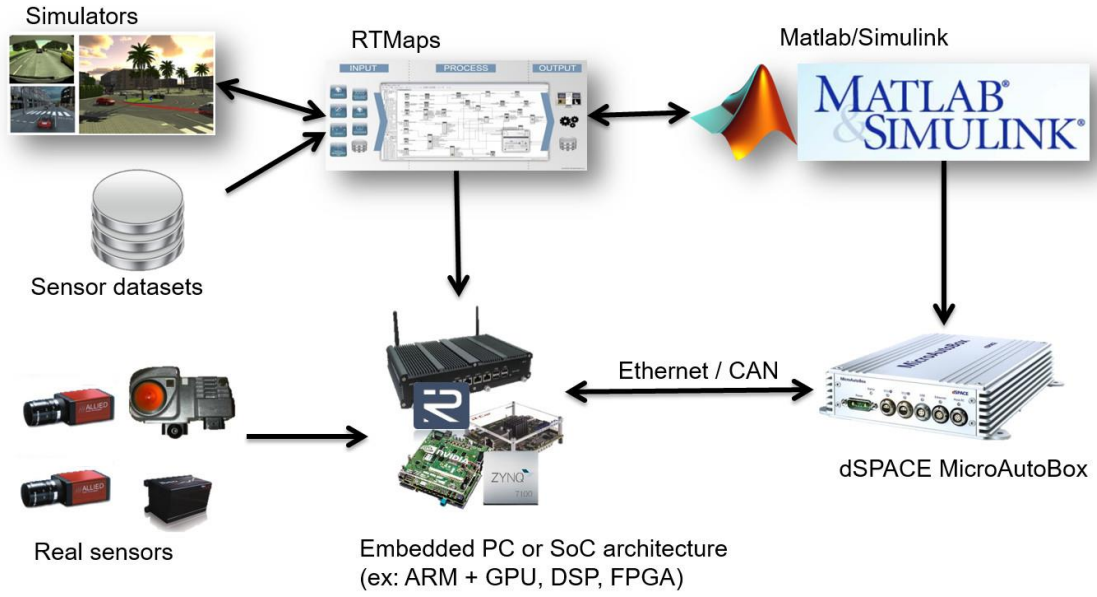


KONTROL SİSTEMLERİ-2 LABORATUVARI DENEY FÖYÜ



Dr. Öğr. Üyesi Güzin ÖZMEN
Arş. Gör. Fehmi SEVİLMİŞ

Konya, 2019

İÇİNDEKİLER

Laboratuvar Kuralları ve Deney Grupları.....	3
Deney-1: MATLAB/Simulink Programının Kullanımı.....	5
Deney-2: DC Motorun Modellenmesi	13
Deney-3: DC Motorun Hız Kontrolü.....	21
Deney-4: DC Motorun Hız Kontrolü	33
Deney-5: Ziegler–Nichols Yöntemi ile PID Parametrelerinin Bulunması.....	37

Kontrol Sistemleri-2 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.

ACIKLAMALAR

- 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	15 Mart	22 Mart	15 Mart	22 Mart
Deney-2	29 Mart	29 Mart	29 Mart	29 Mart
Vize Uygulama Sınavı	1 - 5 Nisan			
Deney-3	26 Nisan	26 Nisan	26 Nisan	26 Nisan
Deney-4	3 Mayıs	10 Mayıs	3 Mayıs	10 Mayıs
Deney-5	17 Mayıs	24 Mayıs	17 Mayıs	24 Mayıs
Final Uygulama Sınavı	27 - 31 Mayıs			

****Deney-2 (B-Grupları), NÖ için 10.30'da, İÖ için 18.30'da yapılacaktır.***

*****Deney-3 (B-Grupları), NÖ için 10.30'da, İÖ için 18.30'da yapılacaktır.***

******Diğer tüm deneyler, ders programında belirtilen saatlerde yapılacaktır.***

DENEY NO: 1

DENEY ADI: MATLAB/Simulink Programının Kullanımı

1.1 DENEYİN AMACI

- MATLAB/Simulink programının kullanılmasını öğrenmek
- Simulink ortamında basit sistemler tasarlayabilmek


