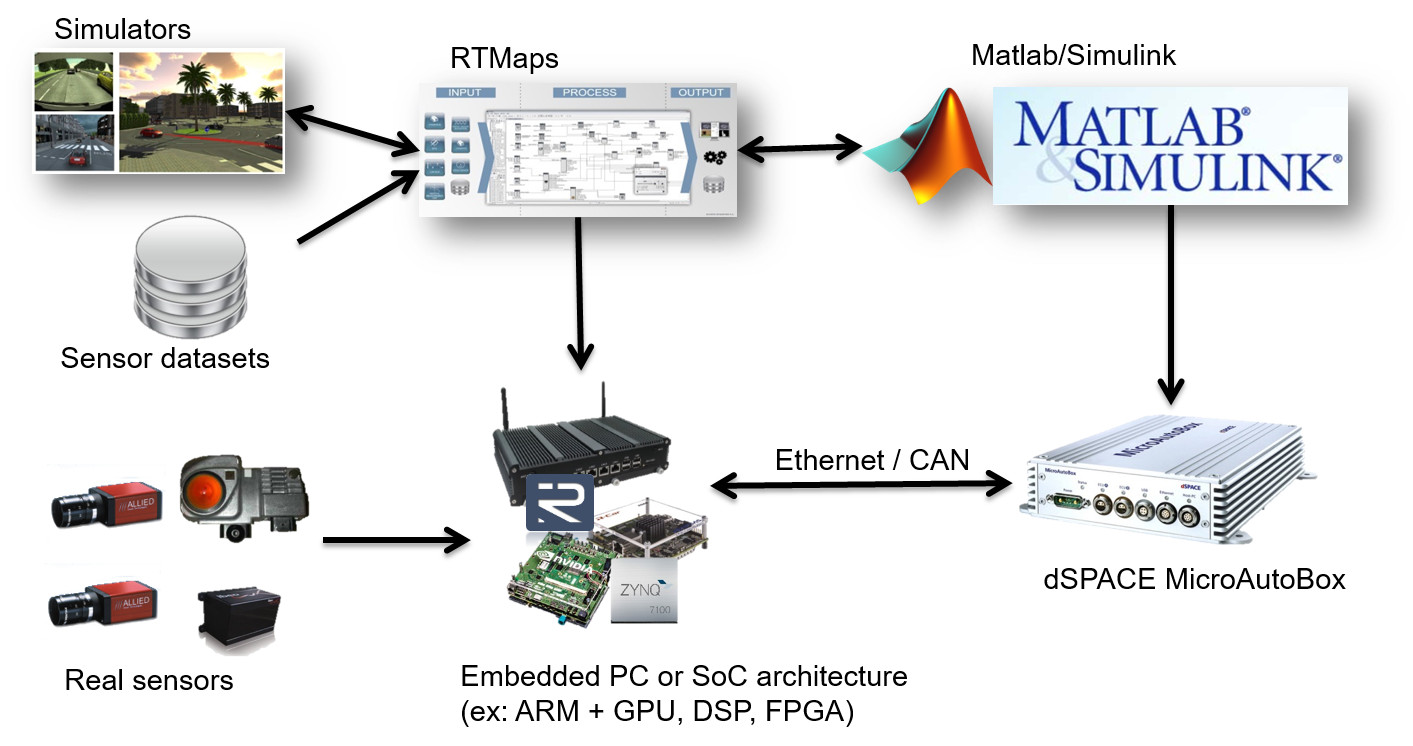


**KONTROL SİSTEMLERİ-1 LABORATUVARI**

**DENEY FÖYÜ**



**Dr. Öğr. Üyesi Güzin ÖZMEN**

**Arş. Gör. Fehmi SEVİLMİŞ**

**Konya, 2018**

**İÇİNDEKİLER**

**Laboratuvar Kuralları ve Deney Grupları**........……………………………………...…3

**Deney-1:** MATLAB Programının Kullanımı…………………………………………..….5

**Deney-2:** Laplace Dönüşümü ve Diferansiyel Denklem Çözümü ……………………....15

**Deney-3:** Transfer Fonksiyonu ve Durum Uzay Gösterimleri …………………….…….19

**Deney-4:** Fiziksel Sistemlerin Modellenmesi ………………………………..………….27

**Deney-5:** Blok Şemaların İndirgenmesi …………………………………………………33

**Kontrol Sistemleri-1 Laboratuvarı Kuralları**

1. Deneyler **iki haftada bir**, **ders programında belirtilen gün ve saatte** yapılacaktır.
2. Laboratuvarın **ders içerisindeki ağırlığı %25**’tir.
3. **Devamsızlık hakkı** 1 deneydir. (2 veya daha fazla deneye gelmeyen laboratuvar notundan sıfır puan alacaktır.)
4. Her öğrenci kendi gün ve saatinde laboratuvara girecektir. Öğrenciler kendi grupları dışındaki herhangi bir grupla deneylere katılamayacaktır.
5. Laboratuvara gününde ve saatinde gelmek **mecburidir.**
6. Laboratuvara gelirken deney föyü ve kalem, silgi vb. dışında bir şey getirmeye gerek yoktur.
7. Laboratuvar vize ve final sınavları, vize ve final haftalarından 1 hafta önce uygulamalı olarak yapılacaktır. Değerlendirme notu bu sınavlara göre verilecektir.

