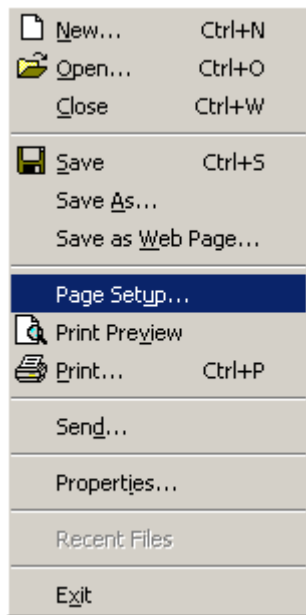
Şekil 167 : MathCAD GUI Hatası-23

DOĞRU 23 : Arayüzde, her etiketten (label) sonra mutlaka (:) işareti kullanılmalıdır.

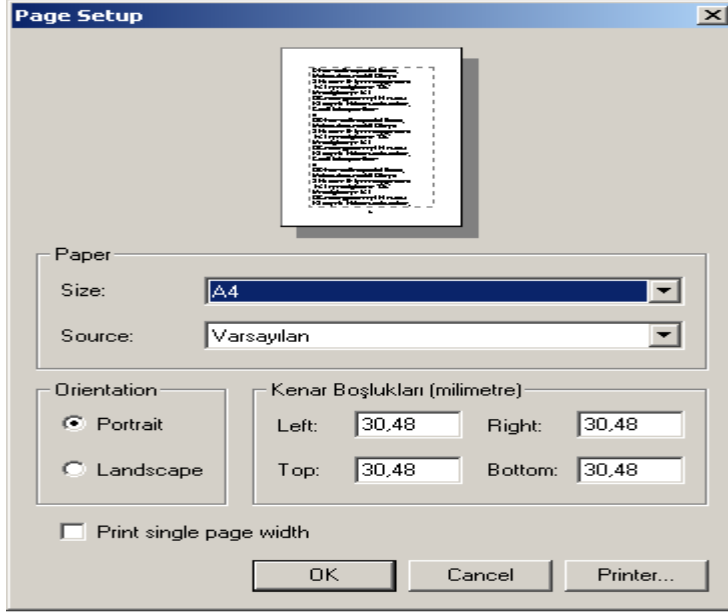


Şekil 168 : MathCAD GUI Hatası-23'ün Doğrusu

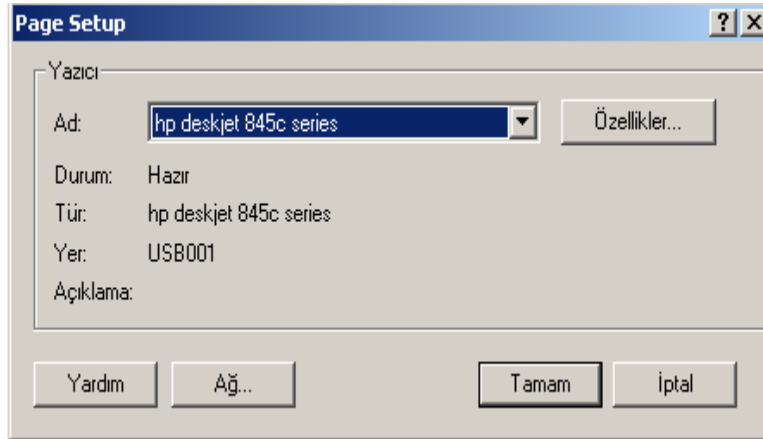
HATA 24 : İki farklı pencerede aynı başlığı kullanmak, hataya sebep olmuştur. Birinci durumda; “File” menüsünün “Page Setup...” alt menüsünü tıkladığımızda açılan pencerenin başlığının adı : “Page Setup” tır. İkinci durumda ise; “Page Setup” penceresi içerisindeki “Printer...” düğmesine (button) tıkladığımızda açılan pencerenin başlığının adı yine : “Page Setup” olmaktadır. Bu durum, kullanıcıların kafalarının karışmasına ve arayüzde karışıklığa yol açmaktadır.



Şekil 169 : MathCAD GUI Hatası-24'ün Birinci Arayüzü



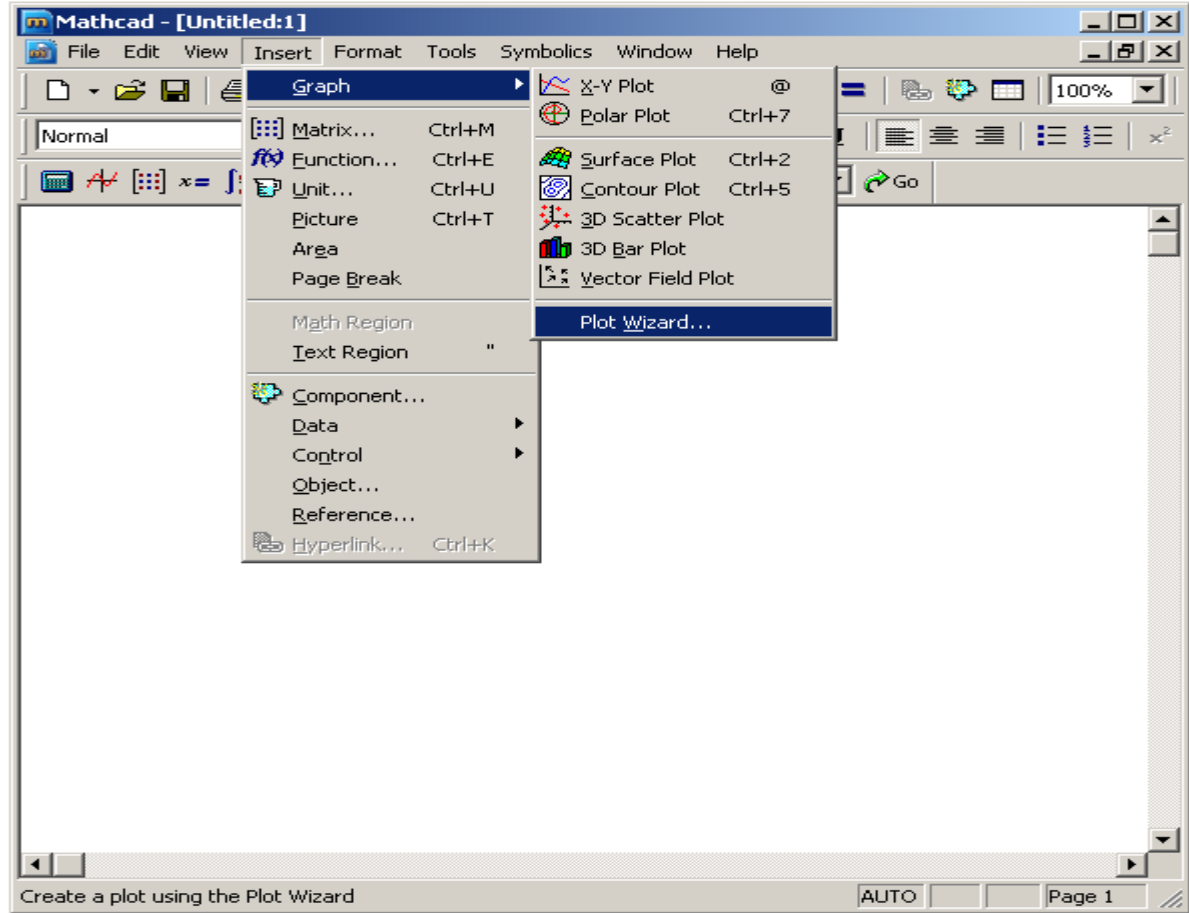
Şekil 170 : MathCAD GUI Hatası-24'ün İkinci Arayüzü