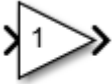



1.2 TEORİK BİLGİLER

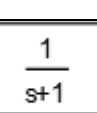
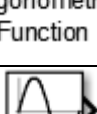
Mühendislik sistemlerinde simülasyonun önemi gün geçtikçe artmaktadır. Sistemlerin tasarımında büyük oranda bilgisayar simülasyonlarından faydalanılmakta, mümkün olduğunca tasarımın test aşamaları da bilgisayarlar yardımıyla yapılmaktadır. Bu da prototiplere olan ihtiyacı azaltarak maliyetlerin büyük oranda düşmesini sağlamaktadır.

Günümüzde mühendislik alanında en çok kullanılan programlardan birisi MATLAB'dir. Simulink, MATLAB ile birlikte bütünleşik olarak çalışan bir simülasyon ortamıdır. Simulink bize karmaşık sistemleri tasarlama ve simülasyon yapma olanağı vermektedir.

Birçok sistem, Simulink'in ihtiva ettiği zengin blok kütüphanesi ve blokları sürükle-bırak yöntemi ile çok kısa bir sürede kurularak simüle edilebilir, değişik durumlardaki cevabı test edilebilir.

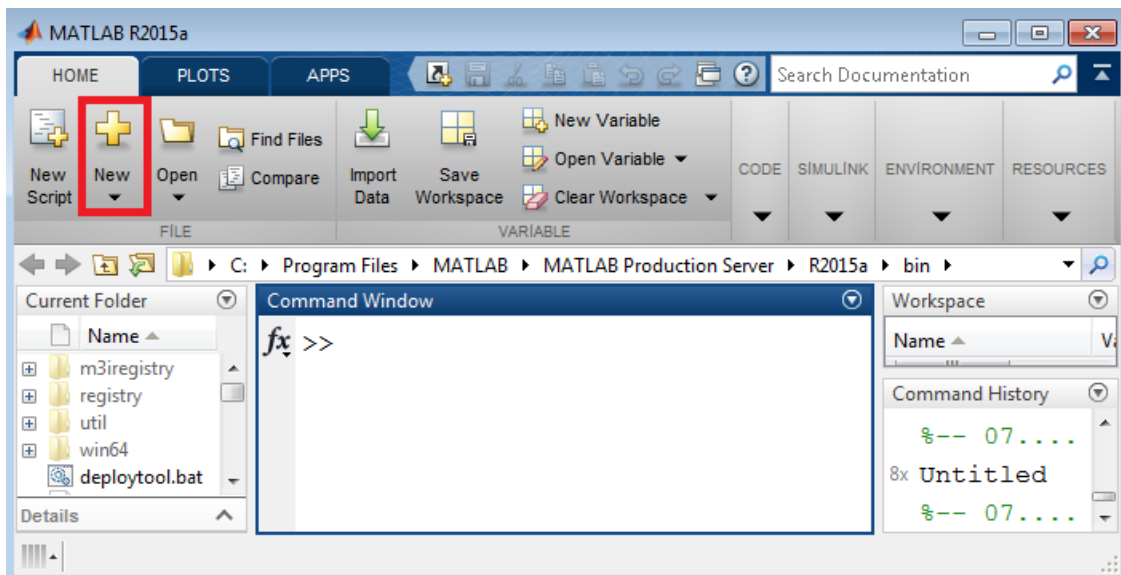
Simulink'te Sık Kullanılan Bloklar

 Integrator	İntegral Alıcı: Giriş sinyalinin zamana göre integralini hesaplar.
 Gain	Kazanç: Bu bloğa gelen giriş sinyali bir kazanç (sayısal bir değer) ile çarpılır.
 Sum	Toplama: Girişlerin cebirsel toplamını veren bir bloktur. Girişlerin sayısı ve her bir girişe uygulanacak işaret (+ / -), blok parametre ayarlarından düzenlenebilir.
 Product	Çarpma: Girişlerin cebirsel çarpımını veren bir bloktur. Girişlerin sayısı, blok parametre ayarlarından düzenlenebilir.
 Mux	Mux: Sonlu sayıda skaler giriş sinyallerini bir çıkış sinyali matrisi üretecek tarzda birleştiren bloktur. Giriş sayısı, blok parametre ayarlarından artırılabilir.