**AÇIKLAMALAR**

1. **Deneyde yaptığınız uygulamalardan ve deney sonrasında verilen çalışmalardan sorumlusunuz.**
2. **Deney sonrası verilen çalışmaları, ödev olarak getirmenize gerek yoktur. Ancak; vize ve final sınavlarında bu çalışmalar da dahildir.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Deneyler** | **Normal Öğretim** | | **İkinci Öğretim** | |
| **A-Grubu** | **B-Grubu** | **A-Grubu** | **B-Grubu** |
| **Deney-1** | 2 Ekim | 9 Ekim | 2 Ekim | 9 Ekim |
| **Deney-2** | 16 Ekim | 30 Ekim | 16 Ekim | 30 Ekim |
| **Deney-3** | 6 Kasım | 13 Kasım | 6 Kasım | 13 Kasım |
| **Vize Uygulama Sınavı** | **19-23 Kasım** | | | |
| **Deney-4** | 11 Aralık | 18 Aralık | 11 Aralık | 18 Aralık |
| **Deney-5** | 25 Aralık | 31 Aralık | 25 Aralık | 31 Aralık |
| **Final Uygulama Sınavı** | **2 - 4 Ocak** | | | |

**DENEY NO:** 1

**DENEY ADI:** MATLAB Programının Kullanımı

* 1. **DENEYİN AMACI**

1. MATLAB programının kullanılmasını öğrenmek
2. MATLAB ortamında basit matematiksel fonksiyonları ifade etmek
   1. **TEORİK BİLGİLER**

MATLAB(**MAT**rix **LAB**oratory, Matris Laboratuvarı); sayısal hesaplama, programlama, bir sistemin modellenmesi gibi işlemler yapan yüksek seviyeli programlama dilidir. C/C++ gibi programlama dillerine göre kullanımı daha kolay ve hızlı bir çözüm sağlar. İşaret işleme, görüntü ve video işleme, kontrol sistemleri, test ve ölçme gibi daha birçok alanda geniş bir uygulama alanına sahiptir. Peki, MATLAB neler yapabilir?

* 20. dereceden bir denklemin köklerini bulabilir.
* 100x100 boyutlu bir matrisin tersini alabilir.
* Bir elektrik motorunu gerçek zamanda kontrol edebilir.
* Bir otobüsün süspansiyon simülasyonunu yapabilirsiniz.

MATLAB, araştırma ve mühendislik alanlarında karşılaşılan problemlere pratik ve hızlı sonuçlar sunmaktadır. Bu nedenle MATLAB, tüm dünyada binlerce endüstri, devlet ve akademik kurumlarda kullanılmaktadır. Özellikle tüm üniversitelerde yaygın olarak kullanılmaktadır. MATLAB kullanıcı şirketler arasında;

* Dünyada Boeing, Motorola, NASA, Texas Instruments, Toyota vb.
* Ülkemizde ise Aselsan, Tofaş, Arçelik, Siemens vb.

şirketler verilebilir.

**MATLAB’ in Tanıtılması**

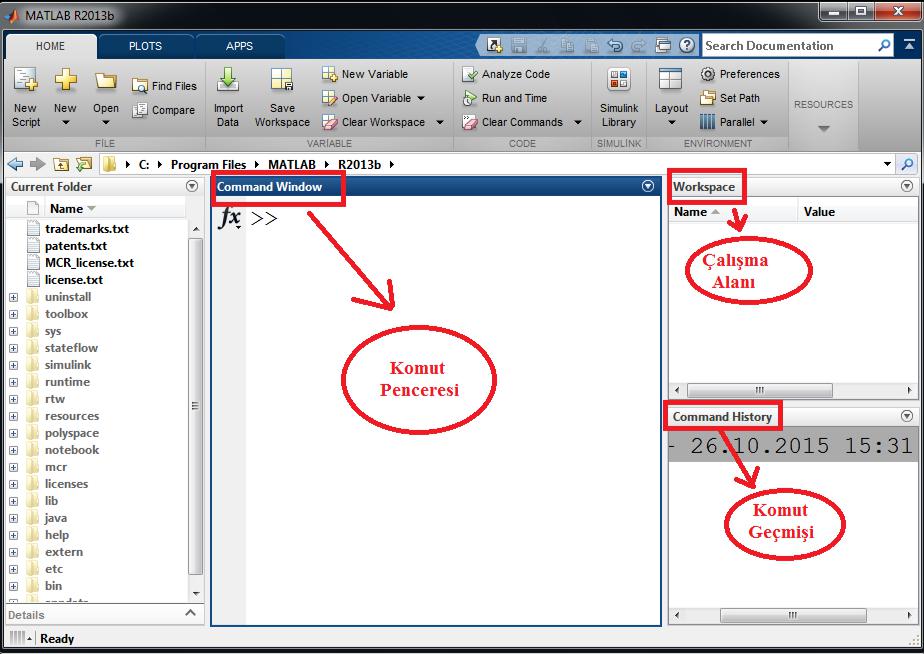
1. **MATLAB Pencereleri**

MATLAB ilk açıldığı zaman diğer sayfada olduğu gibi bir pencere açılır.

**Command Window (Komut Penceresi):** Basit matematiksel işlemlerin yapıldığı,değişkenlerin girildiği ve kod çıktılarının göründüğü ana penceredir. Burada yapılan işlemler, klavyenin **Enter** tuşuna basıldıktan sonra değiştirilmez. Kod yazmaya uygun değildir.

**Workspace (Çalışma** **Alanı):** Kullanılan değişkenler ile ilgili bilgilerin sağlandığı penceredir.

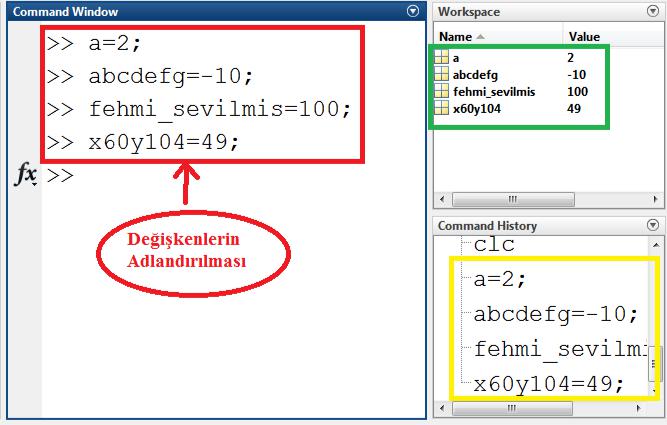
**Command History (Komut History):** Komut penceresinde girilen komutların kaydı tutulur. En üst kısımda tarih ve saat bilgisi yer alır.