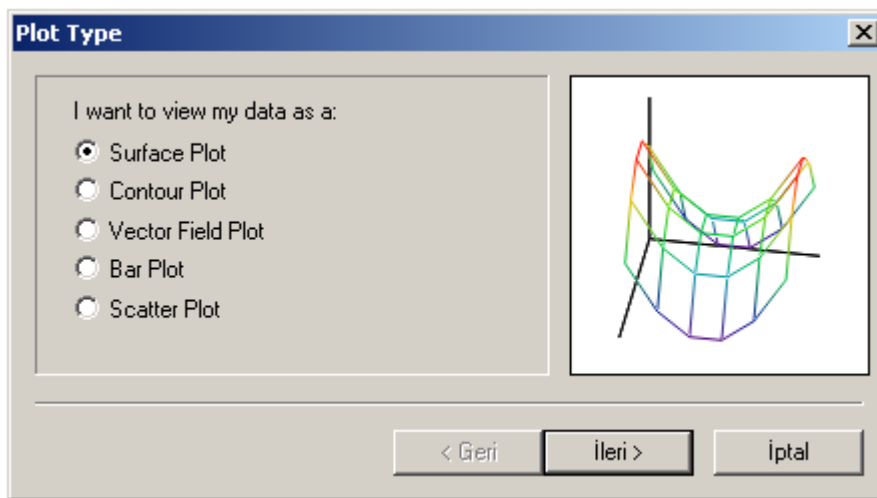
Şekil 171 : MathCAD GUI Hatası-24'ün Üçüncü Arayüzü

DOĞRU 24 : İki farklı pencerede aynı başlık ismi kullanılmamalıdır. Her pencerenin tek ve kendine has bir başlığı olmalıdır. Örneğin; “Printer...” düğmesine (button) tıkladığımızda açılan pencerenin başlığının adı, diğer pencere başlıkları ile karışmasın diye “Page Setup” yerine “Printer Properties” olabilirdi.

HATA 25 : Pencere başlıklarının farklı olması, hataya sebep olmuştur. “Graph” menüsü altındaki “Plot Wizard...” menüsünü tıkladığımızda, yeni açılan pencerenin başlığının “Plot Type” olması sorun yaratmakta ve karışıklığa yol açmaktadır.



Şekil 172 : MathCAD GUI Hatası-25'in Birinci Arayüzü

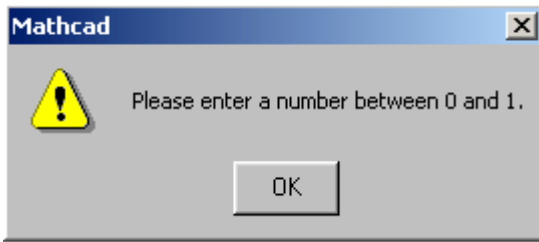


Şekil 173 : MathCAD GUI Hatası-25'in İkinci Arayüzü

DOĞRU 25 : Tıklanılan menü adı ile yeni açılan pencerenin başlığının ismi aynı olmalıdır. Bu durumda, “*Plot Wizard...*” menüsüne tıkladığımızda açılan yeni pencerenin ismi, “*Plot Type*” değil, “*Plot Wizard*” olmalıydı.

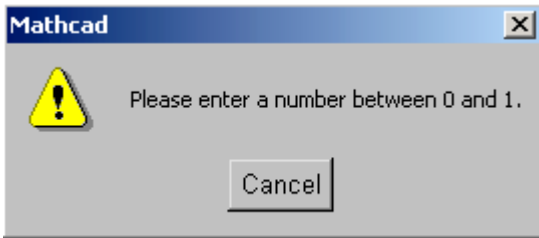
4.5.4 MathCAD’teki Etkileşim Hataları (Interaction Bloopers)

HATA 26 : “*OK*” düğmesinin (button), “*Cancel*” düğmesi gibi kullanılması, hataya sebep olmuştur. Kullanıcı yaptığı hatayı “*OK*” düğmesine basarak zorla kabul etmiş olmaktadır.