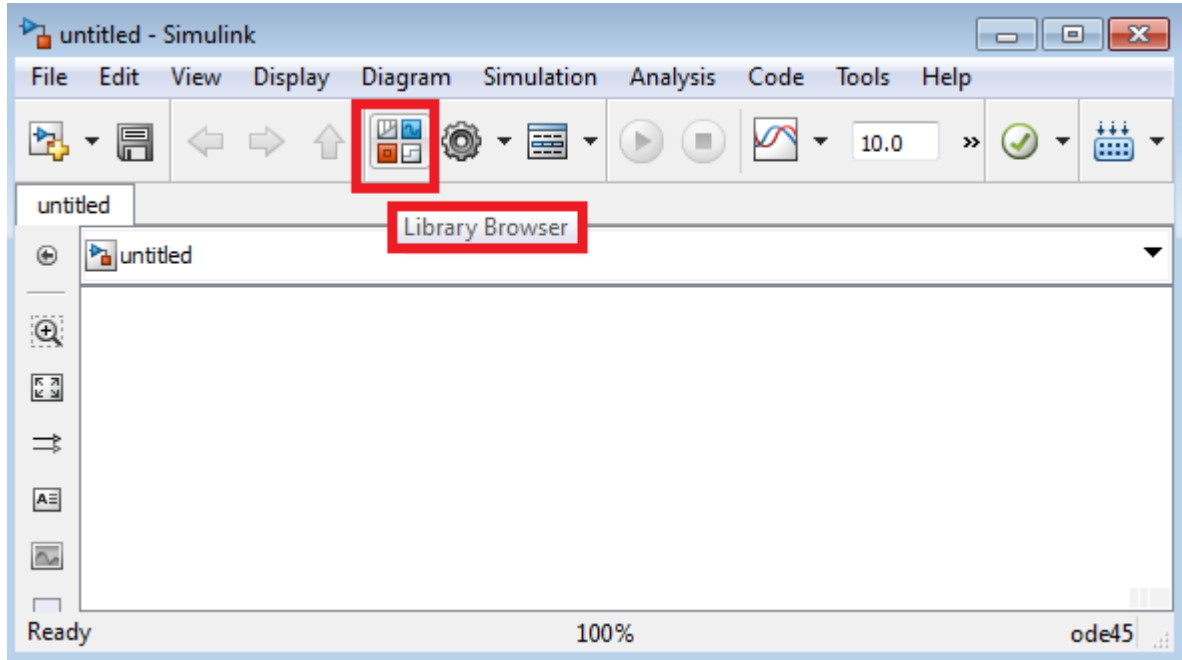
 Display	Gösterge: Giriş sinyalinin o anki değerini gösterir.
 Scope	Görüntüleme: Skaler veya vektör sinyallerini osiloskoptakine benzer tarzda grafik olarak gösteren bloktur.
 Constant	Sabit: Sabit bir sayısal değer üreten bloktur.
 Step	Basamak/Adım: Basamak/Adım fonksiyonu üretir.
 PID Controller	PID Denetleyici: PID kontrolörü ayarlayabilen bloktur.
 Transfer Fcn	Transfer Fonksiyonu: Bir sistemin transfer fonksiyonunu ifade eder.
 Trigonometric Function	Trigonometrik Fonksiyon: Blok parametre ayarlarından, sinüs dışındaki diğer trigonometrik fonksiyonlar ayarlanabilir.
 Sine Wave	AC Kaynak: Sinüzoidal bir AC kaynağı bu blok yardımıyla oluşturulur.