1. **Değişkenlerin Adlandırılması**

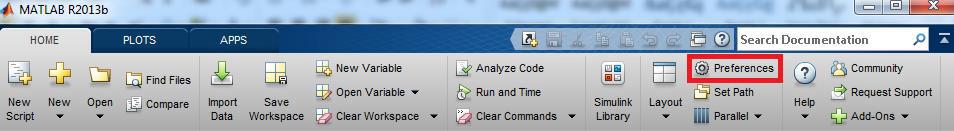
Değişken adları Türkçe karakter olmamak üzere bir harf ile başlaması gerekir ve onu takiben herhangi bir sayıda harfler, rakamlar veya altçizgiden ( \_ ) oluşabilir. Ancak birinci harften sonra boşluk bırakarak devam edilemez.

* Değişkenler, büyük ve küçük harflere duyarlıdır. **( Yani; a ≠ A )**

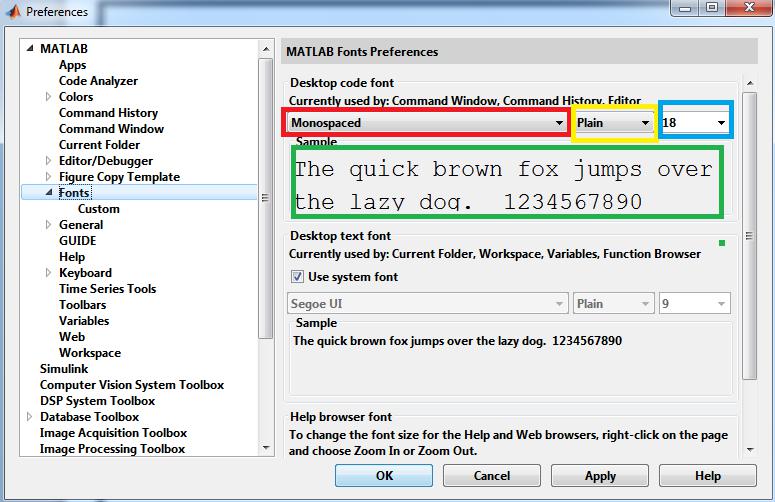


1. **Yazı Tipinin Değiştirilmesi**

En üstte araç çubuklarında yer alan ve çeşitli ayarların yapıldığı **Preferences** kısmına tıklanır.

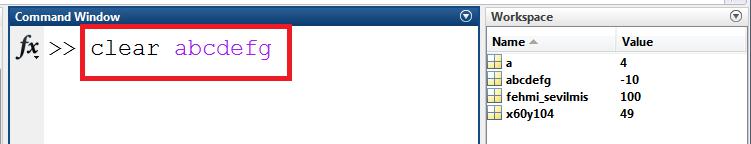


Açılan pencerenin sol kısmındaki **Fonts’ a ( Yazı tipi)** tıklanır. Yazı boyutunu değiştirmek için **Mavi** kutucuk, yazının kalınlığı için **Sarı** kutucuk ve yazı şekli için **Kırmızı** kutucuk tıklanır. **Yeşil** kutucukta ise ayarlar örnek (sample) ile gösterilir.



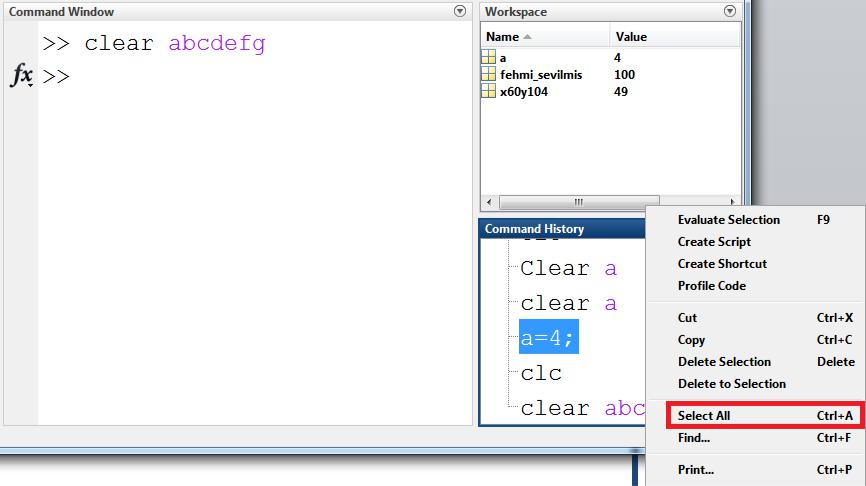
1. **Pencerelerin Temizlenmesi**

* Komut penceresini temizlemek için **clc ( cl**ear **c**ommand window) yazıp Enter’ e basılır.
* Çalışma alanı penceresindeki tüm değişkenleri temizlemek için komut penceresine **clear** yazılır. Sadece bir değişkeni silmek için **clear**’dan sonra bir boşluk bırakarakdeğişken ismi yazılır. **Örnek: clear abcdefg**



Yani; clear komutu, bellekteki tüm değişkenleri siler.

* Komut geçmişi penceresini temizlemek için ise ya pencere üstünde sağ tıklanır **Select** **All (Tümünü Seç)** ya da **Ctrl + A** tuşları kullanılır ve **Delete** tuşuna basılır.