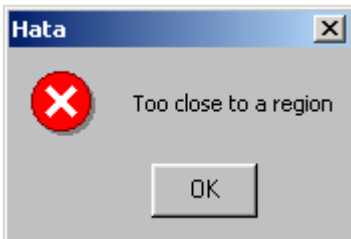
Şekil 174 : MathCAD GUI Hatası-26

DOĞRU 26 : Mesajdaki hatanın giderilmesi için, “*OK*” düğmesi (button), “*Cancel*” gibi kullanılmamalıdır. Mesajda “*OK*” düğmesi yerine, “*Cancel*” düğmesi kullanılmalıdır.



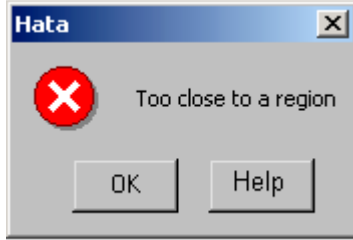
Şekil 175 : MathCAD GUI Hatası-26’nın Doğrusu

HATA 27 : Kullanıcının yaptığı yanlış ile ilgili olarak bilgilendirilmemesi ve yapılan hata hakkında kullanıcıya yardım edilmemesi, hataya sebep olmuştur.



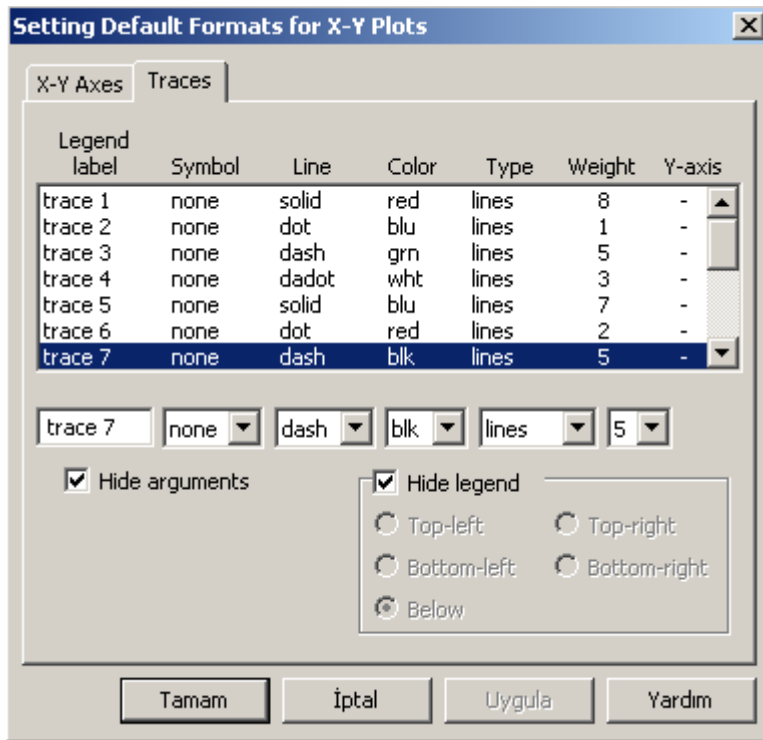
Şekil 176 : MathCAD GUI Hatası-27

DOĞRU 27 : Kullanıcının yaptığı hata karşısında ne yapması gerektiğinin söylenmesi ve kullanıcının uyarılması için, “*Help*” (yardım) düğmesinin (button) de mesajda olması gerekmektedir. Böylece mesajdaki hata, ortadan kaldırılmış olacaktır.



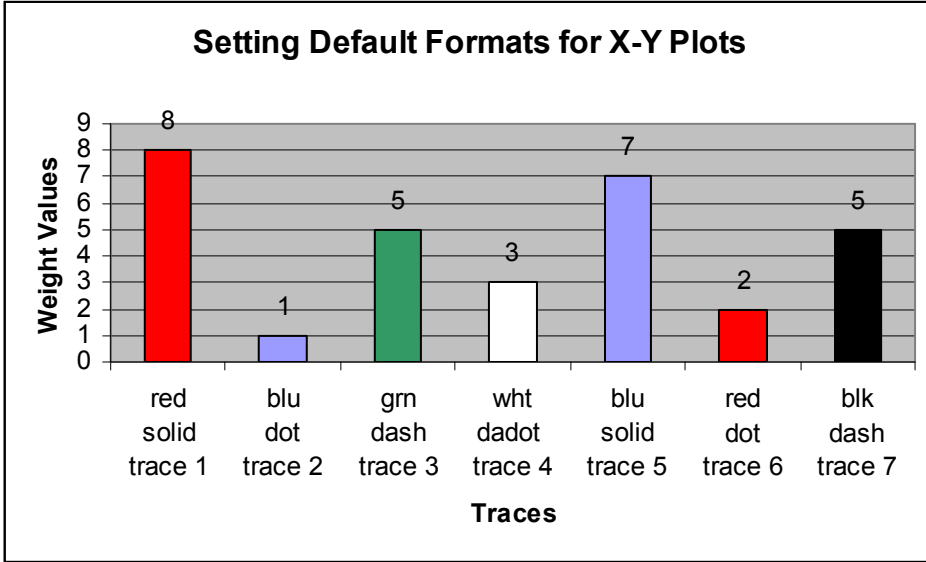
Şekil 177 : MathCAD GUI Hatası-27'nin Doğrusu

HATA 28 : Arayüzde verilerin, liste biçiminde sunulması, hataya sebep olmuştur. Bu şekilde bir gösterim, verilerin anlaşılmasını zorlaştırmaktadır.