1.3 DENEYİN YAPILIŞI

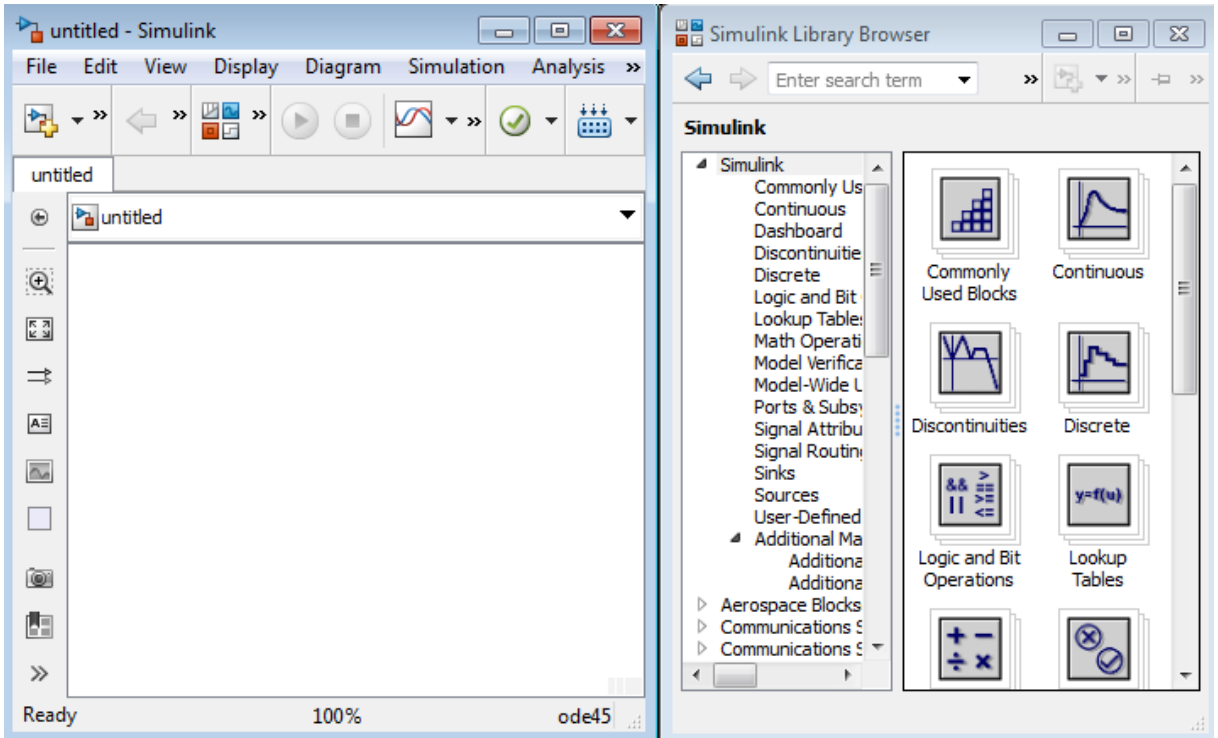
- 1. Adım:** “New” menüsünden  Simulink Model açınız.