1. **Noktalı Virgül ( ; ) Kullanımı**

Değişkenler alt alta yazılabildiği gibi yan yana da yazılabilir. Değişkenleri yan yana yazmak için ya virgül ( , ) ya da noktalı virgül ( ; ) kullanmak gerekir. Virgül kullanıldığı zaman komut penceresinde bir alt satıra değişkene atanan değer gösterilir. Değişkene atanan değeri görmemek için noktalı virgül kullanmak gerekir.

|  |
| --- |
| **UYGULAMA-1: “**a” ve “b” gibi iki değişkene rastgele bir değer ataması yapınız. a'yı b’ye bölünüz ve sonucunu bir başka değişkene atayınız. Değişkenlerin sonuna noktalı virgül ( ; ) kullanma veya kullanmama durumlarını inceleyiniz. |
| **>>**  **>>**  **>>**  **>>** |

* Açıklama eklemek için ( Türkçe karakter dahil her şey yazılabilir) yazının başına % (yüzde) işareti konur. MATLAB bunu komut olarak dikkate almaz. Açıklama, aradan geçecek zaman içerisinde komutların ne amaçla yazıldığı hakkında bilgi vermekten başka bir şey değildir.
* Bir işlem, bir değişkene atanmamış ise MATLAB otomatik olarak sonucu, **ans** **(answer, cevap)** adı verilen özel değişkende saklar.

1. **Sayılar**

* Ondalıklı sayılar Türkçe’ deki **2,5** yerine **2.5** şeklinde nokta ile gösterilir.
* Bilimsel notasyon gösterimi olan e (veya E) harfi 10’nun kuvvetini temsil eder.

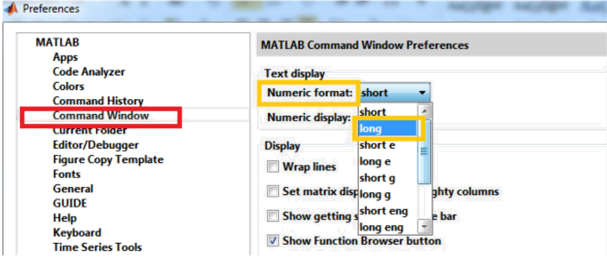
**Ör:** 2e4 = 2.104=2000 veya 1.65e-20 = 1.65.10-20

* Virgülden sonraki ondalık sayısı 4 hanedir. Bunu arttırmak için komut penceresine **format long** yazmak yeterlidir.Ondalık sayısı 15 hane olur. Tekrar 4 hane yapmakiçin **format** veya **format short** yazılır. Diğer bir yol ise Preferences sekmesinden sol taraftan Command Window seçilir ve Numeric Format long (uzun) yapılır.
* Ondalıklı sayıları rasyonel sayı olarak görmek için **format rat** yazılır.

**Ör:** 0.6667 = 2/3

* + - İki ondalıklı sayı için **format bank** yazılır.

**Ör:** 0.67

****

1. **Matematiksel İşlem Yazımının MATLAB Karşılığı**

**y= a.b + c2 – d3c + – +**  - fonksiyonunu MATLAB’ de yazarken **y=a\*b + c^2 – (d^3)\*c + sqrt(b) – exp(-x) + 3/(c-a) – nthroot(a,5)**

şeklinde ifade edilir.

**sqrt:** square root (karekök)

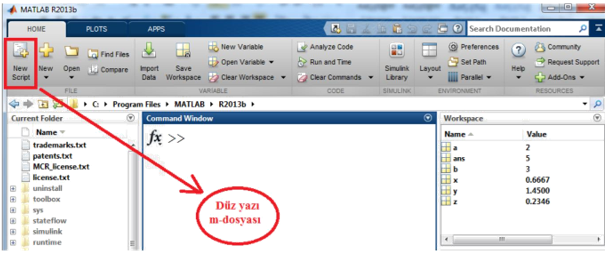
**nthroot(sayı, n.kök):** bir sayının n. dereceden kökünü alır.

|  |
| --- |
| **UYGULAMA-2:** Yukarıdaki “y” fonksiyonunun değerini MATLAB ile bulunuz. Değişkenlere rastgele değer ataması yapınız. |
| **>>**  **>>**  **>>**  **>>** |

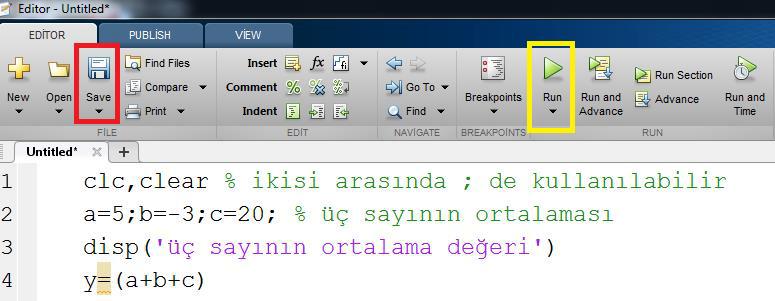
1. **MATLAB’ de Programlama**

MATLAB’ de programlama genel olarak iki yolla yapılır:

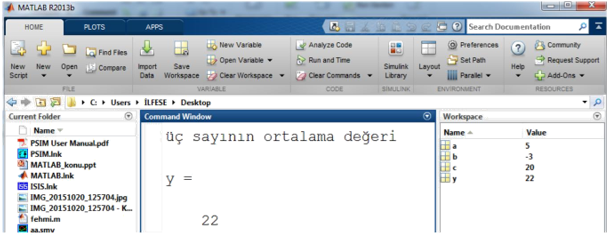
* Komut satırında (in-line) programlama
* m-dosyasıyla (m-file) programlama
  + - Komut satırında programlamayı önceki uygulamalarda anlatıldı. Şimdi m-dosyasıyla programlamaya değinilecek. Bunun için önce **düz yazı (New Script)** m-dosyası açılır.
    - Kod yazmaya ilk olarak **clc** ve **clear** komutlarıyla başlanır. Daha önce de değinildiği gibi clc, komut penceresindeki daha önce yapılmış gereksiz komutları temizler; clear ise işe yaramayacak değişkenleri bellekte boşuna yer tutmaması için silmeye yarar.
    - Daha önce anlatılanlar buraya da uygulanabilir. Komut penceresine yazılan komutlara göre avantajı daha önce yazılmış komutları değiştirebilme imkanıdır. O yüzden daha uzun komutları burada yazmak kolaylık sağlar.