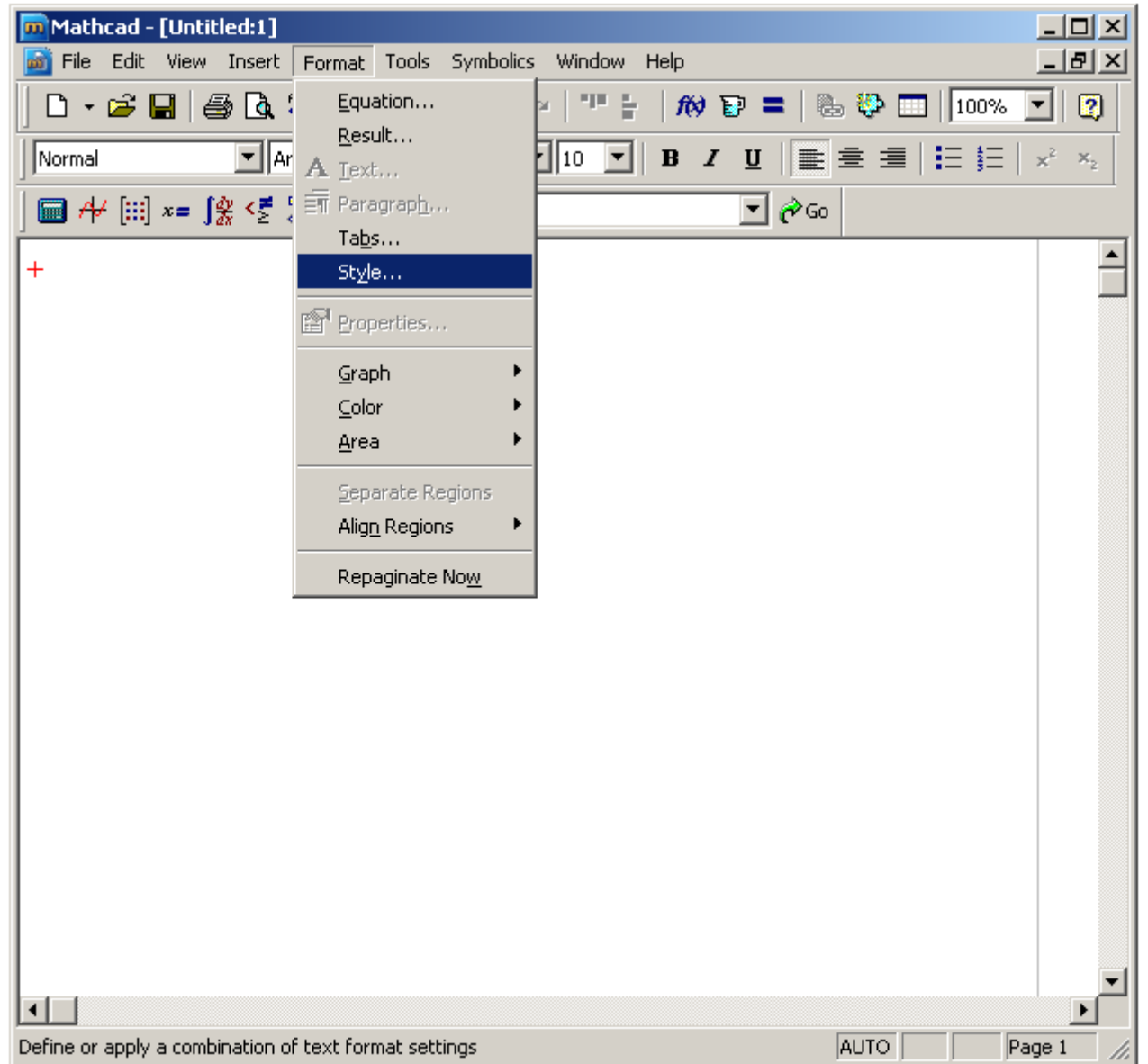
Şekil 178 : MathCAD GUI Hatası-28

DOĞRU 28 : Verileri, grafikler ile göstermek kullanıcının arayüzde anlatılmak istenileni anlamasını kolaylaştıracaktır. Ayrıca, verilerin liste biçiminde gösterildiği arayüzde, verilere birer renk verildiğinden, her veriyi kendisini temsil eden renkle göstermek, kullanıcı açısından daha bilgilendirici olacaktır. Örneğin; arayüzde birinci veri, kırmızı renkle temsil edildiğinden, bu verinin grafikte kırmızı renk ile gösterilmesi, kullanıcı açısından yararlı olacaktır.

