

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**MATLAB PORTFÖY ENİYİLEŞTİRME ARACI VE ÖRNEK
UYGULAMALARIN MATLAB PLATFORMUNDAN BAĞIMSIZ
(STANDALONE) HALDE HAZIRLANMASI**

ERHAN GÜNEY KARADAĞ

KOCAELİ 2012

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

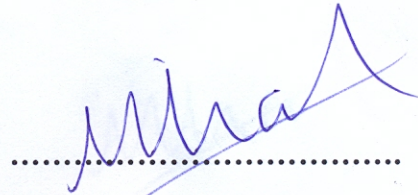
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

MATLAB PORTFÖY ENİYİLEŞTİRME ARACI VE ÖRNEK
UYGULAMALARIN MATLAB PLATFORMUNDAN BAĞIMSIZ
(STANDALONE) HALDE HAZIRLANMASI

ERHAN GÜNEY KARADAĞ

Doç.Dr. M.Melih İNAL
Danışman, Kocaeli Univ.



Doç.Dr. Emine D. BOLAT
Jüri Üyesi, Kocaeli Univ.



Doç.Dr. Hayriye KORKMAZ
Jüri Üyesi, Marmara Univ.



Tezin Savunulduğu Tarih: 05.10.2012

ÖNSÖZ ve TEŞEKKÜR

Matlab, Avrupa ve Amerika'da üniversitelerin dışında endüstrideki kullanım alanları bir yana ortaokul düzeyinde dahi öğrencilere algoritma mantığını vermek için öğretilen bir program olmaya başlamasına rağmen ülkemizde sadece bazı üniversitelerde tanınmakta ve sadece birkaç gelişmiş AR-GE birimi bulunan endüstriyel kuruluş tarafından kullanılmaktadır. Matlab gerek istatistik, matematik gerek ise finans ve mühendislik alanlarındaki öğrencilere öğrenim yaşamlarında ve sonrasında da bilgisayarı kullanabilme ortamı sunmaktadır. Maliyetli bir ürün olmakla birlikte, endüstriyel alanda, gerçek zamanda test edilmesi ve denetim prototiplerinin hazırlanmasındaki kullanım kolaylığı ve önemi inkâr edilemez bir gerçek olmuştur. Dünyada kullanıcı sayısı milyonlara ulaşan MATLAB programının ülkemizde de yaygın kullanılır hale gelmesi bilimsel ilerlememiz açısından önemli ve gerekli olduğu düşüncesindeyim. Bu tezin amacı da, Matlab'ın Grafikselleştirilmiş Kullanıcı Arayüzü (GUI) ve Derleyici (Compiler) kısımlarını inceleyerek bu gayeye katkıda bulunmaktır. Her geçen gün MATLAB'ın birçok derste de yardımcı araç olarak kullanılması kaçınılmazdır.

Sabır, sevgi ve desteklerini hiçbir zaman esirgemeyen annem, babam ve ablama, başta her sorumu bıkmadan usanmadan cevapsız bırakmayan Doç.Dr. Melih İNAL'a, teknik konularda bana yardımcı olan Arman VAROL ve Rüştü UTKAN'a, her anımda yanımda olup bana maddî manevî destek olan Munise SEVİL'e teşekkür ederim.

MAYIS-2012

Erhan Güney KARADAĞ

İÇİNDEKİLER

ÖNSÖZ ve TEŞEKKÜR.....	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	vii
SİMGELER DİZİNİ ve KISALTMALAR	viii
ÖZET	ix
ABSTRACT	x
GİRİŞ	1
1. MATLAB'A GİRİŞ	2
2. MATLAB'DA GUI TABANLI VE FARKLI UYGULAMALARIN TASARIMI	14
2.1. Giriş	14
2.2. Matlab'ta GUI Oluşturma Yöntemleri.....	15
2.2.1. Matlab gui development environment (guide).....	15
2.2.1.1. Gui bileşenleri ve özellikleri	21
2.2.1.2. Guide aracının incelenmesi	23
2.2.1.3. Gui arayüzünün programlanması	27
2.2.1.4. Callback kullanımı.....	29
2.2.1.5. metin biçimindeki callback fonksiyonları.....	33
2.2.1.6. Fonksiyon tutamacı (handle) kullanılarak belirlenen callback fonksiyonlar	35
2.2.1.7. Hücre içerisinde metinsel olarak belirlenen callback fonksiyonlar	35
2.2.2. M-File programlama yöntemi kullanılarak gui oluşturma.....	36
2.2.2.1. Programlama yoluyla nesnelerin eklenmesi.....	37
2.2.2.2. Figure nesnesi.....	37
2.2.2.3. Axes nesnesi.....	39
2.2.2.4. Uicontrol nesnesi	43
2.2.2.5. Uimenu nesnesi	45
2.2.2.6. Uicontextmenu nesnesi	47
2.3. Matlab GUI Uygulamalarında Diyalog Kutuları.....	49
2.3.1. MsgBox fonksiyonu	51
2.3.2. Helpdlg fonksiyonu	53
2.3.3. Errorldg ve warnldg fonksiyonları	54
2.3.4. Questdlg fonksiyonu.....	54
2.3.5. Listdlg fonksiyonu.....	55
2.3.6. Inputdlg fonksiyonu	57
2.3.7. Waitbar fonksiyonu	58
2.3.8. Uigetfile ve uiputfile fonksiyonları	59
2.3.9. Menü fonksiyonu.....	61
3. MATLAB DERLEYİCİSİ	63
3.1. Neden Derleme?	64
3.2. MATLAB Derleyicisinin Yetileri	65
3.3. MATLAB Compiler Kurulumu ve Mbuild.....	65

3.4. M-Fonksiyon Gerekliliği	68
3.4.1. M-dosya m-fonksiyon dönüşümü.....	69
3.5. Derleme Adımları.....	69
3.5.1. Bağımlılık analizi	70
3.5.2. Kod oluşturma.....	71
3.5.3. Arşiv yaratma	72
3.5.4. Derleme.....	72
3.5.5. Birleştirme	72
3.6. Matlab Compiler Kullanımı	73
3.6.1. Deployment tool ile derlemek.....	73
3.6.1.1. Paketlemek (isteğe bağlı)	76
3.6.2. Mcc komutu ile derlemek	77
3.6.2.1. Farklı mcc seçenekleri ile derleme	80
3.6.2.2. Dosya ve izin ekleme seçeneği.....	80
3.6.2.3. DLL oluşturma seçeneği	82
3.6.3. Matlab ve java	82
3.6.4. Alternatif bir dağıtma yöntemi : toolbox oluşturarak dağıtma ...	82
3.7. P-Code Derleme	87
3.8. MATLAB “MEX” Derleme Komutu	88
3.9. MATLAB Component Run-Time Ortamı (MCR)	93
4. UYGULAMALAR	94
4.1. Giriş	94
4.2. Eval Hesap Makinası Uygulaması.....	94
4.2.1. Giriş	95
4.2.2. GUI’de oluşturma.....	95
4.3. MATLAB Builder .NET Uygulaması	96
4.3.1. Giriş	96
4.3.2. Matlab’da kütüphane oluşturma.....	97
4.4. MATLAB Builder EX Uygulaması.....	101
4.4.1. Giriş	102
4.4.2. Excel eklentisi (add-in) oluşturma	102
4.5. Portföy Eniyileştirici Aracı (Portfolio Optimizer Tool) Uygulaması.....	105
4.5.1. Giriş	105
4.5.2. Data import (veri alma)	106
4.5.3. Portföy eniyileştirme (portfolio optimization).....	107
4.5.4. Sonuçlar (Results)	109
4.5.5. Uygulama kodları.....	110
SONUÇLAR VE ÖNERİLER	111
KAYNAKLAR.....	112
KİŞİSEL YAYIN VE ESERLER.....	114
ÖZGEÇMİŞ.....	115

ŞEKİLLER DİZİNİ

Şekil 1.1. Matlab'ın Veri Yapısı.....	4
Şekil 1.2. Matlab programının ana penceresi	6
Şekil 1.3. Değişken Editörü Penceresi	7
Şekil 1.4. Komut Geçmişi (Command History) Penceresi	8
Şekil 1.5. Komut Penceresi (Command Window).....	9
Şekil 1.6. Kullanımdaki Dizin (Current Folder) Penceresi	10
Şekil 1.7. Demolar (Demos) Penceresi	11
Şekil 1.8. Yardım (Help) Penceresi	11
Şekil 1.9. Arama Sonuçları.....	12
Şekil 1.10. Satır İçi Yardım.....	12
Şekil 1.11. Çalışma Alanı (Workspace) Penceresi	13
Şekil 2.1. GUIDE editörü.....	16
Şekil 2.2. GUI with Uicontrols.....	17
Şekil 2.3. GUI with Axes and Menu.....	18
Şekil 2.4. Modal Question Dialog	19
Şekil 2.5. GUIDE LAYOUT Editor (GUIDE Çalışma Alanı).....	20
Şekil 2.6. Bileşen paletinde isimlerin gösterildiği durum.....	20
Şekil 2.7. Bileşenlerin Figür üzerindeki görünümü.....	23
Şekil 2.8. GUIDE Araç Kutusu	23
Şekil 2.9. Align Object (Nesne Hizalama) Penceresi	24
Şekil 2.10. Menu Editor Penceresi	24
Şekil 2.11. Tab Order (Sekme sırası) Editor Penceresi	25
Şekil 2.12. Toolbar (Araç çubuğu) Editor Penceresi	25
Şekil 2.13. Property Inspector (Özellik Denetçisi) Penceresi	26
Şekil 2.14. Object Browser (Nesne Tarayıcısı) Penceresi	27
Şekil 2.15. M-File Editor Penceresi.....	28
Şekil 2.16. Geriçağırım (Callback) Fonksiyonlarına ulaşma	29
Şekil 2.17. Metin Biçimindeki Callback Fonksiyonları.....	33
Şekil 2.18. guney fonksiyonu ekran çıktısı	33
Şekil 2.19. Kod Parçası.....	34
Şekil 2.20. Çizdirelecek Fonksiyon	34
Şekil 2.21. guney fonksiyonu ekran çıktısı	34
Şekil 2.22. Fonksiyon Tutamacı Callback Fonksiyonlar	34
Şekil 2.23. Hücre İçerisinde Metinsel Callback Fonksiyonlar	36
Şekil 2.24. Figure Nesnesi.....	37
Şekil 2.25. Figure Nesnesinin Üzerinde Barındırdığı Nesnelere	38
Şekil 2.26. figure Nesnesinin Eklenmesi	39
Şekil 2.27. Eksen (Axes) Nesnesi.....	39
Şekil 2.28. axes Nesnesinin Eklenmesi.....	43
Şekil 2.29. uicontrol Nesnesinin eklenmesi	45
Şekil 2.30. Uimenu Nesnesi görünümü	46
Şekil 2.31. uimenu Nesnesinin eklenmesi.....	47
Şekil 2.32. Uicontextmenu Nesnesi görünümü	48

Şekil 2.33. uicontextmenu Nesnesinin eklenmesi	49
Şekil 2.34. msgbox Fonksiyonu	51
Şekil 2.35. msgbox görünümü.....	52
Şekil 2.36. msgbox(mesaj,başlık).....	52
Şekil 2.37. msgbox görünümü.....	52
Şekil 2.38. msgbox(mesaj, başlık, simge).....	52
Şekil 2.39. msgbox görünümü.....	53
Şekil 2.40. helpdlg(mesaj, başlık).....	53
Şekil 2.41. helpdlg görünümü	53
Şekil 2.42. errordlg(mesaj, başlık, OluşturmaŞekli).....	54
Şekil 2.43. errordlg görünümü.....	54
Şekil 2.44. questdlg Fonksiyonu.....	55
Şekil 2.45. questdlg görünümü	55
Şekil 2.46. listdlg Fonksiyonu	56
Şekil 2.47. listdlg görünümü	57
Şekil 2.48. inputdlg Fonksiyonu.....	57
Şekil 2.49. inputdlg görünümü	58
Şekil 2.50. Waitbar fonksiyonu	58
Şekil 2.51. waitbar görünümü	59
Şekil 2.52. Filtre	59
Şekil 2.53. uigetfile Fonksiyonu.....	60
Şekil 2.54. Uigetfile ve uiputfile fonksiyonları kullanımı	60
Şekil 2.55. uiputfile fonksiyonu	61
Şekil 2.56. Uigetfile ve uiputfile fonksiyonları kullanımı	61
Şekil 2.57. Menü Fonksiyonu.....	62
Şekil 2.58. Menü görünümü	62
Şekil 3.1. Matlab Derleyicisi Yetileri	65
Şekil 3.2. mbuild –setup komutu	67
Şekil 3.3. mbuild –setup komutu	68
Şekil 3.4. M-Dosya M-Fonksiyon Dönüşümü	69
Şekil 3.5. Bağımlılık Analizi	70
Şekil 3.6. Bağımlılık Analizi Çıktısı.....	71
Şekil 3.7. mcc kodu.....	71
Şekil 3.8. Derleme Adımları.....	73
Şekil 3.9. File->New->Deployment Project.....	74
Şekil 3.10. komut satırını kullanarak	74
Şekil 3.11. Başlat sekmesi ile	74
Şekil 3.12. Deployment Project Giriş Penceresi.....	75
Şekil 3.13. Deployment Tool penceresi	76
Şekil 3.14. Paketleme (Package)	77
Şekil 3.15. Paketleme (Package) MCR ekli	77
Şekil 3.16. Dosya ve Dizin Ekleme	81
Şekil 3.17. Dosya Ekleme kodu.....	81
Şekil 3.18. Dizin Ekleme kodu.....	81
Şekil 3.19. Farklı bir dosya adı ile derlenmek.....	81
Şekil 3.20. DLL Oluşturma	82
Şekil 3.21. info.xml kodları.....	83
Şekil 3.22. Start düğmesi Ayarları.....	84
Şekil 3.23. Başlat menüsünde oluşturulan kişisel araç kutusu	85

Şekil 3.24. HTML dosyası	85
Şekil 3.25. HTML sayfası test mesajı	86
Şekil 3.26. P-Code Derleme	87
Şekil 3.27. P-Code	87
Şekil 3.28. P-Code görünümü	88
Şekil 3.29. Mex Ayarları	88
Şekil 3.30. Mex Kodları	92
Şekil 3.31. Mex Derleme	92
Şekil 3.32. Mex Çalıştırma	92
Şekil 3.33. MCR [10]	93
Şekil 4.1. Eval Hesap Makinası	94
Şekil 4.2. GUI Arayüzü	95
Şekil 4.3. Eval Hesap Makinası Kodları	95
Şekil 4.4. MATLAB Builder .NET Uygulaması	96
Şekil 4.5. Matlabda Fonksiyon Hazırlama	97
Şekil 4.6. NET Assembly	97
Şekil 4.7. Sınıf Ekleme	98
Şekil 4.8. Referans Ekleme	99
Şekil 4.9. Ekleme Kısmı	99
Şekil 4.10. Kod Yazımı	100
Şekil 4.11. Program Çıktısı	101
Şekil 4.12. MATLAB Builder EX Uygulaması	101
Şekil 4.14. .bas Dosyası İçeriği	103
Şekil 4.15. Import File (Dosya Alma)	104
Şekil 4.16. İşlev Ekle	104
Şekil 4.17. Portföy eniyileştirici aracı (Portfolio Optimizer Tool)	105
Şekil 4.18. Portfolio Optimizer Tool (Portföy eniyileştirici aracı) Data Import Sekmesi	108
Şekil 4.19. Portfolio Optimization (Portföy eniyileştirme) sekmesi	108
Şekil 4.20. Results (Sonuçlar) sekmesi	109
Şekil 4.21. Oluşturulan Rapor (Excel ve PDF)	110

TABLolar DİZİNİ

Tablo 2.1. GUI Uygulamalarında Geriçağırım (Callback) Türleri	31
Tablo 2.2. GUI uygulamalarında kullanılabilir etkileşim kutuları	51
Tablo 2.3. listdlg fonksiyonu özellikleri	55
Tablo 3.1. Matlab Derleyicisi Yetileri	64
Tablo 3.2. Matlab Derleyicileri	66
Tablo 3.3. MCC seçenek kodları	79

SİMGELER DİZİNİ ve KISALTMALAR

AES	: Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
ASP	: Active Server Pages (Aktif Sunucu Sayfaları)
CTF	: Component Technology File (Bileşen Teknoloji Dosyası)
DLL	: Dynamic Link Library (Dinamik Bağlantı Kütüphanesi)
EX	: EXCEL
EXE	: Executable (Çalıştırılabilir)
GUI	: Graphical User Interface (Grafiksel Kullanıcı Arayüzü)
GUIDE	: GUI Development Environment (GUI Geliştirme Ortamı)
HTML	: Hyper Text Markup Language (Zengin Metin İşaret Dili)
JA	: JAVA
JNI	: Java Native Interface (Java Yerel Arayüzü)
MATLAB	: Matrix Laboratory (Matris Laboratuvarı)
MCR	: Matlab Component Run-Time (Matlab Bileşen Çalıştırma Ortamı)
MS	: Microsoft
NE	: NET
P-CODE	: PsuedoCode (Sahte Kod)
PDF	: Portable Document Format (Taşınabilir Belge Biçimi)
RSA	: Rivest, Shamir, Adleman (Açık Anahtarlı Şifreleme Yöntemi)
SDK	: Software Development Kit (Yazılım Geliştirme Aracı)
XML	: Extensible Markup Language (Genişletilebilir İşaretleme Dili)

MATLAB PORTFÖY ENİYİLEŞTİRME ARACI VE ÖRNEK UYGULAMALARIN MATLAB PLATFORMUNDAN BAĞIMSIZ (STANDALONE) HALDE HAZIRLANMASI

ÖZET

Bu tez çalışmasında, MATLAB'ın son kullanıcıyı kod karmaşasından kurtarmak için tasarlanan grafiksel kullanıcı arayüzü (GUI) tabanlı uygulamaların hazırlanma metodları ve bu uygulamaların MATLAB kurulu olmayan bilgisayarlarda da kullanılması için platformdan bağımsız (standalone) şekilde hazırlanması anlatılmıştır. Tezin ilk bölümünde Matlab hakkında temel bilgiler, kullanım alanları, diğer programlama dilleri ile kıyaslanması, dezavantajları, veri türleri ve Matlab ortamı ve pencereleri incelenmiştir. Tezin ikinci bölümünde Matlab'ta grafiksel kullanıcı arayüzü hazırlamak için kullanılan GUIDE (GUI Development Environment) metodu ve M-File programlama yöntemi kullanarak GUI oluşturma incelenmiştir. Grafiksel kullanıcı arayüzü (GUI) üzerinde bulunan bileşenlerin görevleri ve bir olay karşısında bazı eylemleri gerçekleştirerek GUI veya bileşen davranışlarının kontrol edilmesinde kullanılan callback fonksiyonları incelenmiştir. Ayrıca yine bu bölümde Matlab GUI uygulamalarında kullanılan diyalog kutuları ve faydaları incelenmiştir. Tezin üçüncü bölümünde ise Matlab'ta hazırlanan uygulamanın ortamdan bağımsız olarak çalıştırılması için yapılan derleme (compile) işlemi incelenmiştir. Matlab Derleyicisinin kurulumu, Derleyici kullanım metodları olan Dağıtım Aracı(Deployment Tool) ve Mcc komutu, Matlab Builder (NE,JA,EX), M-File – M-Function dönüşümü, Bağımlılık Kontrolü, Matlab Bileşen Çalıştırma (Component Run-Time) Ortamı, Toolbox oluşturarak dağıtma, P-Code ve MEX derleme açıklanmıştır. Uygulama olarak Finans Portföy Yönetim GUI programı seçilmiştir. Bu GUI'li Matlab programı MS Excel'den veya her hangi bir internet sayfasından veri olarak grafik çizme, sonuçları pdf ve MS Excel' e aktarabilme özelliklerine sahiptir. Bu uygulamadaki tüm bu bileşenler standalone hale dönüştürülmüş ve sonuçları açıklanmıştır. Sonuç olarak standalone uygulamaların artması ve Matlab ortamından bağımsız program çalıştırabilme özelliği ile Matlab programının yaygınlaşacağı görüşü kaçınılmazdır.

AnahtarKelimeler: Bağımsız Matlab Programlama, Matlab Builder, Matlab Derleyici, Matlab Grafiksel Kullanıcı Arayüzü.

STANDALONE PREPERATION OF THE MATLAB PORTFOLIO OPTIMIZATION TOOL AND SAMPLE APPLICATIONS

ABSTRACT

This study explains the preperation of Graphical User Interface, GUI, based application methods which are designed by Matlab to prevent the user from any confusion. It is also explained independently (standalone) from the platform so that it can be used in computers where Matlab is not installed. First of all, the study gives basic information about Matlab and its usage as well as its interaction with other programmes and its disadvantages. It also examines the settings and windows of Matlab. Afterwards the GUIDE (GUI Development Environment) method, used in Matlab for preperation of graphical user interface, and M-File programming techniques are used to examine the formation of GUI. The assignments of GUI components and the call back functions, that are used to control GUI or component behavior by carrying out the necessary actions it is are described. Afterwards, the compiling procedure of the application that is created with Matlab and run regardless (standalone) of the setting is examined. The installation of Matlab compiler, Deployment Tool and Mcc commands which are compiler usage methods, Matlab builder (NE, JA, EX) , M-File - M Function, dependency control, Matlab Component Run-Time setting, distribution with toolbox, P-Code and MEX compiling are explained. Finance Portolio Management GUI programme is chosen for the application. This Matlab programme with GUI can draw a graph from MS Excel or any web page at the same time. It can also transfer the graph into PDF and MS Excel. All the components in this application were transformed in standalone and its results were explained. Finally it is believed that the increase in standalone applications and widespread of matlab programmes, featuring its standalone running, is unavoidable.

Keywords: Standalone Matlab Programming, Matlab Builder, Matlab Compiler, Matlab Graphical User Interface.

GİRİŞ

"MATLAB" yüksek seviyeli bir teknik programlama dili olmasının yanında algoritma geliştirme, verilerin görselleştirilmesi, veri analizi ve sayısal hesaplamalar için etkileşimli bir yazılım paketidir. MATLAB ile teknik hesaplama problemlerini, C,C++ ve Fortran gibi geleneksel programlama dillerinden daha hızlı bir şekilde çözebilirsiniz. Grafik arayüzler tasarlamak için araçlar MATLAB'ın kullanımı olmayan bir mühendislik alanı yok gibidir. Bu fonksiyonlar MATLAB dilinin temelini oluşturmaktadır. MATLAB ile double, single ve integer(tam sayı) gibi genel veri tipleri üzerinde işlemler gerçekleştirebilirsiniz. Toolbox adı verilen modüller sayesinde işaret işleme, optimizasyon, istatistik ve sembolik matematik gibi birçok özelleştirilmiş fonksiyonları kullanabilirsiniz. Bu modüller MATLAB'a ayrıyeten dahil edilebilmektedirler. Bu modüllere her geçen gün bir yenisı eklenmektedir. Böylece MATLAB'ın kullanım alanları da her geçen gün genişlemektedir. MATLAB birçok klasik algoritmayı tek bir komutta sunmaktadır. Böylece matematiksel hesaplamaların bilgisayarda yapılması diğer programlama dillerinde (C,C++,Fortran gibi) olduğundan daha kısa sürede ve daha az kod yazılarak gerçekleştirilebilmektedir. Böylece problemin bilgisayara uyarlanmasından çok problemin kendisine yoğunlaşmak daha da kolaylaşmaktadır. Eğer matematik ve bilgisayar programcılığı bilginiz iyiye MATLAB'ı öğrenmeniz hiç de zor olmayacaktır. Çünkü MATLAB'ta bildiğiniz matematik formüllerini (lineer cebir, integral, türev, fourier analizi) ve programlama yapılarını (vektörler, matrisler, for, while yapıları) göreceksiniz. MATLAB'ı çok geniş özelliklere sahip bir hesap makinesi olarak da düşünebiliriz. Bir hesap makinesi bize bir şey öğretmez. Sadece işlemlerimizi daha hızlı yapmamızı sağlar. Onu kullanmayı ne kadar iyi bilirsek o kadar yüksek verim elde edebiliriz

Bu tezin amacında MATLAB'da hazırladığımız görsel uygulamaları MATLAB'ın ürünü olan COMPILER(Derleyici) sayesinde derleyip, platformdan bağımsız hale getirerek dağıtımın kolaylaştırılması ve böylece oluşturulan bağımsız programların MATLAB olmayan bilgisayarlarda da kullanabilinmesini sağlamaktır.

1. MATLAB'A GİRİŞ

Matlab[®], temel olarak sayısal hesaplama, grafiksel veri gösterimi ve programlamayı içeren teknik ve bilimsel hesaplamalar için yazılmış birçok disipline ait kullanıma hazır kodlara sahip bir yazılımdır.

Avrupa ve Amerika'da üniversitelerin dışındaki endüstrideki kullanım alanları bir yana ortaokul düzeyinde bile öğrencilere algoritma mantığını kavratmak için öğretilen bir program olmasına rağmen ülkemizde sadece üniversitelerde tanınmakta ve sadece gelişmiş AR-GE birimi bulunan endüstriyel kuruluşlar tarafından kullanılmaktadır [1].

Sayısal problemleri hızlı, zorlanmadan ve eğlenceli bir şekilde çözenin en önemli yollarından biri Matlab'ı kullanmaktır. Matlab'ta bilimsel ve matematiksel hesaplamalar, standart programlama dillerinden (FORTRAN, C, JAVA vb.) daha esnek kodlamalar yaparak çözmek mümkündür. Matlab, gerçek hayattaki problemlerin incelenmesinde inanılmaz kolaylıklar sağladığından kullanıcı dostu bir programdır. Sayısal hesaplamalar dışında grafik çizimlerinde sağladığı kolaylıklar, verdiği sonuçlar ve çizilen cisimler ile ilgili ilginç özellikler sunma başarısı ile faydalı bir programdır. Daha önce bilimsel kitaplarda çözülmüş olan matematik alanındaki soruları Matlab programında tekrardan çözüp sonuçlarını karşılaştırmak mümkündür. Büyük ölçekli projelerde ekip çalışması gerektiren uygulamalar için Matlab programı modüler programlama yapmaya elverişlidir.

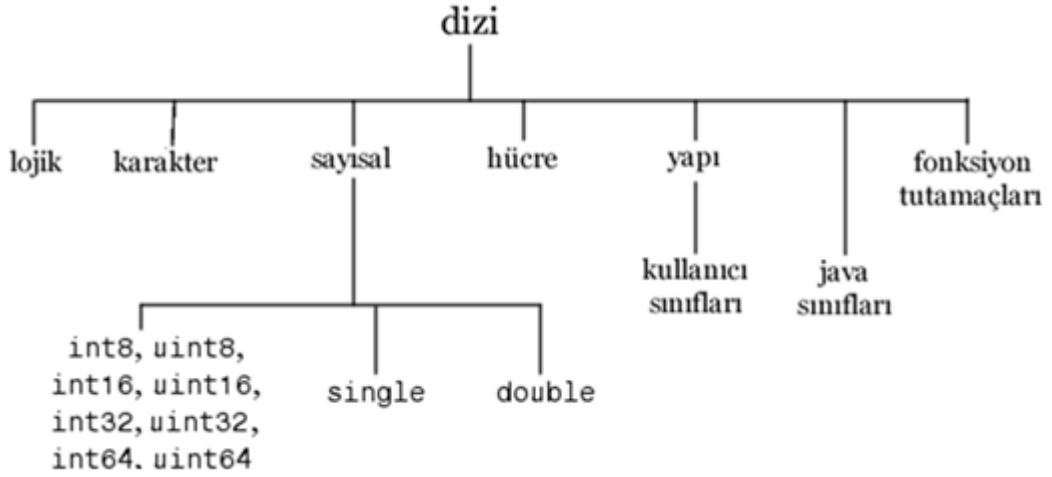
Matlab, ilk uygulamalarında bilim adamlarına problemlerin çözümüne matris temelli teknikleri kullanarak yardımcı olmaktadır. Daha sonraları ise geliştirilen yerleşik kütüphanesi, uygulama ve programlama özellikleri ile gerek üniversite ortamlarında (başta matematik ve mühendislik olmak üzere tüm bilim dallarında) gerekse sanayi çevresinde yüksek verimli hesaplama ve analiz aracı olarak yaygın bir kullanım alanı bulmuştur. Ayrıca işaret işleme, kontrol, bulanık mantık, yapay sinir ağları, genetik algoritmalar gibi bir çok alanda ortaya koyduğu Araçkutusu (Toolbox) adı verilen yardımcı alt programlarla özelleştirilmiş, kolaylaştırılmış ve bir çok esnek programlama yapma imkanları sağlamaya da devam etmektedir. Matlab,

matematiksel hesaplamalar ve programlama başta olmak üzere çeşitli alanlarda kullanılmaktadır. Matlab'ın özünü ve içeriğini oluşturan bu alanlar aşağıda belirtilmiştir [2]:

- Matematik ve hesaplamalar
- Algoritma geliştirme
- Modelleme ve Benzetim (Simülasyon)
- Veri analizi
- Çeşitli özelliklere sahip Grafikler
- Görüntü işleme
- İşaret işleme
- Grafikselle kullanıcı arayüzü (GUI) geliştirme
- Mühendislik ve fen alanında uygulamalar geliştirme
- İşaret Filtreleme

SIMULINK eklentisiyle Matlab görselden matematiğe geçiş yaklaşımını benimsemiştir. SIMULINK'te mühendisler mantıksal süreçlerini sürükle bırak yöntemi ile oluşturabilmektedir. Oluşturulan sistem matematiksel olarak denenebilmektedir. Bu yöntem ile bir otobüsün süspansiyonunu, bir rüzgar türbinini veya dijital ses işlemcilerini tasarlamak mümkündür [3].

Matlab'ı diğer programlama dilleri ile teknik açıdan karşılaştırmak için çok boyutlu diziye dayalı olduğundan söz etmek gerekli olacaktır. Matlab'ta temel olarak double, int8, logical, char vb. 15 farklı veri tipi vardır. Bütün bu veri tipleri dizi biçimindedir. Diziler çok boyutlu olabilir. İki boyutlu olanı ise genel olarak matris olarak isimlendirilir. Bütün veri tiplerinin diziye dayalı olması matematiksel işlemlerin gerçekleştirilmesinde kolaylık sağlar. Matlab'ın ilk ortaya çıkış sebebi matematiksel işlemlerde büyük kolaylık sağlayacak bir değişken (dizi) türü oluşturmak olarak gösterilir. Ancak, bu değişken o kadar kolaylıklar sağlamıştır ki, Matlab şimdi karşımızda başlı başına bir programlama dili olarak duruyor. Matlab'ın veri yapısı Şekil 1.1.'deki gibi modellenebilir [1].



Şekil 1.1. Matlab'ın Veri Yapısı

Diğer bir farklılık da Matlab'ın geleneksel programlama dillerindeki aksine yazılan kodu derleyip (compile), çalıştırılabilir (exe) bir dosya oluşturmak yerine, yorumlayarak (interpret) çalıştırmasıdır. Bu sayede kodun büyüklüğü ile orantılı olarak artan derleme (compile) ve bağlama (link) süreci kısalmır. Ayrıca, bu sürecin kısılması, kodun hatalardan arındırılması (debug) için gereken süreyi kısaltmış ve kolaylaştırmış olur. Bu, doğal olarak Matlab ile yazılan kodun geleneksel programlama dilleri ile yazılan kodlardan daha yavaş çalışmasına sebep olur. Ancak bunun da çözümü basittir; yazılan kod hatasız ve istendiği gibi çalışıyorsa bilgisayarda yüklü olan Borland, Microsoft Watcom veya Matlabın tümleşik C derleyicisi olan Lcc aracılığı ile derlenerek MEX formatına dönüştürülüp çalışma hızı yüksek oranda artırılabilir. MEX, Matlab çalıştırılabilir (Matlab executable) dosyası anlamına gelmektedir. Ayrıca Windows işletim sisteminde bu dosyalar dll (dynamic link library) uzantılı dosyalar ile bağları kurulmuştur.

Öte yandan, Matlab'ın yorumlayıcı mantığı ile çalışması, komut satırı mantığı ile işlemlerin satır satır çalıştırılabilmesi imkanını sağlar. Derleyici aracılığı ile çalıştırılabilir bir dosya üreten geleneksel programlama dillerinde ise bu mümkün değildir. Bu sayede pratik işlemler komut satırından hızlıca gerçekleştirilebilir. Ne yazık ki avantaj sayılabilecek bu özellik Matlab'ın bir programlama dili olarak değil de sadece matematiksel işlemleri gerçekleştirebilen sıradan bir program olarak düşünülmesinin tek sebebidir [1].

Matris işlemlerinde ayrılan fazla hafıza performansı anlamında bir dezavantaj olarak düşünülebilir. Fakat matrisel işlemlerin kolaylıkla yapılması bu dezavantajı gördürebilmektedir. Matlab'ın programı; kodu önce yorumlayıp, daha sonra derlemektedir. Bu durum da performans sorunlarına yol açmaktadır. Matlab programında döngü kullanımı koşulan programı yavaşlatmaktadır. Döngü yerine "Mathworks" Matlab programının resmi firması vektörel işlemleri önermektedir. Vektör işlemleri ile aynı döngü işlemi yirmi kat daha hızlı yapılabilir. Fakat her işlemi kolaylıkla vektör işlemlerine çevirmek döngü kullanımı kadar kolay olmamaktadır.

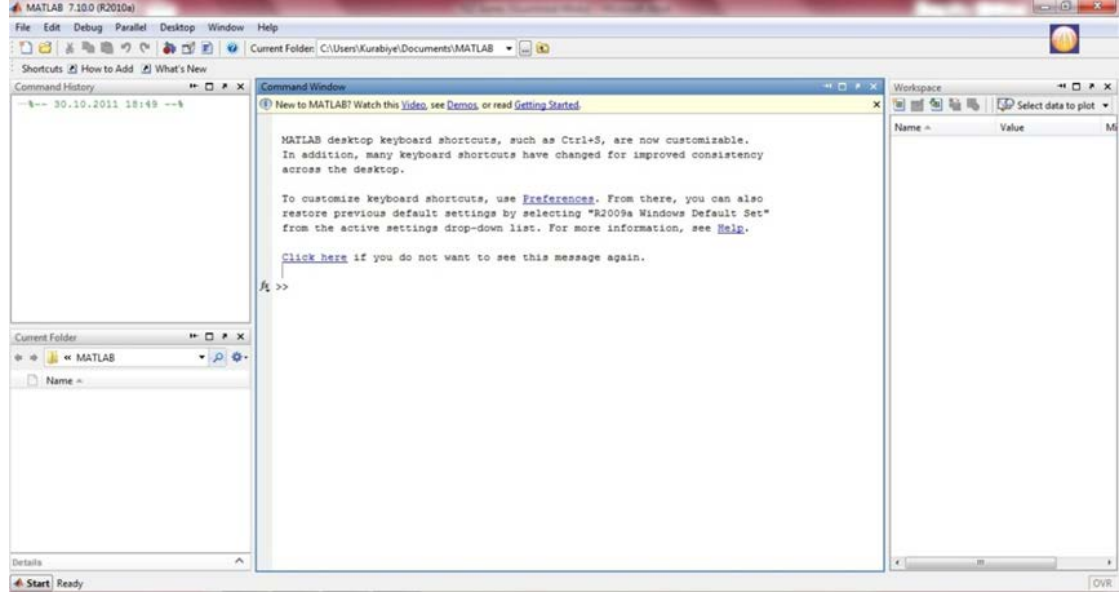
Matlab, C ve FORTRAN programlarına göre lisans yönünden maliyeti daha yüksek görülmektedir. Bu da iş çevreleri için maliyeti arttıran bir unsurdur. Bu durum piyasa dağılımını yavaşlatmasına karşın öğrenciler için ücretsiz sürümleri bulunmaktadır [3]. Matlab'ta her şey bir dizi (array) olarak işleme konur. Bir dizi; skaler, vektör, matris veya metinsel dizi (string; karakter dizisi) olabilir.

Matlab'ta temel dosya türleri aşağıdaki gibi özetlenebilir [5] :

- *.m Matlab program dosyaları
- *.fig Grafik dosyaları ve GUI'lerin grafik parçaları
- *.mat Değişken ve matris dosyaları
- *.p pre-parsed pseudo-code dosyaları (bu dosyaların içeriği görüntülenemez ancak program olarak çağırılabilir, yani Matlab'ta çalıştırılabilir!)

Matlab programının Türkiye dağıtıcısı Figes firmasıdır (www.figes.com). Programın lisanslı bir sürümü, istenen yapılandırma belirtilerek satın alınabilir. Matlab programının deneme sürümünü Mathworks firmasının Web sitesinden (www.mathworks.com) indirmek mümkündür.

Windows işletim sisteminden Matlab programını çalıştırmak için başlat menüsünde veya masaüstünde yer alan Matlab simgesine tıklamak yeterlidir. Matlab programı yüklenince Şekil 1.2.'deki gibi bir pencere açılacaktır.