****

* **Basit bir program yazımı**
  + Değişkenler yan yana ve aralarında noktalı virgül konulmuş. Bundan dolayı komut penceresinde değişkenler yazılmaz.
  + % işareti koyarak gerekli açıklamalar eklenmiş.
  + Komut penceresinde metinsel çıkış gösterimi için **disp (display, gösterme)** komutu kullanılır.
  + y=(a+b+c) komutunun sonunda noktalı virgül olmadığı için komut penceresinde sonuç görünür.



Programı çalıştırmadan önce MATLAB programı kaydetmenizi ister. Kayıt dosyasına isim verme kuralları değişken atama kuralları ile aynıdır. Kaydettikten sonra ise **Run (Koştur)** işaretine veya **F5** tuşuna basılır. Çıktı ise komut penceresinde görünür.

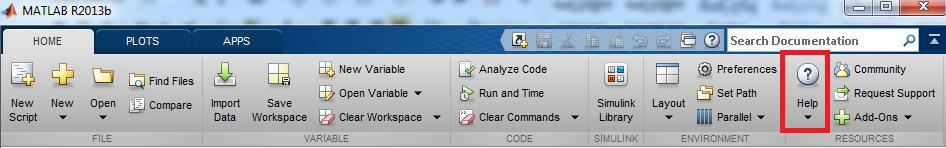
****

1. **Yardım (Help) Menüsü**

Bir komut hakkında bilgi sahibi olmak için komut satırına **help xxx** yazılır. Burada xxx bilgi almak istenilen komuttur.

**Ör:** helpsqrt

Diğer yol ise Help (?) işaretine tıklamak veya F1 tuşuna basmaktır.



* 1. **DENEY SONRASI ÇALIŞMA SORULARI**

**UYGULAMA-1:** Santigrat sıcaklığını, Fahrenhayt sıcaklığına dönüştüren programı MATLAB’de yazınız.

**İstenen-1:** **Santigrat sıcaklığını klavyeden yani dışarıdan girilecek şekilde** yazınız. Örneğin; klavyeden 10 °C girildiği zaman Fahrenhayt sıcaklığı komut penceresinde 50 olarak görünsün.

İPUCU: **MATLAB Kılavuzuna** bakınız. ==> **“input”** komutu

**CEVAP-1:**

**İstenen-2:** Komut penceresi ekranında alttaki gibi metin yazdırınız. Bunun için **disp** ve **fprintf** komutlarını kullanınız.

İPUCU: **MATLAB Kılavuzuna** bakınız.

**CEVAP-2:**

**“disp” komutu sonucunda elde edilen çıktı**

****

**“fprintf” komutu sonucunda elde edilen çıktı**

****

**DENEY NO:** 2

**DENEY ADI:** Laplace Dönüşümü ve Diferansiyel Denklem Çözümü

* 1. **DENEYİN AMACI**

1. Kontrol sistemlerinin analizinde sıkça kullanılan Laplace dönüşümünü MATLAB ortamında gerçekleştirmek
2. Diferansiyel denklemleri MATLAB ortamında çözmek
   1. **TEORİK BİLGİLER VE UYGULAMALAR**

**Symbolic Math Toolbox** içinde tanımlı olan **laplace** ve **ilaplace** komutları ile Laplaceve Ters Laplace dönüşümlerinin doğrudan sembolik çözümlerini yapmak mümkündür. Diferansiyel denklem çözümü için ise **dsolve** komutu kullanılır.

**1-) Laplace Dönüşümü: F=laplace(f)** komutu MATLABortamında tanımlanmış bir **f** fonksiyonunun sembolik Laplace dönüşümünü yapar. Burada Laplace dönüşümünde kullanılan **s** ve **t** değişkenleri ile varsa **a, b** gibi parametrelerin **syms** komutu ile önceden tanımlanması gerekir.

* Laplace dönüşümü sonucunu güzelleştirmek için **“pretty”** komutu kullanılır.

|  |  |
| --- | --- |
| **UYGULAMA-1: f(t) = 3cost - sint** fonksiyonunun Laplace dönüşümünü yapınız. |  |
|  |

**2-) Ters Laplace Dönüşümü: f=ilaplace(F)** komutu MATLABortamında tanımlanmışbir **F** fonksiyonunun sembolik ters Laplace dönüşümünü yapar. Burada Laplace dönüşümünde kullanılan **s** ve **t** değişkenleri ile varsa **a, b** gibi parametrelerin **syms** komutu ile önceden tanımlanması gerekir.

|  |  |
| --- | --- |
| **UYGULAMA-2: F(s) =** fonksiyonunun ters Laplace dönüşümünü yapınız. |  |
|  |

**3-) Kısmi Kesirlere Ayırarak Ters Laplace Dönüşümü: [r p k]=residue(pay,payda)** komutu ile bir fonksiyonun kısmi kesirlerinin **çarpanları(r)**, **kutupları(p)** ve **kalanları(k)** hesaplanabilir. (r, p ve k değişkendir, o yüzden başka harflendirme de yapılabilir.) Sonra elde edilen kısmi kesirlerin ters Laplace dönüşümü alınarak toplam fonksiyonun ters Laplace dönüşümü hesaplanabilir. Kısmi kesir gösterimi:



|  |  |
| --- | --- |
| **UYGULAMA-3: F(s) =** fonksiyonunun kısmi kesir dönüşümünü yapınız. |  |
|  |

**Sonuç olarak;**

**F(s) =**  şeklinde kısmi kesirlerine ayrılır.

**!!!** *Komut penceresindeki r, p ve k değerlerini kullanarak MATLAB’de kısmi kesir formatında yazılabilir mi? Nasıl?*

**Not-1:** Eğer payın derecesi paydanın derecesinden büyük veya eşit olsaydı o zaman k =[ ] boşküme olmazdı. Yani, k, payın paydaya bölünmesiyle oluşan bölüm değeridir.