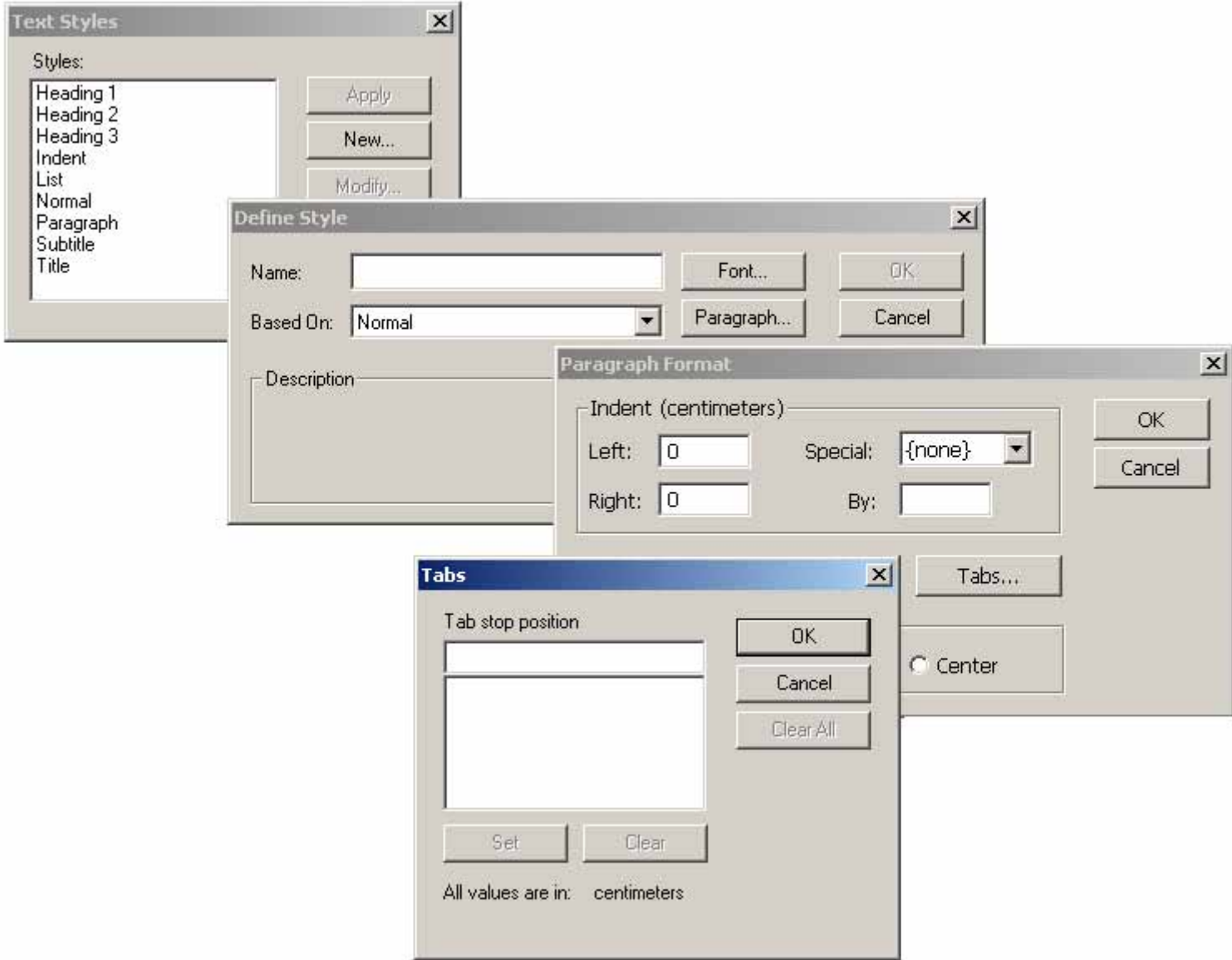


Şekil 179 : MathCAD GUI Hatası-28'in Doğrusu

HATA 29 : Diyalog Kutusu'nda, (Dialog Box) derin bir hiyerarşinin olması, kullanıcıları esas amacından ve yapmak istedikleri işlerden uzaklaştırmakta, kullanıcıların dikkatinin başka yöne çevrilmesine sebep olmakta ve kullanıcıların izledikleri yolu kaybetmelerine yol açmaktadır. Arayüzde “*Tabs*” bölümüne ulaşmak için kullanılan hiyerarşi şöyledir : “*Format*” menüsünün içerisinde “*Style...*” alt menüsü vardır. “*Text Style*” içerisinde “*New...*” bölümü vardır. Daha sonra, “*Define Style*” içerisinde “*Paragraph...*” bölümü vardır. Ve en son olarak, “*Paragraph Format*” içerisinde “*Tabs...*” bölümü vardır.



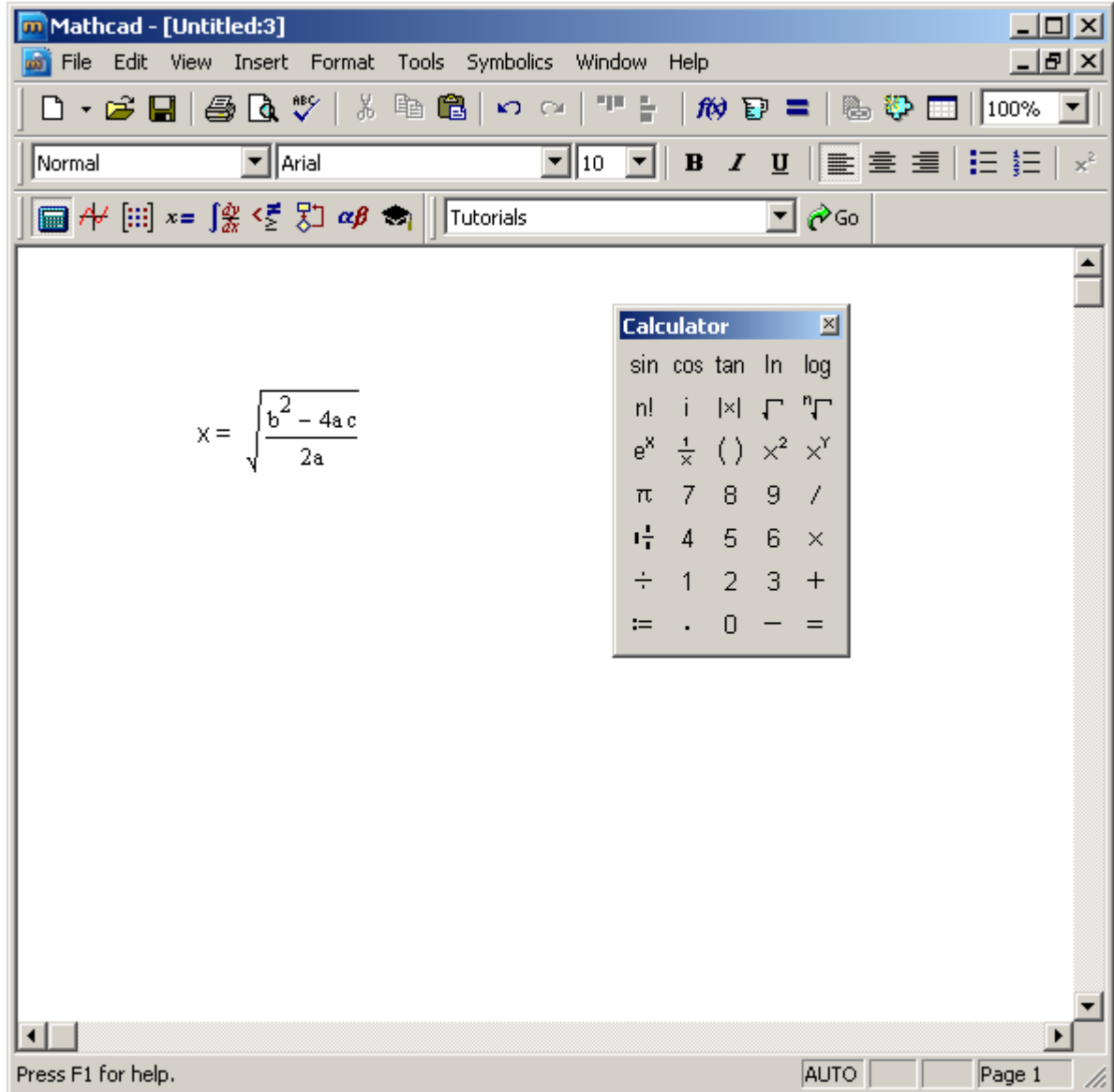
Şekil 180 : MathCAD GUI Hatası-29'un Birinci Arayüzü