Şekil 1.2. Matlab programının ana penceresi

Masaüstü olan adlandırılan Matlab ana penceresi birçok pencereden meydana gelen tümleşik bir ortamdır. Windows işletim sistemindeki gibi sol altta “Start” düğmesi yer alır. Hemen hemen bütün özelliklerine bu düğme sayesinde erişilebilir. Masaüstündeki pencereler ve diğer birçok özellik ihtiyaçlar doğrultusunda istenildiği gibi düzenlenerek kişiselleştirilebilir. Matlab’ın rahat kullanılmasını sağlayan pencereler ve görevleri aşağıdaki gibi özetlenebilir:

- Değişken Editörü (Variable Editor)
- Komut Geçmişi (Command History)
- Komut Penceresi (Command Window)
- Kullanımdaki Dizin (Current Folder)
- Demo’lar (Demos)
- Yardım (Help)
- Çalışma Alanı (Workspace)

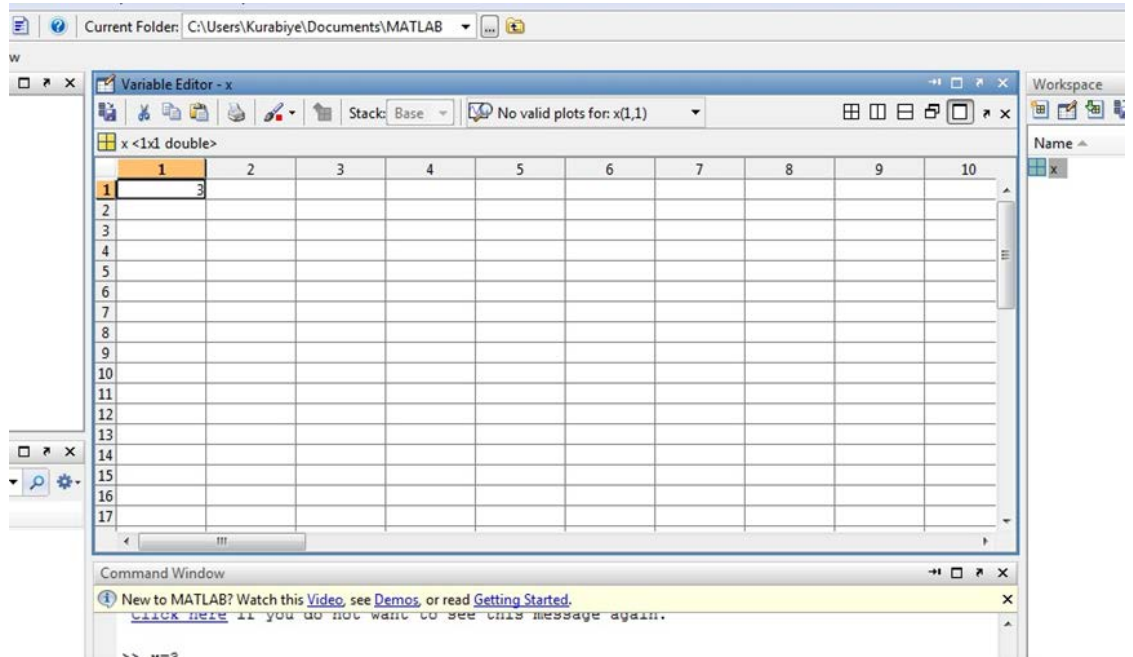
Aşağıda bu pencerelerin görevleri kısaca anlatılmıştır:

Değişken Editörü Penceresi; kullanıcı Çalışma Alanı penceresindeki herhangi bir değişkenin üzerinde çift tıkladığında Şekil 1.3.’te görülen pencere ile karşılaşır. Bu pencere yardımıyla seçilen herhangi bir değişkenin içeriği görülebileceği gibi yine aynı değişkenin içeriği bu pencere yardımıyla değiştirilebilir. Ancak, bazı farklı

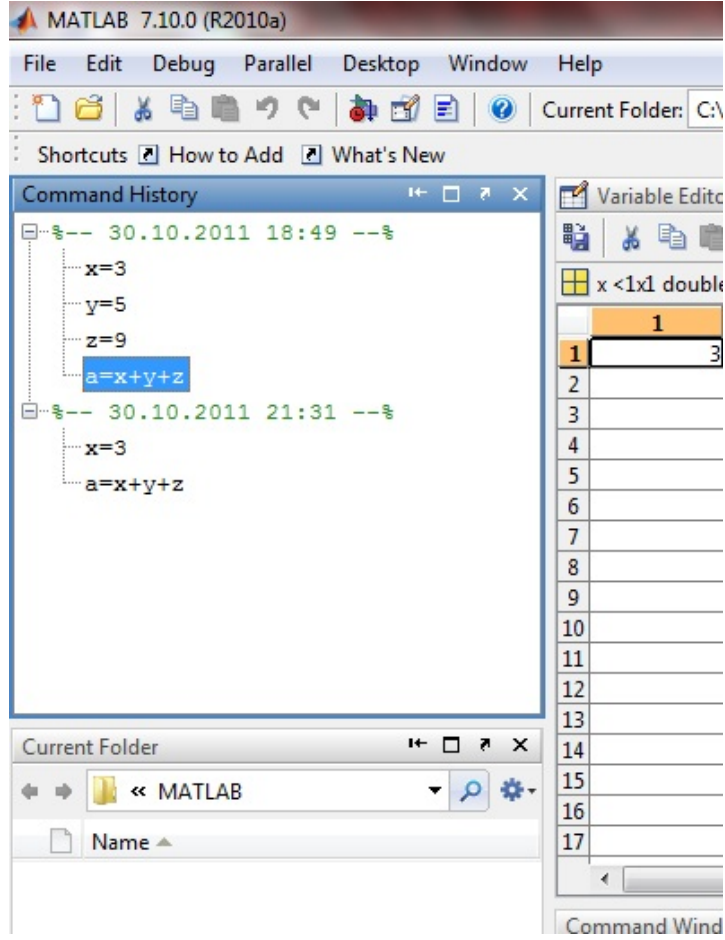
tipteki deęişkenlerin içerięini deęiştirmek mümkün deęildir. Bu durumda gerekli matlab komutu kullanılarak deęişiklik gerçekleştirilebilir.

Komut Geçmiři (Command History) Penceresi: Matlab masaüstünün sol kenarında bir pencere olarak yer almaktadır. Bu pencere sayesinde daha önce kullanılmış olan komutların bir listesine ulaşılabilir. İlgili komut, çift tıklanarak tekrar çalıştırılabilir. Ayrıca, bir grup komut pencere içerisinde seçilerek sağ tuş yapıldığında açılan kısayol menüsündeki M-Dosya Oluştur komutu ile bir M-dosya içerisine alınarak daha sonra kullanılmak üzere kaydedilebilir.

Matlab'ın çok kullanışlı bir dięer özellięi ise daha önce kullanılan komutlar arasında klavyedeki yukarı-aşaęı ok vasıtasıyla gezinme imkanı sağlamasıdır. Bu sayede bir süre önce kullanılmış olan komutlar tekrar yazmaya gerek kalmadan çalıştırılabilir.

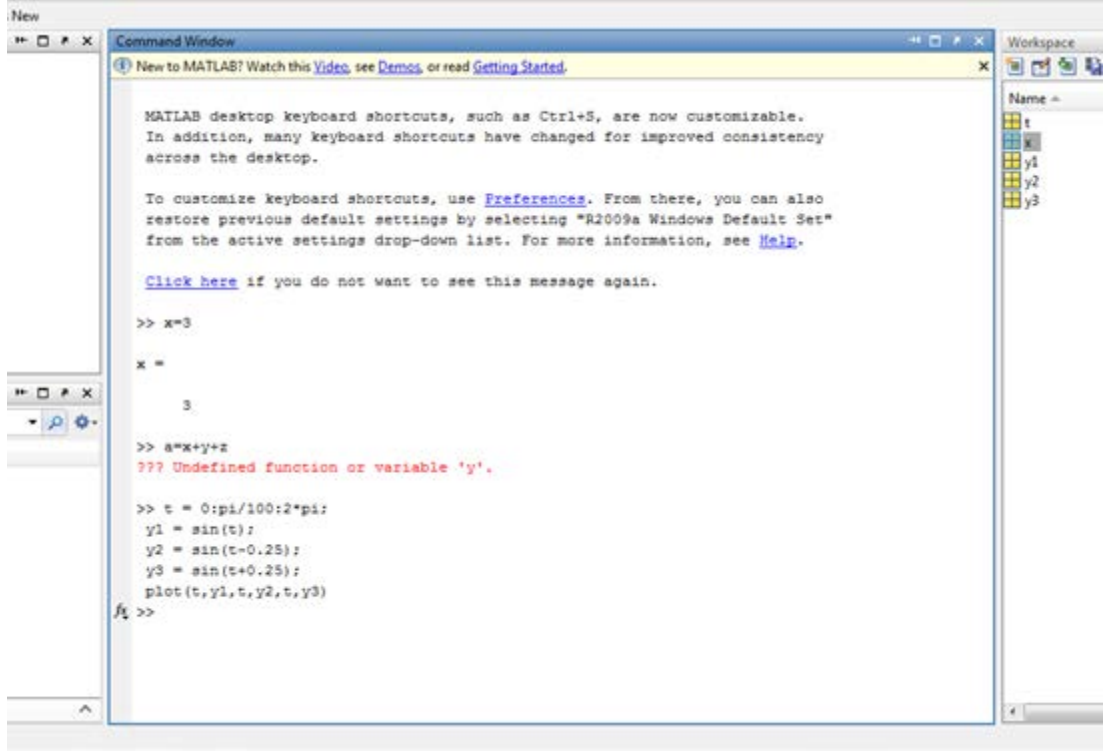


Şekil 1.3. Deęişken Editörü Penceresi



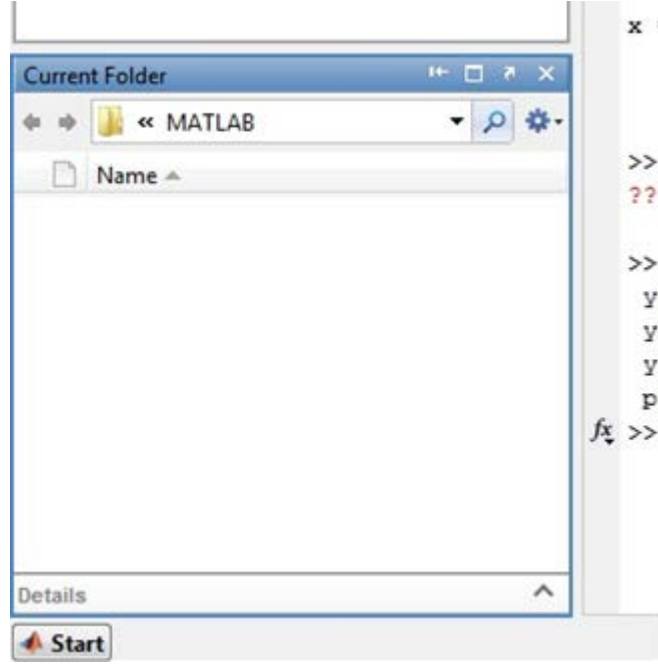
Şekil 1.4. Komut Geçmişi (Command History) Penceresi

Command Window Penceresi; Komut penceresi yerleşik Matlab fonksiyonları ile programcının oluşturduğu makro ve fonksiyonları satır satır çalıştırabileceği bir ortamdır. Daha öncede sözü edildiği gibi Matlab kodu derleyerek değil yorumlayarak çalıştırır. Bu durum işlemlerin satır satır çalıştırılabileceği bir komut penceresinin kullanılabilmesi imkanını sağlar. Komut penceresinde aktif olarak kullanılacak olan satır, yani komut satırı çift büyüktür (>>) işareti ile temsil edilir. Komut satırına yazılan ifade “Enter” tuşuna basıldığı anda otomatik olarak yorumlanır ve hatasız ise çalıştırılır. Bu işlem bir örnek Şekil 1.5.’de görülmektedir.



Şekil 1.5. Komut Penceresi (Command Window)

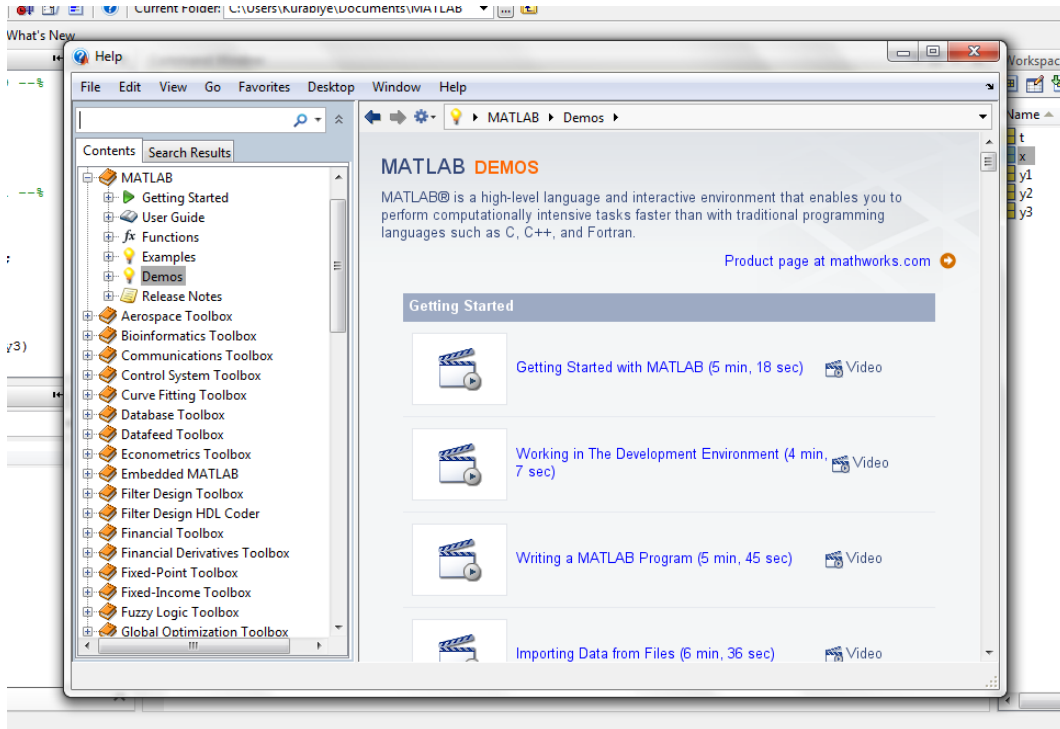
Kullanımdaki Dizin (Current Folder) Penceresi; Matlab'ın herhangi bir anda aktif olarak kullandığı geçerli dizin yolunu değiştirmek, içinde bulunulan klasör içerisinde taşıma, kopyalama ve dosya silme gibi işlemleri gerçekleştirmek veya dosyalar hakkında bilgi edinmek için bu pencere kullanılır. Matlab daima geçerli bir yol üzerinden çalışır. Şekil 1.6.'da görüldüğü gibi aktif dizin olarak MATLAB görülmektedir. Varsayılan olarak bu yol Matlab'ın kurulu olduğu dizin içinde yer alan "Work" klasörüdür. Geçerli klasör içerisinde yer alan kullanıcının tanımladığı fonksiyon dosyaları da bu alandan çağrılır. Eğer kullanılacak fonksiyonlar farklı ise bu pencere yardımıyla geçerli yol tanımı değiştirilmelidir. Ayrıca, Matlab'ın geçerli klasör yolu ana penceredeki araç çubuğunda yer alan Kullanımdaki Dizin (Current Folder) bölümünden de görülebilir veya bu yol tanımı değiştirilebilir.



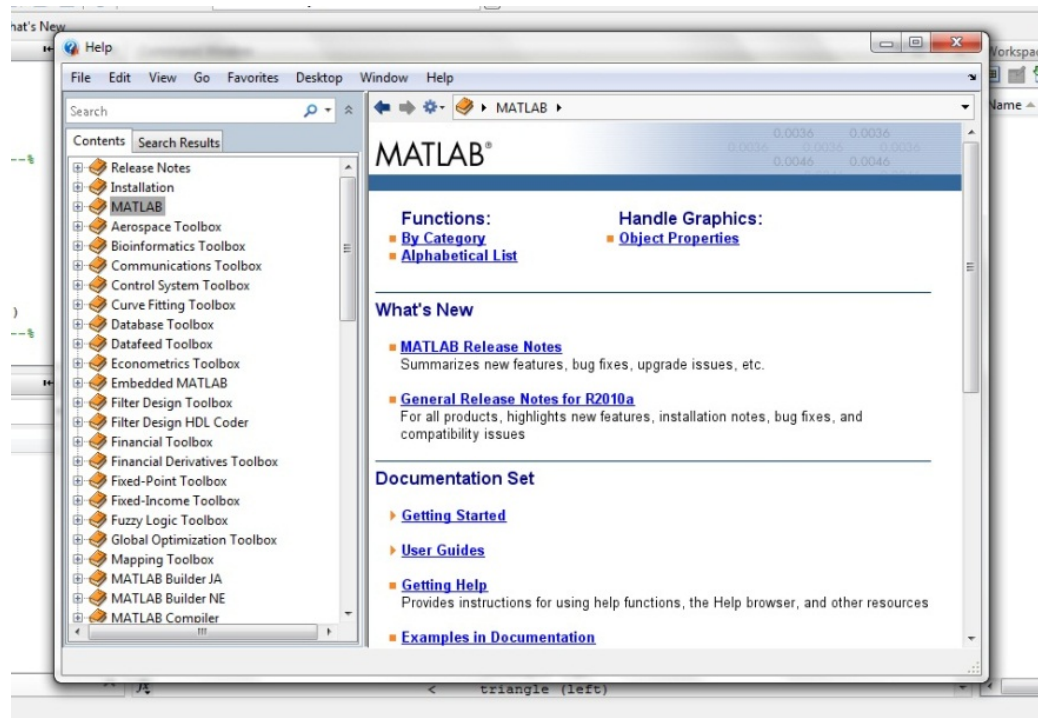
Şekil 1.6. Kullanımdaki Dizin (Current Folder) Penceresi

Demolar (Demos) Penceresi; “Help” menüsünden Demos veya Matlab ana penceresi sol alt köşede yer alan “Start” butonu kullanılarak Demos komutunun verilmesi ile Şekil 1.7.’de görülen demos penceresi açılır. Kullanıcı bu pencere yardımıyla Matlab’ın kendi içinde yer alan hazır uygulamalarını görebilir, kodlara bakabilir, konu ile ilgili bilgiler edinebilir veya internet bağlantısı ile eğitim videoları izleyebilir.

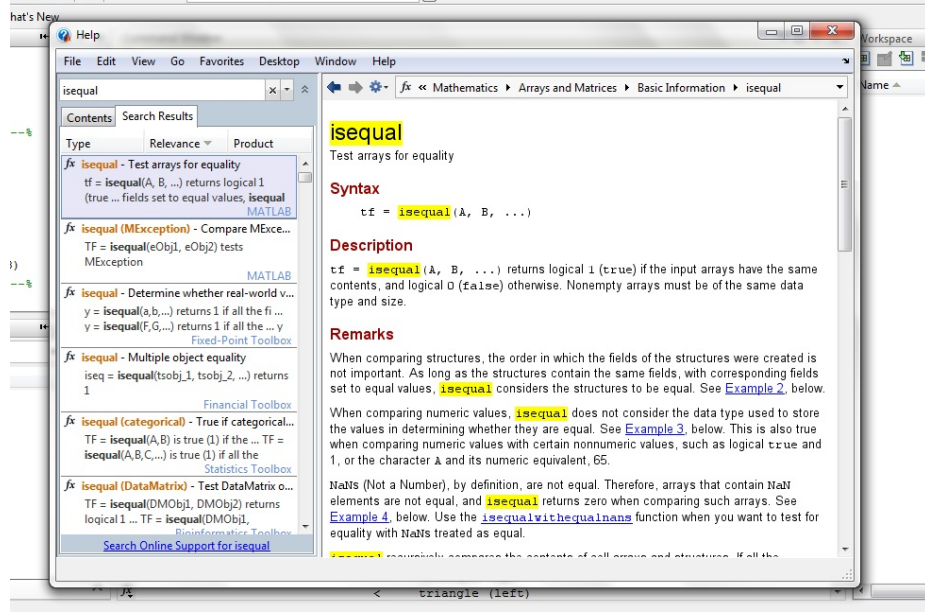
Yardım (Help) Penceresi; Matlab ve araçları hakkında yardım alınabilecek gelişmiş etkileşimli yardım ekranıdır. Yardım dosyaları HTML formatında olduğu için gezinme linkler aracılığı ile çok kolaydır. Şekil 1.8.’deki sol taraftaki pencerede iki adet sekme mevcuttur. Bunlardan ilki bilgisayara yülü olan Matlab bileşenlerinin bir listesini tutan İçerik (Contents) sekmesidir. Listede yardım almak istenen bileşenin üzerine tıklanarak sağ taraftaki ekranda bilgi görüntülenebilir. İkinci sekmede ise arama yapıldıktan sonra sonuçları liste halinde gösteren Arama Sonuçları (Search Results) sekmesi mevcuttur. Şekil 1.9.’da örnek bir arama sonucu gösterilmiştir.



Şekil 1.7. Demolar (Demos) Penceresi

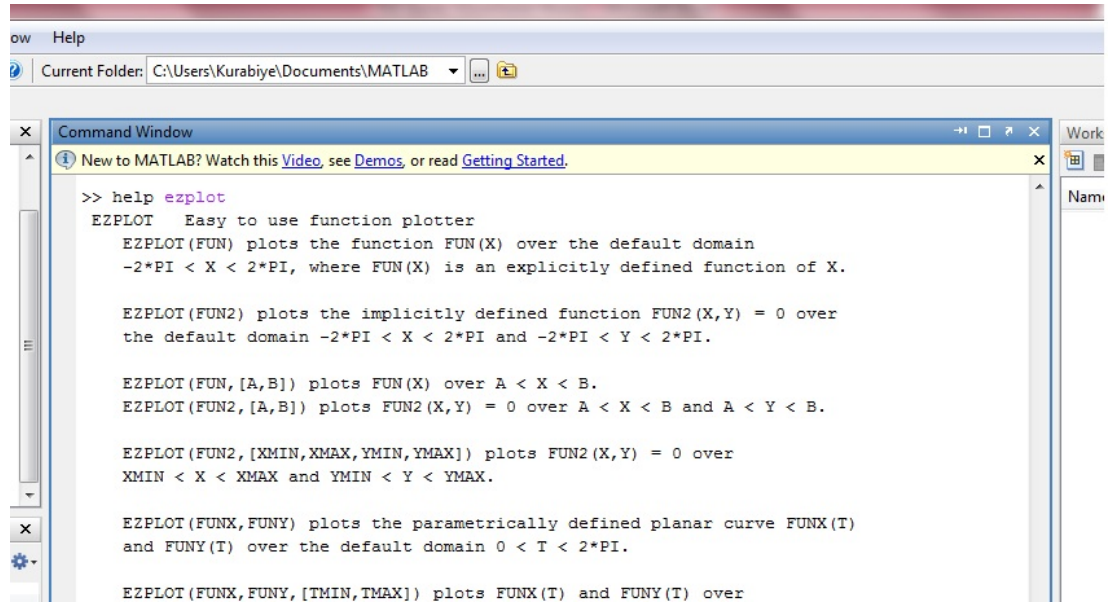


Şekil 1.8. Yardım (Help) Penceresi



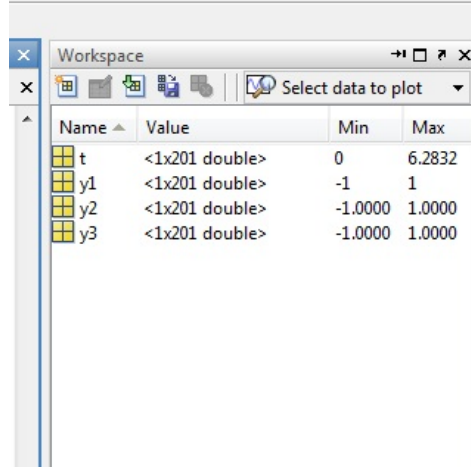
Şekil 1.9. Arama Sonuçları

Ayrıca yardımın komut satırından alınması daha pratik ve hızlı olabilmektedir. Örneğin komut satırına Şekil 1.10'da görüldüğü gibi “help ezplot” yazarak bu konu üzerindeki yardımlara ulaşılabilir.



Şekil 1.10. Satır İçi Yardım

Çalışma Alanı (Workspace) Penceresi: Matlab çalışma oturumunda, o ana kadar oluşturulan veya bir dosyadan yüklenen değişkenler üzerinde düzenleme, silme ve çizdirme gibi işlemleri daha kolay ve hızlı bir şekilde gerçekleştirmek için tasarlanmıştır. Ayrıca Şekil 1.11’de gösterilen değişkenlere çift tıklanarak Variable Editör penceresine ulaşılabilir [1].



Şekil 1.11. Çalışma Alanı (Workspace) Penceresi

2. MATLAB'DA GUI TABANLI VE FARKLI UYGULAMALARIN TASARIMI

2.1. Giriş

İçeriğinde yer alan nesnelerin kullanılması ile kullanıcıya etkileşim sağlayan ve bir programın koşturulmasını sağlayan grafiksel kullanıcı arayüzüdür (Graphical User Interface – GUI).

GUI nesnelere menüler, araç çubukları, radio butonları, liste kutuları veya kaydırıcılar olabilir. Bunların yanında Matlab GUI ile Matlab'ın sunduğu hesaplama olanakları kullanılarak da veri edinimi ve grafik çizimi gibi pek çok işlem gerçekleştirilebilir[6].

Genel olarak kullanıcı arayüzü, herhangi bir programın görsel ve etkileşimli kullanımını 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. Kullanıcı da, fare ve klavye ile arayüzdeki bu seçenekleri seçer ve çalıştırır. Böylece kullanıcı ile görsel program arasındaki etkileşim arayüz ile sağlanmış olur.

Kullanıcı arayüzü; kullanıcıya program üzerinde “pushbutton, textbox, menu, listbox” gibi kontrol birimlerinin bulunduğu bir ortam sağlar. Örneğin bir pushbuttona basıldığında daha önceden yazılmış bir fonksiyonun çalışması sağlanabilir; ya da veri tabanında yer alan verilerin arayüz açıldığında listbox'ta görüntülenmesi sağlanabilir.

Kullanıcılar Matlab GUI'yi temelde beş nedenden dolayı tercih etmektedir:

1. Yüksek seviyeli skript temelli geliştirme yapılabilmesi
2. Matlab'ın matematiksel hesaplama gücü ile uygulamaların uyumunun sağlanması
3. İşletim sistemi bağımsız uygulamalar geliştirilebilmesi
4. Son kullanıcı ile etkileşimli uygulamalar geliştirilebilmesi
5. Gerekliğinde gerçek zamanlı ve görsel sonuçlar gösterilebilmesi

Matlab GUI'lerinde nesne tabanlı tanımlamalar kullanılmaktadır. Matlab "event handling" olay başarımı yaklaşımına sahiptir. Program çalışırken kullanıcı tarafından programa dair gerçekleştirilen tüm hareketlere olay(event) denir. Örneğin kullanıcının programda bir buttona tıklaması, fareyi hareket ettirmesi, programı kapatması vb. şekilde kullanıcının programla etkileşimi sonucunda olay(event) meydana gelir. Event Handling bu olaylarla gerçekleştirilen mekanizmalardır. Uygulama geliştirici GUI üzerinde bir nesne yarattığında bu işlem ilgili ".m" dosyasında "handle graphic object" - grafik nesnesi tutamacı şeklinde otomatik olarak yaratılır. Son kullanıcının GUI üzerinde yaptığı işlemler "handle graphic object" tarafından yakalanır. Bu işlemler için ilgili "handle graphic object" 'e bağlı "callback" - geriçağırım fonksiyonundaki tanımlamalar gerçekleştirilir. Son kullanıcı GUI uygulamalarında anlık olarak görselin veya verinin üzerinde çalışabilir.

GUI fonksiyonları yerel değişkenler ile çalışmak durumundadır. Bunun gerekçesi çoklu GUI yapılarından kaynaklanmaktadır. Uygulama geliştirici çoklu GUI geliştirirken sistematüğını kaybetmemesi için değişkenler fonksiyonlara özel tanımlanmaktadır. Dolayısıyla diğer fonksiyonlar tarafından kullanılan bir değişken olduğunda her bir fonksiyonun başında genel (global) değişken olarak değişkenin tanımlanma zorunluluğunu ortaya çıkmaktadır [3].

2.2.Matlab'ta GUI Oluşturma Yöntemleri

Matlab'ta GUI oluşturmada iki yöntemden söz etmek mümkündür:

- Matlab GUI Development Environment (GUIDE) aracı kullanılarak,
- M-File programlama yöntemi kullanılarak.

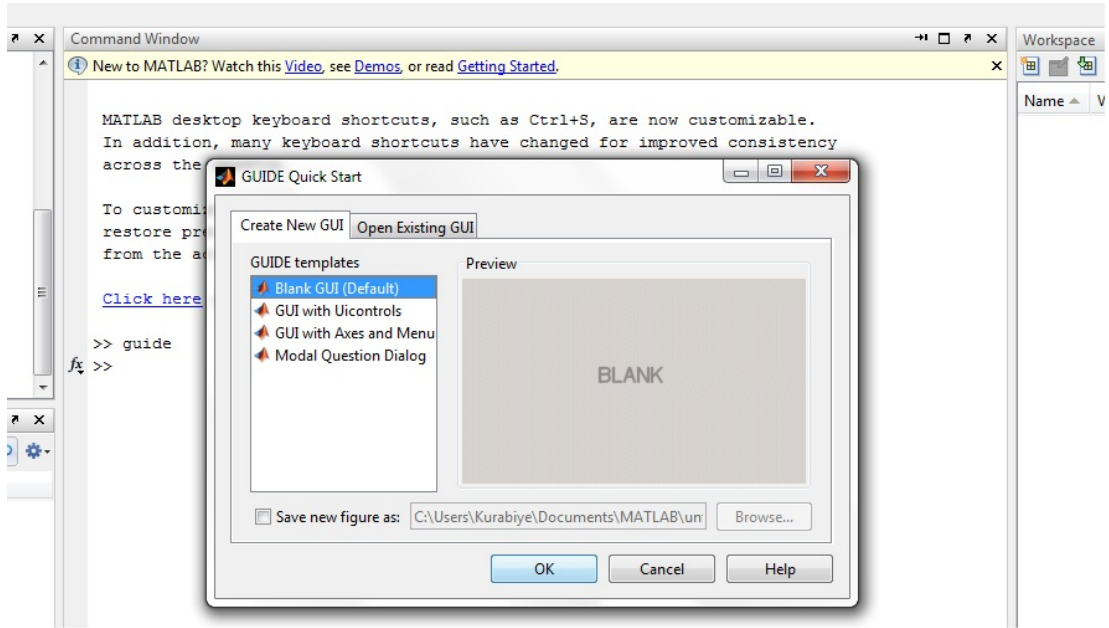
2.2.1. Matlab guidevelopment environment (guide)

Özellikle GUI tasarımında hızlı arayüzler tasarlamak için ve bu işe ilk başlayan programcılar için Matlab GUIDE aracının kullanılması büyük bir kolaylık sağlar. Bu aracın kullanılması ile GUI arabirimi kolaylıkla ve yorulmadan sürükle bırak ve açılan pencerelerde özelliklerin değiştirilmesine dayanan bir yöntem kullanılır. Ayrıca, bu yöntemi kullanmanın ileride var olan bir GUI'nin düzenlenmesi ve değişiklik yapılması bakımından da çok yararlıdır.

M-File programlama yönteminde tüm GUI tasarımları ve geriçağırım (callback) program parçalarının yazılması tamamı ile programlama kodları kullanılarak yapılır. Burada tasarımcı her şeye hakimdir ve bu teknik uzman bir programlama bilgisi gerektirir. Bu yöntem ile tasarım zamanı uzamasına rağmen programcı her türlü manipülasyonu yapabildiği için programcı açısından çok yararlıdır [6].

Kullanıcı “GUIDE” alt programı vasıtasıyla sürükle bırak yöntemini kullanarak “radio button”, “text box”, “slider”, “run button”, “panel”, “axes” ve “static text” gibi nesnelere arayüz tasarlayabilir. GUIDE ilgili nesnenin özelliklerini göstererek yazı tipi, renk, görünürlük, saydamlık, katman ve aksiyon tanımlamaları yapılmasına imkan sağlar. İşlem sonuçlarını GUI içerisinde tanımlanabilen eksen (axes) nesnesi içerisinde tablo, grafik veya film olarak gösterilebilir.

Şekil 2.1.’de gösterilen GUIDE editörü, komut penceresinde ‘guide’ yazılarak ya da Matlab penceresinde ‘File’ menüsünden ‘New’ ve daha sonra ‘GUI’ seçilerek ya da Start düğmesi tıklanarak Matlab/GUIDE komutu işletilerek açılabilir [1].



Şekil 2.1. GUIDE editörü

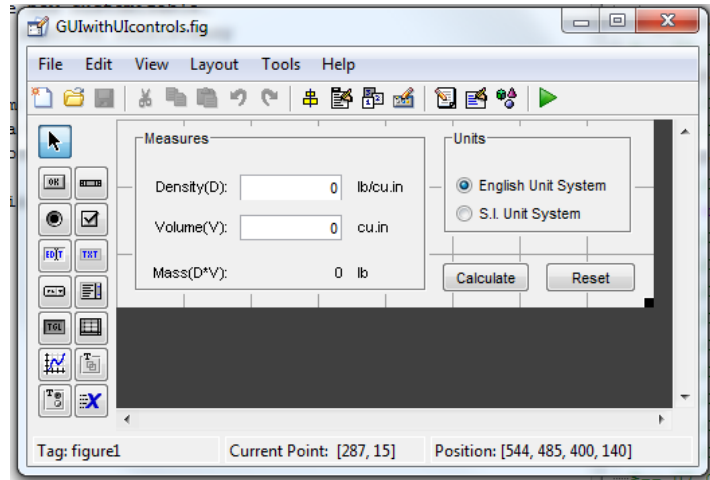
Şekil 2.1.’de gösterilen pencerede eğer yeni bir GUI tasarımı yapılacak ise “Blank GUI” seçeneği seçilebilir. Yoksa önceden yapılmış bir tasarımı açmak istenirse “Open Existing GUI” sekmesinden sonra istenilen dosya seçilir.

Matlab GUIDE aracı GUI tasarımcılarına ilk başlayanlar için kendi içinde hazır örnek uygulamalar; şablonlar (templates) sunmaktadır. Bu uygulamaların arayüzü ve nesnelerin geriçağırımı (callback)'larını kullanan komut satırlarını yazma hususunda programcı ve tasarımcıya önbilgi vermesi bakımından çok yararlıdır.

Bu pencereden şablon olarak karşımıza üç seçenek çıkmaktadır:

- “GUI with Uicontrols” (Uicontrolleri ile GUI)
- “GUI with Axes and Menu” (Uicontrolleri ile GUI)
- “Modal Question Dialog” (Tipik Soru Diyalogu)

GUI with Uicontrols uygulamasında amaç seçilen birim sistemine göre yoğunluk ve hacim değerleri girilen bir cismin kütle değerini Yoğunluk*Hacim ($D*V$) formülünden yola çıkarak hesaplanmasını sağlamak ve kullanıcıya hesaplanan değeri sunmaktır. Burada push butonların ve text kutuların program kodları ve callback parçalarının kullanımına yönelik olarak programcıya bilgiler verilmektedir. Şekil 2.2.'deki gibi görünmektedir.