**Not-2:** Pay veya payda, (s + a)(s + b) şeklinde çarpanlarına ayrılmış şekilde ifade edildiyse; s2 + (a + b)s + ab şekline dönüştürmeye gerek kalmadan **conv([1 a],[1 b])** biçiminde yazılabilir. Yani, **conv (convolution)** komutu çarpma işlemi yapmaya yarar.

**4-) Diferansiyel Denklem Çözümü: dsolve** komutu ile diferansiyel denklemler çözülebilir.Komut içinde **d/dt** ifadesi **“ D”** ile temsil edilir. Bu komutun çözümünden elde edilen fonksiyon içinde bir sabit bulunması istenmiyorsa bu durumda başlangıç koşullarını program içinde belirtilmesi gerekir. Aksi takdirde çözüm fonksiyonu c sabitlerini bulunduracaktır. Alttaki formatta olduğu gibi yazılır.

y = dsolve(‘D.türevin derecesi.y + …’, ‘başlangıç değeri1=, başlangıç değeri2=, …’)

**Örneğin; y’nin üçüncü türevi için “D3y” yazılmalıdır.**

|  |  |
| --- | --- |
| **UYGULAMA-4:** diferansiyel denkleminin çözümünü y(0)=0 ve y’(0)=1 (1. türev) başlangıç koşulları altında çözünüz. |  |
|  |

* 1. **DENEY SONRASI ÇALIŞMA SORULARI**

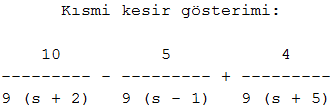
**UYGULAMA-1:** Uygulama-3’teki fonksiyonu kısmi kesirlerine ayırmıştık. Sonuç olarak r, p ve k değerleri altta olduğu gibi elde edilmişti. MATLAB’te **matris işlemlerini** kullanarak fonksiyonu ekrana kısmi kesir formatında yazdırınız. *Örneğin; r(2,1) = 2’dir.*

|  |  |
| --- | --- |
|  | **Kısmi kesir formatı**  *\*Komut penceresinde alttaki gibi görünecektir.*  **F(s) =** |

**CEVAP-1:**

**UYGULAMA-2: F(s) =** fonksiyonunun kısmi kesir dönüşümünde payın ve paydanın katsayılarını elde ederken **“conv” komutunu** kullanınız. (**!!!** *Not-2’de ifade edilmişti*.)

**CEVAP-2: Komut penceresindeki ekran görüntüsü**

****

**DENEY NO:** 3

**DENEY ADI:** Transfer Fonksiyonu ve Durum Uzay Gösterimleri

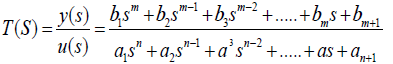
* 1. **DENEYİN AMACI**

1. Verilen sisteme ait transfer fonksiyonlarını farklı biçimlerde elde etmek
2. Verilen sistemin durum uzay gösterimlerini farklı biçimlerde ifade etmek
   1. **TEORİK BİLGİLER VE UYGULAMALAR**

Bir sistem doğrusal ve zamanla değişmez olduğu sürece transfer fonksiyonu belirlenebilir. Doğrusal sistemlerde sistemin giriş ve çıkış büyüklüklerinin Laplace dönüşümü alınır ve birbirine oranlanır ise sisteme ilişkin transfer fonksiyonu elde edilir.

Transfer fonksiyonu [ **T(s)** ], kısaca çıkışın[ **y(s)** ] girişe[ **u(s)** ] oranlanmasıyla elde edilir.

n ≥ m olmak üzere;



biçiminde ifade edilir. Polinom şeklinde verilen transfer fonksiyonu değişik şekillere sokulabilir. Aşağıda bir sistemin **Sıfır-Kutup-Kazanç** şeklinde gösterimi verilmiştir.

* z’ler sistemin sıfırları
* p’ler sistemin kutupları
* k ise sisteme ait bir kazanç katsayısıdır.

1. **Transfer Fonksiyonlarının Tanımlanması ve Analizi:**

MATLABortamında transferfonksiyonu nesnelerini tanımlamak için **“*tf*** ***”*** veya **“*zpk”*** fonksiyonları kullanılır.

* 1. **tf komutu ile transfer fonksiyonu tanımlama:**

|  |
| --- |
| **UYGULAMA-1:** Transfer fonksiyonu T(s) = olan sistemi ***polinom biçiminde*** MATLABortamında tanıtınız. |
|  |

* 1. **zpk komutu ile transfer fonksiyonu tanımlama:**

|  |
| --- |
| **UYGULAMA-2:** Transfer fonksiyonu T(s) = olan sistemi ***sıfır-kutup-kazanç biçiminde***MATLABortamında tanıtınız. |
|  |

1. **Transfer Fonksiyonları Gösterim Biçimlerinin Birbirine Dönüşümü**

MATLABortamında transferfonksiyonu gösterimi biçimlerini birbirine dönüştürmek için **“*tf2zp”*** veya **“*zp2tf”*** fonksiyonları kullanılır.