Şekil 181 : MathCAD GUI Hatası-29'un İkinci Arayüzü

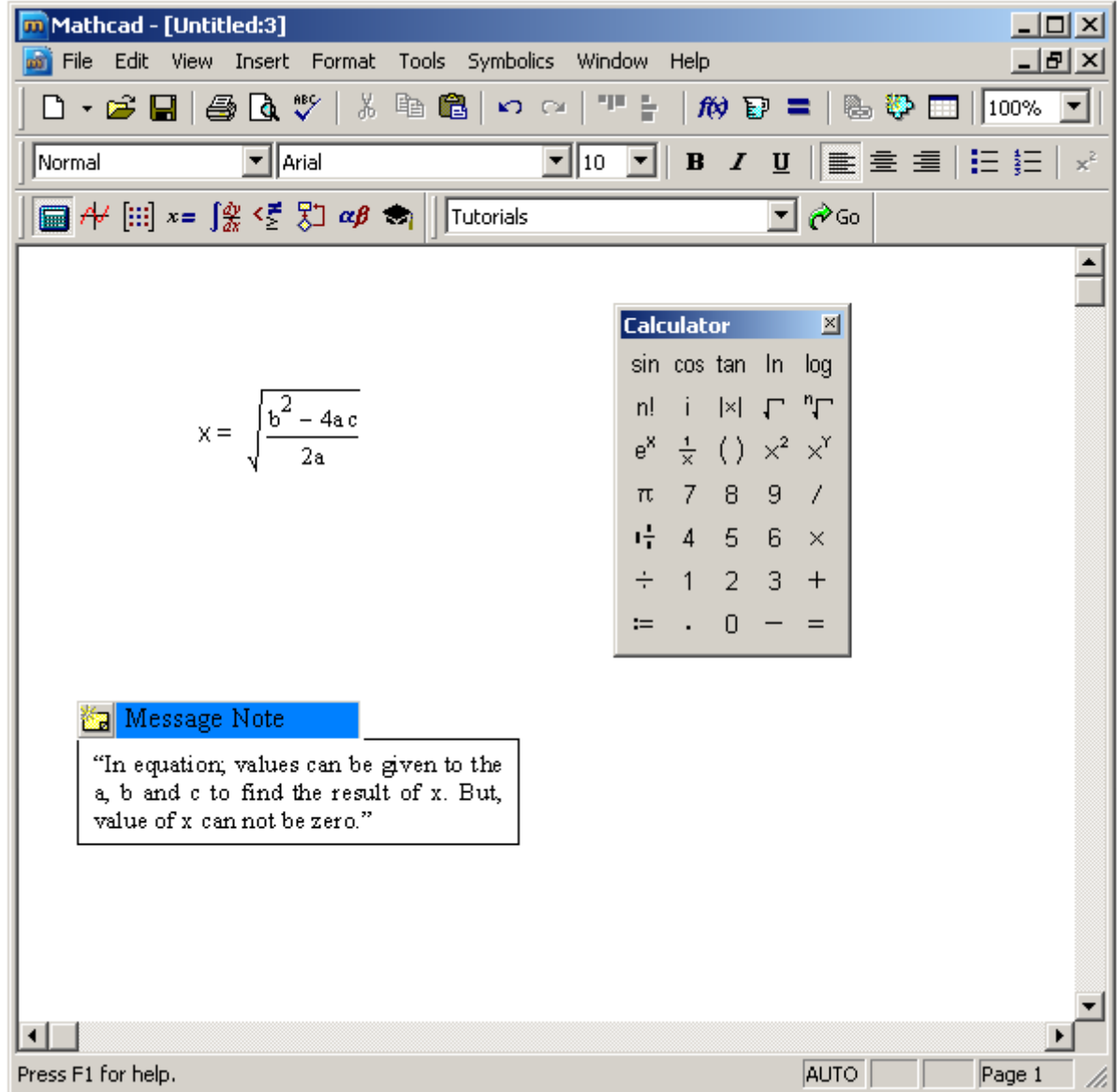
DOĞRU 29 : Diyalog Kutu'sundaki (Dialog Box) derin bir hiyerarşiyi önlemek için, diyalog kutusu'nun bölüm ve aşamalarının ikiden fazla olmaması gerekmektedir. Diyalog Kutusu, kullanıcının yolunu kaybettirmeyecek ve amacına ulaştıracak şekilde basit bir yapıda olmalıdır.

HATA 30 : İfadenin ne anlama geldiğinin açıklamasının yapılmamış olması, hataya sebep olmuştur. Formülün amacının ne olduğunun, kullanıcılar tarafından bilinmemesi sorun yaratmaktadır.



Şekil 182 : MathCAD GUI Hatası-30

DOĞRU 30 : Kullanıcılara ifadeyi açıklayıcı şekilde, mesaj verilmelidir. Arayüzde pencerenin altında uyarıcı bilgi olması gerekmektedir. Mesaj notu ile kullanıcı, yapılmak istenen işlem hakkında bilgilendirilmelidir.