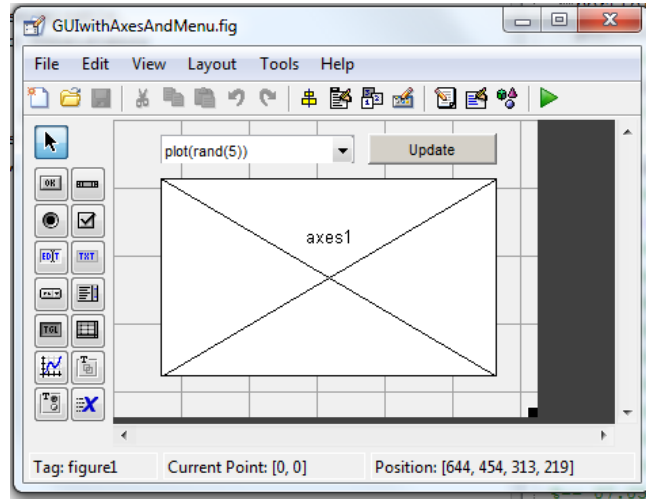


Şekil 2.2. GUI with Uicontrols

GUI with Axes and Menu çalışmasında amaçlanan liste kutusundan seçilen elemanlara göre farklı formülleri kullanılarak yenile (update) butonuna tıklandığında grafik nesnesi üzerinde bulunan sonuçlara göre çizimin yapılmasını sağlamaktır. Bu uygulama ile programcıya grafik çizimi (axes) nesnesinin nasıl kullanıldığı ve liste kutuları ile ilgili geriçağırımı (callback) satırlarının nasıl işletildiği hususunda bilgiler verilmektedir. Ayrıca, bu uygulamada “rand(5)” komutunun kullanımı ve rastgele

değerler içeren dizilerin nasıl üretildiği de gösterilmiştir. Bu uygulamanın bir başka yönü de menü içermesi ve menülerin programlanması konusunda tasarımcıya ön bilgi vermesidir. Menülerin kullanımı ile ilgili olarak aşağıda belirtilen konularda programcıya programlama teknikleri sunulmaktadır:

- `file = uigetfile('*.*fig')` komutu ile bir başka `.fig` dosyanın bir grafik çizim nesnesinde nasıl aktarılacağı,
- `printdlg(handles.figure1)` komutu ile var olan bir çizimin yazıcıdan nasıl çıktı alınacağı,
- `close` komutu ile aktif bir GUI ekranının nasıl kapatılacağı açıklanmaktadır. Şekil 2.3.'deki gibi görünmektedir.

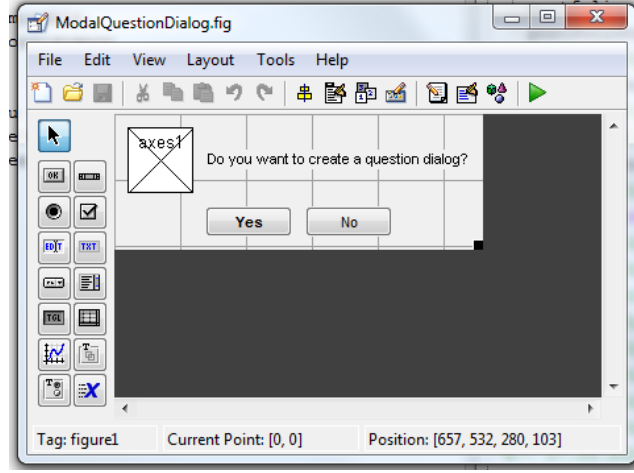


Şekil 2.3. GUI with Axes and Menu

Modal Question Dialog Şekil 2.4.'daki gibi kendisine komut satırından verilen dış parametreleri alarak kendi içerisinde yorumlamak ve buna göre oluşturulacak GUI uygulama ekranının içeriğini (burada pencere başlığı ile metin kutusunun karakter dizisi (string) değerini) değiştirerek kullanıcıya sunmaktır. Ayrıca, bu uygulama ile programcı şu konularda bilgilendirilmektedir:

- `varargin` ve `varargout` komutlarını kullanarak dışarıya bilgi gönderme ve dışarıdan verilen parametreleri yorumlamak
- `uiwait(handles.figure1)` komutu ile bir uygulama penceresinin aktif konumunun nasıl bloklandığı ve ne iş yaptığı,
- `uiresume(handles.figure1)` komutu ile bloklanan bir GUI penceresinin nasıl eski haline getirildiği,

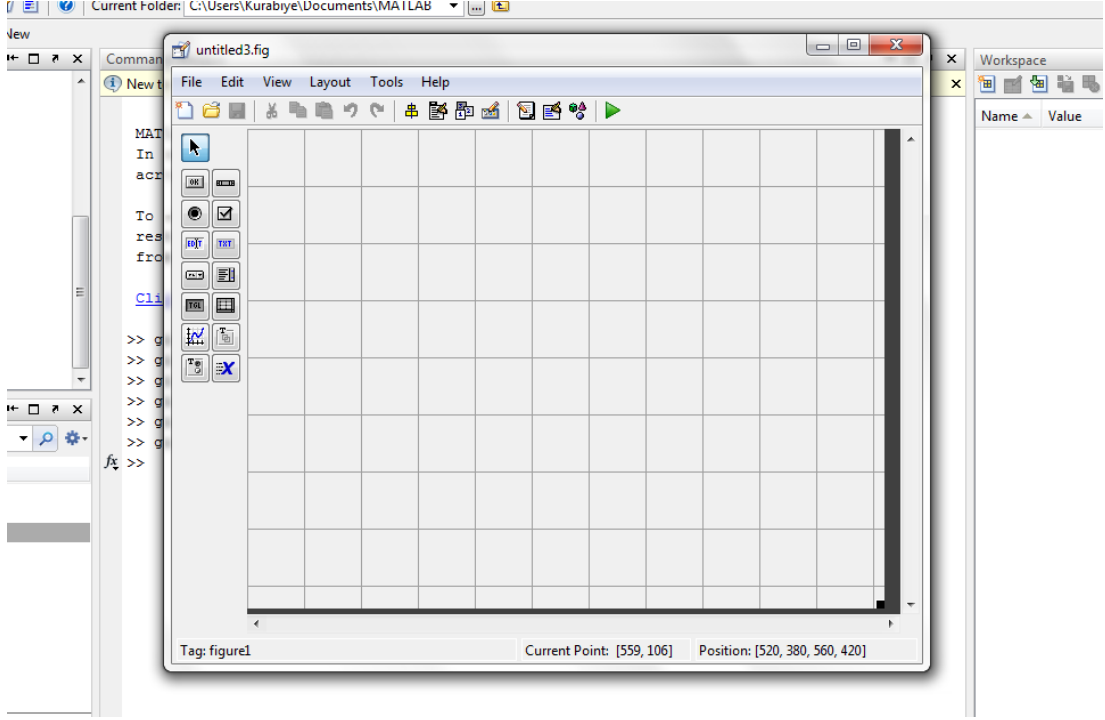
- “modal” programlama tekniđi ile tasarlanan bir GUI uygulamasında açılan bir pencerenin arka taraftaki bir başka pencerenin aktif kontrolünü ele geçirmesi için gerekli programlamanın nasıl yapıldığıdır.



Şekil 2.4. Modal Question Dialog

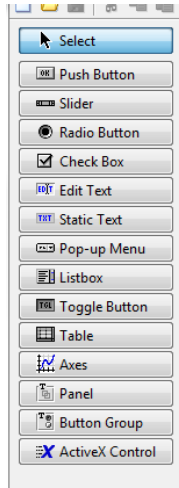
Ayrıca, bu uygulama Matlab GUI tasarımlarında birden fazla form ile nasıl çalışılabileceđi konusunda bir ipucu niteliđi taşımaktadır. Bu uygulama ile giriş ve çıkış parametreleri kullanan GUI uygulama pencereleri vasıtasıyla kendi içinde birden fazla GUI arayüzü içeren GUI uygulamalarının programlanma tekniđi tanıtılmaktadır [6].

Yeni bir tasarım oluşturmak için; varsayılan olarak boş bir GUI seçip OK düğmesine basılarak Şekil 2.5.’deki GUIDE LAYOUT Editor (GUIDE Çalışma Alanı) penceresine ulaşılmaktadır.



Şekil 2.5. GUIDE LAYOUT Editor (GUIDE Çalışma Alanı)

Bu adımdan sonra File/Prefences/Guide yolunu kullanılarak gelen pencereden Show names in component palette (Bileşen paletinde isimleri göster) seçeneğini tıklayıp OK düğmesine basılır. Böylelikle soldaki bileşenler artık isimleri ile Şekil 2.6.'daki gibi görülebilir.



Şekil 2.6. Bileşen paletinde isimlerin gösterildiği durum

2.2.1.1. Gui bileşenleri ve özellikleri

Aşağıda maddeler halinde GUI'nin bileşenleri ve özellikleri kısaca açıklanmıştır, ayrıca Şekil 2.7.'de bileşenlerin figür üzerindeki görünüşleri gösterilmiştir[7].

Push Button: Fare ile üzerine tıkladığında basılmış görünümü alan, fare bırakıldığında eski konumuna dönen, yaylı buton özelliğindeki bileşendir. Basılı konumdan normal konuma dönerken geriçağırım (callback) fonksiyonu işleme konulur.

Slider: Fare ile oklara veya çubuğa tıklanarak yukarı-aşağı ya da sağa-sola hareket edilmesini sağlayan, görüntüyü nümerik değerlere karşılık gelecek şekilde kaydıran bileşendir.

Radio Button: Tıkladığında yuvarlak kutucuğa işaret koyan, tekrar tıkladığında işareti kaldıran bileşendir. GUI çıktısını, işaretlenen duruma göre sonuçlandırmak için kullanılır.

Checkbox: Tıkladığında işaret koyan, tekrar tıkladığında işareti kaldıran işaretleme butonudur. İşaretleme butonlarına atanan özellikler, işaretli olduğunda işleme konulur, aksi durumda işleme konulmaz.

Edit Text: Kullanıcının karakter dizisi girmesini sağlar. Kullanıcının metin girmesi gereken yerlerde kullanılır.

Static Text: Başlık atamak, isim vermek ya da açıklama yapmak için kullanılan ve kullanıcı tarafından değiştirilemeyen kutucuklardır.

Popup Menü: Ok bölümüne tıkladığında birçok seçeneğin görüntülediği, menü düzenindeki bileşendir. Farklı gruplarda ve birbiriyle bağlantılı seçeneklerin, yer tasarrufu yaparak görüntülenmesine imkan sağlar.

Listbox: Kullanıcı için birçok seçeneği kutucuk içinde görüntülemeyi sağlayan bileşendir. Kullanıcının bir seçeneği veya birkaç seçeneği aynı anda seçmesi sağlanabilir.

Toogle Button: Fare ile üzerine tıkladığında basılı konuma geçen, tekrar tıkladığında eski konumuna dönen buton özelliğindeki bileşendir. Üzerine

tıklandığında yani her konum deęişiminde geriçaęırım (callback) fonksiyonu isleme koyulur.

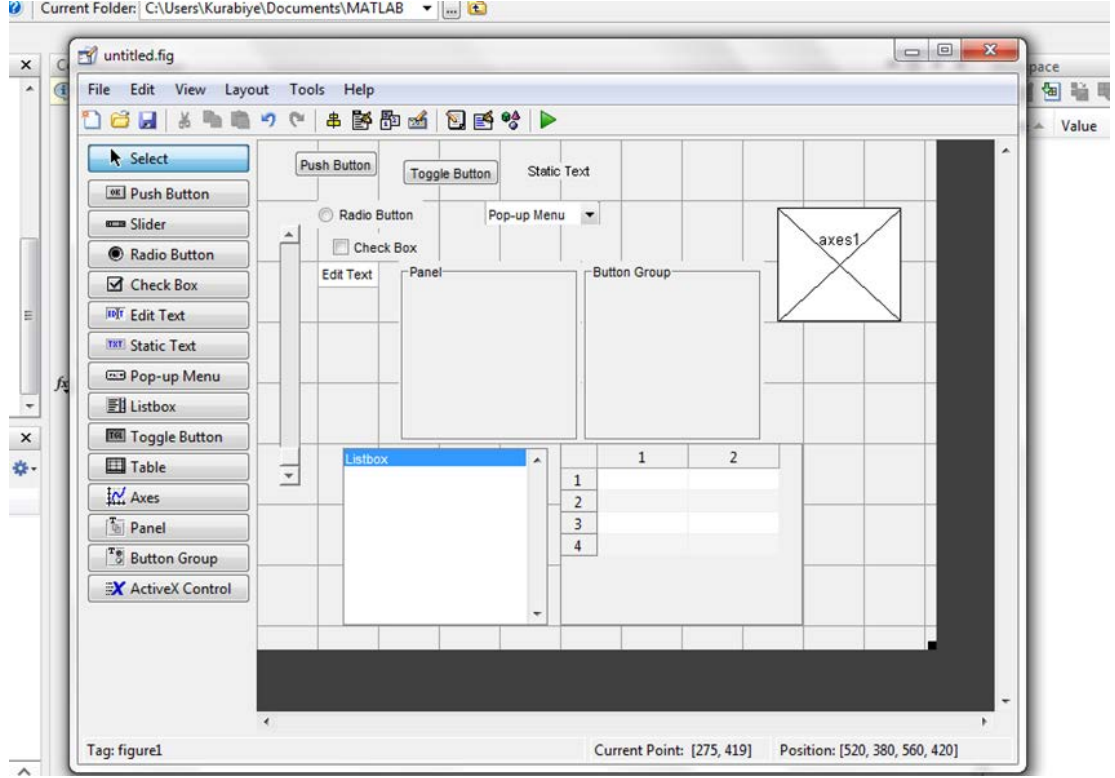
Table: Bu bileşen kullanıcıya verileri 2 boyutlu bir tablo içerisinde sunmaya imkan sağlar. İlk kez Matlab 2008b sürümü ile kullanılmaya başlanmıştır.

Axes: Grafiklerin görüntülenmesi için kullanılan, eksenleri ve grafik alanı içeren bileşendir.

Panel: GUI yüzeyi nesnelere kullanıcıya daha anlamlı ve güzel görünmesini sağlayan, ayrıca tasarımcıya GUI tasarımında kolaylık sunan bir nesne olup, GUI yüzeyi nesnelere gruplanması ve bir arada gösterilmesi amacıyla kullanılır.

Button Group: Radio veya toggle tipteki buton nesnelere bir arada kullanılarak kullanıcının birden fazla seçenekten sadece bir tanesini seçmesini sağlamak amacıyla kullanılan bir nesnedir

ActiveX Component: Matlab GUI tasarımları sadece yukarıda belirtilen nesnelere ile sınırlı değildir. Tasarımcı ayrıca, ActiveX adı verilen ve deęişik alternatifleri olan nesnelere kullanılmalarına da olanak verir. Böylece hem tasarımcı hem tasarlanılacak GUI arayüzünün kullanımı bakımından kullanıcıya esneklik sağlanmış olur.



Şekil 2.7. Bileşenlerin Figür üzerindeki görünümü

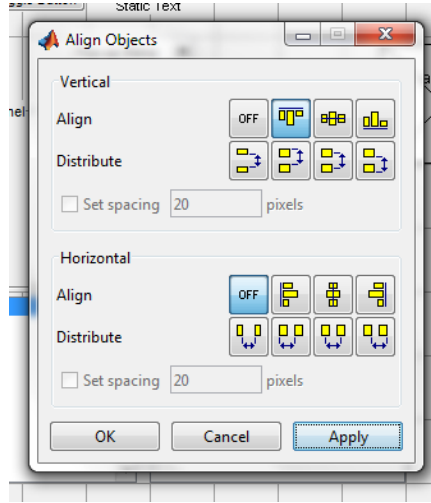
2.2.1.2. Guide aracının incelenmesi

Aşağıda maddeler halinde GUIDE aracındaki araç kutusunun özellikleri kısaca açıklanmıştır, ayrıca Şekil 2.8.'de araç kutusunun GUIDE üzerindeki görünümü gösterilmiştir.



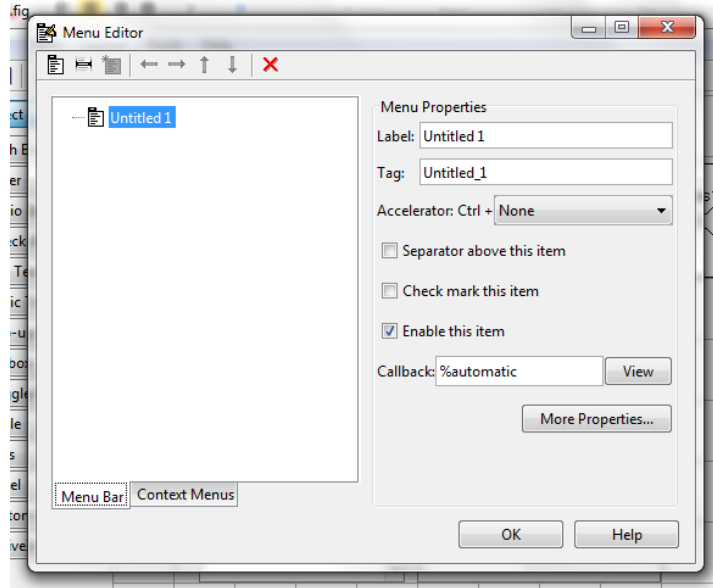
Şekil 2.8. GUIDE Araç Kutusu

Align Objects (Nesne Hizalama) : Şekil 2.9.'da da görüldüğü gibi bu araç sayesinde GUI çalışma alanına eklenen nesnelerin yatay ya da dikey olarak hizalanması işlemleri gerçekleştirilir.



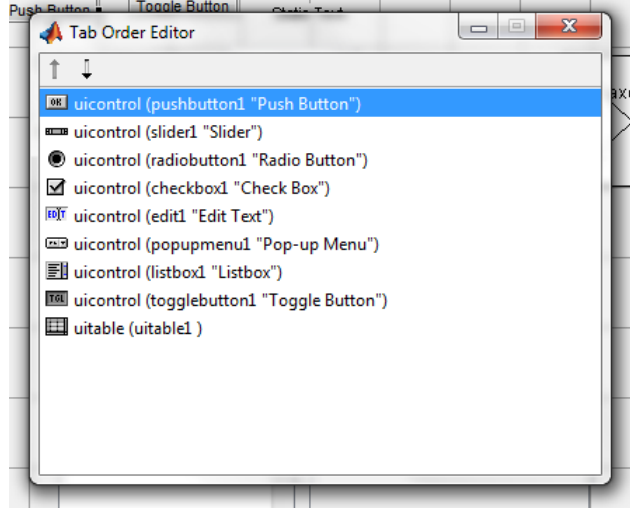
Şekil 2.9. Align Object (Nesne Hizalama) Penceresi

Menu Editor: GUI uygulamasına istenilirse File, Edit... v.b. gibi menü içeren programlarda olduğu gibi bir menü eklenmesi ve eklenen menü ile ilgili işlemlerin yapılması bu araç vasıtasıyla sağlanır ve Şekil 2.10.'da görülmektedir



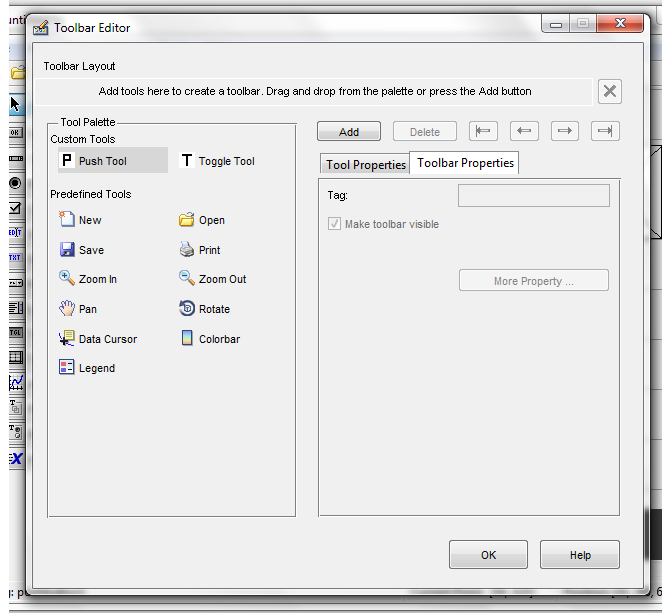
Şekil 2.10. Menu Editor Penceresi

Tab Order (Sekme sırası) Editor: Tab Order (sekme sırası) Editor kullanılarak Şekil 2.11'deki gibi GUI yüzeyindeki nesnelerin birinden diğerine tab tuşu ile geçiş sırası (örneğin bir buton seçili ve aktif iken bir başka butona ya da bir liste kutusuna tab tuşu kullanılarak geçilmesi gibi.) değiştirilebilir.



Şekil 2.11. Tab Order (Sekme sırası) Editor Penceresi

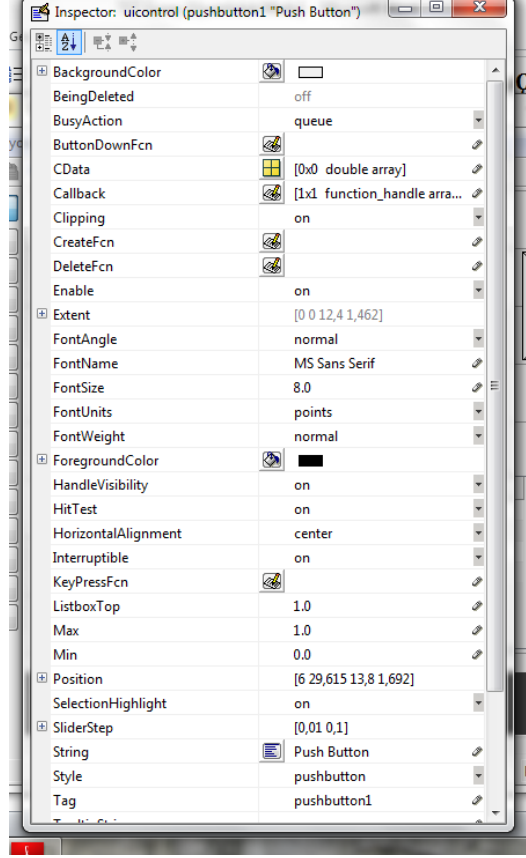
Toolbar (Araç Çubuğu) Editor: Şekil 2.12.'de görüldüğü gibi bir GUI uygulamasına araç çubuğu eklemek için kullanılır. Kullanıcının o uygulamayı kullanım kolaylığını artırır.



Şekil 2.12. Toolbar (Araç çubuğu) Editor Penceresi

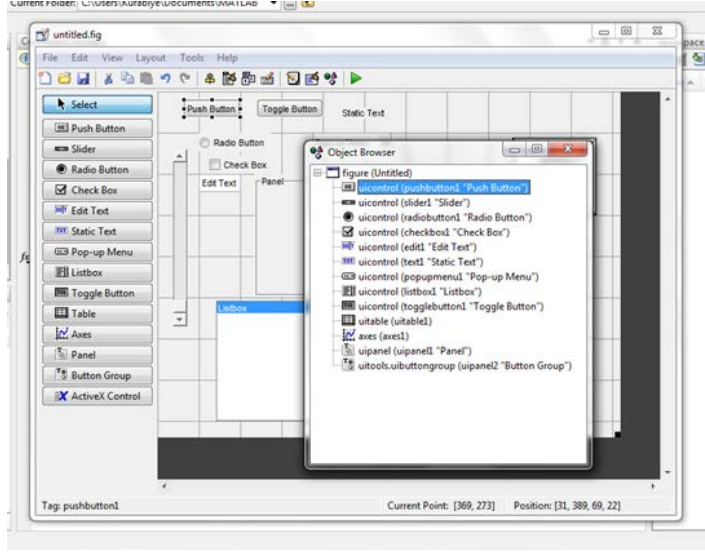
M-File Editor: Hazırlanmış olan GUI uygulaması ile ilgili komutları görebilmek ve üzerinde değişiklik yapabilmek için bu araç kullanılır.

Property Inspector (Özellik Denetçisi) : Bu pencere sayesinde de GUI uygulamasına eklenen nesnelerin özellikleri değiştirilebilir ya da var olan özelliklerinin ve değerlerinin neler olduğu Şekil 2.13.'deki gibi gözlenilebilir.



Şekil 2.13. Property Inspector (Özellik Denetçisi) Penceresi

Object Browser (Nesne Tarayıcısı) :Yerleştirilen bileşenlerin hiyerarşik düzeni GUIDE editöründeki “Object Browser” kısayoluna tıklanarak Şekil 2.14.’deki gibi görülebilir. GUI içerisinde kullanılan bileşenler oluşturulma sırasına göre yukarıdan aşağıya dizilir. İlk oluşturulan “pushbutton” ve en son oluşturulan bileşen “Button Group” bileşenidir.

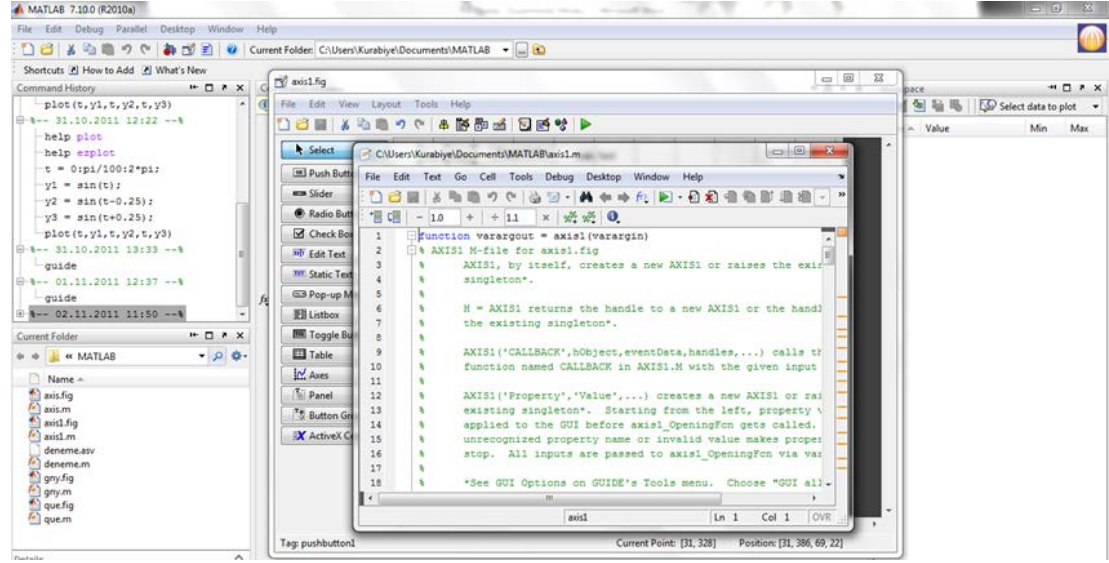


Şekil 2.14. Object Browser (Nesne Tarayıcısı) Penceresi

Run Figure: Bu buton yardımı ile hazırlanmış olan bir GUI uygulaması çalıştırılabilir. Bu şekilde tasarımcı hazırlamış olduğu GUI'yi test etme imkanına sahiptir[6].

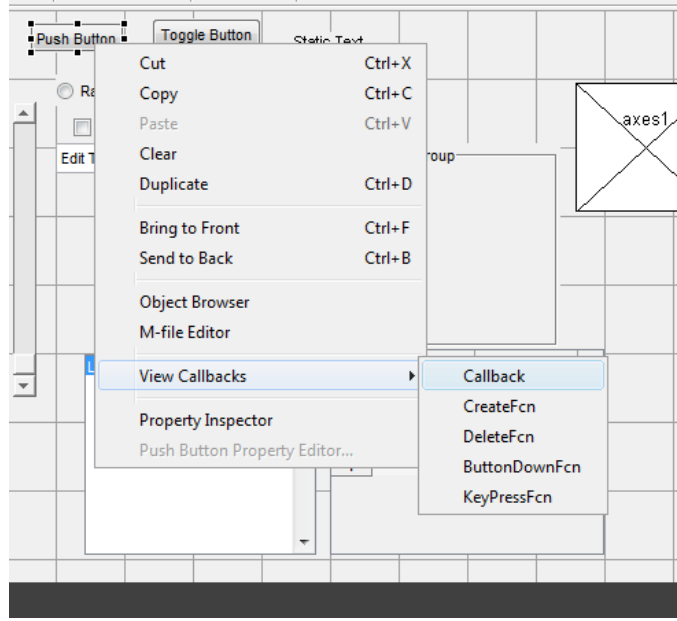
2.2.1.3. Gui arayüzünün programlanması

Bir GUI arayüzünün programlanması demek o çalışmanın kaydedildiği isimle aynı zamanda oluşturulan .m uzantılı dosya içerisine kodlama satırlarının eklenmesi demektir. Bu dosyanın içine görebilmek, değişiklik yapabilmek için GUIDE çalışma ekranı penceresinden View/M-File Editor komutu işletilebilir. Ardından karşımıza Şekil 2.15.'deki gibi bir pencere gelecektir.



Şekil 2.15. M-File Editor Penceresi

Şekil 2.15.'deki pencerede hazırlamış olduğumuz GUI tasarımına ait kodlar görülmektedir. Burada pek çok kodun hazır eklenmiş olduğu görülecektir. Bu kodlar otomatik olarak Matlab GUIDE tarafından eklenmiştir. Biz burada ilgili butonlara ve liste kutularına ya da istenilen bir nesneye ait callback isimli alt program parçalarına ilgili kodları yazacağız. Bir nesneye ait callback in bulunduğu satıra gitmek için araç çubuğunda yer alan f simgeli butona tıklanır ve açılan listeden ilgili nesneye ait callback in ismi seçilir. Bu durum yukarıdaki pencerede de görülmektedir. Ayrıca, GUIDE çalışma ekranından da direk istenilen bir callback satırına gidilebilir. Bunun için Şekil 2.16.'daki gibi ilgili nesne üzerinde sağ tıklanır ve açılan pencereden View Callbacks menüsünden ilgili callback tıklanması ya da ilgili nesne seçilip View/View Callbacks yolu üzerinden açılan listeden gidilmek istenilen callback tıklanması yeterlidir[6].



Şekil 2.16. Geriçağırım (Callback) Fonksiyonlarına ulaşma

2.2.1.4. Callback kullanımı

Callback fonksiyonu, GUI'nin belirli bir bileşeni veya GUI'nin kendisi ile ilişkilendirmek için kullanılan fonksiyondur. Callback, bileşeni için bir olay karşısında bazı eylemleri gerçekleştirerek GUI veya bileşen davranışlarını kontrol eder. Olay bir düğme, menü seçimi tuşuna basma, bir fare tıklaması olabilir.

Callbackler önceki konularda da bahsedildiği üzere oluşan herhangi bir olaya bağlı olarak her nesne için ayrı ve olayın türüne göre icra edilen alt program parçalarıdır. Aşağıda sunulan tabloda kullanılan callbacklerin işlevi ve hangi nesneler ile birlikte kullanıldığı belirtilmiştir.

Bir m file dosya yapısı gereği bir GUI uygulaması tasarımında da aynı dosya ismini taşıyan fonksiyon ismi ile başlayan bir m function yapısına sahiptir. function varargout = DeneGui (varargin)

M function ilk satırında yer alan (yukarıda da ifade edilen deyimde de görüldüğü üzere) parametrelerinin dinamik olmasına dikkat edilmelidir. Yani, “varargin” deyimini ile giriş parametreleri hücre dizisi formatında birden fazla olabilir. Aynı, şekilde GUI uygulamasının kapatılacağı zaman aynı bir fonksiyon mantığı ile “varargout” dışarıya aktarılacak parametreler bu değişkene aktarılabilir.

Dışarıdan fonksiyon içerisine gönderilen giriş parametreleri ile ilgili bilinmesi gereken iki değişken vardır. Bunlar nargin ve varargin değişkenleridir.

- nargin değişkeni: Fonksiyona (ya da GUI uygulamasına) dışarıdan gönderilen toplam parametre sayısını tutar.
- varargin değişkeni: Fonksiyona gönderilen parametrelerin alınmasını sağlar. Hücre dizisi yapısında olduğundan parametrelerin alınması için dizi indislerinin “{“ ve ”}” işaretleri arasında yazılması gerekir.

Fonksiyonunun içinden dışarıya GUI uygulaması sonlandırılırken gönderilen çıkış parametreleri ile ilgili bilinmesi gereken iki değişkeni vardır. Bunlar nargout ve varargout değişkenleridir.

- nargout değişkeni: Fonksiyona (ya da GUI uygulamasına) dışarıdan gönderilen toplam parametre sayısını tutar.
- varargout değişkeni: Fonksiyondan dışarıya parametrelerin gönderilmesini sağlar. Hücre dizisi yapısında olduğundan parametrelerin bu değişkene atanması sırasında dizi indislerinin “{“ ve ”}” işaretleri arasında yazılması gerekir.

Tablo 2.1.’de GUI Uygulamalarında Geriçağırım (Callback) Türleri gösterilmiştir.

Tablo 2.1. GUI Uygulamalarında Geriçağırım (Callback) Türleri [6]

Callback Türü	Tetiklendiği Olay	Kullanıldığı Nesnelere	
ButtonDownFcn	Fare göstergesi Figure veya bir nesnenin kenarlarından 5 piksel içerde olduğunda fare butonu tıklandığında oluşur. UI control için Enable özelliğinin true olması gerekmektedir.	axes	figure
		buttongroup	panel
		userinterface controls	
Callback	Nesnenin temel olayı. Örneğin bir push button tıklandığında ya da bir menü öğesi seçildiğinde oluşur.	contextmenu	menu
		userinterface controls	
CreateFcn	Bir nesnenin create edilmesi anında oluşur. Nesne oluşturulduğunda initializing için kullanılabilir. Bu olay nesne create edilince ancak nesne GUI alanında gözükmeden önce icra edilir.	axes	figure
		buttongroup	contextmenu
		menu	panel
		userinterface controls	
DeleteFcn	Bir nesnenin kaldırılması anında oluşur. Herhangi bir nesne veya figure yok edilmeden önce temizlemeye dayalı operasyonlarda kullanılabilir.	axes	figure
		buttongroup	contextmenu
		menu	panel
		userinterface controls	

Tablo 2.1. GUI Uygulamalarında Geriçağırım (Callback) Türleri (Devam) [6]

CallbackTürü	Tetiklendiği Olay	KullanıldığıNesneler
KeyPressFcn	Figure veya bir nesne focus (aktif) olduğunda veya klavyeden herhangi bir tuşa basıldığında oluşur.	figure userinterface controls
ResizeFcn	Panel, button group veya figure nesnelerinin boyutları kullanıcı tarafından değiştirildiğinde oluşur. Ayrıca, bu durumun gerçekleşmesi için figure'e ait "Resize" özelliği "on" olmalıdır.	buttongroup figure panel
WindowButtonDownFcn	Fare işaretçisi figure penceresi üzerinde iken farenin tuşuna basıldığında oluşur.	figure
WindowButtonUpFcn	Faretuşu bırakıldığı zaman oluşur.	figure

Not: User interface controls push button, slider, radio button, check box, editable textbox, static text, listbox ve toggle butonları içeren genel bir tanımlama ismidir. Bazen uicontrols olarak da isimlendirilebilirler.

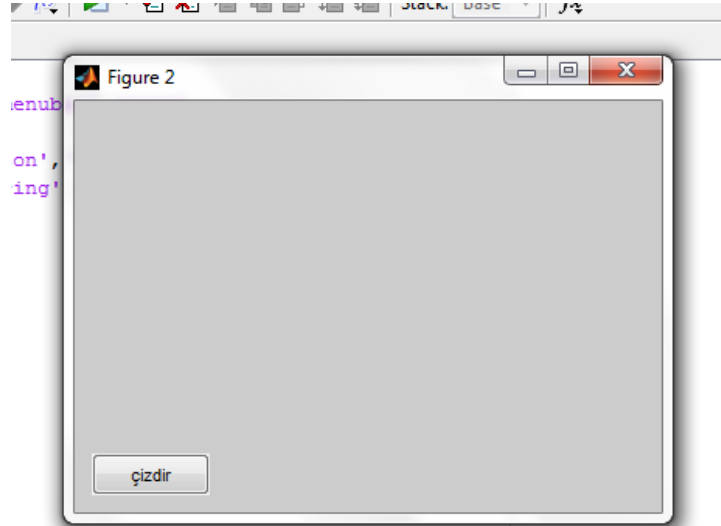
Bir olay karşısında tetiklenecek olan Geriçağırım (Callback) fonksiyonu nesne ile ilişkilendirmek için kullanılabilir 3 farklı yöntem mevcuttur. İlişkilendirme işlemi,

- Metin veya metinsel olarak çağırılacak M-dosyasının adı kullanarak
- Fonksiyon tutamacı veya hücre içerisinde oluşturulmuş fonksiyon tutamacı ile birlikte geri çağırılmalı fonksiyona gönderilecek diğer parametreler kullanılarak.
- Hücre içerisinde metinsel olarak M-dosyanın adı ve varsa parametreleri kullanılarak, yapılır.