1. **Polinom biçiminden sıfır-kutup-kazanç biçimine dönüşüm (*tf2zp*)**

|  |  |
| --- | --- |
| **UYGULAMA-3:** Transfer fonksiyonu T(s) =  polinom biçiminde olan sistemi ***sıfır-kutup-kazanç biçimine*** dönüştürünüz. | |
| 1.  2.  3.  T(s) = 5. şeklinde ifade edilir. |  |

1. **Sıfır-kutup-kazanç biçiminden polinom biçimine dönüşüm (*zp2tf*)**

|  |  |
| --- | --- |
| **UYGULAMA-4:** Transfer fonksiyonu T(s) = sıfır-kutup-kazanç biçiminde olan sistemi ***polinom biçimine*** dönüştürünüz. | |
|  | |
| T(s) = elde edilmiş olur. |  |

1. **Durum Uzay Gösterimleri**

Lineer veya lineer olmayan tek girişli tek çıkışlı veya çok girişli çok çıkışlı sistemlerin analiz ve tasarımında matris özelliklerinden faydanıldığı durum denklemleri kullanılır.

**Not: Burada bilinmesi gereken, durum uzayı gösteriminin tek olmadığıdır. Yani aynı sistemi temsil eden pek çok durum uzayı gösterimi bulunabilir.**

Kontrol sistemlerindeki durum uzay (**S**tate-**S**pace) modeli aşağıdaki gibidir.

1. **Transfer fonksiyonu verilen sistemlerde durum uzay modelinin elde edilmesi (*tf2ss*)**

Polinom şeklinde ifade edilen bir sistemin durum-uzay modeli elde edilebilir.

**[ a b c d ] = tf2ss ( pay, payda ):** Bu komut,polinom şeklinde verilen bir transferfonksiyonunun yardımıyla sistemin durum uzay modelini bulur.

|  |  |
| --- | --- |
| **UYGULAMA-5:** T(s) = şeklinde transfer fonksiyonu verilen bir sistemin durum uzaymodelini elde ediniz. | |
| Yanda elde edilen sonuçlara göre sisteme ilişkin durum uzay modelinin matrissel gösterimi aşağıdaki gibi olur. |  |

1. **Sıfır-Kutup-Kazanç biçiminde transfer fonksiyonu verilen sistemlerde durum uzay modelinin elde edilmesi (*zp2ss*)**

**[ a b c d ] = zp2ss (z, p, k ):** Bu komut sistemin sıfır-kutup-kazanç şeklinde verilentransfer fonksiyonu yardımı ile sistemin durum uzay modelini bulur. a, b, c, d matrisleri durum uzay modeline ait matrislerdir.

|  |  |
| --- | --- |
| **UYGULAMA-6:** T(s) = şeklinde transfer fonksiyonu verilen bir sistemin durum uzaymodelini elde ediniz. | |
| Yanda elde edilen sonuçlara göre sisteme ilişkin durum uzay modelinin matrissel gösterimi aşağıdaki gibi olur.  **Açıklama:** Bir önceki uygulama ile karşılaştırıldığında durum uzay modeli farklı çıkmasınarağmen her ikisi de aynı sistemi ifade eder. |  |

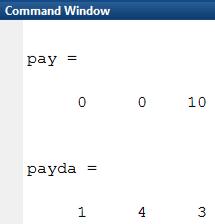
1. **Durum** **uzay** **modeli** **verilen** **sistemlerde** **transfer** **fonksiyonunun** **elde edilmesi (*ss2tf*)**

Eğer sisteme ilişkin durum uzay modeli biliniyor ise buradan hareketle transfer fonksiyonuna ulaşılabilir. İlk başta A, B, C ve D matrisleri tanımlanır ve ***ss*** komutuyla durum uzay modeli oluşturulur. Daha sonra ***tf*** komutu ile transfer fonksiyonu bulunur.

|  |
| --- |
| **UYGULAMA-7:** Sistemin durum uzaymodelinden transfer fonksiyonunu elde ediniz. |
|  |

Ayrıca, **[pay, payda]= ss2tf(A, B, C, D):** Bu komut sistemin durum uzay modeli yardımıyla transfer fonksiyonunu bulur. A, B, C, D durum uzay modeline ilişkin matrisleri gösterir. Pay vektörü T transfer fonksiyonunun payına ilişkin polinomun katsayılarını, payda vektörü ise T transfer fonksiyonunun paydasına ilişkin polinomun katsayılarını gösteren satır vektörlerdir.

|  |  |
| --- | --- |
| **UYGULAMA-8:** Bir önceki uygulamayı tekrar ediniz. | |
|  |  |

****

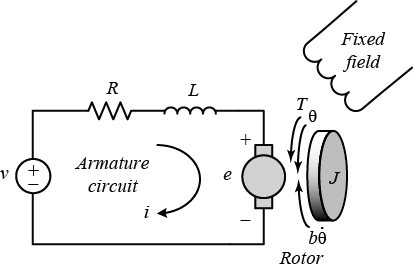
**DENEY NO:** 4

**DENEY ADI:** Fiziksel Sistemlerin Modellenmesi

* 1. **DENEYİN AMACI**

1. DC motorun elektriksel ve mekanik denklemlerini çıkartmak
2. DC motoru MATLAB ortamında modellemek
   1. **TEORİK BİLGİLER VE UYGULAMALAR**

Kontrol sistemlerinde çok kullanılan sistemlerden biri de DC motordur. DC motorun elektriksel ve mekanik modeli aşağıda görülmektedir.



Bu modelden yola çıkarak DC motora ilişkin diferansiyel denklemleri yazabiliriz. Motorun ürettiği moment (Tork, T), endüvi (armatür) akımı (i) ile Kt sabiti ile çarpımına eşittir. Ters emk kuvveti (e) ile açısal hız (ω) arasında da şu ilişki vardır:

**(1)**

**(2)**

Burada Kt tork sabiti, Kb ise zık emk sabitidir.

Modelin elektriksel devresine Kirchoff Gerilimler Kanunu, mekanik devresine Newton Kanunu uygulanarak alttaki gibi matematiksel denklemler elde edilir.