Şekil 183 : MathCAD GUI Hatası-30'un Doğrusu

4.6. Genel Değerlendirme

Görülüyor ki; çok yaygın kullanılmakta olan MATLAB ve MathCAD programlarında bir çok grafiksel kullanıcı arayüz (GUI) hatası bulunmaktadır. Bu tez çalışmasında bu hatalara işaret edilmiş ve hatalar tek tek incelenmiştir. Bu hatalar doğrultusunda, kullanıcılar açısından uygun ve yararlı bir grafiksel kullanıcı arayüzü oluşturmak için gerekli olan açıklamalar yapıp, bilgiler verilmiş ve bu kullanıcı arayüz hatalarının doğrularının nasıl olması gerektiği belirtilmiştir. Bu bakımdan tasarımcıların bu bilgiler ekseninde programlar geliştirmeleri önerilir. Tezde MATLAB ve MathCAD programlarının incelenmesi sonucu, bu programlardaki GUI hata türleri sayısı ve toplam grafiksel kullanıcı arayüz (GUI) hata sayısı, aşağıdaki tabloda gösterilmiştir :

Grafiksel Kullanıcı Arayüz (GUI) Hata Türleri Sayısı		MATLAB	MathCAD
	1. GUI Elemanlarındaki Hataların Sayısı	8	9
	2. Düzen ve Görünüm Hatalarının Sayısı	18	12
	3. Metinsel Hataların Sayısı	6	4
	4. Etkileşim Hatalarının Sayısı	12	5
TOPLAM GUI HATA SAYISI	44	30	

Tablo 10 : MATLAB & MathCAD GUI Hata Türleri Sayısını Gösteren Liste

MATLAB & MathCAD programlarındaki GUI hata türleri sayısını gösteren liste'yi incelediğimizde; MathCAD programındaki GUI hatalarının sayısı MATLAB programına göre daha azdır. Tezde yapılan grafiksel kullanıcı arayüz (GUI) hataları incelemelerinde; MATLAB'te kırkdört GUI hatasına rastlanırken, MathCAD'te ise otuz GUI hatasına rastlanmıştır. Bunun en büyük nedeni ise; MATLAB programının içerisinde kendine özgü bir özelliği olan ve çeşitli fonksiyonların biraraya gelmesi ile oluşan araç kutuları (*Toolbox*) adı verilen yapıların olmasıdır. Grafiksel kullanıcı arayüz (GUI) hataları tek tek incelenip MATLAB ile MathCAD programlarını kıyasladığımızda; MATLAB'te GUI elemanlarındaki hataların sayısı sekiz iken, MathCAD'teki GUI elemanları hata sayısı dokuz'dur. Ayrıca, MATLAB'teki düzen ve görünüm hatalarının sayısı onsekiz iken, MathCAD'teki düzen ve görünüm hatalarının sayısı oniki'dir. Bununla birlikte, MATLAB'teki metinsel hataların sayısı altı iken, MathCAD'teki metinsel hataların sayısı dört'tür. Son olarak, MATLAB'teki etkileşim hatalarının sayısı oniki iken, MathCAD'teki etkileşim hatalarının sayısı beş'tir.

Tezde MATLAB ve MathCAD programlarındaki grafiksel kullanıcı arayüz (GUI) hata türleri ele alınıp, incelendiğinde ortaya çıkan en önemli sonuçlardan biri şudur : MATLAB ve MathCAD programlarındaki grafiksel kullanıcı arayüz elemanlarında yapılan hatalar ele alındığında; menülerde yapılan hatalar, kısayol tuşlarında yapılan hatalar ve arayüz elemanlarının fonksiyonlarının yanlış ve farklı kullanılmasından doğan hatalar, MATLAB ve MathCAD programlarının her ikisinde de benzerlik göstermektedir. MATLAB ve MathCAD programlarındaki düzen ve görünümde yapılan hatalar ele alındığında; GUI elemanlarının düzen bozuklukları, arayüz elemanlarının iç içe sokulması, etiketlerin dizilme hataları, yazı tiplerinin ufak olması ve arayüz arka plan renginin görünümü bozması gibi hatalar, MATLAB ve MathCAD programlarının her ikisinde de benzerlik göstermektedir. MATLAB ve MathCAD programlarındaki metinsel hatalar ele alındığında; sadece pencere başlıklarının farklı olmasının yarattığı hata, MATLAB ve MathCAD programlarının her ikisinde de benzerlik göstermektedir. Son olarak, MATLAB ve MathCAD programlarındaki etkileşim hataları ele alındığında; bilginin çok kötü ve fakirce sunulması, veri sonuçlarının liste biçiminde sunulmasının yarattığı belirsizlik, kullanıcılara ifade ve işlemleri açıklayıcı bilginin verilmemesi ve uyarı mesajlarında yapılan hatalar, MATLAB ve MathCAD programlarının her ikisinde de benzerlik göstermektedir.