2.2.1.5. Metin biçimindeki callback fonksiyonları

```
function guney
fig=figure('Units','centimeters','menubar','none',...
    'position',[10 10 10 7]);
dugme = uicontrol('style','pushbutton','Units','centimeters',...
    'position',[0.3 0.3 2 0.7],'string','çizdir')
```

Şekil 2.17. Metin Biçimindeki Callback Fonksiyonları



Şekil 2.18. guney fonksiyonu ekran çıktısı

Bu fonksiyonu çalıştırdığımızda karşımıza Şekil 2.18'deki gibi bir şekil penceresi çıkar. Ancak düğmenin üzerine tıkladığında hiçbir şey olmaz çünkü gerçekleşen olay karşısında yapılması gerekenler belirlenmemiştir. Düğmeye tıkladığında

$y=0.6x^3-12x^2+3x-5$ fonksiyonunun grafiğini $x=0$ ile 20 arasındaki değerler için çizdirelim. Grafiksel bir arabirim tasarlıyor olmasaydık, bu işlemi

```
>>x=0:0.5:20; y= 0.6*x.^3-12*x.^2+3*x-5; plot(x,y);
```

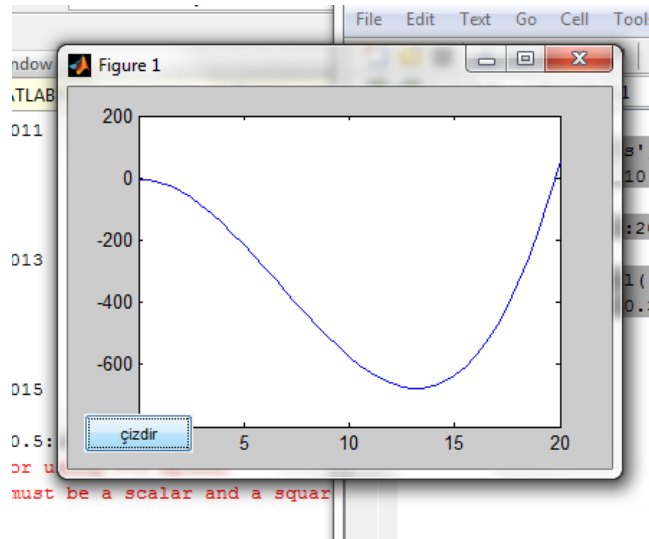
Şekil 2.19. Kod Parçası

Şekil 2.19'daki gibi tek bir satırda gerçekleştirebilirdik. Ancak, amacımız bu işlemin düğmeye basıldığı anda gerçekleşmesi. Bu yüzden düğmenin callback özelliğine bu işlemi yapacak bir değer atmalıyız.

```
function guney
fig=figure('Units','centimeters','menubar','none',...
'position',[10 10 10 7]);
ifade = 'x=0:0.5:20; y= 0.6*x.^3-12*x.^2+3*x-5; plot(x,y)';
dugme = uicontrol('style','pushbutton','Units','centimeters',...
'position',[0.3 0.3 2 0.7],'string','çizdir','Callback',ifade);
```

Şekil 2.20. Çizdirelecek Fonksiyon

Şekil 2.20'de çizdirmeyi amaçladığımız fonksiyon ifade isimli bir değişkende oluşturuldu. Uicontrol nesnemizde ise callback özelliğine atandı. Bu sayede düğme tıklandığında istediğimiz grafik Şekil 2.21'deki gibi çizdirilecektir.



Şekil 2.21. guney fonksiyonu ekran çıktısı

2.2.1.6. Fonksiyon tutamacı (handle) kullanılarak belirlenen callback fonksiyonlar

```
function guney
fig=figure('Units','centimeters','menubar','none',...
    'position',[10 10 10 7]);
dugme = uicontrol('style','pushbutton','Units','centimeters',...
    'position',[0.3 0.3 2 0.7],'string','çizdir','Callback',@cizdir);
function cizdir (~,~)
x=0:0.5:20;
y= 0.6*x.^3-12*x.^2+3*x-5;
plot(x,y);
```

Şekil 2.22. Fonksiyon Tutamacı Callback Fonksiyonlar

Öncelikle fonksiyonun düğme kısmında callback özelliğine çizdir isimli fonksiyonun tutamacı atanıyor. Bu durumda düğmeye tıklandığı anda bu fonksiyon çağrılacak. Daha sonra çizdir isimli fonksiyon oluşturuluyor. Bu fonksiyonun aynı m-dosya içerisinde yer alması gerekli değildir. Başka bir m-file içerisinde de kodlanabilir. Hatta başka bir m-file içerisinde alt fonksiyon olarak da kodlanabilir. Burada dikkat edilmesi gereken konu çizdir fonksiyonu kurallara göre 2 parametre almalıdır. İlki, fonksiyonu tetikleyen grafik nesnesinin tutamacıdır. Bu arabirim tasarımı içim çok önemli bir parametredir. Diğer parametre ise Matlab'ın gelecekteki sürümlerinde kullanılmak üzere ayrılmış bir değerdir. Şu an için bu değer bizim için bir önemi yoktur. Ancak kurallar gereği Şekil 2.22.'deki gibi kullanılmalıdır.

2.2.1.7. Hücre içerisinde metinsel olarak belirlenen callback fonksiyonlar

Metin biçimindeki geri çağırılmalı fonksiyonların en büyük dezavantajı kullanılan ifadelerin Matlab çalışma alanında hesaplatılmasıdır. Bu yüzden, kullanılan değişkenler Matlab çalışma alanında oluşturulurlar. Bu problemi çözmek için Callback fonksiyonu hücre içerisinde metinsel olarak belirlenebilir. Kullanımı fonksiyon tutamaçları kullanarak belirlenen callback fonksiyona çok benzerdir. Ancak, tutamacı kullanılarak belirlenen callback fonksiyonlardan farklı olarak ayrı

bir m-dosya içerisine kodlanmalılardır. Bu durumda callback fonksiyonlar olarak aynı veya farklı bir m-dosya içerisinde yer alan alt fonksiyonlar kullanılamaz.

```
function guney
fig=figure('Units','centimeters','menubar','none',...
    'position',[10 10 10 7]);
dugme = uicontrol('style','pushbutton','Units','centimeters',...
    'position',[0.3 0.3 2 0.7],'string','çizdir','Callback',{'cizdir'});
Düğme ile ilişkilendirilen çizdir fonksiyonu aşağıdaki gibi ayrı bir m-dosya
olarak oluşturulmalıdır.
function cizdir (~,~)
x=0:0.5:20;
y= 0.6*x.^3-12*x.^2+3*x-5;
plot(x,y);
```

Şekil 2.23. Hücre İçerisinde Metinsel Callback Fonksiyonlar

Bu sayede Şekil 2.23.'deki gibi geri çağırılmalı fonksiyon içerisindeki değişkenler yerel olduğu için Matlab çalışma alanı ile bir ilişkileri söz konusu değildir. Ancak, alt fonksiyonlar kullanılmadığı için bütün grafik nesnelere callback fonksiyonları ayrı ayrı m-dosyalar içerisine kodlanmalıdır. Bu ise, büyük bir uygulama geliştirirken kullanışsızdır. Dolayısı ile bir grafik nesnesinin callback fonksiyonunu belirlemenin en kullanışlı yolu fonksiyon tutamaçlarıdır [1].

2.2.2. M-File programlama yöntemi kullanılarak guioluşturma

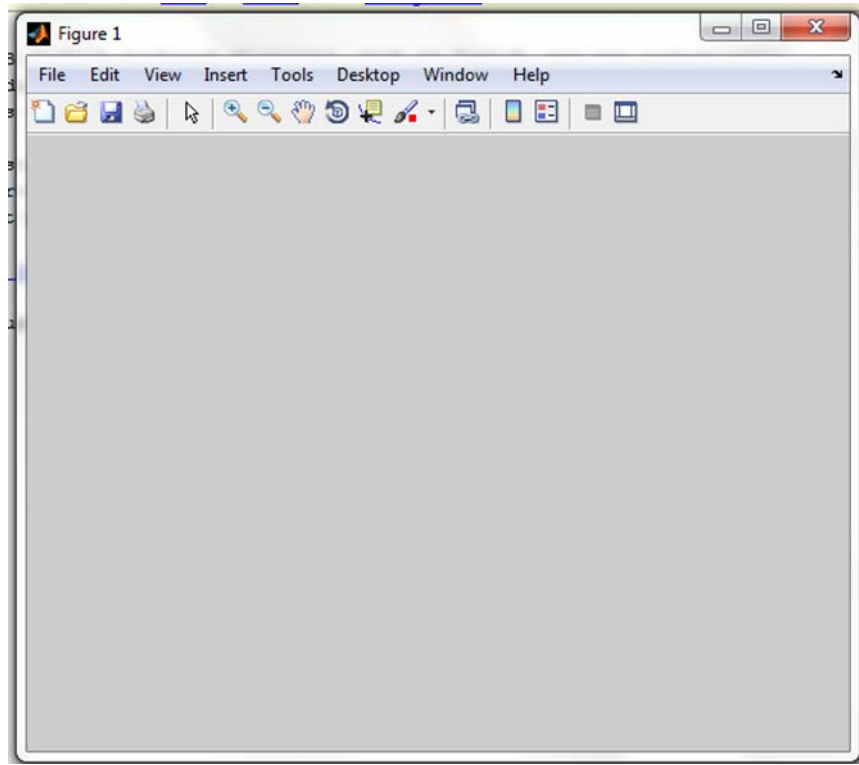
Diğer programlama dilleri hakkında az da olsa bilgi sahibi olanlar, bir programın arabiriminin bir form üzerinde tasarlandığını ve kodun arka planda çalışarak, kullanıcı ile form üzerindeki düğme, onay kutusu, liste kutusu gibi nesnelere arasındaki ilişkiyi sağladığını bilirler. İşte Matlabın bütün arabirim tasarımlarını oluşturmak üzere size sunduğu formun adı şekil (figure) penceresidir. Bu şekil penceresi, üzerinde programlarımızın kullanıcılar ile irtibatını sağlayacak olan arabirimleri bu sefer M-dosyalar içerisinde grafiksel araç kutusu fonksiyonlarını kullanarak oluşturacağız. Oluşturduğumuz arabirim program çalıştığında şekildekine benzer bir şeyler olacak.

2.2.2.1. Programlama yoluyla nesnelerin eklenmesi

Burada GUIDE gibi bir tasarım aracı kullanılmaz. Sadece kod satırları yazılarak hem GUI arayüzü hem de bu arayüzün koşturduğu komut satırları aynı dosya içerisinde yazılır. Bu dosyalar .m uzantısına sahiptirler[1].

2.2.2.2. Figure nesnesi

Figure nesnesi bütün programlarımızın grafiksel arabirimlerini ve veri görselleştirme işlemlerimizi üzerinde gerçekleştireceğimiz şekil penceresidir. Şekil 2.24 örnek bir figür nesnesi gösterilmiştir.



Şekil 2.24. Figure Nesnesi

figure Nesnesinin Bazı Özellikleri:

Children: Şekil penceresinin çocuk (axes, uicontrol, uimenu, uicontextmenu gibi) nesnelerinin tutamaçlarını tutan dizidir.

Color: Şekil penceresinin zemin rengini düzenlemeye yarayan 1x3 boyutlu bir dizidir. Dizinin elemanları yerine 0 ile 1 arasında değişen reel sayılar yazılabilir ki bu değerler tanımlanan rengin RGB renk oranlarını gösterir.[0 0 0] değeri siyah rengi, [1

1 1] değeri beyaz rengi, [1 0 0] değeri kırmızıyı, [0 1 0] değeri yeşili, [0 0 1]değeri maviyi, [0.6 0.7 0.8] değeri açık maviyi belirtir.

Menubar: Şekil penceresinin üst tarafında bulunan standart menüleri kapatıp-açmaya yarar.

Name: Şekil penceresi üstündeki başlığı düzenlemeye yarayan özelliktir.

NextPlot: Grafik işlemleri aktif pencere üzerinde nasıl gerçekleşeceğini belirleyen özelliktir. Değeri add, replace veya replacechildren olabilir.

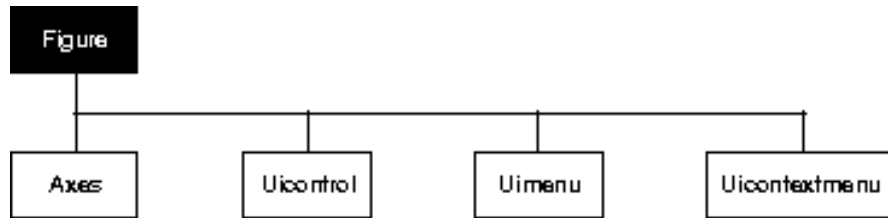
NumberTitle: Şekil penceresinin üst bölümündeki şekil penceresi numarasının görünüp görünmemesini sağlayan özelliktir. Değeri off yaparak numaranın görünmemesi, on yaparak görünmesi sağlanır.

Pointer: Pencere üzerine geldiğinde, farenin biçimini belirleyen özelliktir. Bu değerler crosshair, fullcrosshair, arrow, ibeam, watch, right, bottom, circle, cross gibi olabilir.

Position: Şekil penceresinin ekran üzerindeki konumunu belirleyen 1x4 boyutlu bir diziyi düzenleyen özelliktir.

Resize: Şekil penceresinin boyutlarını değiştirmeye yarayan özelliktir. Değeri on olursa şekil penceresi boyutları değiştirilebilir, off olursa değiştirilemez.

Units: Şekil penceresinin uzunluk ve konum bilgilerinin birimini tutan özelliktir, centimeters, pixels , inches, characters ... gibi değerler alabilir.



Şekil 2.25. Figure Nesnesinin Üzerinde Barındırdığı Nesnelere

Figure nesnesi Şekil 2.25'te görüldüğü gibi üzerinde 4 temel nesne barındırabilir. Bunlar şekilde görüldüğü gibi üzerinde çizim işlemlerinin gerçekleştirileceği eksen (axes) nesnesi, liste kutusu, onay kutusu, düzenleme kutusu ve düğme gibi kullanıcı ile ilişki kuracak olan grafiksel arabirim kontrolleri (uicontrol), kullanıcının çeşitli

fonksiyonları çalıştırabilmesi için düzenlenmiş menü (uimenu) ve fareye sağ tıklayarak erişilebilen bir kısayol menüsüdür (uicontextmenu).

Örnek bir figure Nesnesinin Eklenmesi Şekil 2.26.'daki gibidir.

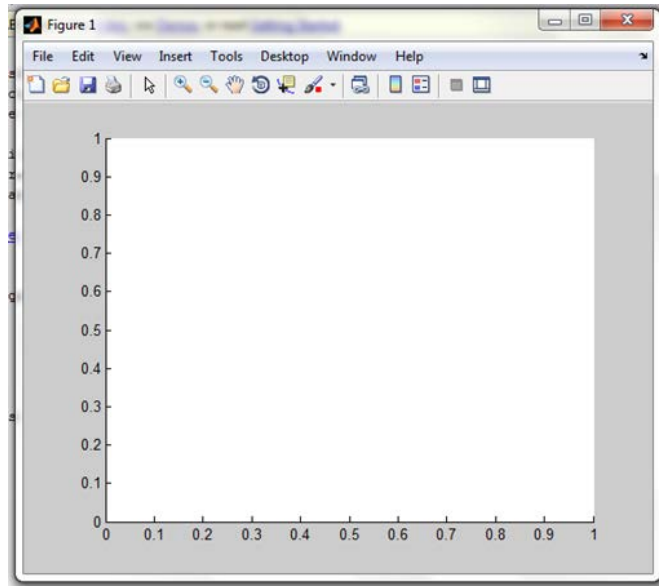
```
fig1 = figure( 'units','pixels',...  
              'position',[800 200 200 260],...  
              'menubar','none',...  
              'name','GUI_1',...  
              'numbertitle','off',...  
              'resize','off');
```

Şekil 2.26. figure Nesnesinin Eklenmesi

Nesneleri figure üzerine ekleyeceğimiz için diğer nesnelere de oluşturduğumuz fig1 üzerine ekleyelim.

2.2.2.3. Axes nesnesi

Çizim, animasyon, resim ve video görüntüleme gibi birçok kabiliyete sahip eksen nesnesi. Axes nesnesi Şekil 2.27.'deki gibi ancak bir figure nesnesi üzerinde oluşturulabilir. 2 boyutlu ve 3 boyutlu veriyi görselleştirebilir. Ayrıca çocuk nesnelere sahip olan tek nesnedir.



Şekil 2.27. Eksen (Axes) Nesnesi

axes Nesnesinin Bazı Özellikleri:

Box: Grafiğin, iki boyutlu çizimlerde bir dikdörtgen içinde, üç boyutlu çizimlerde bir küp içinde görünüp görünmemesini sağlayan özelliktir. Değeri on ya da off olabilir.

CameraPosition: Eksen koordinatlarında kameranın sahneyi gördüğü pozisyonu tanımlayan 1x3 lük bir diziyi belirleyen özelliktir.

CameraPositionMode: Kamera pozisyonu ayarını düzenleyen özelliktir. Değeri auto olduğunda Matlab pozisyonu otomatik belirler, manual olduğunda ise CameraPosition daki değer geçerlidir.

CameraTarget: Eksen koordinatlarında kameranın baktığı noktayı tanımlamaya yarayan 1x3 lük diziyi tutan özelliktir.

CameraTargetMode: Kamera hedefini tayin etmeye yarayan özelliktir. Değeri auto olduğunda Matlab kamera hedefini otomatik belirler, manual olduğunda ise CameraTarget daki değer geçerlidir.

CameraUpVector: Kameranın CameraTarget ve CameraPosition ile tayin edilen bakı doğrultusu etrafındaki dönmeyi belirleyen 1x3 lük vektördür.

CameraUpVectorMode: Kameranın bakı doğrultusu etrafındaki dönmesini tayin etmeye yarayan özelliktir. Değeri auto olduğunda Matlab kameranın bakı doğrultusu etrafındaki dönmeyi otomatik belirler, manual olduğunda ise CameraUpVector daki değer geçerlidir.

CameraViewAngle: Görme alanını belirleyen açıdır.0° ile 180° arasında bir değer olabilir.

CameraViewAngleMode: Görme alanını belirleyen açının modunu belirleyen özelliktir. Değeri auto olduğunda Matlab kameranın görme açısını bütün sahneyi görecekt biçimde en küçük açığa ayarlar, manual olduğunda ise CameraViewAngle daki değer geçerlidir.

Children: Eksen üzerinde oluşturulmuş çocuk nesnelerinin (line, light, patch, surface, rectangle ... gibi) tutamaçlarını tutan dizidir.

Color: Eksenin arka rengini tanımlayan 1x3 lük dizidir. Ayrıca değeri none de olabilir. Bu durumda eksen şeffaflaşarak arkasındaki nesnelere de gösterir.

DataAspectRatio: Eksen çizgilerinin (x, y ve z) birbirlerine göre oranını belirleyen 1x3 lük bir diziyi tutan özelliktir.

DataAspectRatioMode: Eksen çizgilerinin birbiri ile ilgili oranını belirlemeye yarayan özelliktir. Değeri auto olduğunda Matlab eksen çizgilerinin birbirine oranını otomatik olarak ayarlar, manual olduğunda ise DataAspectRatio daki değer geçerlidir.

FontAngle: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı biçiminin eğimini düzenlemeye yarayan özelliktir, .normal, italic veya oblique değerlerini alabilir.

FontName: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı fontunu düzenlemeye yarayan özelliktir.

FontSize: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı tipi boyutunu düzenlemeye yarayan özelliktir.

FontUnits: Fontsize özelliği ile belirlenen yazıtipi büyüklüğünü birimini belirlemeye yarayan özelliktir. Değeri centimeters, inch, points veya normalized olabilir.

FontWeight: Eksen çizgilerinin etiketleri ve çentik değerleri için kullanılan yazı tipinin ağırlığını düzenlemeye yarayan özelliktir. normal, bold veya light olabilir.

GridLineStyle: Çizim alanını eksen çizgilerine dik olarak bölen çizgilerin (grid lines) stilini belirleyen özelliktir. Değeri; -, --, :, -. ve none olabilir.

Layer: Eksen çizgilerinin eksen çizim alanının üst ya da alt tarafında olmasını sağlayan özelliktir. Değeri top veya bottom olabilir.

LineStyleOrder: Çoklu bir grafik çizimi yapıldığında, sırasıyla grafik çizgilerini belirlemeye yarayan özelliktir. Örneğin ilk grafik * lardan, ikincisi - lardan üçüncüsü + lardan oluması isteniyorsa; bu özelliğe atanan değer '*|-|+' olmalıdır.

LineWidth: Eksen çizgilerinin kalınlıklarını düzenlemeye yarayan özelliktir. Varsayılan değer 0.5 noktadır.

NextPlot: Grafik işlemlerinin aktif eksen üzerinde nasıl gerçekleşeceğini belirleyen özelliktir. Üç farklı değer alabilir bunlar: add, replace ve replacechildren dir.

Position: Eksen nesnesinin şekil penceresi üzerindeki pozisyonunu belirleyen 1x4 boyutlu bir diziyi düzenlemeye yarayan özelliktir.

Projection: Eksen üzerindeki grafiğin üç boyutlu görüntüleme yöntemini belirleyen özelliktir. İki değer alabilir, bunlar: ortographic ve perspective dir.

Tag: Eksen nesnesine bir etiket ismi vermeye yarayan özelliktir. get fonksiyonu ile değerini öğrenebilir set fonksiyonu ile istediğimiz bir değeri atayabiliriz.

Title: Eksenin başlık metninin tutamacıdır.

Units: Eksen nesnesinin uzunluk ile ilgili birimini düzenlemeye yarayan özelliktir. Değeri centimeters, pixels, inches, characters ... olabilir.

Visible: Eksen nesnesinin görünüp-görünmemesini düzenleyen özelliktir. Değeri on ise eksen görünür, off ise görünmez.

XColor, YColor, ZColor: x, y, z eksenlerinin renklerini düzenleyen özelliktir.

XDir, YDir, ZDir: X,y,z eksen değerlerinin artı yönünü belirler. Değeri normal ise artı normal yöndedir, reverse ise artı ters yöndedir.

XGrid, YGrid, ZGrid: Çizimaları eksen çizgilerine dik yönde bölen çizgilerin (gridlines) görünüp görünmemesini belirleyen özelliktir. Değeri on ya da off olabilir.

XLabel, YLabel, ZLabel: Eksen çizgi etiketlerini düzenlemeye yarayan özelliktir.

XLim, YLim, ZLim: Eksenin x, y ve z çizgilerinin sayısal aralıklarını belirleyen özelliktir. Örneğin x çizgisinin sayısal değeri 10 ile 120 arasında olmasını istersek; XLim özelliğinin değeri [10 120] olmalıdır.

XLimMode, YLimMode, ZLimMode: Eksen çizgi değerlerinin sayı aralıklarını düzenlemeye yarayan özelliktir. Değeri auto olduğunda Matlab eksen çizgilerinin sayısal aralıkları otomatik olarak belirlenir, manual olduğunda XLim, ZLim veya ZLim özelliğinin değeri geçerlidir.

XScale, YScale, ZScale: Eksen çizgilerinin sayısal ölçeğini belirlemeye yarayan özelliktir. Değeri lineer olduğunda sayı artışı düzgün artar, log olduğunda logaritmik artar.

Örnek Bir axes Nesnesinin Eklenmesi Şekil 2.28.'deki gibidir.

```
eksen = axes('units','pixels',...  
            'position',[30 30 160 160],...  
            'fontsize',8,...  
            'parent',fig1);
```

Şekil 2.28. axes Nesnesinin Eklenmesi

2.2.2.4. Uicontrol nesnesi

Uicontrol nesnesi, bir şekil(figure) penceresi üzerinde kullanıcı ile iletişimi sağlayacak olan liste kutusu, onay kutusu, komut düğmesi ve bunun gibi grafiksel kontrolleri sağlayan nesnedir. Bu nesne sadece style özelliği aracılığı ile aşağıdaki listede gördüğümüz bütün kontrolleri bünyesinde barındırır.

- pushbutton (komut düğmesi)
- togglebutton (açma-kapama düğmesi)
- radiobutton (radyo düğmesi)
- checkbox (onay kutusu)
- edit (metin düzenleme kutusu)
- text (sabit metin kutusu)
- listbox (liste kutusu)
- popupmenu (açılır liste kutusu)
- slider (kaydırma çubuğu)
- frame (çerçeve)

uicontrol Nesnesinin Bazı Özellikleri:

BackgroundColor: Nesnenin arka plan rengini belirlemeye yarayan özelliktir.

Callback: Kontrol düğmesinin fare ile tıklandığında tetiklenecek geri çağırılmalı fonksiyonu belirleyen özelliktir.

CData: Komut düğmesi ya da açma-kapama düğmesi üzerinde görüntülenebilecek gerçek renklerden oluşan resmi tanımlayan bir matristir.

Enable: Kontrolün kullanılıp-kullanılmayacağını belirleyen özelliktir. Değeri on ise düğme kullanılabilir ve normal renktedir, off ise düğme kullanılamaz ve rengi siliktir, inactive ise düğme yine kullanılamaz fakat rengi normaldir.

FontAngle: Nesnenin string özelliğinin tuttuğu yazı biçimini düzenlemeye yarayan özelliktir, .normal, italic veya oblique değerlerini alabilir.

FontName: Nesnenin string özelliğinin tuttuğu yazı fontunu düzenlemeye yarayan özelliktir.

FontSize: Nesnenin string özelliğinin tuttuğu yazı tipi boyutunu düzenlemeye yarayan özelliktir.

FontUnits: Fontsize özelliği ile belirlenen yazıtipi büyüklüğünü birimini belirlemeye yarayan özelliktir. Değeri centimeters, inch, points veya normalized olabilir.

FontWeight: Nesnenin string özelliğinin tuttuğu yazı tipinin ağırlığını düzenlemeye yarayan özelliktir. Normal ,bold veya light olabilir.

ForegroundColor: Nesnenin string özelliğinin tuttuğu yazının rengini (RGB) tutan 1x3 boyutunda bir diziyi belirlemeye yarar.

HorizontalAlignment: Metin kutusu ve statik metin nesnelerinin etiket metnini hizalamaya yarayan özelliktir. left, right veya center değerlerini alabilir.

Position: Nesnenin şekil penceresi üzerindeki pozisyonunu belirleyen 1x4 boyutlu bir diziyi düzenlemeye yarayan özelliktir.

String: Nesnenin üzerinde bulunması gereken yazıyı düzenlemeye yarayan özelliktir.

Style: Kontrol nesnesinin tipini belirleyen özelliktir. Değeri pushbutton olduğunda nesne komut düğmesi, togglebutton olduğunda nesne açma-kapama düğmesi, radiobutton olduğunda nesne radyo düğmesi, checkbox olduğunda nesne onay kutusu, edit olduğunda nesne metin düzenleme kutusu, text olduğunda nesne sabit metin kutusu, listbox olduğunda nesne liste kutusu, popupmenu olduğunda nesne açılır liste

kutusu, slider olduđunda nesne kaydırma çubuđu, frame olduđunda nesne çerçeve nesnesine dönüşür.

Tag: Nesneye bir etiket ismi vermeye yarayan özelliktir. Bu özellik sayesinde nesne findobj komutuyla bulunabilir.

TooltipString: Fare nesnenin üzerine geldiđinde bir ipucu ya da açıklama metni görüntülemeye yarayan özelliktir.

Units: Nesnenin uzunluk ile ilgili birimini düzenlemeye yarayan özelliktir. Deđeri centimeters, pixels, inches, characters olabilir.

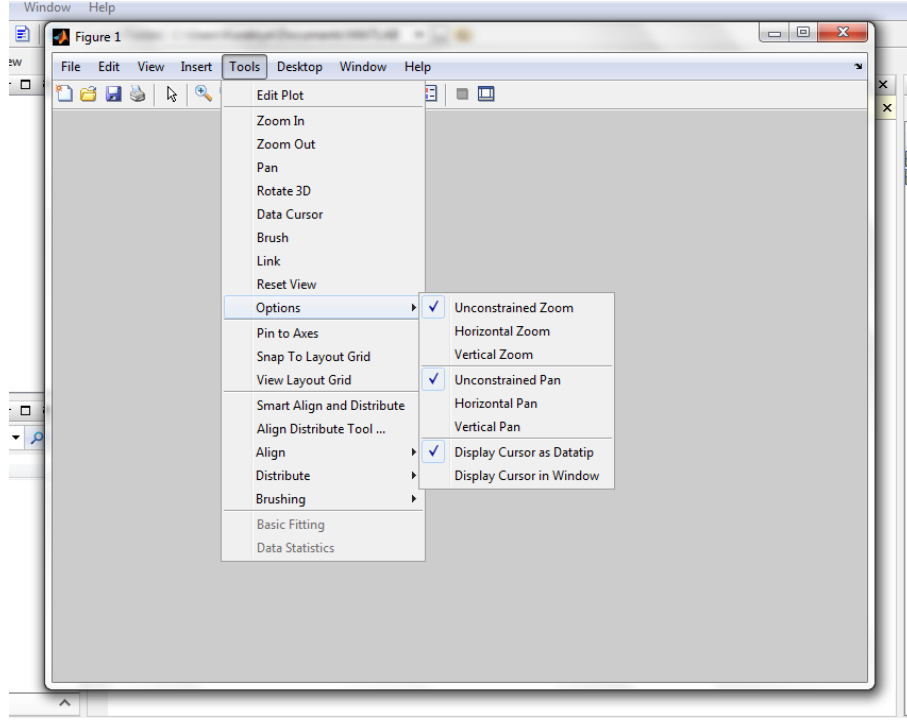
Örnek Bir uicontrol Nesnesinin eklenmesi Şekil 2.29.'daki gibidir.

```
list1 = uicontrol('style','list',...  
                'unit','pix',...  
                'position',[10 110 180 180],...  
                'min',0,'max',2,...  
                'fontsize',14,...  
                'parent',fig1,...  
                'string',{'bir';'iki';'üç';'dört'});
```

Şekil 2.29. uicontrol Nesnesinin eklenmesi

2.2.2.5. Uimenu nesnesi

Şekil(figure) nesnesi üzerinde oluşturulabilen menü nesnelidir. Menüler üst seviye ve alt seviye olmak üzere ikiye ayrılır. Üst seviye menülerin ebeveyni şekil penceresidir. Alt seviye menülerin ebeveyni ise başka bir menüdür.



Şekil 2.30. Uimenu Nesnesi görünümü

Şekil 2.30’da File menüsü üst seviye bir menüdür. Unconstrained Zoom ve altındaki seçeneklerin yer aldığı menü ise alt seviye bir menüdür ve ebeveyni Tools menüsüdür. Tools->Brushing menü seçeneğinin hemen altında yer alan ve menü seçeneklerini birbirinden ayırmak için kullanılan çizgi ayırıcıdır (separator). Ayrıca Unconstrained Zoom menü seçeneği gibi, bir menü tıklandığı takdirde onay işareti ile seçili hale gelebilir ya da Basic Fitting menüsü seçeneği gibi kullanılmaz yapılabilir.

uimenu Nesnesinin Bazı Özellikleri:

Callback: Menü nesnesinin fare ile tıklanarak seçildiğinde tetiklenecek geri çağırılmalı fonksiyonu belirleyen özelliktir.

Checked: Menü seçeneği tıklandığında, sol tarafında seçildiğine dair onay işareti belirlenmesini sağlayan bir özelliktir. Değeri on olduğunda onay işareti konur, off olduğunda konmaz.

Enable: Menü seçeneğinin kullanılıp kullanılmayacağını belirleyen özelliktir. Değeri on ise seçenek kullanılabilir ve normal renktedir, off ise düğme kullanılmaz ve rengi siliktir.

ForegroundColor: Menü seçeneğinin string (etiket yazısının) rengini belirlemeye (RGB kodlarını tutan 1x3 boyutunda bir dizi) yarayan özelliktir.

Label: Menü seçeneğinin üstünde görüntülenecek metni belirlemeye yarayan özelliktir. Menü açıkken, kısa yoldan çalışmasını istersek, ilgili kısayol karakterinin önüne & işareti konmalıdır.

Position: Menü seçeneğinin, ebeveyni içerisindeki göreceli pozisyonunu belirlemeye yarayan özelliktir. İlk menü değeri 1 dir.

Separator: Menü nesnesi üzerinde bir ayırıcı işaret konulmasını düzenler. Değeri on ise ayırıcı işaret konur, off ise konmaz.

Tag: Menüye bir etiket ismi vermeye yarayan özelliktir. Bu özellik sayesinde nesne findobj komutuyla bulunabilir.

Visible: Menü'nün görünüp-görünmemesini düzenleyen özelliktir. Değeri on ise menü görünür, off ise görünmez.

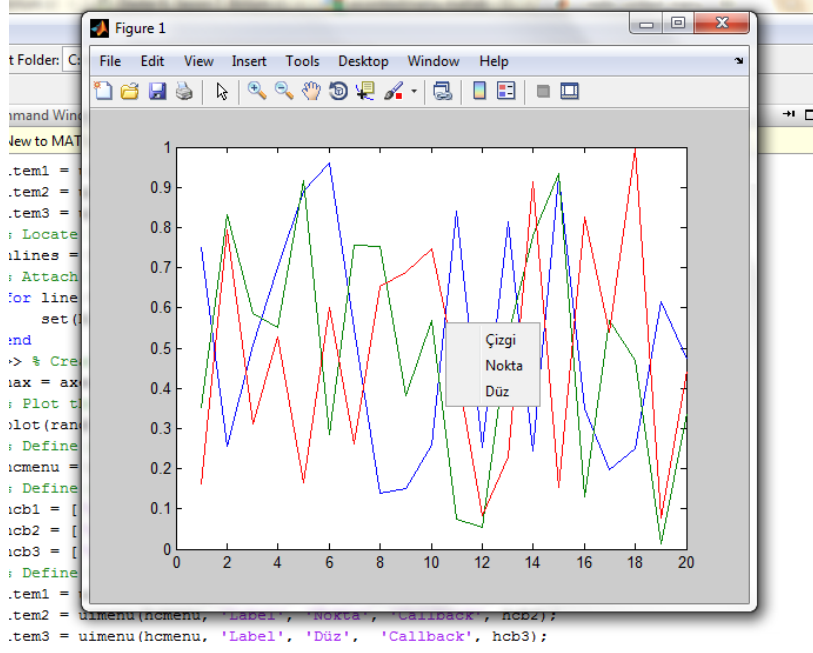
Örnek Bir uimenu Nesnesinin eklenmesi Şekil 2.31.'deki gibidir.

```
menu = uimenu('Label','Workspace');  
uimenu('Label','New Figure','parent',menu);  
uimenu('Label','Save','parent',menu);  
uimenu('Label','Quit',...  
        'Separator','on','Accelerator','Q','parent',menu);  
set(fig1,'uimenu',menu)
```

Şekil 2.31. uimenu Nesnesinin eklenmesi

2.2.2.6. Uicontextmenu nesnesi

Uicontextmenu, Şekil 2.32'deki gibi bir grafik nesnesi üzerinde sağ tıklanarak erişilen bir kısayol menüsü oluşturmak için kullanılan nesnedir.



Şekil 2.32. Uicontextmenu Nesnesi görünümü

Uicontextmenu Nesnesinin Bazı Özellikleri:

Callback: Kısayol menüsüne atanmış bir grafik nesnesi üzerinde menüyü açmak üzere farenin sağ düğmesine basıldığı anda tetiklenecek geri çağırılmalı fonksiyonu belirleyen özelliktir.

Children: Kısayol menü nesnesinin çocuklarının (uimenu nesneleri) tutamaçlarını tutan dizidir. Bir kısayol menüsü içerisinde birçok menü olabilir. Bu menülerin tutamaçları, bir dizi içerisinde kısayol menüsünün Children özelliğine atanmalıdır.

Parent: Kısayol menü nesnesinin ebeveynini belirleyen özelliktir. Bir kısayol menü nesnesinin ebeveyni figure, axes, line, light, rectangle, patch, surface veya text grafik nesnelere biridir.

Type: Grafik nesnesinin türünü tutar. Kısayol menü nesnesi için bu özellik 'uicontextmenu' olacaktır.

Örnek Bir uicontextmenu Nesnesinin eklenmesi Şekil 2.33.'deki gibidir.

```
menu = uicontextmenu;  
alt1 = uimenu( 'label','Menü 1',...  
              'parent',menu);  
alt2 = uimenu( 'label','Menü 2',...  
              'parent',menu);  
set(fig1,'uicontextmenu',menu)[1]
```

Şekil 2.33. uicontextmenu
Nesnesinin eklenmesi

2.3. Matlab GUI Uygulamalarında Diyalog Kutuları

Matlab GUI ile görsel tasarım hem programcının tasarımı kolay kılmakta, hem de kullanıcı yapacağı işleri görerek ve kolaylıkla birkaç tıklama ile dahi gerçekleştirebilmektedir. GUI uygulamalarının kullanıcılara sunduğu kolaylıklardan biri de kullanıcıların yapılan veya yapılacak işler ilgili uyarılması veya bilgilendirilmesi amacıyla kullanılan etkileşim kutularının kullanılmasıdır. Örneğin bir hata oluştuğunda bu durumu en temel anlamı ile GUI ara yüzünde bir nesne kullanarak ve bu nesneye ait bir özelliği (örneğin renk gibi) değiştirerek kullanıcı bilgilendirilebilir. Ancak, bu yöntem etkin bir uyarma aracı olarak düşünülemez. Bunun yerine az önce bahsedildiği gibi örneğin bir hata penceresi ile hem kullanıcının başka işlem yapması engellenebilir (yani, uygulamanın arka plandaki ara yüzü uyarı anında kilitlenebilir), hem de daha gerçekçi ve sade bir diyalog pencere görüntüsü kullanıcının Windows benzeri GUI tabanlı ortamlarda alışkanlıkların dışında tasarım ekranları ile karşılaşmamış olur. Ayrıca, etkileşim kutularının kullanımı ile programcının da GUI ara yüzünde herhangi bir tasarım değişikliğine gitmesine fırsat verilmez, dolayısıyla daha etkin GUI tabanlı uygulama geliştirme imkanı sağlanmış olmaktadır [6]. Tablo 2.2.'de GUI uygulamalarında kullanılacak etkileşim kutuları verilmiştir.