**(3)**

**(4)**

1. ve (2) nolu denklemleri (3) ve (4)’te yerine koyarak alttaki denklemler elde edilir.

**(5)**

**(6)**

Sistemin durum uzay modelini elde etmek için türevli parametreler durum değişkenleri olarak belirlenir ve genel durum uzay gösterimi şeklinde yazılır.

Aynı DC motor Laplace dönüşümü uygulanarak da modellenebilir. Bunun için alttaki adımlar uygulanır.

1. **Adım:** (5) ve (6) nolu denklemler kullanılarak Laplace dönüşümü alınır.

**(7)**

**(8)**

1. **Adım:** (7) nolu denklemden ***I(s)***, (8) nolu denklemden ise ***W(s)*** çekilir. (Durum uzayda olduğu gibi yine türevli kısımlar (durum değişkenleri) yalnız bırakılır.)

**(9)**

**(10)**

|  |
| --- |
| **UYGULAMA-1:** V(s) giriş, W(s) çıkış olmak üzere (9) ve (10) nolu denklemleri kullanarak sistemin blok şemasını çiziniz. |
|  |

|  |
| --- |
| **UYGULAMA-2:** V(s) giriş, W(s) çıkış olmak üzere sistemin durum uzay modelinden faydalanarak blok şemasını çiziniz. |
|  |
| **UYGULAMA-3:** V(s) giriş, W(s) çıkış olmak üzere sistemin durum uzay modelinden faydalanarak transfer fonksiyonunu MATLAB ortamında bulunuz. |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | R=2 Ω | L=0,5 H | J=0,01 kg.m2 | B=0,06 N.m.s | Kb=0,02 V/rad/sec | Kt=0,03 N.m/A |   1-  2-  3-  4-  5- |
|  |
| **UYGULAMA-4:** Uygulama-1’deki blok şemayı kullanarak transfer fonksiyonu elde ediniz.  Her bir bloğun transfer fonksiyonunu bulun. Geri besleme için ***“feedback”*** komutunu kullanın. |
| 1-  2-  3- |
| **UYGULAMA-5:** Hem durum uzay modelinde hem de transfer fonksiyonunda sistemin girişine birim basamak ***(step)*** sinyali uygulayınız ve hız-zaman grafiklerini çizdiriniz.  Eksenlere ve grafiklere başlık ekleyiniz. x-ekseni 🡪 **xlabel(‘…..’)**, y-ekseni 🡪 **ylabel(‘…..’),** başlık **🡪 title(‘…..’)**  x-ekseni aralığı **0-3 saniye**, y-ekseni aralığı **0-0.5 rad/s** olacak şekilde ayarlayınız. ***(axis)*** |
| 1-  2-  3-  4-  5-  6- |

**DENEY NO:** 5

**DENEY ADI:** Blok Şemaların İndirgenmesi

* 1. **DENEYİN AMACI**

1. Kontrol sistemlerine ait blok şemaların indirgemesini yapmak
   1. **TEORİK BİLGİLER VE UYGULAMALAR**

MATLAB ortamında blok şema indirgeme işlemleri için “**series”, “parallel”** ve “**feedback”** komutları kullanılır.

**1-) Seri Bağlı Blokların İndirgenmesi (Çarpma İşlemi)**

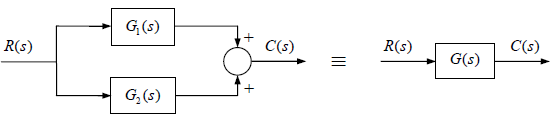
Seri bağlı blokların indirgenmesinde “**series(sistem1, sistem2)”** komutu veya çarpma ( \* ) işlemi kullanılır.



|  |
| --- |
| **UYGULAMA-1:** G1(s) ve G2(s) verildiğine göre G(s)’yi MATLAB ortamında bulunuz. |
| 1-  2-  3-  4-  5- |

**2-) Paralel Bağlı Blokların İndirgenmesi (Toplama veya Çıkarma İşlemi)**

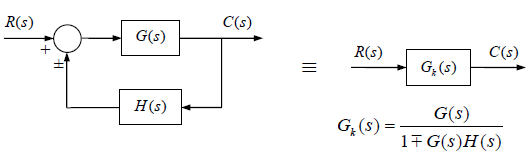
Paralel bağlı blokların indirgenmesinde “**parallel(sistem1, sistem2)”** komutu veya toplama/çıkarma (**+/-**) işlemi kullanılır.



|  |
| --- |
| **UYGULAMA-2:** G1(s) ve G2(s) verildiğine göre G(s)’yi MATLAB ortamında bulunuz. |
| 1-  2-  3-  4-  5-  6-  7-   |  |  | | --- | --- | |  |  | |

**3-) Geri Besleme ile Bağlı Blokların İndirgenmesi**

Pozitif veya negatif geri besleme ile bağlı blokların indirgenmesinde “**feedback(sistem1,sistem2)”** komutu kullanılır.

****

|  |
| --- |
| **UYGULAMA-3:** G(s) ve H(s) verildiğine göre Gk(s)’yi MATLAB ortamında bulunuz. |
| 1-  2-  3-  4-  5-  6-  7-   |  |  | | --- | --- | | **Negatif Geri Besleme** | **Pozitif Geri Besleme** | |

**4-) Karmaşık Blok Şemaların İndirgenmesi**

**NOT:** Blok indirgeme yaparken **iç döngüden dış döngüye doğru** işlem yapılır.

|  |
| --- |
| **UYGULAMA-4:** Aşağıda blok şeması verilen sistemi indirgeyerek transfer fonksiyonunu bulunuz. |
| 1-  2-  3-  4-  5-  6-  7- |

|  |
| --- |
| **UYGULAMA-5:** Aşağıda blok şeması verilen sistemi indirgeyerek transfer fonksiyonunu bulunuz. |
| 1-  2-  3-  4-  5-  6-  7- |