5. KAYNAKÇA

- [1] <http://www.ceturk.com/dersoku.asp?id=197>
- [2] <http://www.teknohaber.net/makale.php?id=30808>
- [3] <http://www.papatya.info/matlab.htm>
- [4] www.math.mtu.edu/~msgocken/intro/node2.html#SECTION00020000000000000000
- [5] <http://www.math.ufl.edu/help/matlab-tutorial/matlab-tutorial.html#SEC13>
- [6] <http://www.mathworks.com/products/matlab/>
- [7] <http://www.indiana.edu/~statmath/math/matlab/overview.html>
- [8] <http://www.yildiz.edu.tr/~inan/uq.edu.au.pdf>
- [9] Tom Huber, MATLAB'in Temelleri, Envision It Workshop, Şubat 1997
- [10] http://www.mathworks.com/access/helpdesk/help/techdoc/learn_matlab/ch1intro.html#12671
- [11] <http://ktef.karaelmas.edu.tr/bilgisayar/kutuphane/kutuphane.htm>
- [12] [http://www.hrs.co.nz/\(3a5z1c55tcs52ozt2phoz4q4\)/prod_overview.aspx?ProductID=89](http://www.hrs.co.nz/(3a5z1c55tcs52ozt2phoz4q4)/prod_overview.aspx?ProductID=89)
- [13] <http://www.mathworks.com/products/matlab/description5.html>
- [14] <http://www.mathworks.com/products/matlab/description4.html>
- [15] http://www.rit.edu/~pnveme/MATLAB_Course/MATLAB_1.html#What%20is%20MATLAB
- [16] <http://www.genie.uottawa.ca/~mcg4345/matlabguidelines.htm>
- [17] <http://charlotte.at.northwestern.edu/bef/MATLAB/Toolboxes.html>
- [18] <http://www.math.siu.edu/MATLAB/tutorial1.pdf>
- [19] <http://www.mathworks.com/products/matlab/description2.html>
- [20] http://www.rit.edu/~pnveme/MATLAB_Course/MATLAB_vars.html
- [21] http://www.yildiz.edu.tr/~inan/Temel_Bilgiler.doc
- [22] Robert DeMoyer, Sistem Mühendisliği Bölümü, U.S. Naval Akademisi, MATLAB GUIDE(Graphical User Interface Development Environment) Kullanımı, ASEE/IEEE Frontiers in Education Conference, November 1999, <http://fie.engrng.pitt.edu/fie99/papers/1120.pdf>
- [23] http://www.m-tutor.usask.ca/about/1998_m-tutor_evaluation.pdf
- [24] <http://www.mpassociates.gr/software/distrib/science/mathsoft/usma.html>
- [25] Michelle T. Lehmann, Matematik, Mühendislik Öğrencileri MathCAD'e Sıcak Bakıyor, UW Gazette, 10 Kasım 1999, <http://www.adm.uwaterloo.ca/infoipa/Gazette/1999/nov10/mathcad.html>
- [26] <http://www.adeptscience.co.uk/products/mathsim/mathcad/academic/discipline.html>
- [27] <http://darkwing.uoregon.edu/~pchem412/math.htm>

- [28] <http://nats-www.informatik.uni-hamburg.de/~tempus/abstracts1/mathcad.html>
- [29] <http://www.mpassociates.gr/software/distrib/science/mathsoft/pro2000.html>
- [30] <http://www.chip.com.tr/yazilim/urun.asp?ID=49>
- [31] <http://www.chempute.com/mathcad.htm>
- [32] <http://www.mpassociates.gr/software/distrib/science/mathsoft/engsci.html>
- [33] Bernie Buelow, Mathsoft Engineering & Education Inc.,
Stefanie Guzikowski , Beaupre & Co. Public Relations Inc.,
MathCAD' in Genel Görünümü, Mathsoft – Cambridge,MA,
http://www.mathsoft.com/news/presskit/pdfs/Mathcad_12_Overview_04.pdf
- [34] Boris Sedacca, Electrical Products and Panel Building Dergisi,
MathSoft - MathSoft Engineering & Education, Inc., 2002.
- [35] <http://www.eng.fsu.edu/~ecollins/dynamics/mathcad.html>
- [36] <http://www.mpassociates.gr/software/distrib/science/mathsoft/corporatebenefits.html>
- [37] Miroslav Dolozilek, Anna Ryndova, Brno Üniversitesi, Makine Mühendisliği,
www.bme.hu/ptee2000/papers/doloz.pdf
- [38] www.unf.edu/~mzhan/chapter1.pdf
- [39] <http://www.benkoltd.com/yazilim/MathCAD/MathCAD.htm#pro>
- [40] Sidney H. Young, Kuzey Alabama Üniversitesi, Kimya Bölümü,
Theresa Julia Zielinski, Monmouth Üniversitesi, Kimya Bölümü, 20 Ocak 1996,
http://www.che.lsu.edu/links/computing/tutorials/intro_to_mathcad.pdf
- [41] <http://www.MathCAD.com/comparisonChart.html>
- [42] www.stetson.edu/~wgrubbs/pchem1/mathcadintro.doc
- [43] G.Vaughan, MathCAD' in Kullanımına Giriş, 17.9.2001,
www.aber.ac.uk/physics/docs/winmca.pdf
- [44] K. P. Brannan, J. A. Murden, From C++ to MathCAD,
ASEE National Conference Proceedings, 1998.
- [45] http://www.benkoltd.com/yazilim/MathCAD/MathCAD_universite.htm
- [46] www.soc.napier.ac.uk/module.php3?op=getlecture&cloaking=no&lectureid=420159
- [47] http://www.webopedia.com/TERM/u/user_interface.html
- [48] Terry Winograd, İnsan Bilgisayar Etkileşimi (HCI), Prentice-Hall, 1990
- [49] <http://www.english.mankato.msus.edu/stc/archives/2003/03Nov/Harris.htm>
- [50] <http://www.medicalcomputingtoday.com/0agui.html>
- [51] Bob Baxley, Kullanıcı Arayüz Modeli, ACM, 2003

- [52] Jesse James Garrett, Kullanıcı Arayüz Elemanları, 30 Mart 2000,
www.jjg.net/elements/pdf/elements.pdf
- [53] <http://www.edtech.vt.edu/edtech/id/interface/>
- [54] <http://cfg.cit.cornell.edu/cfg/design/elements.html>
- [55] Scott W. Ambler, Kullanıcı Arayüz Dizayn Teknikleri,
Cambridge Üniversitesi Yayınevi, 26 Ekim 2000,
<http://www.ambysoft.com/userInterfaceDesign.pdf>
- [56] James Hobart, Grafiksel Kullanıcı Arayüz (GUI) Dizayn Prensipleri,
Ekim 1995, http://www.classicsys.com/classic_site/articles/article_10-95.html
- [57] Tamas Daboczi, Grafiksel Kullanıcı Arayüzü Nasıl Test Edilir?,
IEEE Instrumentation & Measurement Magazine, Eylül 2003
- [58] <http://cfg.cit.cornell.edu/cfg/design/process.html>
- [59] George Kondylis, Grafiksel Kullanıcı Arayüzü (GUI), Mart 2004,
www.lancs.ac.uk/ug/kondylis/FYP%20REPORT.doc
- [60] Jeff Johnson, Grafiksel Kullanıcı Arayüz (GUI) Hataları, Morgan Kaufmann
Yayınevi, 2000
- [61] Roger S.Pressman, Yazılım Mühendisliği, McGraw-Hill, 5. Baskı, Sayfa 437-538