Tablo 2.2. GUI uygulamalarında kullanılacak etkileşim kutuları [1]

Diyalog Kutusu	Açıklama
axlimdlg	Eksen çizgilerinin sayısal aralıklarını düzenleyen diyalog kutusunu düzenler.
dialog	Belirtilen özelliklere sahip diyalog kutusunu görüntüler.
errordlg	Hata diyalog kutusunu görüntüler.
helpdlg	Yardım diyalog kutusunu görüntüler.
imageview	Şekil penceresi üzerinde, resmi yakınlaştırılabilir bir şekilde görüntüleyen diyalog kutusunu gösterir.
inputdlg	Kullanıcı giriş diyalog kutusunu görüntüler.
listdlg	Seçim listesi diyalog kutusunu görüntüler.
menu	Kullanıcı girişi için menü seçenekleri oluşturan diyalog kutusunu görüntüler.
movieview	Şekil penceresi üzerinde film görüntüleyen diyalog kutusunu görüntüler.
msgbox	Mesaj kutusu görüntüler.
pagedlg	Sayfa konumu düzenleme diyalog kutusunu görüntüler.
pagesetupdlg	Sayfa yapısı düzenleme diyalog kutusunu görüntüler.
printdlg	Yazdırma diyalog kutusunu görüntüler.
printpreview	Şekil penceresi ön görünümü diyalog kutusunu görüntüler (yazdırmak için).
questdlg	Soru diyalog kutusunu görüntüler.
soundview	Şekil penceresi üzerinde, ses dosyası içerisindeki veriyi görüntü ile birlikte çalan diyalog kutusunu görüntüler.
uigetpref	Özellik ayarlama desteği ile birlikte soru diyalog kutusunu görüntüler.
uigetfile	Standart dosya aç diyalog kutusunu görüntüler.
uiputfile	Standart dosya kaydet diyalog kutusunu görüntüler.

Tablo 2.2. GUI uygulamalarında kullanılabilecek etkileşim kutuları (Devam) [1]

Diyalog Kutusu	Açıklama
uigetdir	Standart klasör aç diyalog kutusunu görüntüler.
uisetcolor	Renk seçim diyalog kutusunu görüntüler.
Uisetfont	Yazı tipi seçim diyalog kutusunu görüntüler.
uiopen	Dosya aç diyalog kutusunu görüntüler ve seçilen dosyayı open fonksiyonu ile birlikte açar.
uisave	Dosya kaydet diyalog kutusunu görüntüler ve çalışma alanında tanımlı değişkenleri seçilen dosyaya save fonksiyonu ile kaydeder.
uiloadd	Dosya aç diyalog kutusunu görüntüler ve seçilen dosya içerisindeki değişkenleri load fonksiyonu ile yükler.
uiimport	Veri yükleme sihirbazını görüntüler.
waitbar	Üzerinde ilerleme çubuğu(progress bar) yer alan bir diyalog kutusunu görüntüler.
warndlg	Uyarı diyalog kutusunu görüntüler.

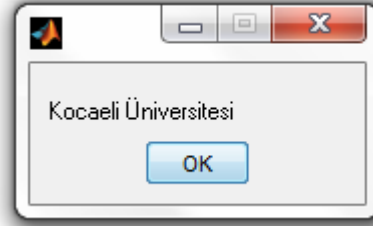
2.3.1. MsgBox fonksiyonu

msgbox(mesaj)

Şekil 2.34'deki kullanım biçimi ile parametre olarak gönderilen mesajı gösteren uygun boyutlarda bir mesaj kutusu (şekil penceresi) görüntülenir. Mesaj metinsel bir vektör, matris veya hücre olabilir. Şekil 2.35'de, 'Kocaeli Üniversitesi' mesajı görüntüleniyor.

```
>>msgbox('Kocaeli Üniversitesi');
```

Şekil 2.34. msgbox Fonksiyonu



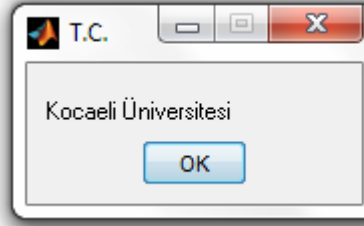
Şekil 2.35. msgbox görünümü

msgbox(mesaj, başlık)

Şekil 2.36.'daki kullanım biçimi ile mesaj kutusunun başlığı düzenlenebilir. (Şekil 2.37.)

```
>>msgbox('Kocaeli Üniversitesi','T.C.');
```

Şekil 2.36. msgbox(mesaj, başlık)



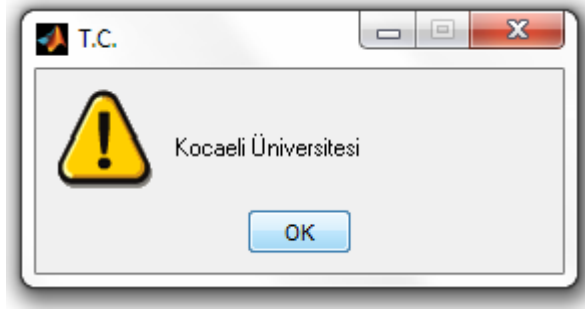
Şekil 2.37. msgbox görünümü

msgbox(mesaj, başlık, simge)

Şekil 2.38.'deki kullanım biçimi ile mesaj kutusu üzerinde belirtilen simge görüntülenebilir. Simge parametresi 'none','error','help','warn' veya 'custom' olabilir. Simge belirtilmezse varsayılan olarak none geçerlidir(Şekil 2.39).

```
>>msgbox('Kocaeli Üniversitesi','T.C.','warn');
```

Şekil 2.38. msgbox(mesaj, başlık, simge)



Şekil 2.39. msgbox görünümü

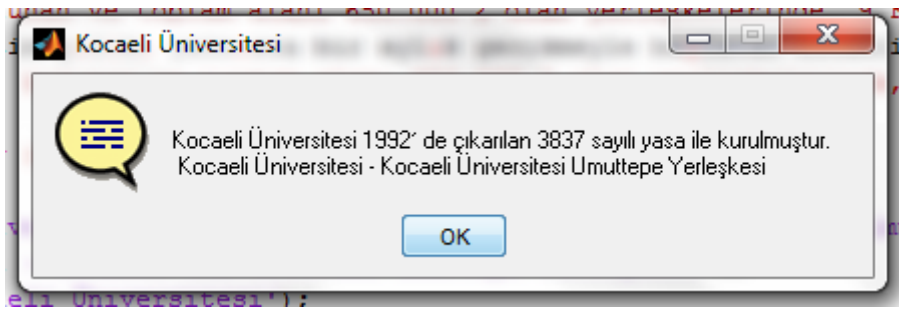
2.3.2. Helpdlg fonksiyonu

helpdlg(mesaj, başlık);

Şekil 2.40.'daki kullanım biçimi ile parametre olarak gönderilen mesajı gösteren uygun boyutlarda bir yardım diyalog kutusu (şekil penceresi) görüntülenir. Diyalog kutusunun başlığı parametre olarak gönderilen başlık metni ile düzenlenebilir. Aynı başlığa sahip bir yardım diyalog kutusu zaten açıksa, aktive edilerek ekranda öne getirilir. Mesaj hücre olarak oluşturularak metin farklı satırlara yazılabilir(Şekil 2.41.).

```
>>mesaj={'Kocaeli Üniversitesi 1992' de çıkarılan 3837 sayılı yasa ile kurulmuştur.';  
' Kocaeli Üniversitesi - Kocaeli Üniversitesi Umuttepe Yerleşkesi'};  
>> helpdlg(mesaj, 'Kocaeli Üniversitesi');
```

Şekil 2.40. helpdlg(mesaj, başlık)



Şekil 2.41. helpdlg görünümü

2.3.3. Errordlg ve warndlg fonksiyonları

errordlg(mesaj, başlık, OluşturmaŞekli);

warndlg(mesaj, başlık, OluşturmaŞekli);

Şekil 2.42'deki kullanım biçimleri ile parametre olarak gönderilen mesajı gösteren uygun boyutlarda hata veya uyarı simgesi ile diyalog kutuları görüntülenir. Mesaj, metinsel bir vektör, matris veya hücre olabilir. Diyalog kutusunun başlığı parametre olarak, gönderilen başlık metni ile düzenlenebilir. Oluşturma şekli ise msgbox fonksiyonundaki kullanımı ile aynıdır(Şekil 2.43).

```
>>mesaj={'Dosya Açılmadı'};  
>>errordlg(mesaj, 'Dosya Hatası', 'modal');
```

Şekil 2.42. errordlg(mesaj, başlık, OluşturmaŞekli)



Şekil 2.43. errordlg görünümü

2.3.4. Questdlg fonksiyonu

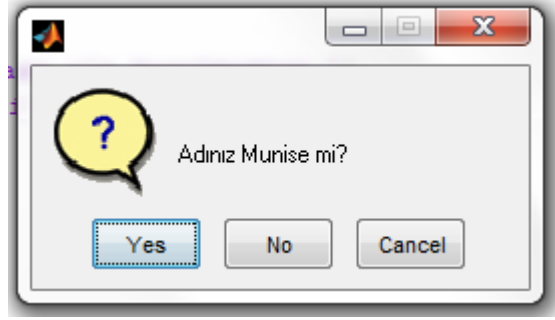
cevap = questdlg(soru);

Şeklindeki kullanım biçimi ile üzerinde soru metni ile birlikte, Yes, No, Cancel düğmeleri yer alan bir diyalog kutusu görüntülenir. Hangi düğmeye basılırsa, fonksiyon geriye düğme üzerinde yer alan metni döndürür. Şekil 2.45.'daki örnek ile kullanıcıya adının Munise olup olmadığı soruluyor. Yes düğmesine basıldığı için fonksiyon geriye 'Yes' metnini döndürüyor.

```
>>>cevap = questdlg('Adınız Munise mi?');
```

Şekil 2.44. questdlg Fonksiyonu

cevap = Yes



Şekil 2.45. questdlg görünümü

2.3.5. Listdlg fonksiyonu

```
[Seçimİndisi, Düğme] = listdlg('ListString',ListeMetni);
```

Şeklindeki kullanım biçimi ile bir liste kutusu içerisinde 2'nci parametre ile hücrel olarak oluşturulmuş liste metnini gösteren ve kullanıcıdan seçim yapmasını isteyen bir diyalog kutusu görüntülenir. Kullanıcı bir seçim yaptıktan sonra OK ya da Cancel düğmelerine basabilir. OK düğmesine basılırsa, fonksiyondan geri dönen düğme değeri 1'e eşit olur. Cancel düğmesine basılırsa düğme değeri 0'a eşit olacaktır. Seçim indisi ise liste kutusu içerisinde seçilen seçeneğin indisidir. Bu değer çoklu seçimlerde vektördür. Hiçbir seçim yapılmadıysa boş bir dizi olarak geri döndürülür(Şekil 2.47.).

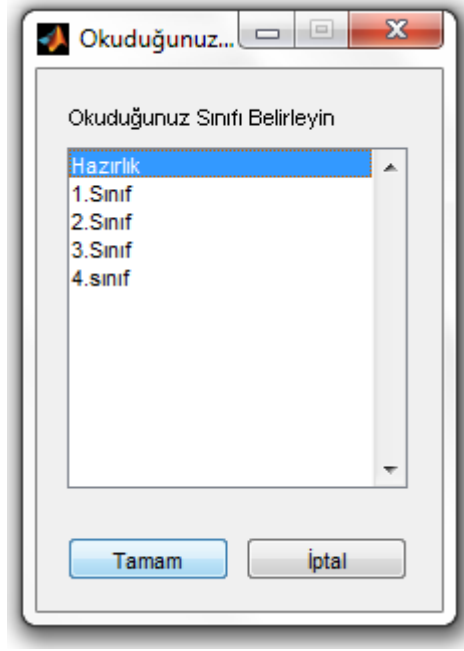
ListString özelliğinin yanında tablodaki özelliklerin değerleri de düzenlenebilir. Tablo 2.3.'te listdlg fonksiyonu özellikleri verilmiştir.

Tablo 2.3. listdlg fonksiyonu özellikleri

Özellik	Açıklama
'SelectionMode'	'single' ya da 'multiple' olabilir. 'single' da liste kutusundan sadece tek bir seçenek seçilebilir. 'multiple' da ise Ctrl ve Shift tuşu ile birlikte çoklu seçim yapılabilir.
'ListSize'	Listenin genişliğini ve yüksekliğini belirleyen vektördür. Birim olarak pixel kullanılır. Standart olarak [160 300] pixel'dir.
'InitialValue'	Diyalog kutusu görüntülediğinde seçili vaziyette olan seçeneklerin indislerini tutan vektördür.
'Name'	Diyalog kutusu başlığını belirler.
'PromptString'	Liste kutusu üzerinde görüntülenecek olan metni belirler.
'OKString'	OK düğmesinin metnini belirler.
'CancelString'	Cancel düğmesinin metnini belirler.
'uh'	OK ve Cancel düğmelerinin pixel cinsinden yüksekliğini belirler. Standart olarak 18 pixeldir.
'fus'	Çerçeve ile kontrol nesneleri arasındaki mesafeyi pixel cinsinden belirler. Standart olarak 8 pixeldir.
'ffs'	Çerçeve ile şekil penceresi arasındaki mesafeyi pixel cinsinden belirler. Standart olarak 8 pixeldir.

```
>>liste = {'Hazırlık'; '1.Sınıf'; '2.Sınıf'; '3.Sınıf'; '4.sınıf'};
>>[indis] = listdlg('ListString',liste,...
    'SelectionMode','single',...
    'ListSize',[170 170],...
    'InitialValue',1,...
    'Name','Okuduğunuz Sınıf',...
    'PromptString','Okuduğunuz Sınıfı Belirleyin',...
    'OKString','Tamam',...
    'CancelString','İptal');
```

Şekil 2.46. listdlg Fonksiyonu



Şekil 2.47. listdlg görünümü

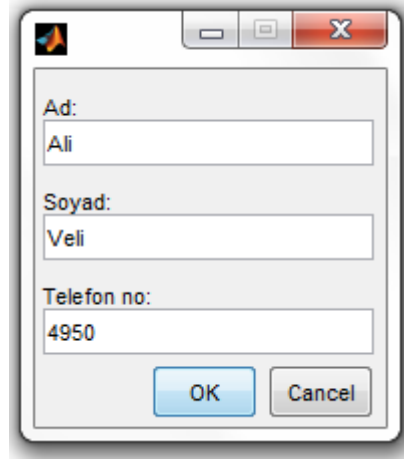
2.3.6. Inputdlg fonksiyonu

```
cevaplar = inputdlg(Sorular)
```

Şeklindeki kullanım biçimi ile kullanıcıya, sorular hücresi içerisindeki her bir soruya cevap yazması için metin kutuları ile birlikte bir diyalog kutusu görüntülenir. Kullanıcının metin kutularına yaptığı girişler ise bir hücre yapısı içerisinde geri döndürülür. Örneğin, kullanıcıdan adını, soyadını ve telefon numarasını almak isteyelim. Bunun için şekil 2.49'deki gibi bir sorular hücresi olarak kullanıcı girişi için diyalog kutusu görüntüleyebiliriz.

```
>> sorular = {'Ad:','Soyad:','Telefon no:'};
>> cevaplar = inputdlg(sorular);
cevaplar =
    'Ali'
    'Veli'
    '4950'
```

Şekil 2.48. inputdlg Fonksiyonu



Şekil 2.49. inputdlg görünümü

2.3.7. Waitbar fonksiyonu

tutamac = waitbar (oran, başlık)

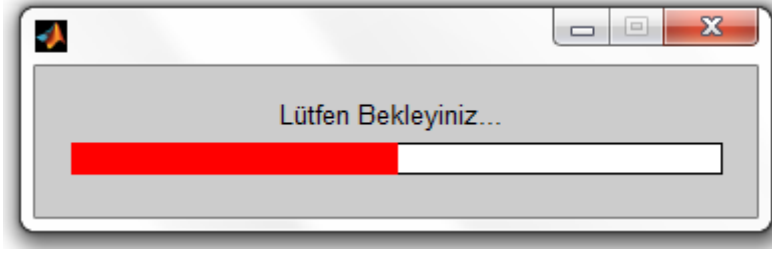
Şeklindeki kullanım biçimi ile belirtilen doluluk oranında ilerleme çubuğuna sahip bir diyalog kutusu oluşturularak tutamacı geriye döndürülür. Ardından

Waitbar(oran, tutamac, başlık);

Şekil 2.50.'deki bir ifade ile tutamacı belirtilen ilerleme diyalog kutusunun oranı güncellenebilir. İlerleme diyalog kutuları genel olarak for veya while döngüleri içerisinde döngü değişkeninin değeri ile kullanılırlar. (Şekil 2.51)

```
h = waitbar(0, 'Lütfen Bekleyiniz...')  
  
    for i=1:10  
  
        waitbar(i/10,h);  
  
        pause(0.5);  
  
    end  
  
    delete(h);
```

Şekil 2.50. Waitbar fonksiyonu



Şekil 2.51.waitbar görünümü

2.3.8. Uigetfile ve uiputfile fonksiyonları

Uigetfile dosya aç diyalog kutusunu görüntülerken uiputfile dosya kaydet diyalog kutusunu görüntüler. Her iki fonksiyonun kullanım biçimi aşağıdaki bi aynıdır.

[DosyaAdı, Dizin, Filtreİndisi] = uigetfile (FiltreTanımı, Başlık)

[DosyaAdı, Dizin, Filtreİndisi] = uiputfile (FiltreTanımı, Başlık)

uigetfile fonksiyonu filtre tanımı hücresi içerisindeki uzantılara sahip dosyaları listeleyen ve belirtilen başlığa sahip bir dosya aç diyalog kutusu görüntüler. Diyalog kutusundaki Dosya adı (File name) alanına yazılarak açılmak istenen dosya mutlaka mevcut olmalıdır. Halbuki uiputfile fonksiyonu ile görüntülenen dosya kaydet diyalog kutusunda mevcut olmayan bir dosya adı belirlenebilir.

Fonksiyonlardan geriye, seçilen dosyanın adı, dosya yolu (dizini) ve filtre tanımı hücresi içerisindeki kaçınıcı uzantıya sahip olduğunu belirten bir indis döndürülür.

```
>> filtre = {'*.txt','metin dosyası(*.txt);*.dat','Veri dosyası (*.dat)'}
```

Şekil 2.52. Filtre

Şekil 2.52.'deki filtre tanımını kullanarak, kullanıcıya uigetfile fonksiyonu ile dosya aç diyalog kutusunu görüntülersek, seçilen dosyanın adını, dizinini ve uzantı indisini elde edebiliriz.

```

>> [dosya, dizin, filtre_indisi] = uigetfile(filtre,'Açılacak Dosyayı Seçin')

dosya =

readme.txt

dizin =

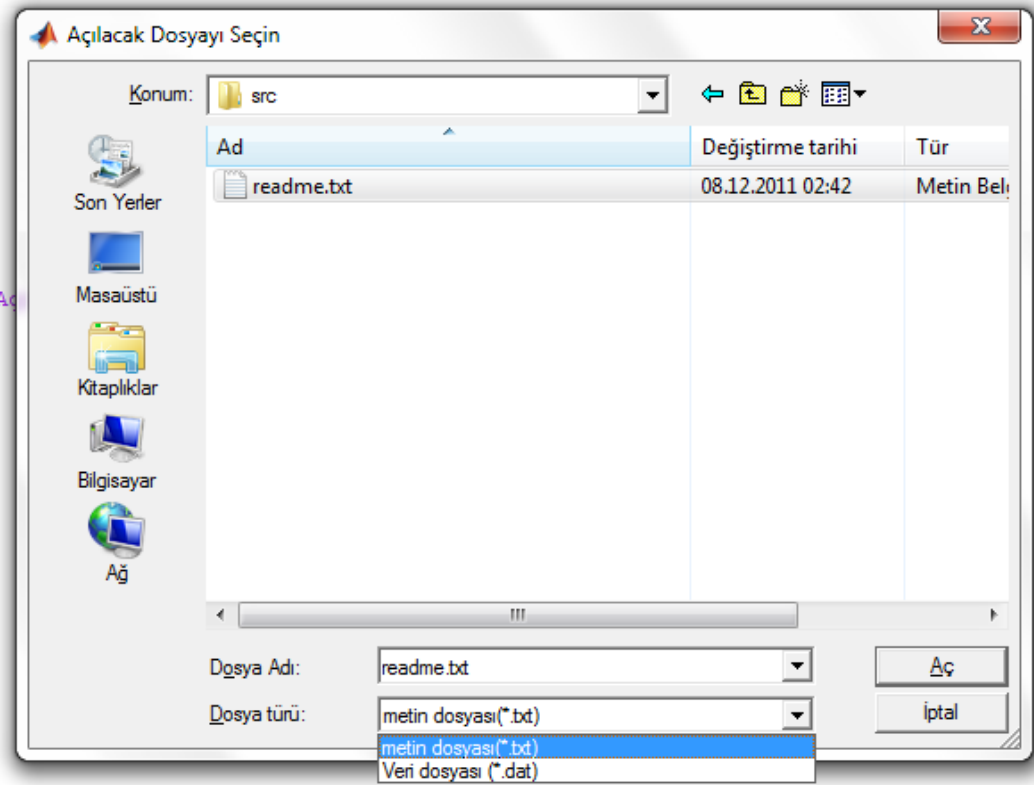
C:\Users\Kurabiye\Documents\Matlab\Ornek\src\

filtre_indisi =

1

```

Şekil 2.53. uigetfile Fonksiyonu



Şekil 2.54. Uigetfile ve uiputfile fonksiyonları kullanımı

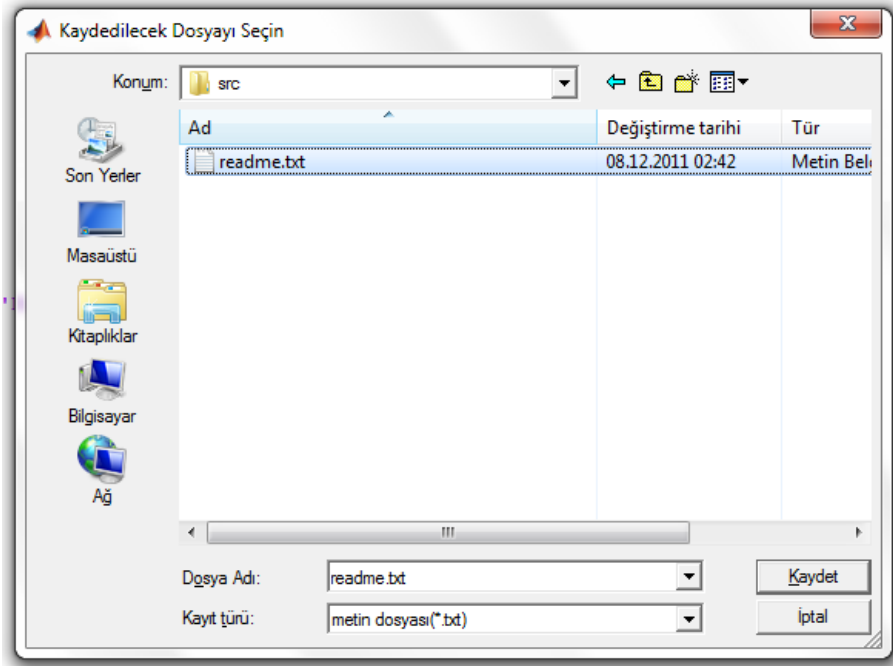
Dosya aç diyalog kutusunda İptal düğmesine basılarak işlem sona erdirilirse, geriye döndürülen dosya adı, dizin ve filtre indisi değerlerinin hepsi 0'a eşit olacaktır.

Aynı şekilde, yazdığımız bir program içerisinde bir bilginin dosyaya kaydedilmesi gereksin. Bu işlem için kullanıcıdan bilgiyi kaydetmek istediği, dosyanın adını, dizinini ve uzantısını almamız gereklidir. Bilginin kaydedileceği dosya mutlaka txt ya da dat uzantılı olsun. Bu durumda daha önce oluşturduğumuz dosya uzantı

filtresini kullanarak dosya uzantı filtresini kullanarak dosya kaydet diyalog kutusunu görüntüleyebiliriz.

```
>> filtre = {'*.txt','metin dosyası (*.txt)','*.dat','Veri dosyası (*.dat)'}  
  
>> [dosya, dizin, filtre_indisi] = uiputfile(filtre,'Kaydedilecek Dosyayı Seçin')  
dosya =  
  
readmse.txt  
  
dizin =  
  
C:\Users\Kurabiye\Documents\Matlab\Ornek\src\  
  
filtre_indisi =  
  
1
```

Şekil 2.55. uiputfile fonksiyonu



Şekil 2.56. Uigetfile ve uiputfile fonksiyonları kullanımı

2.3.9. Menü fonksiyonu

Liste diyalog kutusuna çok benzer. Ancak kullanıcıya sunulacak olan seçenekler liste kutusu içerisinde listelenmek yerine düğmeler üzerine yerleştirilirler. Geriye de

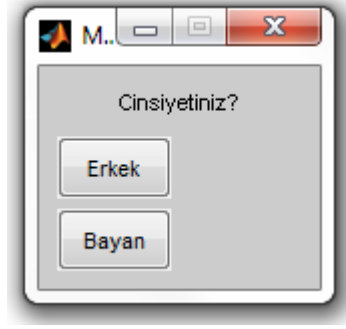
kullanıcının tıkladığı düğmenin numarası döndürülür. Fonksiyonun kullanımı şu şekildedir:

Seçenek = menu(SoruMetni, DüğmeMetni1, DüğmeMetni2, ...)

Örneğin kullanıcının cinsiyetini öğrenmek isteyelim, Bunu, menü fonksiyonu ile Şekil 2.57'deki gibi gerçekleştirebiliriz [1].

```
>>cinsiyet = menu('Cinsiyetiniz?','Erkek','Bayan')  
  
cinsiyet =  
  
1
```

Şekil 2.57. Menü Fonksiyonu

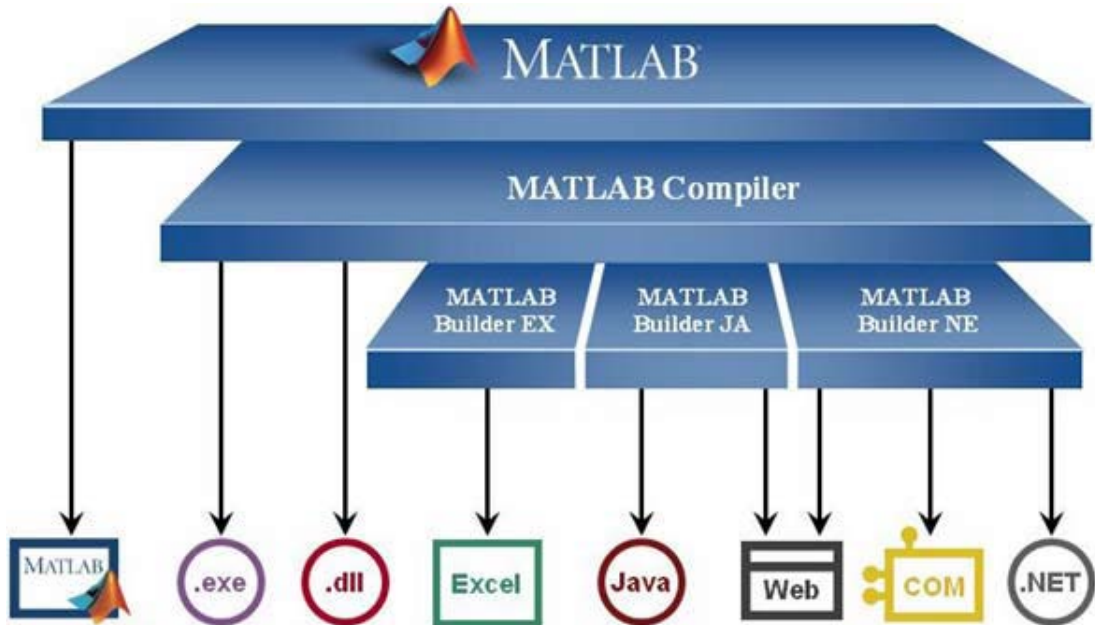


Şekil 2.58. Menü görünümü

3. MATLAB DERLEYİCİSİ

Bir programlama dili ile yazılan bir programın, hazırlanan ortamdan bağımsız olarak çalıştırılması için, yapılan işleme derleme (compile) denir. Böylece ilgili dilde yazılan komut ve deyimler, işletim sisteminin yorumlayıp çalışmasını sağlayan komutlar topluluğu haline gelir ki, bu dosyaların genel olarak uzantısı exe (executable) olur. Aynı mantıkla, Matlab dosyalarını da derleyerek, matlab ortamı dışında çalıştırılabilir dosyalar biçimine getirmek için derlemek gerekir. Ancak matlab m fonksiyon dosyaları derlenebilir, m dosyaları derlenemezler[8].

Yazdığımız kodları C yada C++ 'a entegre etmek istiyorsak MATLAB Compiler komutlarımızı Shared Library olarak kullanmak için bir arayüz sağlamaktadır. Eğer kodlarımızı diğer diller ile kullanmak istiyorsak, Matlab Builder kodlarımızı yazılım bileşeni olarak paketlememizi sağlar. Bu paketlerimizi Java sınıfları, .NET bileşenleri ya da Microsoft Excel eklentisi olarak kullanabiliriz[9]. Tablo 3.1. ve Şekil 3.1.'de Matlab Derleyicisi Yetileri verilmiştir



Şekil 3.1. Matlab Derleyicisi Yetileri

Tablo 3.1. Matlab Derleyicisi Yetileri

Ürün	Hedef	Standalone?	Kütüphaneleri?	Fonksiyon	Uygulamalar?	Grafiksel	Uygulamaları?	Web	Web Şekilleri?
MATLAB Compiler	C ve C++ Standalone dosyaları.	Evet	Evet	Evet	Hayır	Hayır	Hayır	Hayır	
MATLAB Builder NE	C# .NET bileşenleri Visual Basic COM bileşenleri	Hayır	Evet	Evet	Evet	Evet	Evet	Evet	
MATLAB Builder JA	Java bileşenleri	Hayır	Evet	Evet	Evet	Evet	Evet	Evet	
MATLAB Builder EX	Microsoft Excel eklentileri	Hayır	Evet	Evet	Hayır	Hayır	Hayır	Hayır	

3.1. Neden Derleme?

MATLAB'da gerçekleştirilen derleme sonucunda uygulama;

- MATLAB'a ihtiyaç duymadan herhangi bir bilgisayar (derleyicinin desteklediği) üzerinde çalışabilir.
- Geliştirildiği platformdan bağımsız olduğu için yazılım maliyetlerini azaltılmasına,
- Dağıtılarak, çoklu kullanım imkanı sunulmasına,
- Uygulamanın işletim sistemi uyumluluğunun denetlenmesine, olanak sağlar[11].

3.2. MATLAB Derleyicisinin Yetileri

Matlab derleyicisi ile ařađıda belirtilen ıkıřlar retebilir.

- Stand-alone (yrtlebilir, platform bađımsız) uygulamalar.
- C ve C++ paylařım ktphaneleri: Diđer yazılımlarda kullanılmak zere dinamik bađlantılı ktphaneler (Dynamic Linked Libraries - DLL).
- Excel eklentileri: Bu iřlem iin "MATLAB Builder for Excel" ara kutusu gerekir.
- COM objeleri: Bu iřlem iin "MATLAB Builder for COM" ara kutusu gerekir[11].

3.3. MATLAB Compiler Kurulumu ve Mbuild

Matlabın 32Bit versiyonlarında 64Bit versiyonlarından farklı olarak Matlab'la birlikte gelen lcc C derleyicisi bulunmaktadır. Ancak Matlab 64Bit versiyonlarında herhangi bir compiler seeneđi bulunmamaktadır. Bu yzden Compiler yklemeden nce mathworks.com (Tablo 3.2.) dan 64Bit iin desteklenen derleyiciler kısmındaki tablodan uygun olanı ya da olanları indirilmelidir.

Tablo 3.2. Matlab Derleyicileri

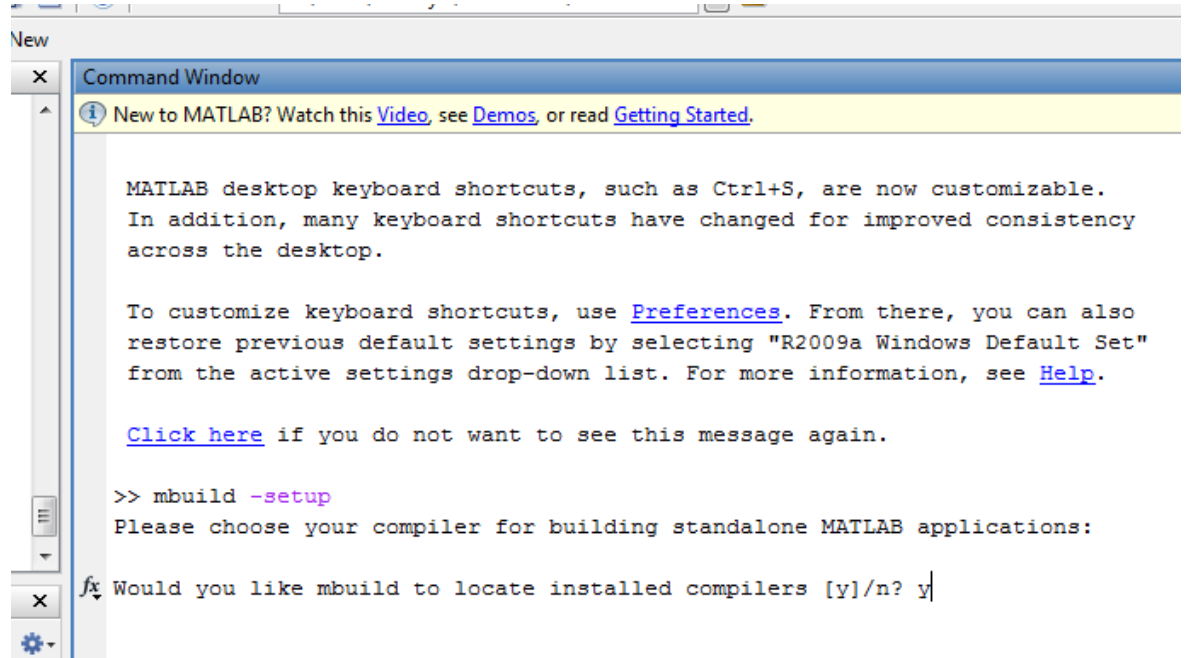
Derleyici	MATLAB Compiler	MATLAB Builder EX	MATLAB Builder NE	MATLAB Builder JA
Microsoft Windows SDK 7.1	√	√	√	
Microsoft Visual C++ 2010 Express with Microsoft Windows SDK 7.1	√	√	√	
Microsoft Visual C++ 2010 Professional	√	√	√	
Microsoft Visual C++ 2008 Professional SP1 and Windows SDK 6.1	√	√	√	
Intel C++ Composer XE 2011	√			
Intel C++ Compiler 11.1	√			
Intel Visual Fortran Composer XE 2011	√			
Intel Visual Fortran 11.1	√			
Microsoft .NET Framework SDK 2.0, 3.0, 3.5			√	
Java Development Kit (JDK) 1.6				√

Mbuild Matlab Compiler tarafından üretilen paylaşılan kütüphaneleri derler ve birbirine bağlar.

Mbuild komutu ile aşağıdakiler için option dosyası oluşturulabilir:

1. Desteklenen her bir derleyici için, geçerli derleyiciyi ve bağlantı seçeneklerini ayarlamak,
2. Derleyiciyi ya da derleyici seçeneklerini değiştirmemize olanak sağlamak,
3. Uygulamamızı derleme imkanı sağlamak [9].

Eğer daha önce deploytool ya da mcc komutu kullanılmadıysa, Matlab derleme için kullanacağı derleyiciyi belirlemenizi ister. Bunun için komut satırına mbuild -setup yazmak gerekir. Yazdıktan sonra Matlab Şekil 3.2.'de görüldüğü gibi "Yüklü olan derleyiciler listelensin mi?" sorusunu sorar.



Şekil 3.2. mbuild –setup komutu

Bu soruya y cevabını verdikten sonra mbuild bilgisayarda yüklü olan derleyicilerin listesini çıkarır. Bu listeden uygun seçeneğin numarasını yazdığımız takdirde Şekil 3.3.'de görüldüğü gibi bizden onaylamak istediğimizi sorar bu soruya da y cevabı verilerek compiler yükleme işlemi bitirilir.

```
>> mbuild -setup
Please choose your compiler for building standalone MATLAB applications:

Would you like mbuild to locate installed compilers [y]/n? y

Select a compiler:
[1] Microsoft Visual C++ 2008 Express in C:\Program Files (x86)\Microsoft Visual Studio 9.0
[0] None

Compiler: 1

Please verify your choices:

Compiler: Microsoft Visual C++ 2008 Express
Location: C:\Program Files (x86)\Microsoft Visual Studio 9.0

Are these correct [y]/n?
```

Şekil 3.3. mbuild –setup komutu

3.4. M-Fonksiyon Gerekliği

- MATLAB Compiler, ancak başlangıç ibaresi olarak function belirteci içeren M-Dosya'ları (dolayısıyla M-Fonksiyon'ları) derler. Bundan da anlaşılacağı gibi, MATLAB Compiler ile M-Dosya'lar derlenemez. MATLAB ile yapılan çalışmalarda, M-Fonksiyon hazırlamaya yatkınlık çok önemlidir.
- M-Dosya'ların derlenebilmeleri için, öncelikle M-Fonksiyon'lara dönüştürülmeleri gerekir[11].

3.4.1. M-dosya m-fonksiyon dönüşümü

M-Dosya:

```
% Ortalama Programı
```

```
S1=input('İlk sayıyı giriniz: ');
```

```
S2=input('İkinci sayıyı giriniz: ');
```

```
S3=input('Üçüncü sayıyı giriniz: ');
```

```
ortalama=(S1+S2+S3)/3;
```

```
ortalama
```

M-Fonksiyon:

```
function sonuc = ortalama
```

```
% Ortalama Programı
```

```
    S1=input('İlk sayıyı giriniz: ');
```

```
    S2=input('İkinci sayıyı giriniz: ');
```

```
    S3=input('Üçüncü sayıyı giriniz: ');
```

```
    ort =(S1+S2+S3)/3;
```

```
    sonuc = ort
```

Şekil 3.4. M-Dosya M-Fonksiyon Dönüşümü

3.5. Derleme Adımları

- Bağımlılık Analizi
- Kod Oluşturma
- Arşiv Yaratma
- Derleme
- Birleştirme

3.5.1. Bağımlılık analizi

İlk Aşama programa bağlı olan M-Dosyaları, Mex dosyaları ve P dosyalarındaki tüm fonksiyonlara karar vermektir. Bu liste verilen dosyaların çağırdığı M-dosyaları çağırdığı gibi bu dosyaların çağırdıklarını da çağırır. Ayrıca tüm yapı fonksiyonları ve MATLAB nesnelere de içerir.

Hazırlanan MATLAB uygulamalarının birden çok M-Fonksiyon içerdiği durumlar söz konusu olabilir. Örneğin; bir işlem fonksiyonu; topla ve carp fonksiyonları gibi farklı fonksiyonları çağırıp işlem gerçekleştiriyor olabilir[12].

islem.m:	topla.m
-----	-----
function s=islem	function s1=topla(a,b);
a=2;	S1 =a+b;
b=3;	
s1=topla(a,b);	carp.m:
s2=carp(a,b);	-----
load veriler.mat	function s2=carp(a,b);
s=s1/s2+c;	S2 =a*b;
disp(s);	

Şekil 3.5. Bağımlılık Analizi

Şekil 3.5'deki gibi uygulamaları derlemek için tüm fonksiyonları çalıştıran ana M-Fonksiyonun (islem.m) derlenmesi yeterlidir. islem M-Fonksiyonu, topla ve carp M-Fonksiyon'larını çalıştırdığı için bu M-Fonksiyonlara bağımlıdır. Farklı bir deyişle, bağımlılık içeren M-Fonksiyon'un derlenmesi yeterlidir.

Bağımlılık kontrollerinin denetlenmesi için depfun MATLAB fonksiyonundan yararlanır. depfun fonksiyonu, belirtilen fonksiyon çalıştırılırken hangi fonksiyonlara ihtiyaç duyduğunu bir rapor halinde sunar [11].

```
T=depfun ( 'islem')  
  
'C:\Uygulama\islem.p'  
'C:\MATLAB\toolbox\matlab\datafun\@double\prod.bi'  
'C:\MATLAB\toolbox\matlab\datafun\@single\prod.bi'  
'C:\MATLAB\toolbox\matlab\datafun\prod.m'  
'C:\MATLAB\toolbox\matlab\datatypes\@opaque\char.m'  
'C:\MATLAB\toolbox\matlab\datatypes\@opaque\double.m'  
'C:\MATLAB\toolbox\matlab\datatypes\cell.bi'  
'C:\MATLAB\toolbox\matlab\strfun\strcmp.m:  
'C:\MATLAB\toolbox\matlab\strfun\strncmp.m'  
'C:\MATLAB\toolbox\matlab\strfun\strrep.bi'  
'C:\MATLAB\toolbox\matlab\timefun\datenumrnx.dll'  
'C:\Uygulama\carp.m'  
'C:\Uygulama\topla.m'
```

Şekil 3.6. Bağımlılık Analizi Çıktısı

(Verilen liste daha uzundur.)

Bu listenin sonunda islem M-Fonksiyonunun, carp ve topla M-Fonksiyon'larına bağlı olduğu görülür.

Bu çok dosyalı uygulamayı derlemek için;

```
>>mcc -m islem
```

Şekil 3.7. mcc kodu

komutunu kullanmak yeterlidir. mcc komutu her derlenecek fonksiyon için depfun fonksiyonunu otomatik olarak arkaplanda çalıştırır.

3.5.2. Kod oluşturma

Bu basamak aşağıdaki maddelerdeki hedef parçaları oluşturmak için ihtiyaç duyulan tüm kaynak kodları üretmek için kullanılır.

- C/C++ arayüzü kodu, komut satırı (portfoliotool_main.c)'daki M-Fonksiyonlar için kullanılır. Kütüphaneler ve bileşenler için, bu dosya üretilen tüm arayüzleri içerir.
- M-kodunun run-timedda çalıştırma bilgilerini içeren bir bileşen veri dosyası gereklidir. Bu veri dosyası, yol bilgisi ve bileşenlerin CTF arşivinde depolanan M-kodunu yüklemek için gerekli kriptolama bilgilerini içerir.

3.5.3. Arşiv yaratma

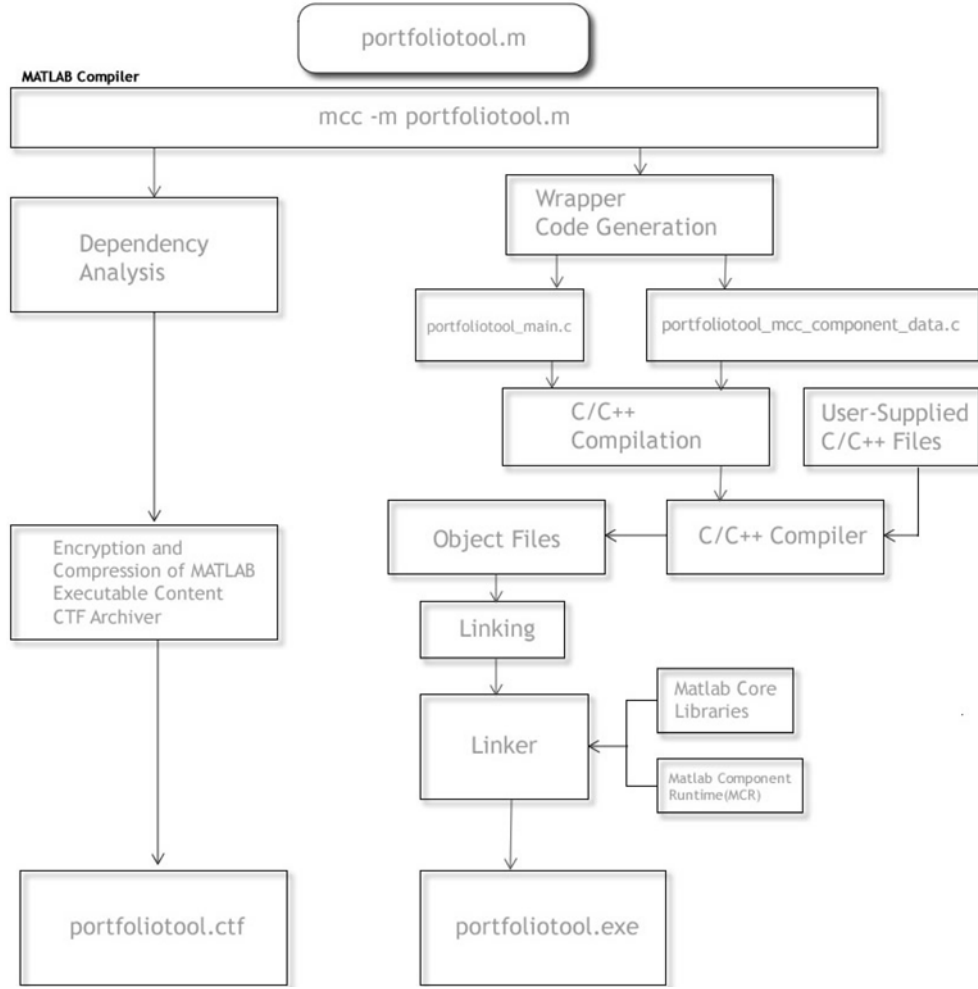
Bağımlılık analizi sırasında üretilen matlab dosyalarının(M-Files ve MEX Files) listesi, run-time sırasında eksiksiz olarak çalıştırılan bileşenlere ihtiyaç duyan bir CTF arşivi oluşturmak için kullanılır. Dosyalar şifreli ve dağıtım için tek bir dosya içerisine sıkıştırılmıştır. Bu dosyada izin bilgisi de bulunmaktadır böylece içerik hedef makineye eksiksiz bir biçimde yüklenebilir.

3.5.4. Derleme

Bu basamak sarıcı kod üreticisinden üretilen C/C++ dosyalarını nesne koduna derler.-mcc komut satırında kullanıcı tarafından sağlanan C/C++ kodu içerenleri destekleyen hedefler için bu kod yine bu basamakta derlenir.

3.5.5. Birleştirme

Son aşama üretilen nesne dosyalarını ilgili MATLAB kütüphaneleriyle birleştirerek ürünün bitmiş hali elde edilir. C/C++ derlemesi ve birleştirme adımları MATLAB derleyicisinde bulunan mbuildi kullanır.



Şekil 3.8. Derleme Adımları

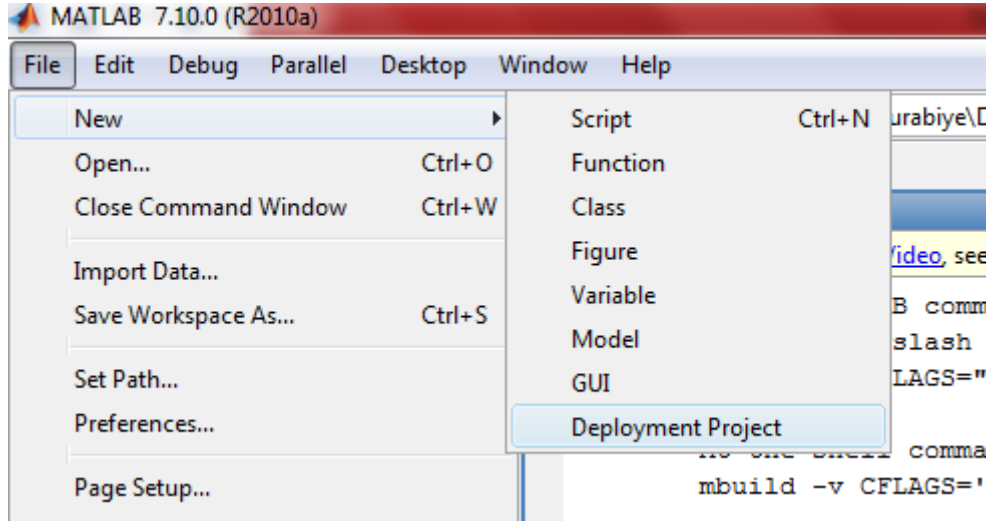
3.6. Matlab Compiler Kullanımı

Matlab Compiler ürününü 2 şekilde kullanabiliriz;

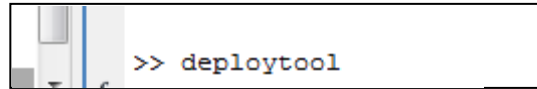
1. Deployment Tool
2. mcc komutu ile

3.6.1. Deployment tool ile derlemek

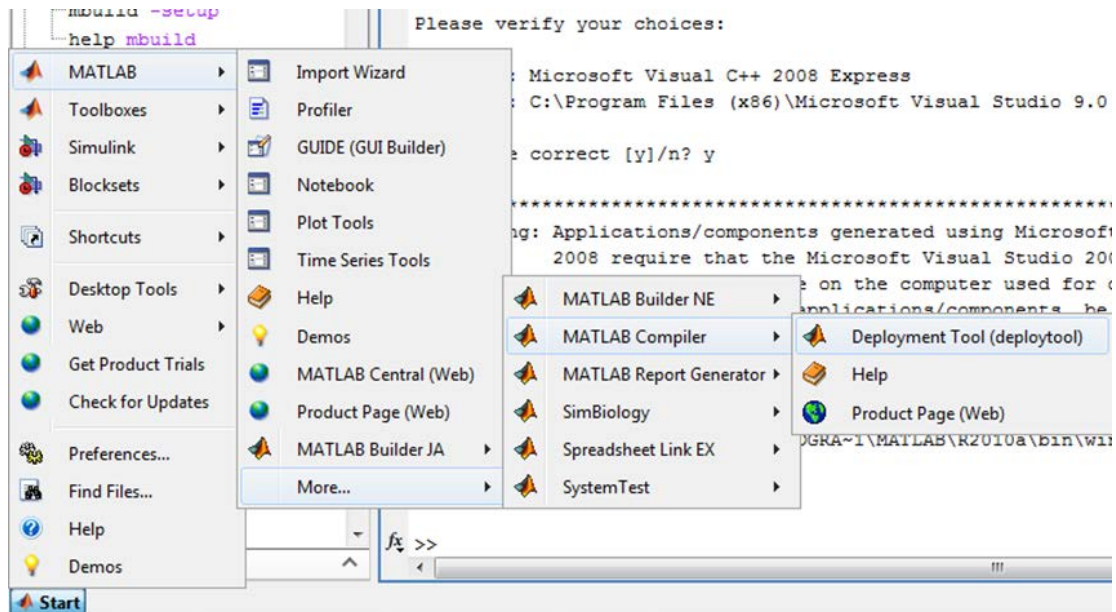
Deployment tool grafik arabirimine istersek File-New-Deployment Project(Şekil 3.9.) ile istersek komut satırına deploytool yazarak (Şekil 3.10.) istersek de Matlab'ın sol altta bulunan Başlat sekmesinden(Şekil 3.11.) ulaşabiliriz.



Şekil 3.9. File->New->Deployment Project

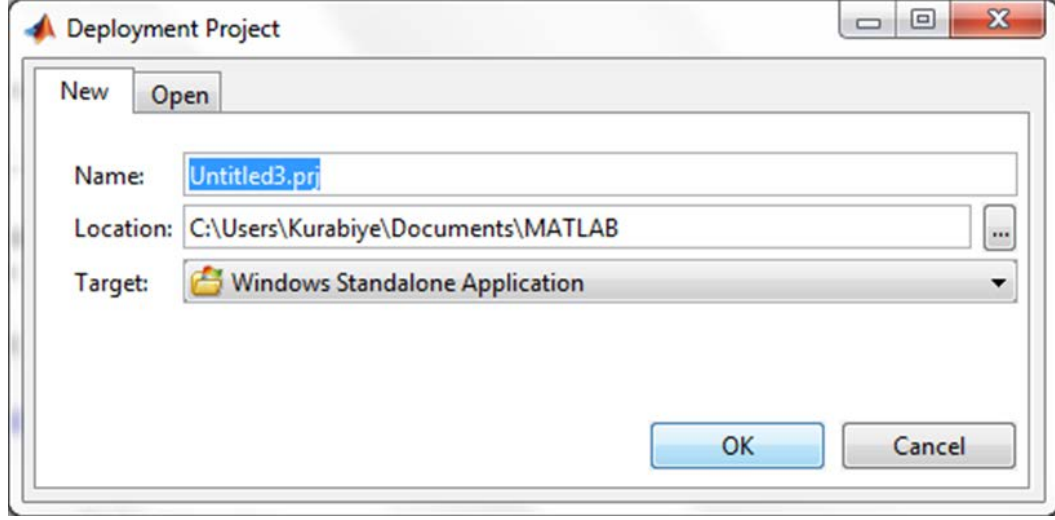


Şekil 3.10. komut satırını kullanarak



Şekil 3.11. Başlat sekmesi ile ulaşmak.

Deployment Tool'u ilk çalıştırdığımızda Şekil 3.12'deki gibi bir pencere karşınıza gelecektir.

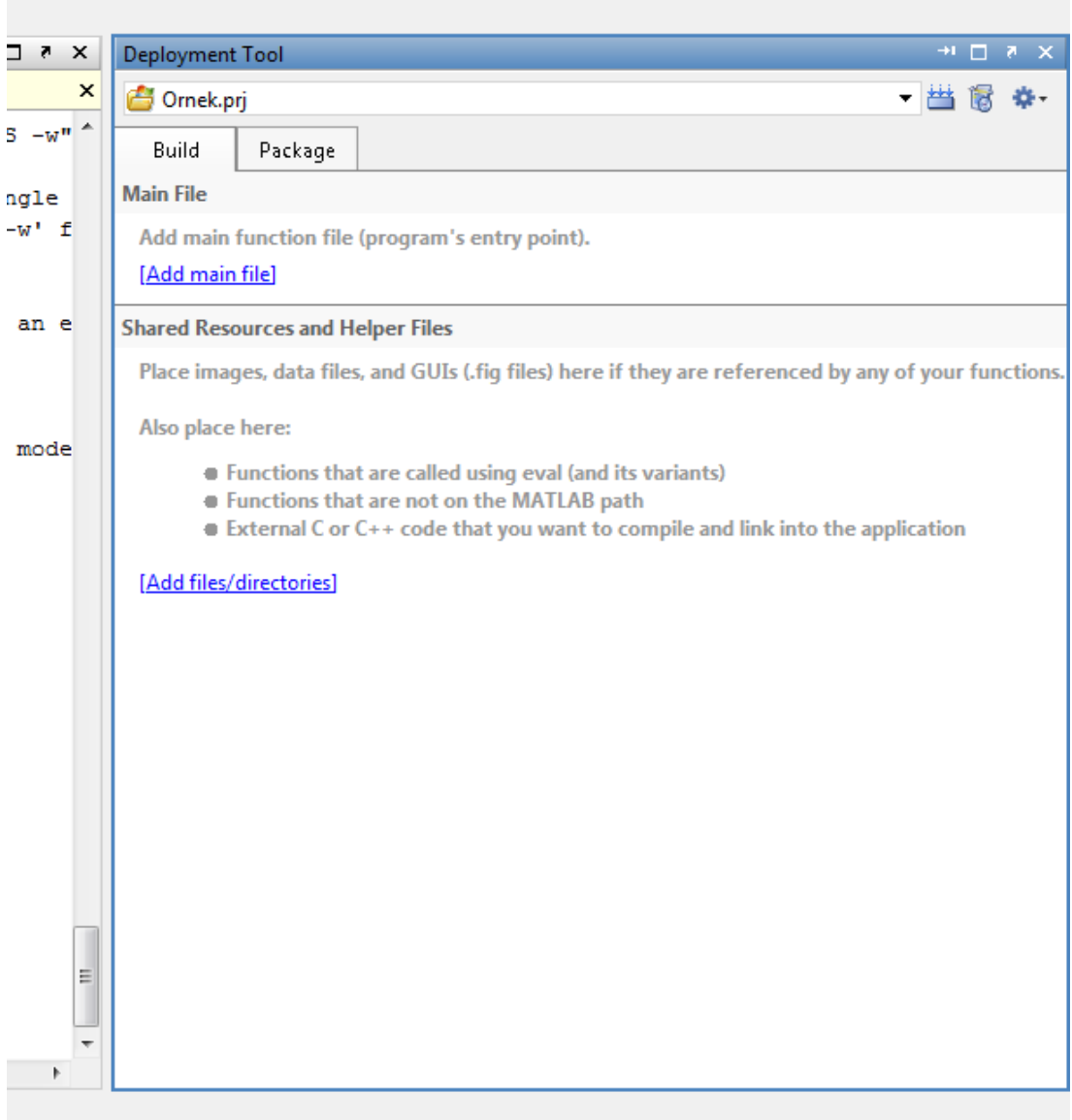


Şekil 3.12. Deployment Project Giriş Penceresi

Burada;

- Name alanına projemizin ismini,
- Location alanına hangi klasörde oluşturulmasını istediğimizi,
- Target alanındaki drop-down menüden ise projeyi hangi alanda kullanacağımızı seçeriz.

OK tıklandığında tekrar matlab penceresine dönülür ve ekranın sağ tarafında beliren Deployment Tool penceresinden (Şekil 3.13.) işleme devam edilir.

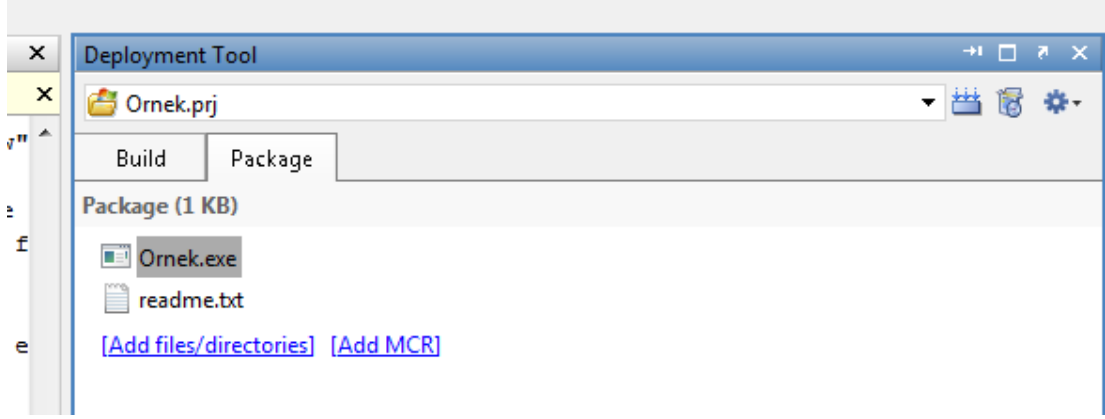


Şekil 3.13. Deployment Tool penceresi

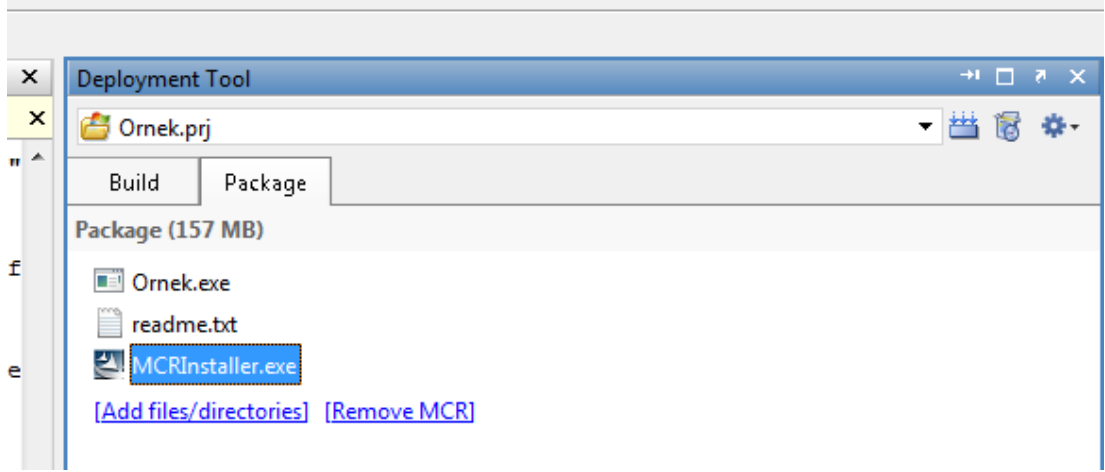
Burada derlemek istediğimiz m-file seçilir. Eğer programımızda yardımcı dosyalarda kullandıysak bu dosyaları da Shared Resources and Helper Files kısmından eklememiz gerekmektedir. Dosya ekleme işlemlerimiz bittikten sonra Build düğmesine tıklayarak işlemimizi sonlandırırız.

3.6.1.1. Paketlemek (isteğe bağlı)

Programı standalone hale getirdikten sonra isteğe bağlı olarak paketleyebiliriz. Paketlemekteki amaç gerekli bütün programları bir balya haline getirerek kullanıcıya ulaştırmaktır (Şekil 3.14). Ayrıca Add MCR seçeneğini işaretleyerek MCR' yide balyamıza ekleyebilme imkanı sunar (Şekil 3.15.)[9].



Şekil 3.14. Paketleme (Package)



Şekil 3.15. Paketleme (Package) MCR ekli

İşlemleri tamamladıktan sonra Package düğmesi tıklanır. Açılan Kaydetme penceresinde ise isteğe bağlı olarak *.exe ya da *.zip formatı seçilebilir.

3.6.2. Mcc komutu ile derlemek

mcc Matlab Compilery tetikleyen bir matlab komutudur. Komut satırında kullanımı aşağıdaki gibidir.

mcc –seçenek mdosyaadi Örneğin: mcc –m guney.m

Derlenme sonucunda iki adet dosya yaratılır. Birinci dosya çalıştırılabilir dosyadır, ikinci dosya ise “CTF Component Technology File” dosyasıdır. “CTF” içerisinde sıkıştırılmış ve şifrelenmiş “.m” dosyaları ile “.m” dosyasının çalıştırılabilmesi için gerekli olan diğer dosyalar bulunur. “CTF” şifreleme algoritması olarak “AES” kullanır. Şifrelenmiş dosyalar simetrik iki adet bin yirmi dört bit “RSA” anahtarı ile

korunmaktadır. MATLAB programı “MCR” yüklü bir makinede ilk defa çalıştırıldığında “CTF” dosyası yeni bir klasör içerisine açılır. Böylelikle “.m” dosyası çalışmak için ihtiyaç duyduğu tüm dosyaları bu klasörden çağırarak kullanır[3].

Seçenek kısmına ise Tablo 3.3.’de bulunan ifadeler yazılabilir.

Tablo 3.3. MCC seçenek kodları[9]

Seçenek	Açıklama	Yorum
-a <i>filename</i>	CTF arşivine bir dosya ekler	
-b	Excel uyumlu formül fonksiyonu üretir.	MATLAB Builder EX’e ihtiyaç duyar.
-B filename[:arg[,arg]]	mcc komut satırındaki -B filename ile filename’nin içeriğini yer değiştirir.	Dosya sadece mcc komut satırı seçeneklerini içermelidir.
-c	C wrapper kodunu üretir.	-T codegen komutu ile aynı görevi yerine getirmektedir.
-C	Mcc’yi C/C++’da gömülü olmayan CTF arşivine yönlendirir.	
-d directory	Çıktıyı belirlenmiş dosyaya yönlendirir.	
-e	Standalone bir uygulama üretirken MS-DOS komut penceresinin görünümünü engeller.	MS-Dos komut penceresini deploytool’u kullanarak ta görünmesini engelleyebiliriz.
-g	Hata ayıklama bilgisini üretir.	
-G	-g ile benzerdir.	

Tablo 3.3. MCC seçenek kodları (Devam)

Seçenek	Açıklama	Yorum
-I directory	MATLAB dosyaları için arama yoluna dosya ekler.	Matlab 'tan çalıştırıldığı zaman Matlab yolu otomatik olarak ekli gelir, ancak DOS/UNIX tabanında geçerli değildir.
-K	Mcc'yi herhangi bir hatadan dolayı derleme erken bittiğinde çıkış dosyalarını silmemesi için yönlendirir.	Mcc'nin varsayılan davranışı, komut eğer başarısız olduysa çıkış dosyalarını ortadan kaldırmaktır.
-l	Fonksiyon kütüphanesi oluşturmak için bir makrodur.	-W lib -T link:lib ile aynı işlevi yerine getirmektedir.
-m	Standalone uygulama üreten makrodur.	-W main -T link:exe ile aynı işlevi yerine getirmektedir.
-M string	Mbuilde geçiş karakteri.	Tanımlı derleme zamanı seçeneklerini kullan.
-N	Gerekli dosya yollarını temizler	
-o outputfile	Final çıkış dosyasını isimlendirir.	Uygun uzantıyı ekler.
-p directory	Sıra-duyarlı bir şekilde derleme yoluna dosya ekler.	-N seçeneği gereklidir.
-R <i>option</i>	MCR için run-time seçeneklerini belirler.	option =-nojvm -nodisplay -logfile filename -startmsg -completemsg filename
-S	MCR oluşturur	MATLAB Builder NE gereklidir.
-u	Sadece çalışılan makinedeki kullanıcılar için COM bileşeni kayıtlı eder.	COM bileşenleri ve Microsoft Excel eklentileri için geçerlidir.

Tablo 3.3. MCC seçenek kodları (Devam)

Seçenek	Açıklama	Yorum
-v	Ayrıntı; Derleme adımlarını gösterir.	
-w <i>option</i>	Dikkat mesajlarını görüntüler.	option = list level level: string where level = disable enable error error [off:string on:string]
-W <i>type</i>	Fonksiyon wrapper'larının üretimini kontrol eder.	type = main cpplib:<string> lib:<string> none com:compname,cname,version
-Y licensefile	MATLAB Compiler lisansını teslim etmek için lisans dosyasını kullanır.	
-?	Yardım mesajı görüntüler.	

3.6.2.1. Farklı mcc seçenekleri ile derleme

Dosya ve dizin ekleme seçeneği, DLL oluşturma seçeneği, toolbox oluşturarak dağıtma gibi farklı mcc seçenekleri de bulunmaktadır.

3.6.2.2. Dosya ve dizin ekleme seçeneği

-a seçeneği mcc komutu ile derlenen uygulamalara dosya eklemede kullanılır.

Şekil 3.16.'da verilen işlem.m fonksiyonu;

<pre> c = 3; save veriler.mat c islem.m: ----- function s=islem a=2; b=3; s1=topla(a,b); s2=carp(a,b); load veriler.mat s=s1/s2+c; disp(s); </pre>	<p>şeklinde olduğu varsayalım. Görüleceği gibi, a ve b değişkenleri tanımlanmasına rağmen c değişkeni tanımlanmamıştır. c değişkenin değeri veriler.mat dosyasından gelmektedir. Bu uygulamanın çalışabilmesi için <code>mcc -m islem</code> komutu yeterli değildir. Çünkü <code>depfun</code> fonksiyonu; <code>.mat</code>, <code>.txt</code> vb. gibi uzantılı dosyaların bağımlılıklarını görmez. Derlemeye bu dosyaların katılması gerekir, katılmadığı takdirde; uygulama çalıştırıldığında böyle bir dosya veya dizin yok (No such file or Directory) hatası verir.</p>
--	---

Şekil 3.16. Dosya ve Dizin Ekleme

veriler .mat dosyasının derleme arşivine katılması ve her çalıştırılışında sorunsuz yürütülebilmesi için Şekil 3.17.'deki;

```
» mcc -m islem -a veriler.mat
```

Şekil 3.17. Dosya Ekleme kodu

komutu kullanılır. Her eklenecek dosya için `-a dosya_adi` yapısı kullanılmalıdır. Birçok dosyanın ekleneceği durumlarda `-d` seçeneği ile bir dizin de arşive eklenebilir. Bu durum için girilecek örnek kod Şekil 3.18.,

```
» mcc -m islem -d C:\MATLAB\work\uygulama\alt dizin
```

Şekil 3.18. Dizin Ekleme kodu

şeklindedir. Uygulama farklı bir dosya adı ile derlenmek istenirse komut;

```
» mcc -m islem -o yeni_dosya_adi
```

Şekil 3.19. Farklı bir dosya adı ile derlenmek

şeklinde kullanılarak. `islem.exe` yerine, `yeni_dosya_adi.exe` oluşturulur.

3.6.2.3. DLL oluřturma seeneđi

-l seeneđi ile MATLAB fonksiyonlarından DLL uzantılı ortak paylařım kütüphaneleri oluřturulabilir. Bu DLL dosyaları ile, farklı uygulamaların iine MATLAB komutları gmülebilir.

» mcc -l islem.m

řekil 3.20. DLL Oluřturma

řekil 3.20'deki komut ile islem.dll dosyası oluřturulur. Bu derleme seeneđinin avantajı; örneđin C ile kodlaması geliřtirilen bir uygulamada, kodlamanın C kullanarak zor, fakat MATLAB ile ok daha kolay gerekleřtirilebilir olması durumlarında, DLL desteđi ile bu noktada ekleme yapılabilir.

Ayrıca dll dosyaları, Active Server Pages (ASP) gibi web uygulamalarında da MATLAB uygulamalarını kullanma imkanı sunar. Ötesinde bir ok farklı programlama dili ile etkileřim imkanı sunar.

3.6.3. Matlab ve java

MATLAB tarafından derlenen uygulamalar C kodlarına da evrilir. Bu C kodlarını Java Native Interface (JNI) ile Java'ya aktarmak mümkündür. Bu sayede Java programlama dilinde, MATLAB uygulamaları kullanılabilir. Böylelikle MATLAB'ın yetenekleri ok fazla geniřletilebilir. Örneđin, Java Applet'ler kullanım gibi[11].

3.6.4. Alternatif bir dađıtma yöntemi: toolbox oluřturarak dađıtma

M-Dosya ve M-Fonksiyon'lar bir araya getirilerek derleme sonucu paket programlar oluřturulabileceđi gibi, MATLAB iinde bu dosya grupları bir Ara Kutusu (Toolbox) olarak da tanımlanabilir. Bir ara kutusu oluřturulurken, birbirleri ile iliřki Fonksiyonlar bir araya getirilir ve bir dizin altında toplanır. Bu dizin bir sonraki adım olarak, MATLAB Dizin Yolu'na (Set Path) eklenir.

MATLAB Veri Yolu'na eklenen bu dizin iin MATLAB Bařlat Düđmesinde Toolbox seeneđi altında bir ara kutusu menüsü aılması istenirse, uygulamanın klasörü iinde bir XML dosyası tanımlamak gerekir. XML bir meta dilidir ve bilgiyi tanımlayan bir yapıya sahiptir. řekil 3.21'de gösterilen bu XML dosyası ile

MATLAB Başlat Düğmesi'nde oluşturulmak istenen seçeneklerin bilgileri tanımlanacaktır.

```
1 <productinfo
2   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="http://www.mathworks.com/namespace/info/v1/info.xsd">
4 <?xml-stylesheet type="text/xsl" href="http://www.mathworks.com/namespace/info/v1/info.xsl"?>
5
6
7 <matlabrelease>14</matlabrelease>
8 <name>Araç Kutum</name>
9 <type>toolbox</type>
10 <icon>ApplicationIcon.MATLAB</icon>
11 <help_location>C:/guney/index.html</help_location>
12
13 <list>
14
15 <listitem>
16 <label>Uygulamam</label>
17 <callback>islem.m</callback>
18 <icon>ApplicationIcon.GENERIC_GUI</icon>
19 </listitem>
20
21 <listitem>
22 <label>Web Sitesi</label>
23 <callback>web http://www.google.com -browser;</callback>
24 <icon>ApplicationIcon.PROFILER</icon>
25 </listitem>
26
27 </list>
28
29 </productinfo>
```

Şekil 3.21. info.xml kodları

Verilen kodlamada ilk 4 satır ve son satır değiştirilmemelidir. Bu satırlar arasına başlat menüsüne eklenecek araç kutusunun bilgileri girilir:

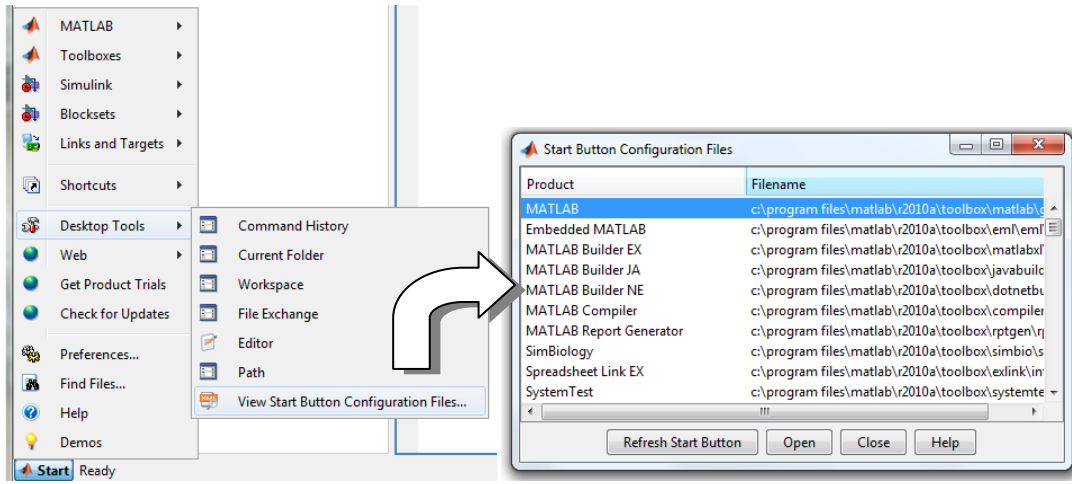
-matlabrelease- etiketleri arasına, uygulamanın hazırlandığı MATLAB sürüm numarası girilmelidir.

- -name- etiketleri arasına, araç kutusunun adı girilmelidir.
- -type- etiketleri arasına, başlat menüsüne eklenecek uygulamanın; bir araç kutusu (toolbox) veya blok seti (blockset) veya MATLAB fonksiyonu (matlab) veya simulink modeli (simulink) seçeneklerinden hangisi olduğu belirtilmelidir. Bu belirtim sonucu başlat menüsünde, belirtilen konumda tanımlama oluşur.
- -icon- etiketleri arasına, başlat menüsünde araç kutusunun yanında hangi resmin yer alacağı belirtilmelidir.
- -help location- etiketleri arasına, uygulamaya ait yardım dosyalarının (HTML formu) konumu belirtilir. Bu işlem helpdesk'ten yardım dosyaları görüntülenmesi için gereklidir.

Bu kısımdan sonrası, list etiketleri arası, menü seçeneklerinin listelendiği kısımdır.

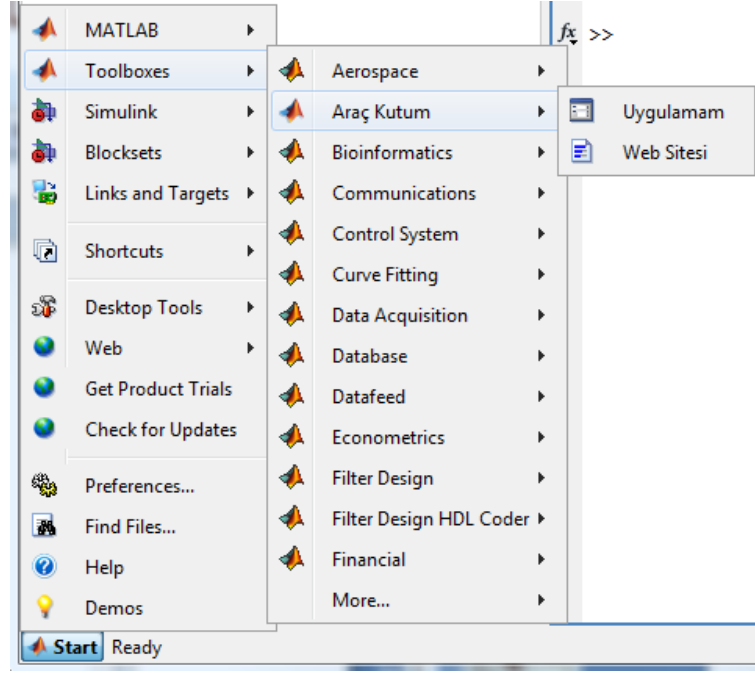
- -listitem- etiketleri arasına seçenek bilgileri gelir.
- -label- etiketleri arasına menü seçeneğinin adı gelir.
- -callback- etiketleri arasına menü seçeneğine tıklanması halinde yürütülecek işlevler gelir.
- -icon- etiketleri arasına bu seçeneğin yanındaki resim tanımlanır. \$toolbox, MATLAB'ın kurulu olduğu dizindeki toolbox klasörünü temsil eder.

Bu info.xml dosyası, başlat düğmesine tıklanarak Şekil 3.22'de gösterilen yoldan tanımlanır.



Şekil 3.22. Start düğmesi Ayarları

MATLAB Dizin Yolu'na eklenen info.xml dosyanın tanımlanması için "Refresh Start Button" düğmesine tıklanır. Bu seçim sonucu, listeye Şekil 3.23'deki gibi "Araç Kutum" gelir.



Şekil 3.23. Başlat menüsünde oluşturulan kişisel araç kutusu

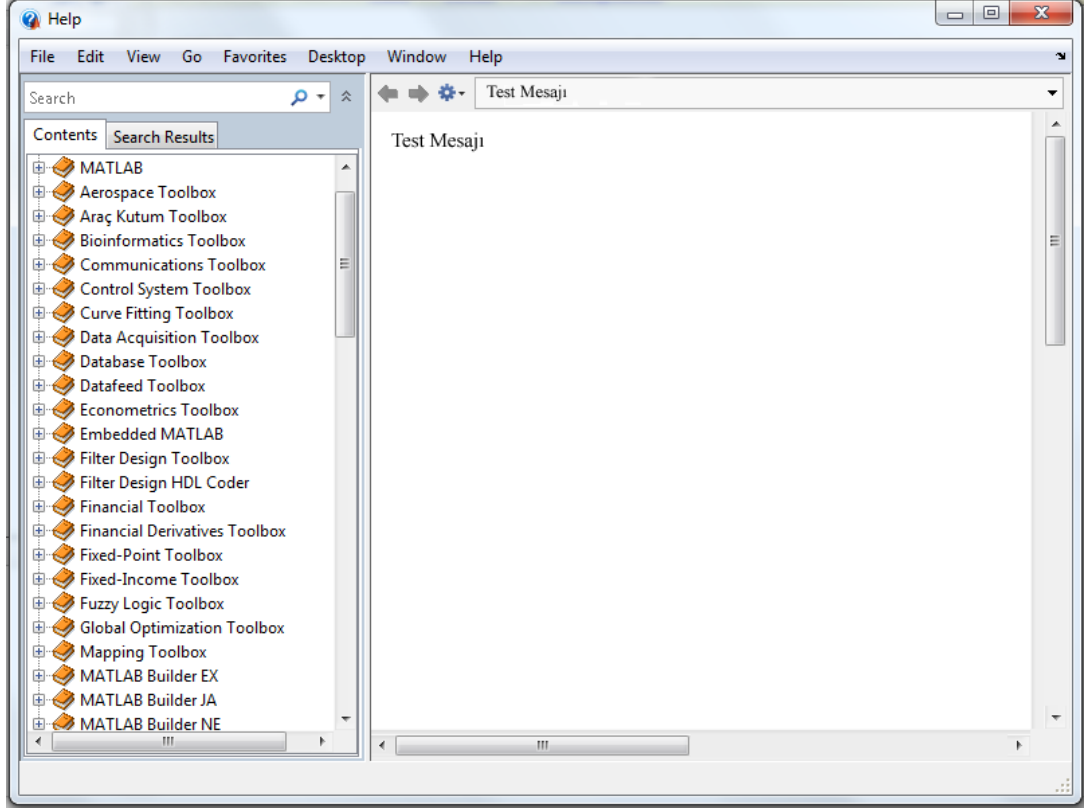
(Menünün adının "Araç Kutum" olacağı, iki alt seçeneğinin olacağı ve Toolboxes ana menüsün altına geleceği XML içinde belirtilmiştir).

Ayrıca, Yardım Masası'nda (helpdesk) uygulamaya ait bir yardım dosyası görüntülenmesi istenirse, XML dosyasında tanımlanan HTML dosyası oluşturulmalıdır. Örnek bir index.htm dosyası Şekil 3.24'de verilmiştir:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
    <title>Test Mesajı</title>
  </head>
  <body>
    Test Mesajı.
  </body>
</html>
```

Şekil 3.24. HTML dosyası

helpdesk açılarak, Araç Kutum adlı Toolbox'a tıklanırsa, oluşturulan HTML sayfası Şekil 3.25'teki gibi görüntülenir.



Şekil 3.25. HTML sayfası test mesajı

Özetlenecek olursa, M-Dosya ve/veya M-Fonksiyonların bir araya getirildiği uygulama klasörünün içeriği aşağıda örneklenmiştir:

Kok Dizin C:

|-----> Uygulama

|-----> info.xml

|-----> index.htm

|-----> uygulama adi.m

|-----> Varsa diğer M-Dosya veya M-Fonksiyonlar [11]

3.7. P-Code Derleme

P-Code kısaltması, psuedocode'dan gelmektedir. P-Code dosyaları, MATLAB programında hazırlanan M-Fonksiyonların pcode komutu ile derlenmesi sonucu oluşur. P-Code olarak derlenen

M-Fonksiyonların içeriği MATLAB Editörü veya MS Note Pad gibi içerik görüntüleyici programlar ile görüntülenemez. Çünkü P-Code olarak derlenen M-Fonksiyon'ların içeriği şifrelenerek gizlenmektedir.

Bu gizleme işlemi P-Code dosyalarının yapısından kaynaklanmaktadır. P-Code dosyaları, içerdiği komutları bir algoritma şemasına dönüştürerek, bu Dosyalara iletilen değerleri algoritma şemasında işleyerek sonuç üretir.

Örneğin Şekil 3.26'daki M-Fonksiyon dosyası ele alınsın;

```
ort.m:  
function sonuc = ort(x1,x2,x3)  
toplam = x1 + x2 + x3;  
sonuc = toplam/3  
Bu uygulama için komut satırından;  
» pcode ort
```

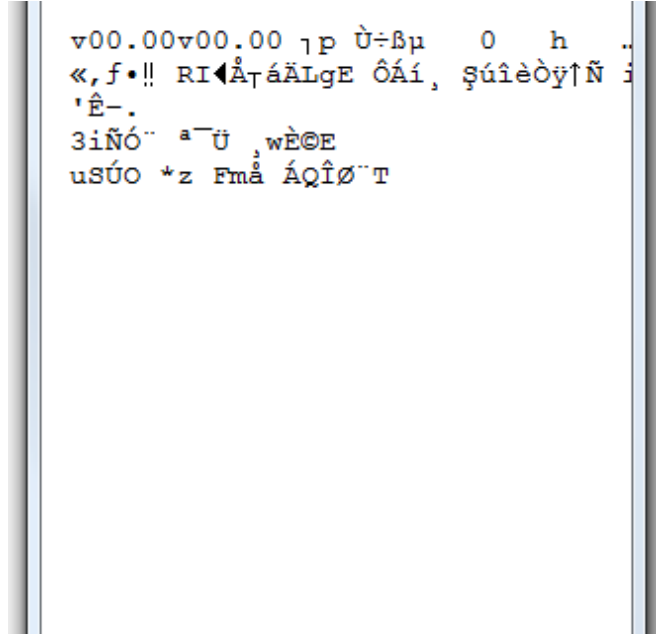
Şekil 3.26. P-Code Derleme

komutu ile bir P-Code dosyası üretilir. Üretilen P-Code dosyasının adı ort.p'dir. Bu dosya MS Note Pad ile görüntülenmek istendiğinde Şekil 3.28' deki gibi görünür. P-Code dosyaları orijinal M-Fonksiyon dosyasına ihtiyaç duymadan çalışır. Çalışma dizinindeki (work) ort.m dosyası silinip, komut satırına Şekil 3.27.'deki;

```
» ort(1,2,3)
```

Şekil 3.27. P-Code

girildiğinde, sonuc=2çıkışı ile karşılaşılır. P-Code dosyalarının dağıtımı, M-Fonksiyon dosyalarının içeriğinin görüntülenmemesi arzulandığı durumlarda sıklıkla kullanılır [11].



Şekil 3.28. P-Code görünümü

3.8. MATLAB “MEX” Derleme Komutu

MATLAB derleyicisinde bulunan mex komutu; C veya başka dillerde yazılmış fonksiyon dosyalarının, MATLAB altında hali hazırda bulunan bir fonksiyonmuş gibi çalıştırılmasını sağlar. Bu işlem DLL ile farklı ortamlara MATLAB fonksiyonu aktarmanın tam tersidir.

Herhangi bir C veya Fortran fonksiyonu, MATLAB fonksiyonu gibi kullanılabilir. Bunun için mex derleyicisinin ayarlanması gerekmektedir. Ayarlama için komut satırın Şekil 3.29’daki gibi;

```
»mex -setup
```

Şekil 3.29. Mex Ayarları

yazılır. mex derleyicisi ayarları, derlenecek fonksiyonun kodlama diline göre seçilir. Örneğin; C dili ile kodlanan bir uygulama için Lcc veya Microsoft Visual C/C++ seçilebilir.

Uygulama:

yprime adlı C ve Fortran tabanlı uygulama, MATLAB içinde kullanılabilir hale getirilmek isteniyor. Bu uygulama MATLAB dizininde extern\examples\mex klasörü altında yer almaktadır.

```
yprime.c
/*=====
*
* YPRIME.C Sample .MEX file corresponding to YPRIME.M
*   Solves simple 3 body orbit problem
*
* The calling syntax is:
*
*   [yp] = yprime(t, y)
*
* You may also want to look at the corresponding M-code, yprime.m.
*
* This is a MEX-file for MATLAB.
* Copyright 1984-2006 The MathWorks, Inc.
*
*=====*/

/* $Revision: 1.10.6.4 $ */

#include<math.h>
#include "mex.h"

/* Input Arguments */

#define T_IN  prhs[0]
#define Y_IN  prhs[1]

/* Output Arguments */
```

Şekil 3.30. Mex Kodları

```

#define YP_OUT plhs[0]

#if !defined(MAX)
#define MAX(A, B) ((A) > (B) ? (A) : (B))
#endif

#if !defined(MIN)
#define MIN(A, B) ((A) < (B) ? (A) : (B))
#endif

staticdouble mu = 1/82.45;
staticdouble mus = 1 - 1/82.45;

staticvoid yprime(
double yp[],
double *t,
double y[]
)
{
double r1,r2;

(void) t; /* unused parameter */

r1 = sqrt((y[0]+mu)*(y[0]+mu) + y[2]*y[2]);
r2 = sqrt((y[0]-mus)*(y[0]-mus) + y[2]*y[2]);
/* Print warning if dividing by zero. */
if (r1 == 0.0 || r2 == 0.0){
mexWarnMsgTxt("Division by zero!\n");
}
yp[0] = y[1];

```

Şekil 3.30. Mex Kodları (Devam)

```

yp[1] = 2*y[3]+y[0]-mus*(y[0]+mu)/(r1*r1*r1)-mu*(y[0]-mus)/(r2*r2*r2);
yp[2] = y[3];
yp[3] = -2*y[1] + y[2] - mus*y[2]/(r1*r1*r1) - mu*y[2]/(r2*r2*r2);
return;
}

void mexFunction( int nlhs, mxArray *plhs[],
int nrhs, const mxArray*prhs[] )

{
double *yp;
double *t,*y;
mwSize m,n;

/* Check for proper number of arguments */

if (nrhs != 2) {
mexErrMsgTxt("Two input arguments required.");
} elseif (nlhs > 1) {
mexErrMsgTxt("Too many output arguments.");
}

/* Check the dimensions of Y. Y can be 4 X 1 or 1 X 4. */
m = mxGetM(Y_IN);
n = mxGetN(Y_IN);
if (!mxIsDouble(Y_IN) || mxIsComplex(Y_IN) ||
(MAX(m,n) != 4) || (MIN(m,n) != 1)) {
mexErrMsgTxt("YPRIME requires that Y be a 4 x 1 vector.");
}

/* Create a matrix for the return argument */

```

Şekil 3.30. Mex Kodları

```

    YP_OUT = mxCreateDoubleMatrix(m, n, mxREAL);
/* Assign pointers to the various parameters */
yp = mxGetPr(YP_OUT);
    t = mxGetPr(T_IN);
    y = mxGetPr(Y_IN);
/* Do the actual computations in a subroutine */
yprime(yp,t,y);
return;
}
-----

```

Şekil 3.30. Mex Kodları

Özel bir diferansiyel denklem çözen bu uygulamayı MATLAB altında çalışabilir hale getirmek için komut satırına Şekil 3.31.'daki gibi

```
»mex yprime.c
```

Şekil 3.31. Mex Derleme

yazılır. Böylelikle çalışma dizininde yprime.mexw32 adlı bir dosya oluşacaktır. Bu dosya MATLAB tarafından okunabilir ve değişken değerleri ile yorumlanıp sonuç üretebilir bir fonksiyon olmuştur. Hazırlanan yprime uygulaması komut penceresi altında Şekil 3.32.'daki gibi çalıştırıldığında;

```
»yprime(3,[1 2 3 4])
```

Şekil 3.32. Mex Çalıştırma

Ürettiği sonuç;

```
ans=          2.000          8.9685          4.0000         -1.0947
```

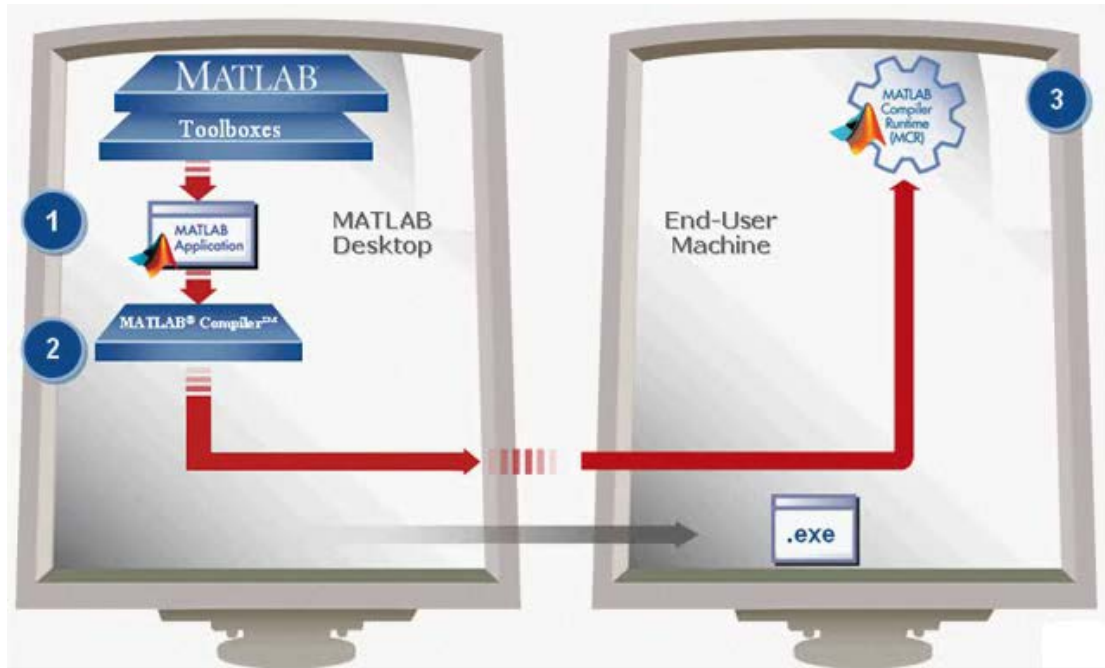
şeklinde olacaktır[11].

3.9. MATLAB Component Run-Time Ortamı (MCR)

MATLAB “Component Run-Time” Ortamı MATLAB kurulu olmayan bir Windows platformunda çalıştırılabilir hale getirilmiş MATLAB programlarını kullanabilmeyi amaç edinmektedir. Çalışma düzeni “Java Run Time” ortamına benzemektedir.[16]

“MCR” kullanmanın avantajlarını şu şekilde sıralayabiliriz;

1. “GUI” uygulamaları için idealdir
2. Her son kullanıcı için MATLAB kurulmasına gerek yoktur. Lisans maliyet avantajı sağlar
3. Son kullanıcının MATLAB bilgisine sahip olması gerekmemektedir
4. Son kullanıcı uygulamanın gizli kalması gereken kod detaylarına erişemez



Şekil 3.33. MCR [10]

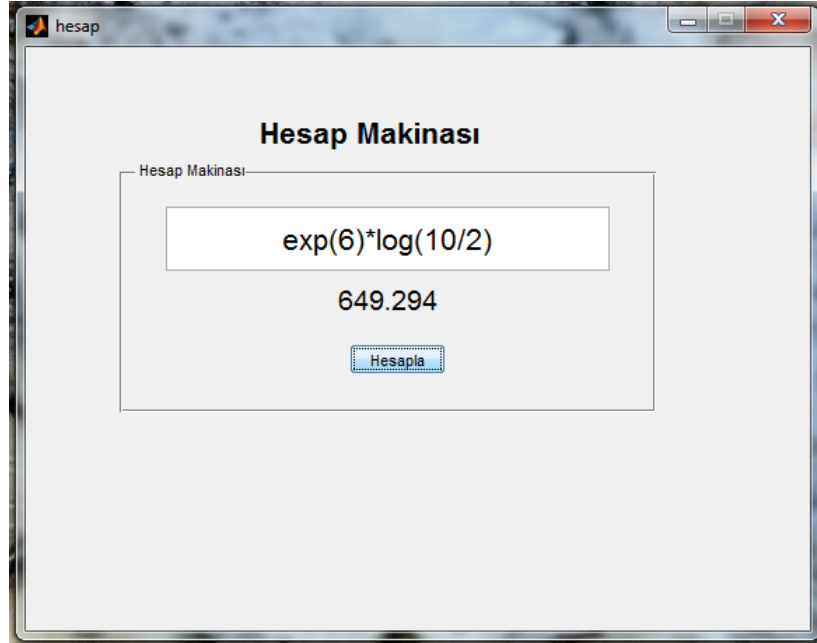
MATLAB Component Runtime çalışma ortamının bazı kısıtlamaları bulunmaktadır. Uygulama geliştiricinin yazmış olduğu “.m” dosyasının mutlaka bir fonksiyon olması gerekmektedir. “Signal Processing Toolbox” fonksiyonları gibi gömülü fonksiyonları desteklememektedir. Uygulamanın çalışabilmesi için hedef sistemde mutlaka MCR yüklü olması gerekmektedir[3].

4. UYGULAMALAR

4.1. Giriş

Uygulama olarak GUI bileşenleri kullanılarak hazırlanan, eval mantığı ile çalışan ve bu mantıkla log, exp gibi matematiksel işlemleri kolayca yapabilen, karmaşık olmayan bir hesap makinası, MATLAB Builder kullanarak Matlab'da hazırlanmıştır. Uygulamanın Microsoft Visual Studio'da Dinamik Bağlantı Kütüphanesi (DLL) ve MS Excel'de eklentiye (Add-in) dönüştürülerek bu ortamlarda kullanılması incelenmiştir. Ayrıca GUIDE ile hazırlanan Portföy Eniyileştirici Aracı (Portfolio Optimizer Tool) anlatılmıştır. Bu program da tek başına çalıştırılabilir şekilde hazırlanmıştır.

4.2. Eval Hesap Makinası Uygulaması

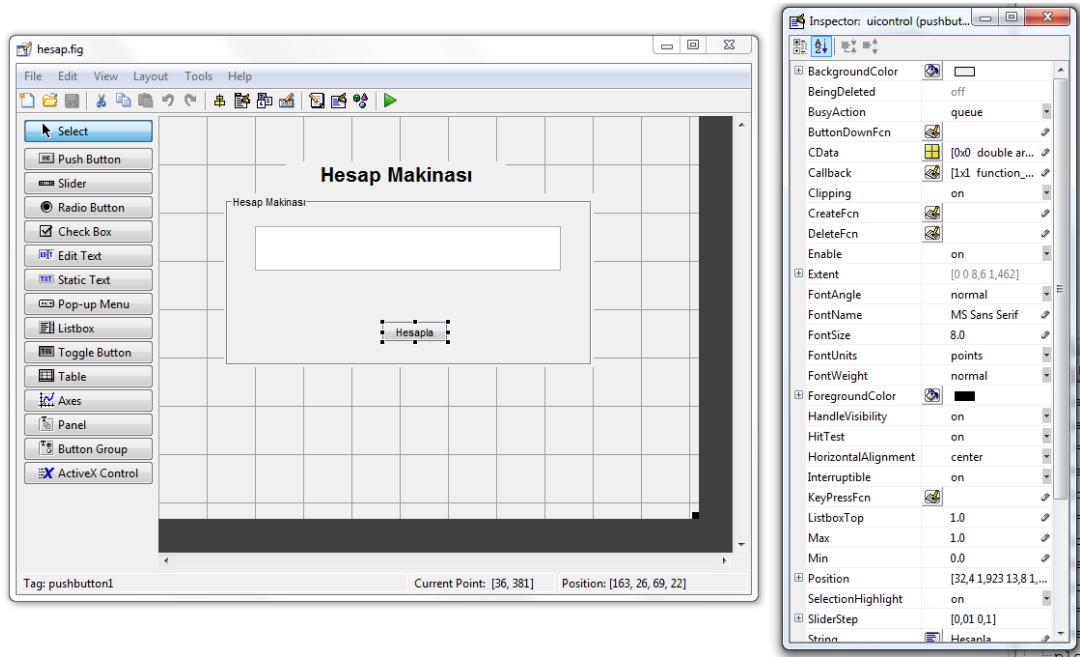


Şekil 4.1. Eval Hesap Makinası

4.2.1. Giriş

Şekil 4.1.'de görülen uygulama eval mantığı ile hazırlanmıştır. Matlab'da karakter katarı (string) şeklinde tanımlanan komutların icra edilmesi için kullanılan Eval fonksiyonu; programdan ya da metin kutusu (TextBox) aracılığı ile aldığı string işlemleri bir komut gibi çalıştırır. Elbette string içinde yazım hataları olması durumunda komut hata verecektir.

4.2.2. GUI'de oluşturma



Şekil 4.2. GUI Arayüzü

Şekil 4.2.'de görülen uygulamada Bölüm 2'de anlatılan bileşenler ile bir Eval Hesap Makinası hazırlanmıştır. Uygulamada Label, Panel, Edit Text ve Button bileşenlerinin yanı sıra Buttonun tıklama olayına(event) aşağıdaki kod parçası yazılmıştır[20].

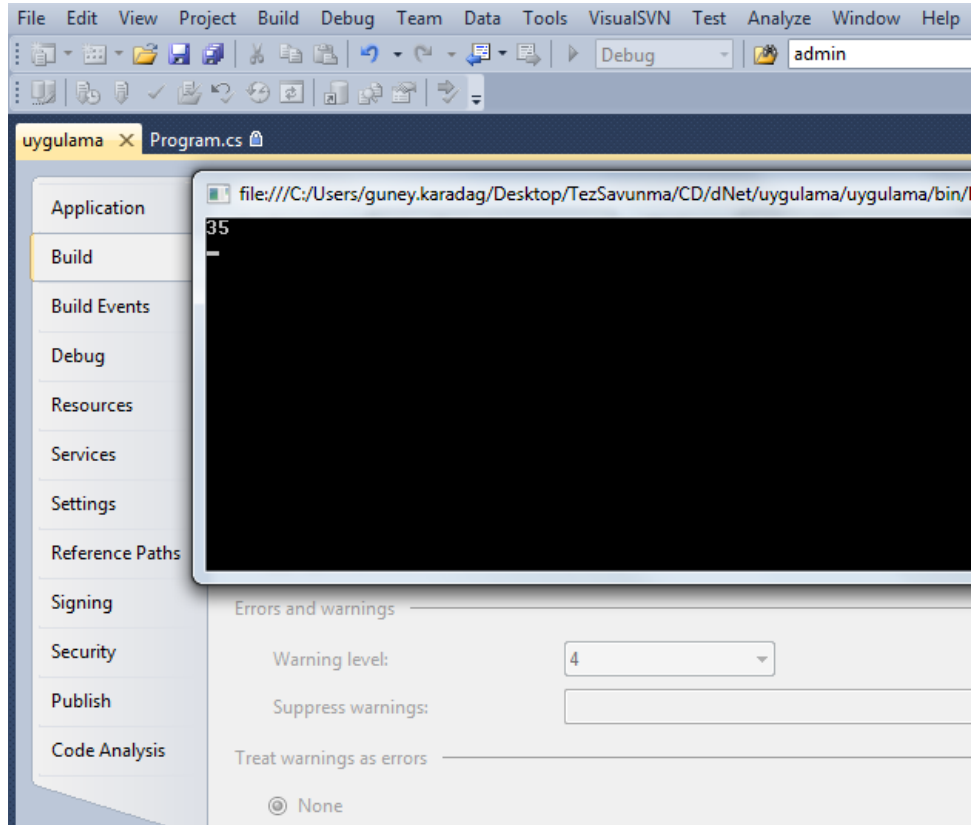
```
function pushbutton1_Callback(hObject, eventdata, handles)
% --- Executes on button press in pushbutton1.
set(handles.text2,'String',eval(get(handles.edit1,'String')));
```

Şekil 4.3. Eval Hesap Makinası Kodları

Bu kod parçası Hesapla düğmesine tıklandığında, EditText'ten aldığı string veriyi eval fonksiyonu ile yorumlanabilir hale getirip çalıştırarak, hesapladığı değeri, Hesapla düğmesinin üzerinde, değer barındırmayan bir etiket (Label) ile kullanıcıya sunmayı amaçlamıştır. Programın m-file, .fig ve derlenmiş. exe hali ekte bulunan CD'de bulunmaktadır.

4.3. MATLAB Builder .NET Uygulaması

Bu bölümde Matlab programında tanımlı Matlab Builder .NET ile iki sayının çarpımı şeklinde basit bir uygulamanın yapım aşamaları açıklanmıştır.



Şekil 4.4. MATLAB Builder .NET Uygulaması

4.3.1. Giriş

Matlab'ın, MATLAB Builder. NET eklentisi ile Matlab'da hazırlanan fonksiyon halindeki uygulamalar “dll” kütüphaneleri şeklinde derlenerek bu kütüphaneler Microsoft Visual Studio'da referans şeklinde kullanılabilir. Böylelikle Matlab'ın

nimetlerini Microsoft Visual Studio ortamında geliştirilen uygulamalarla birleştirilebilir.

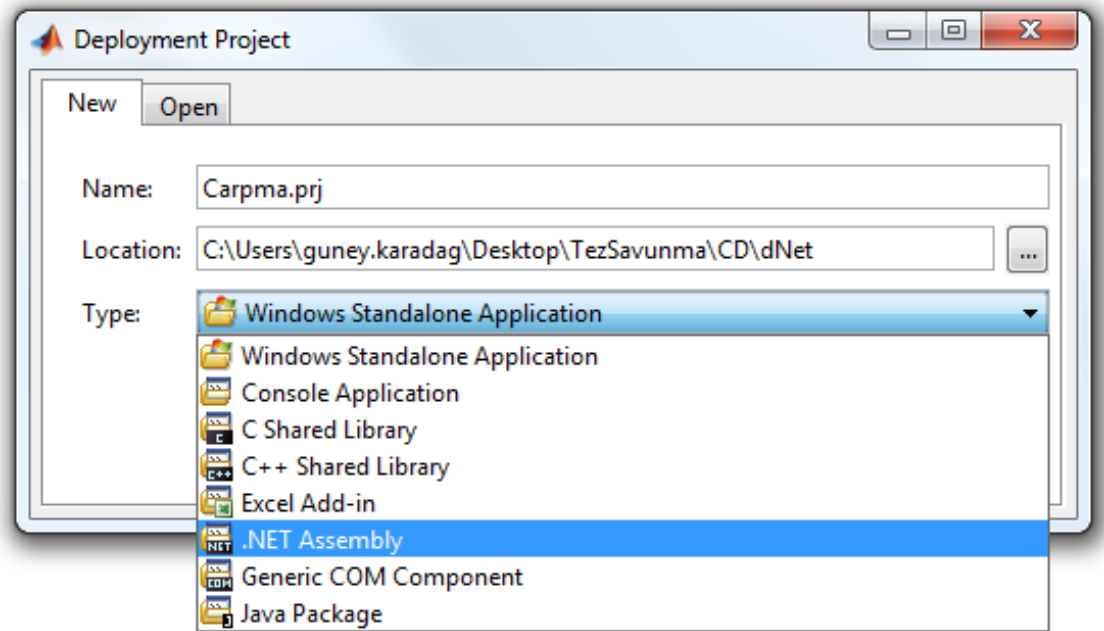
4.3.2. Matlab’da kütüphane oluşturma

Öncelikle dikkat edilmesi gereken durum kütüphane şeklinde derlenecek olan m-file’in function (fonksiyon) formatında hazırlanması gerektiğidir. Böylelikle bu fonksiyona parametreler aracılığı ile değer gönderebilir ve geriye değer döndürebilir. Şekil 4.5.’de basit bir çarpma fonksiyonu hazırlanmıştır.

```
function sonuc=carpmak(a,b)
sonuc=a*b;
```

Şekil 4.5. Matlabda Fonksiyon Hazırlama

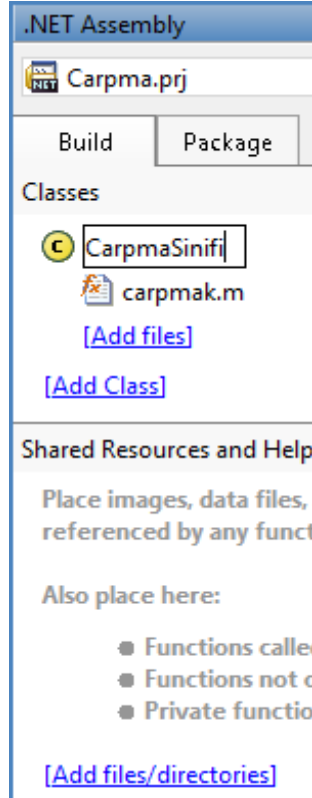
Bu fonksiyon aldığı a ve b parametrelerini çarparak sonuc değişkenine atmakta ve bu değeri döndürmektedir. Bundan sonraki adım File->New->Deployment Project komut dizisi ile .NET Assembly seçeneğini seçmektir.



Şekil 4.6. .NET Assembly

.NET Assembly penceresi Şekil 4.7.’deki gibi ekranın sağ tarafında belirecektir. Burada dikkat edilmesi gereken husus tanımlanan fonksiyonunun dâhil olacağı sınıfın adını girmek olacaktır. Girilen bu sınıf ismini daha sonra Microsoft Visual

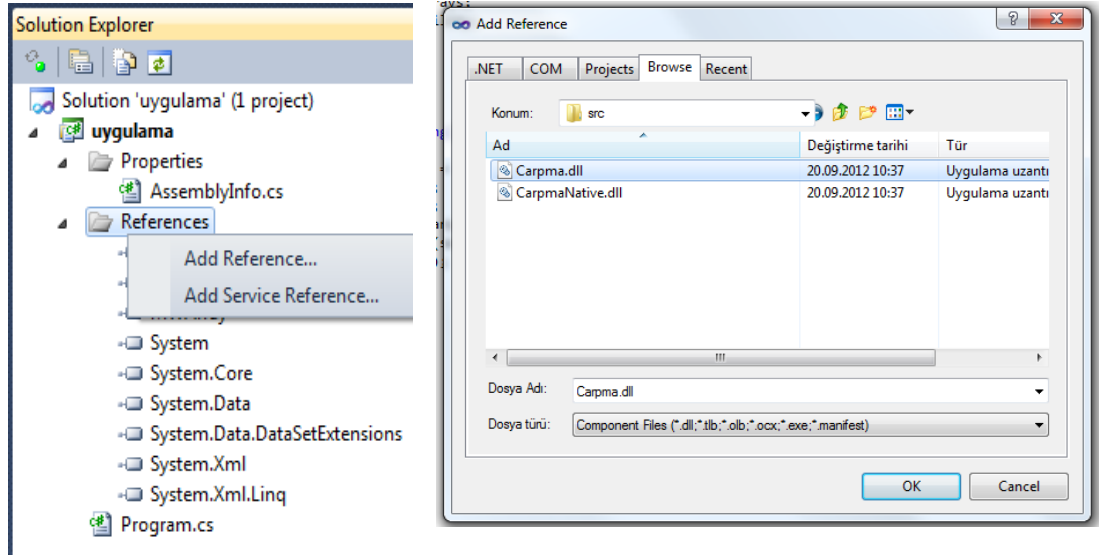
Studio'da yazılacak kodlar ile tanımlana fonksiyonu çağırılmakta kullanılacaktır. Bu uygulamada sınıf adı olarak CarpmaSinifi ismi kullanılmıştır. Bu işlemleride yaptıktan sonra sağ üst kısımda bulunan Build düğmesi ile derleme işlemi başlatabilir. Hatasız bir şekilde derleme işlemi bittikten sonra artık oluşturulan Carpma.dll dosyası Microsoft Visual Studio'da kullanılmak üzere hazır hale gelmiştir. DLL dosyaları bilindiği üzere şifrelidir. Harici bir program vasıtasıyla DLL dosyalarının içeriği görünemez. Yalnızca barındırdığı fonksiyonların adına ulaşılabilir. Bu sayede yazılan kodların da güvenliğini sağlanmış olur. DLL dosyalarının hız, zaman ve hafızadan tasarruf sağladığından da bahsedebiliriz. Programı tek bir .exe şeklinde derlemektense .dll'lere bölerek derlemek ciddi bir hız ve hafıza faydası sağlayacaktır.



Şekil 4.7. Sınıf Ekleme

DLL dosyası hazır olduğuna göre artık Microsoft Visual Studio'da kullanılabilir. Öncelikle File->New->Project komut dizisi ile Visual C# ağacı altından Console Application seçeneği seçilmelidir. Daha sonra hazırlanan uygulamaya bir isim vererek Tamam düğmesine basılmalıdır. Böylelikle Program.cs dosyasına kod yazılmaya başlanabilir. Kod yazmaya başlamadan önce Referans yani .dll dosyası

programa dahil edilmelidir. Öncelikle Microsoft Visual Studio'nun Matlab değişkenlerini tanması için örneğin C:\Program Files\MATLAB\R2012a\toolbox\dotnetbuilder\bin\win32\v2.0\ klasöründe bulunan MWArray.dll dosyası Referans olarak eklenmelidir. Daha sonra yine aynı yolla hazırlanan Carpma.dll dosyası da uygulamaya eklenmelidir. Referans eklemek için Microsoft Visual Studio ekranının sağ tarafında bulunan Solution Explorer penceresindeki References klasörüne sağ tıklanır ve Add Reference seçeneği seçilir.



Şekil 4.8. Referans Ekleme

Açılan pencerede Şekil 4.8.'da görüldüğü gibi Browse sekmesinden istenilen dll dosyası uygulamaya eklenebilir. Bu aşamadan sonra artık programda bu dosyaları çağırarak barındırdığı sınıfları ve fonksiyonları kullanmak mümkündür. Bunun için kod sayfasının ilk satırlarına aşağıdaki kodları eklenmelidir:

```
using Carpma;  
using MathWorks.MATLAB.NET.Arrays;  
using MathWorks.MATLAB.NET.Utility;
```

Şekil 4.9. Ekleme Kısmı

Bu işlemler de tamamlandıktan sonra kodlar yazılmaya başlanabilir. Kod yazımı da Şekil 4.10.'daki gibidir:

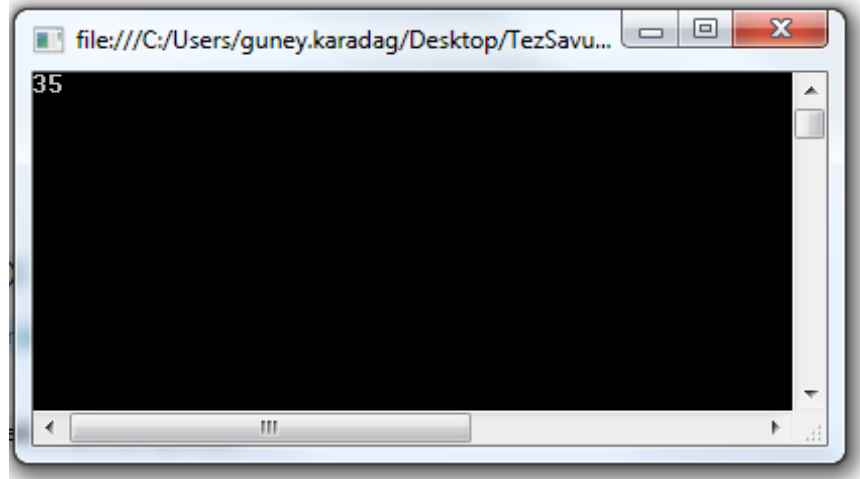
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Carpma;
using MathWorks.MATLAB.NET.Arrays;
using MathWorks.MATLAB.NET.Utility;

namespace uygulama
{
class Program
{
static void Main(string[] args)
{
CarpmaSinifi carp = new CarpmaSinifi();
MWArray sayi1 = 5;
MWArray sayi2 = 7;
MWArray sonuc = carp.carpmak(sayi1,sayi2);
Console.WriteLine(sonuc);
Console.ReadLine();

}
}
}
```

Şekil 4.10. Kod Yazımı

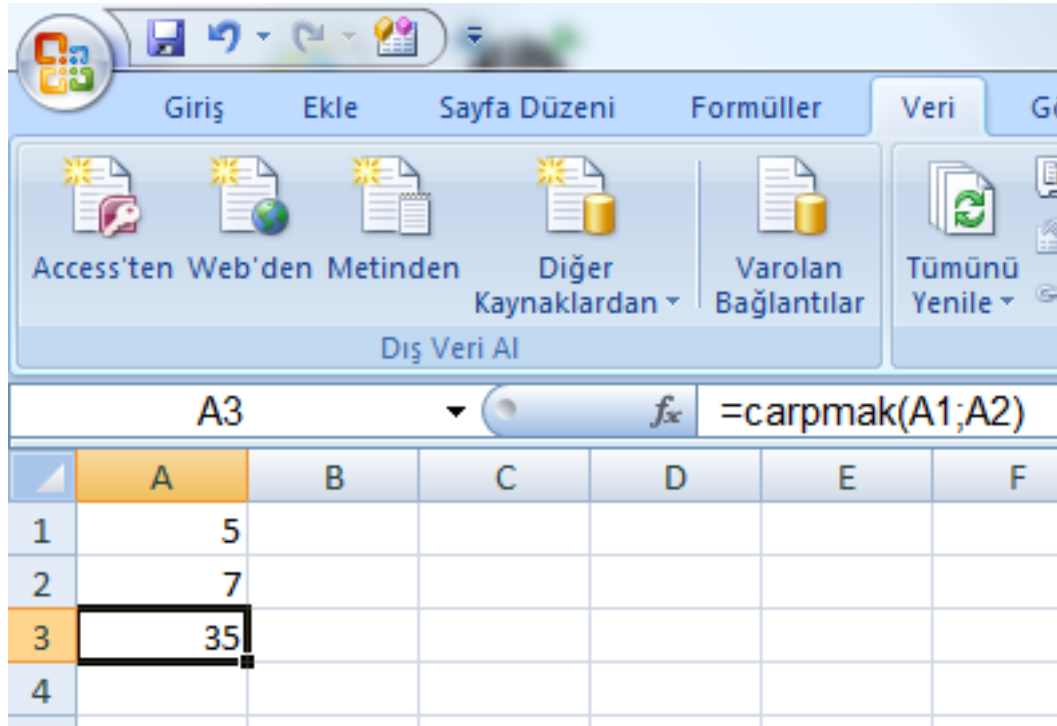
Burada öncelikle yapılması gereken Matlab'da oluşturulan CarpmaSinifi'ndan yeni bir nesne üretmektir. Çünkü üretilen bu nesneyi daha sonraki kısımda sınıfın barındırdığı fonksiyonu çağırmak için kullanılacaktır. Matlab'da değişkenler dizi şeklinde tutulduğu için bu programda da tanımlamalar MWArray tipinde yapılmalıdır. Örnek olarak sayi1 ve sayi2 değişkenlerine 5 ve 7 değerleri atansın. Sonraki adımda iki dizinin çarpımı yeni bir dizi olacağı için sonuc değişkenini de MWArray tipinde tanımlayarak ilk başta üretilen carp nesnesine bağlı carpmak fonksiyonuna parametre olarak 5 ve 7 değerleri gönderildi. Son adımda sonuç ekrana yazdırılarak, sonucun kullanıcıya görünmesi için `Console.ReadLine();` komutu ile kullanıcıdan bir tuşa basması istenmiştir. Programın çalışan son hali Şekil 4.11.'deki gibidir.



Şekil 4.11. Program Çıktısı

4.4. MATLAB Builder EX Uygulaması

Bu bölümde Matlab programında tanımlı Matlab Builder EX ile excel'de iki sayının çarpımı yapılacak şekilde basit bir uygulamanın yapım aşamaları açıklanmıştır[18].



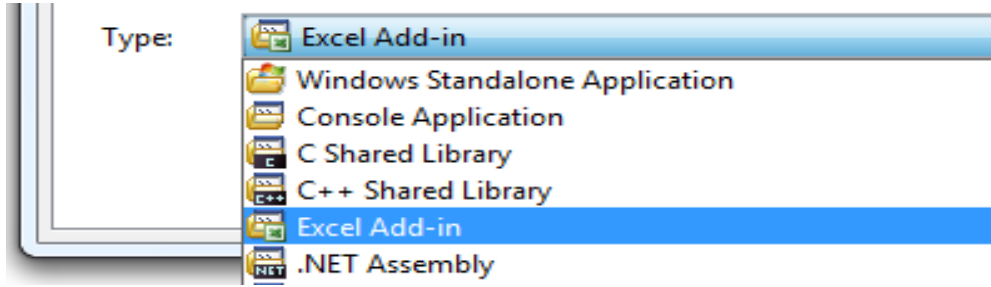
Şekil 4.12. MATLAB Builder EX Uygulaması

4.4.1. Giriş

MATLAB Builder EX eklentisi Matlab aracılığı ile hazırlanan uygulamaları Microsoft Excel'de Eklenti olarak kullanılmasına olanak sağlar. Böylelikle Microsoft Excel'de tanımlı olmayan fonksiyonlar ya da daha karmaşık işlemler Matlab aracılığı ile çözüme ulaştırılabilir.

4.4.2. Excel eklentisi (add-in) oluşturma

Şekil 4.13.'de görüldüğü gibi Microsoft Excel eklentisi oluşturmak, Microsoft Visual Studio için dll oluşturmaya benzemektedir. Derlenirken Matlab'ta File->New->Deployment Project sekmesi ile açılan pencereden Type olarak "Excel Add-in" Seçilir. İşlem sonucunda ise .bas uzantılı bir dosya oluşturulur.



Şekil 4.13. Excel Add-in seçeneği

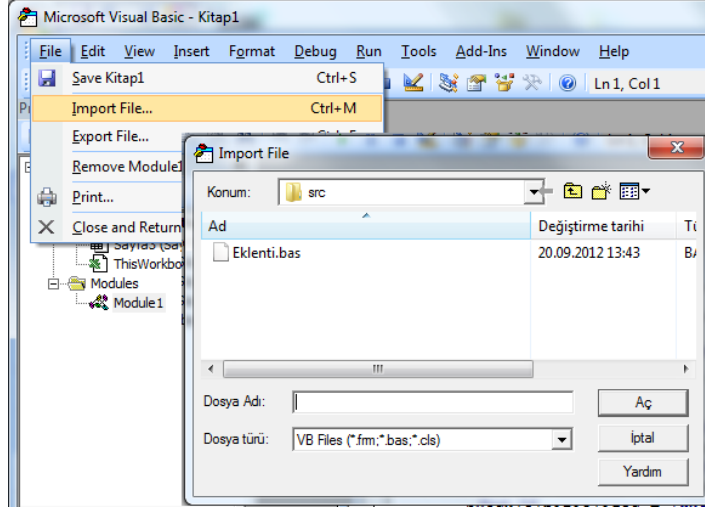
.bas dosyası içeriği Şekil 4.14'deki gibidir:

```
Dim MCLUtil As Object
Dim bModuleInitialized As Boolean
Dim CarpmaSinifi As Object
    Private Sub InitModule()
    If Not bModuleInitialized Then
    On Error GoTo Handle_Error
    If MCLUtil Is Nothing Then
    Set MCLUtil = CreateObject("MWComUtil.MWUtil7.17")
    End If
    Call MCLUtil.MWInitApplication(Application)
    bModuleInitialized = True
    Exit Sub
    Handle_Error:
    bModuleInitialized = False
    End If
    End Sub
    Function carpmak(Optional a As Variant, Optional b As Variant) As
Variant
    Dim sonuc As Variant
    On Error GoTo Handle_Error
    Call InitModule
    If CarpmaSinifi Is Nothing Then
    Set CarpmaSinifi = CreateObject("Eklenti.CarpmaSinifi.1_0")
    End If
    Call CarpmaSinifi.carpmak(1, sonuc, a, b)
    carpmak = sonuc
    Exit Function
    Handle_Error:
    carpmak = "Error in " & Err.Source & ": " & Err.Description
    End Function
```

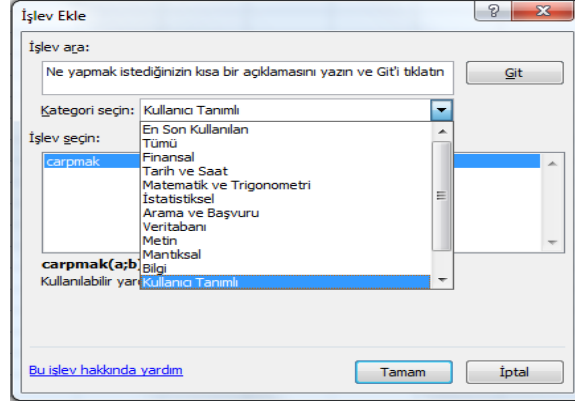
Şekil 4.14. .bas Dosyası İçeriği

Matlab, MATLAB Builder EX sayesinde uygulamamızı Visual Basic kodlarına dönüştürerek Excel'in anlayabileceği konuma getirir.

Hazırlanan ve yukarıda gösterilen .bas dosyası artık Microsoft Excel'e eklenti olarak eklenebilir. Bunun için öncelikle Microsoft Excel'de Visual Basic geliştirme ortamına ulaşılmalıdır. Bu ortama Excel Seçenekleri->Özelleştir->Geliştirici Sekmesinden Visual Basic'i seçmek gerekmektedir. Açılan Visual Basic geliştirme ortamında ise File->Import File seçeneği ile Matlab'da oluşturulan .bas dosyasını seçerek uygulama eklenti olarak Excel'e alınabilmektedir.



Şekil 4.15. Import File (Dosya Alma)

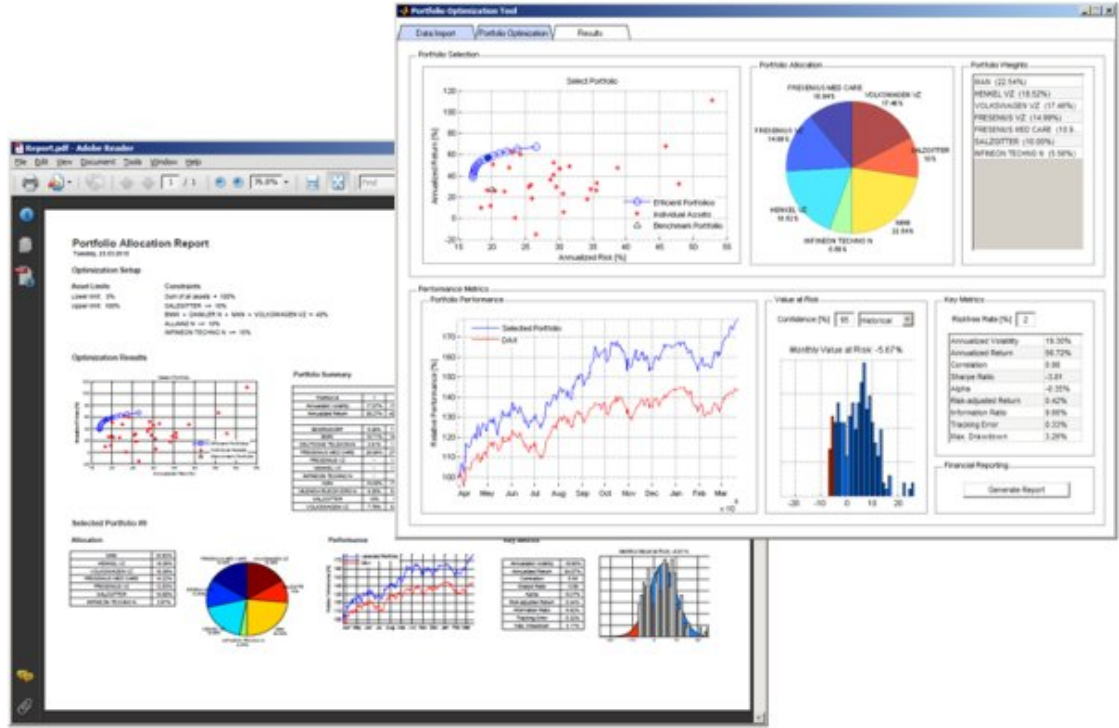


Şekil 4.16. İşlev Ekle

Sonraki adımda ise File->Save Kitap1 seçeneği ile bu eklentiye Şekil 4.15'te gösterildiği gibi kaydetmektir. Dikkat edilmesi gereken durum dosyayı "Excel 97-2003 Eklentisi (*.xla)" formatında kaydetmek olmalıdır. Artık kullanıcı tanımlı fonksiyon kullanılabilir hale gelmiştir. Microsoft Excel'de formül düğmesine basarak Matlab'da tanımlanmış olan fonksiyona, "Kategori seçin" kısmından Şekil 4.16'de görülen "Kullanıcı Tanımlı" seçeneği seçilerek ulaşabilir. Kullanım özellikleri ve alacağı değerler fonksiyon seçildiğinde alt kısmında açıklanmaktadır ve kullanımı diğer formüllere benzemektedir. Böylelikle Matlab sayesinde Microsoft Excel'de Kullanıcı Tanımlı Fonksiyonlar oluşturulmuştur.

4.5. Portföy Eniyileştirici Aracı (Portfolio Optimizer Tool) Uygulaması

Bu bölümde Matlab programında GUIDE kullanılarak hazırlanmış olan grafiksel pencerelerin yoğun olduğu bir uygulamaya bir takım yeni eklentiler yapılarak bu uygulamanın tek başına çalıştırılabilir bir uygulamanın yapım aşamaları açıklanmıştır[19].



Şekil 4.17. Portföy eniyileştirici aracı (Portfolio Optimizer Tool) ekran görünümü

4.5.1. Giriş

Portföy eniyileştirici aracı (Portfolio Optimizer Tool) Matlabda geliştirilmiş bir GUI tabanlı uygulamadır. Bu uygulama çeşitli veri kaynaklarından fiyat serilerini alabilir ve bu seriler üzerinde karışık eniyileştirme sabitlerini çıkarır. Ayrıca eniyileştirme sonuçları ve anahtar risk ölçümleri üzerine geniş interaktif görüntüleme yetenekleri sağlar.

Sonuç olarak bütün ilgili bilgiler Excel ve PDF formatında bağımsız bir şekilde raporlanabilir.

Bu uygulama aşağıdaki konuları içermektedir.

- Veri Alma (Data Import)
- Etkileşimli grafik bileşenleri (GUI-Building/implementation of interactive graphical components)
- Eniyileştirme ve Risk Ölçümleri (Optimization/Risk metrics)
- Nesneye Yönelimli Programlama (Object-oriented programming)
- Excel'e raporlama (Report to Excel)
- Bağımsız Bileşenler (Standalone components)

Portföy eniyileştirici aracı (Portfolio Optimizer Tool) 3 ana sekmeden oluşmaktadır. Bunlar sırasıyla;

- Data Import (Veri alma)
- Portfolio Optimization (Portföy eniyileştirme)
- Results (Sonuçlar)

4.5.2. Data import (veri alma)

The screenshot displays the 'Data Import' tab of the Portfolio Optimizer Tool. The 'Data Source' dropdown menu is open, showing 'Excel' as the selected option. The 'Data Import' section contains several dropdown menus for 'Prices Series', 'Prices Labels', 'Benchmark Index', 'Benchmark Label', 'Dates', and 'Date Format', all currently set to 'No data available'. An 'Import Data' button is located at the bottom right of the 'Data Import' section. The 'Imported Series' section below it contains a large empty area and a checkbox labeled 'First column as benchmark'.

Şekil 4.18. Portfolio Optimizer Tool (Portföy eniyileştirici aracı) Data Import Sekmesi

Portföy eniyileştirici aracı (Portfolio Optimizer Tool) isteğe bağlı olarak 3 şekilde veri alma seçeneğine sahiptir. Bunlar sırasıyla aşağıda verilmiştir;

- MATLAB Workspace
- Excel

MATLAB Workspace (çalışma alanı) seçeneği ile çalışma alanındaki verileri uygulamaya aktarmak mümkündür. Eğer daha önceden çalışılan veriler bir .mat dosyasına kayıt edilmiş ise, önce bu .mat dosyasının yüklenmesi (Load) gerekmektedir. Yükleddikten sonra gelen değişkenlerden hangileri Fiyat serisi (Prices Series) hangileri Fiyat etiketleri (Prices Labels) vb. gibi biçimleri Pop-up menüleri kullanarak belirlemek gerekmektedir.

Sonraki aşamada ise Import Data düğmesine basılarak verileri almak gerekmektedir. Alınan veriler pencerenin aşağısında bulunan Alınmış Seriler (Imported Series) penceresinde görülecektir. Bu adımdan sonra Veri Kabul Et (Accept Data) düğmesi ile Veri alma işlemini sonlandırmak mümkündür. Ayrıca kriter olarak ilk sütun (First column as benchmark) seçeneği de isteğe bağlı olarak işaretlenebilir.

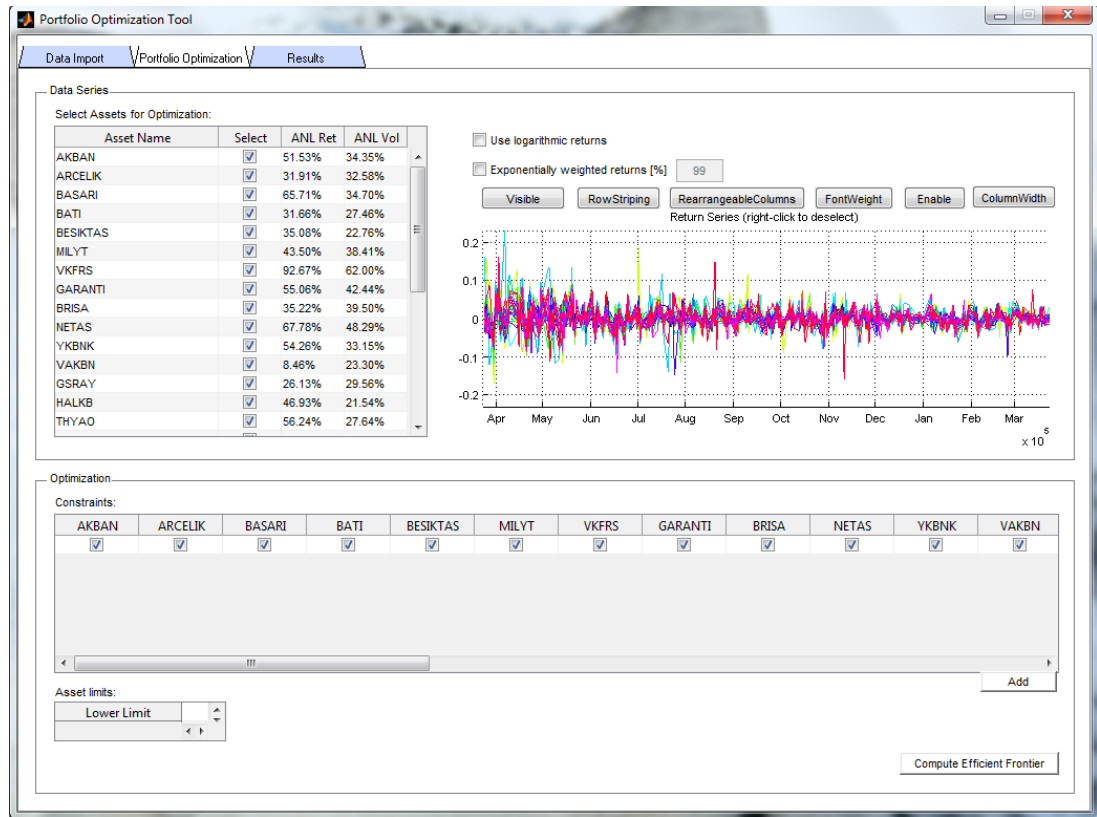
Excel seçeneği ile harici bir Excel dosyasında bulunan veriler uygulamaya aktarılabilir. Browse seçeneği ile Excel dosyası seçildikten sonra yine çeşitli ayarlamalar yapılarak Import Data ile verileri Imported Series penceresinde görebilir. Daha sonra Accept Data ile verileri uygulamaya alınabilir.

4.5.3. Portföy eniyileştirme (portfolio optimization)

Bu sekmede ticari varlıkların Asset Name (Varlık Adı), Annualized Return (yıllık getiri) ve Annualized Volatility (Yıllık Volatilité) değerlerini görülebilir. Getiri serilerini yıllık bazda (logaritmik veya ağırlıklı katlanarak) grafiksel olarak inceleyebilir, eniyileştirmeye dahil edilecek varlıkların seçim ayarları yapılabilir. Ayrıca Şekil 4.14'te solda görülen Assets tablosunda uygulama için seçilmiş 6 özellik de bu sekmede değiştirebilir. Bu özellikler aşağıda sıralanmıştır:

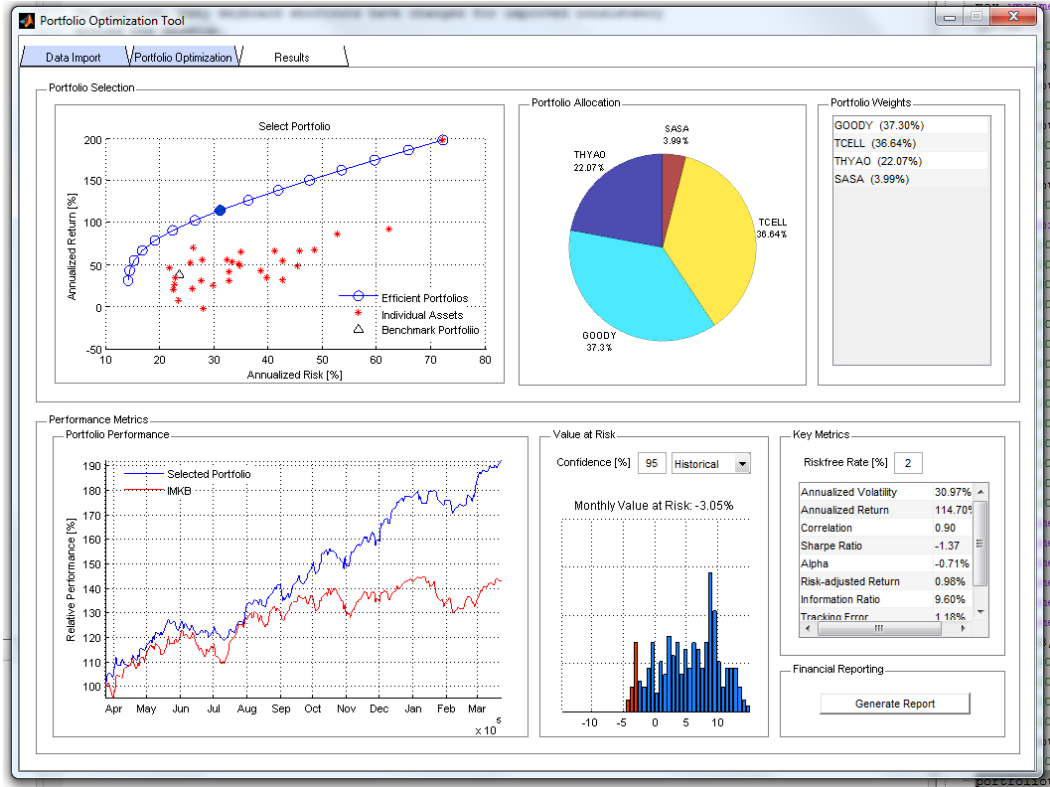
- Visible (Görünürlük)
- RowStriping (Tablonun kolay okunabilmesi için satır vurgulaması)
- RearrangeableColumns (Sütunların yerlerini değiştirme imkanı)
- FontWeight (Karakterleri belirginleştirmesi)
- Enable (Tabloyu seçime kapama)
- ColumnWidth (Sütunları Hizalamak)

Ayarlamalar yapıldıktan sonra Compute Efficient Frontier (etkin sınırı hesapla) düğmesine basılarak Result sekmesine geçilebilir.



Şekil 4.19.Portfolio Optimization (Portföy eniyileştirme) sekmesi

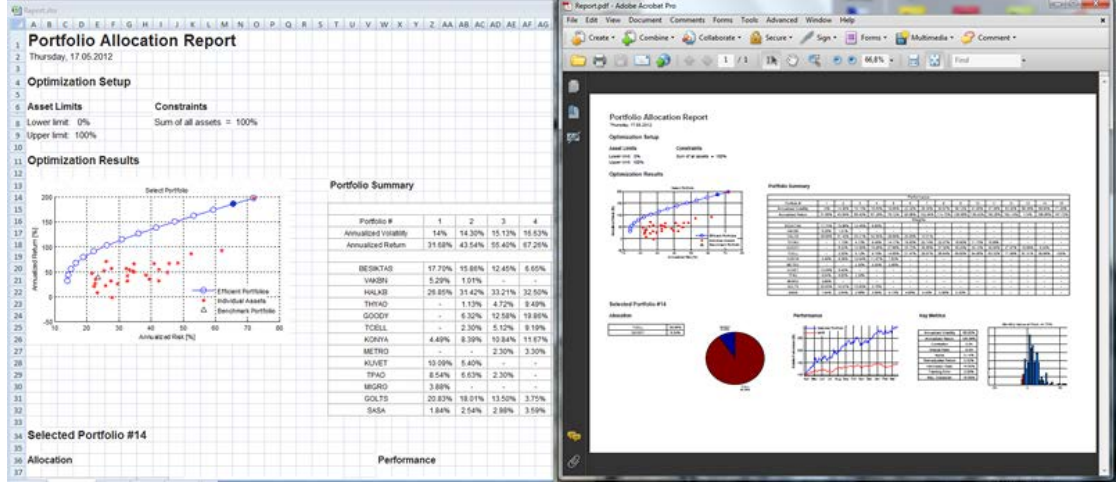
4.5.4. Sonular (Results)



Şekil 4.20. Results (Sonular) sekmesi

Sonular (Results) sekmesinde hesaplanan etkin sınırı grafiksel olarak grlebilir, bu grafik zerindeki varlıkların portfy zerindeki daėılımlarını pasta grafikte incelenebilir, portfydeki aėırlıkları ğrenilebilir. Ayrıca performans lmlerini de (portfy performansı, riske maruz deėer, risk free oranı) incelemek mmkndr.

Generate Report (Rapor Oluştur) dėmesi seilen bir portfy iin harici bir rapor hazırlamaktadır. ncelikle Excel formatında hazırlanan ve kaydedilen bu rapor daha sonra PDF formatında da kaydedilir ve alıřma klasrne atılır. Şekil 4.21.'de oluřturulan rnek rapor grlmektedir.



Şekil 4.21. Oluşturulan Rapor (Excel ve PDF)

4.5.5. Uygulama kodları

Tez çalışmasının uygulama kodları CD olarak ekte sunulmuştur.

SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, mühendislik alanında sıkça kullanılan MALTAB programının, Grafikselle Kullanıcı Arayüz'ü kullanılarak hazırlanan Portföy Eniyileştirme Aracı uygulaması incelenmiştir. Uygulama MATLAB Derleyici'si ile programdan bağımsız çalışacak şekilde derlenmiş ve dağıtıma hazır hale getirilmiştir. Buradan anlaşılıyor ki MATLAB, temel bilgisayar kullanıcılarını bir programcı yapmakta ve kendi programlarını hazırlamalarına ve dağıtımına izin vermektedir. Kullanıcıların Pascal, Fortran, ve C bilmeden kullanıcıların program yazabilmesi, bunların C ve C++ dillerine çevirebilmesi ve bileşen dağıtımını yapabilmesi sağlanır.

Gelecek açısından bakıldığında ülkemizde hava ve savunma sanayi, haberleşme sanayi, elektrik/elektronik sanayi, otomotiv sanayi ve finansal kurumlar (ASELSAN, HAVELSAN, TÜBİTAK, TOFAŞ, Merkez Bankası, Vakıfbank, İş Bankası, Garanti Bankası vb.) başta olmak üzere MATLAB kullanımının yaygın olduğu alanların bu kadarla sınırlı kalmaması MATLAB ve MATLAB tarafından derlenen bağımsız programların kullanımının yaygınlaşması amaçlanmıştır. MATLAB teknik programlama dilinin yetkinliği gün geçtikçe daha iyi anlaşılacaktır.

Sonuç olarak son kullanıcıyı kod karmaşasından kurtaran Grafik Kullanıcı Arayüzleri hazırlamaya imkan vermesi, bu uygulamaları MATLAB olmayan bilgisayarlarda çalıştırma yeteneği, ürettiği kodları Java, Excel ve .NET uygulamalarında kullanılmasına izin vermesi, kodların paylaşılması istenmeyen durumlarda şifreleyerek dağıtması, çeşitli programlama dillerinden MATLAB'a yada MATLAB'tan çeşitli programlama dillerine çevirim yapabilmesi gibi birçok özelliği bulunun MATLAB ve MATLAB Derleyicisinin kullanılması kurumlar, kuruluşlar ve öğrenciler açısından kaçınılmazdır.

KAYNAKLAR

- [1] Çetin A. E., *Matlab 7 Herkes İçin*, Alfa Yayınları, İstanbul, 2006.
- [2] Övüç H., 3D YEE Modelinin Matlab ile Gerçeklenmesi, Yüksek Lisans Tezi, Beykent Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2010.
- [3] [http://brahms.emu.edu.tr/fbayramoglu/fatos%20documents/math405/MATLAB AGIRIS1.pdf](http://brahms.emu.edu.tr/fbayramoglu/fatos%20documents/math405/MATLAB%20AGIRIS1.pdf) (Ziyaret Tarihi: 08.09.2011).
- [4] http://www.yildiz.edu.tr/~caydin/publications_dosyalar/Matlab_basic.pdf (Ziyaret Tarihi: 10.09.2011).
- [5] Savaş K., Kontrol Sistemleri İçin Matlab'da Gui Uygulamaları Tasarımı, Lisans Tezi, Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Bölümü, İstanbul, 2007.
- [6] Arı N., Barkana B., Matlab 6.0 GUI Kullanıcı Arayüzü Tasarımı, Osmangazi Üniversitesi Basımevi, Eskişehir, 2003.
- [7] <http://www.ifl.k12.tr/projedosyalar/MATLAB.pdf> (Ziyaret Tarihi: 12.09.2011).
- [8] Mathworks, MATLAB Compiler User's Guide 1.2, 2.8, 1.17, A(4-5-6-7)
- [9] <http://www.mathworks.com/products/demos/compiler/deployapps/index.html> (Ziyaret Tarihi: 22.09.2011).
- [10] Uzunoğlu M., Geçer T., Eren K., Kızıl A., Onar Ö. Ç., "Matlab ile Risk Yönetimi", Türkmen Kitabevi, İstanbul, 2005.
- [11] http://www.kxcad.net/cae_MATLAB/toolbox/compiler/f10-998688.html (Ziyaret Tarihi: 12.03.2012).
- [12] <http://www.scribd.com/doc/39741363/MATLAB-Lectures-1-%C4%B0TU-DR-AD%C4%B0L-YUCEL> (Ziyaret Tarihi 03.02.2011)
- [13] http://www.peteryu.ca/tutorials/matlab/image_in_3d_surface_plot_with_multiple_colormaps (Ziyaret Tarihi: 17.09.2011).
- [14] <http://www.mathworks.com/company/events/webinars/index.html> (Ziyaret Tarihi: 17.05.2012).
- [15] <http://www.mathworks.com/help/compiler/working-with-the-mcr.html> (Ziyaret Tarihi: 27.04.2012).

- [16] <http://www.mathworks.com/help/matlab/ref/polar.html> (Ziyaret Tarihi: 15.09.2011).
- [17] <http://www.mathworks.com/videos/matlab-builder-ex-overview-68784.html> (Ziyaret Tarihi: 17.04.2012).
- [18] http://www.youtube.com/watch?v=ZeY4_9IPvyA (Ziyaret Tarihi: 19.04.2012).
- [19] <http://www.mathworks.com/matlabcentral/fileexchange/29807> (Ziyaret Tarihi: 03.02.2011)
- [20] <http://www.mathworks.com/help/matlab/ref/eval.html> (Ziyaret Tarihi: 17.05.2012).

KİŞİSEL YAYIN VE ESERLER

- [1] **KARADAĞ E. G.**, İNAL M., Grafik Kullanıcı Arayüzüne Sahip Matlab Programlarının Tek Başına Çalıştırılabilir Halde Hazırlanması, *6th International Computer & Instructional Technologies Symposium*, Gaziantep, 04-06 Ekim 2012

ÖZGEÇMİŞ

08 Ekim 1986 tarihinde Ankara'da doğdu. İlk ve Orta Okulu Ankara Aydınlikevler İlköğretim Okulu'nda tamamladı. Liseyi Ankara Atatürk Anadolu Meslek Lisesi'nde tamamladı.2005 yılında girdiği Kocaeli Üniversitesi Bilgisayar Öğretmenliği Bölümünden 2009 yılında mezun oldu. 2009 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Bilgisayar Eğitimi Anabilim Dalında Yüksek Lisans eğitimine başladı. Şu an T.C. Başbakanlık Basın Yayın ve Enformasyon Genel Müdürlüğü Bilgi-İşlem Müdürlüğünde Çözümleyici olarak görev yapmaktadır ve Ankara'da yaşamına devam etmektedir.