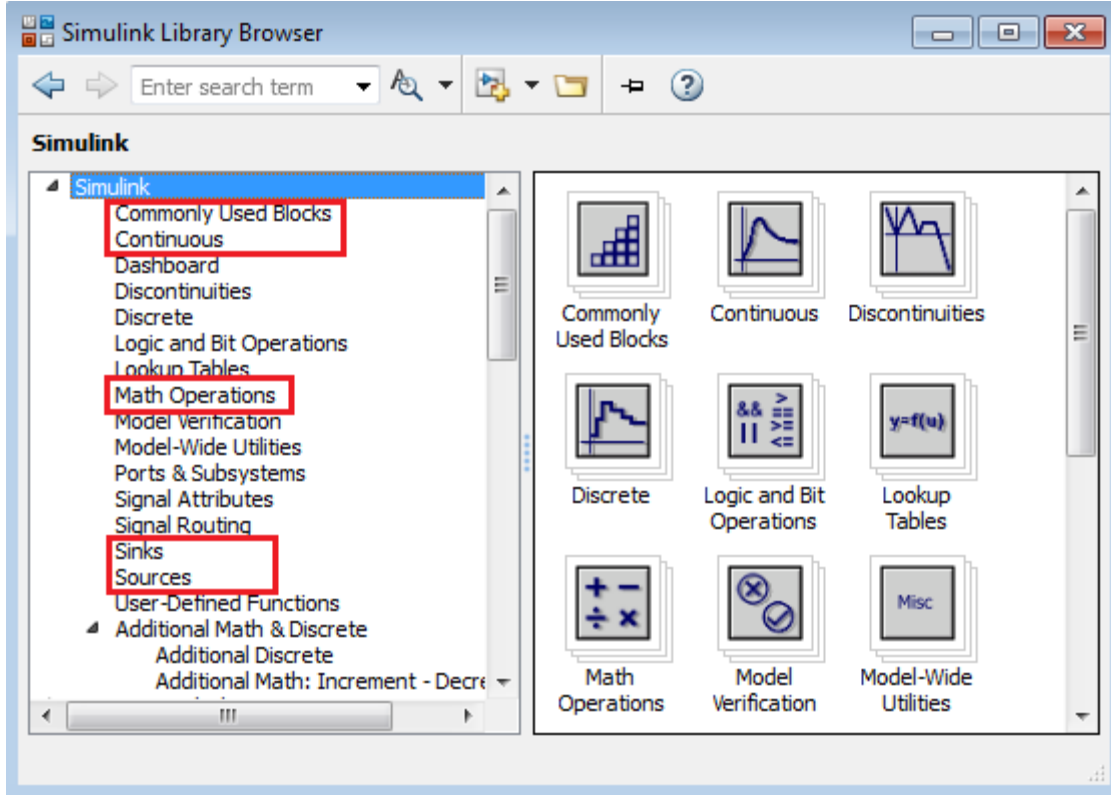
2. Adım: Simulink penceresinde “Kütüphane simgesine” tıklayınız.



3. Adım: “Simulink çalışma penceresi” ile “kütüphane penceresi” yan yana gelecek şekilde bilgisayar ekranını ayarlayınız.

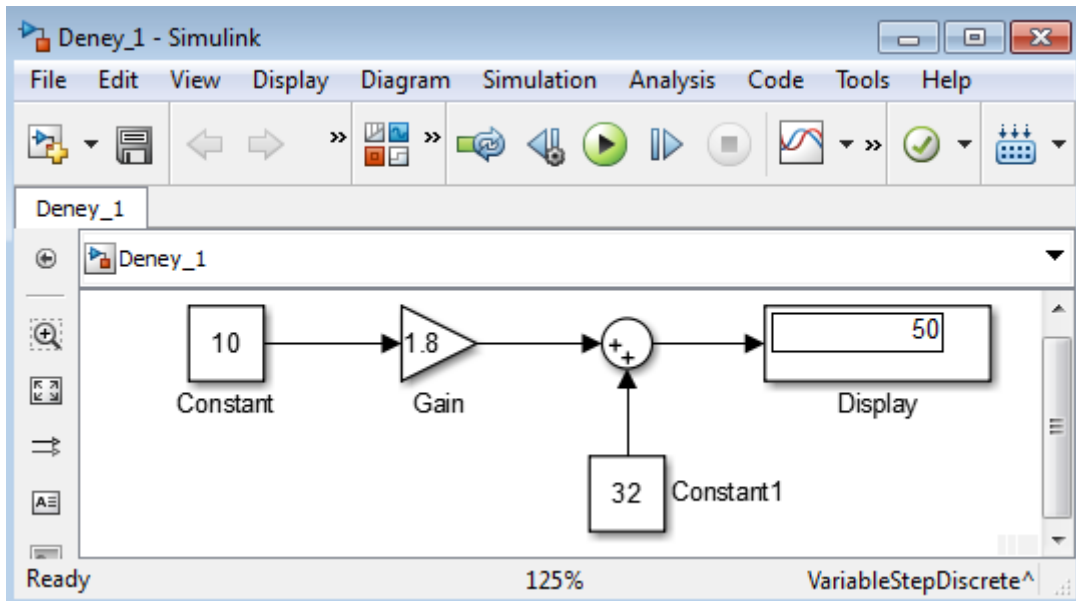


4. **Adım:** 5. Adımdaki örnek uygulamaya geçmeden önce, çok sık kullanacağınız blokların kütüphaneleri altta kırmızı kutu içine alınmıştır. Buradan hareketle bloklarınızı seçebilirsiniz.

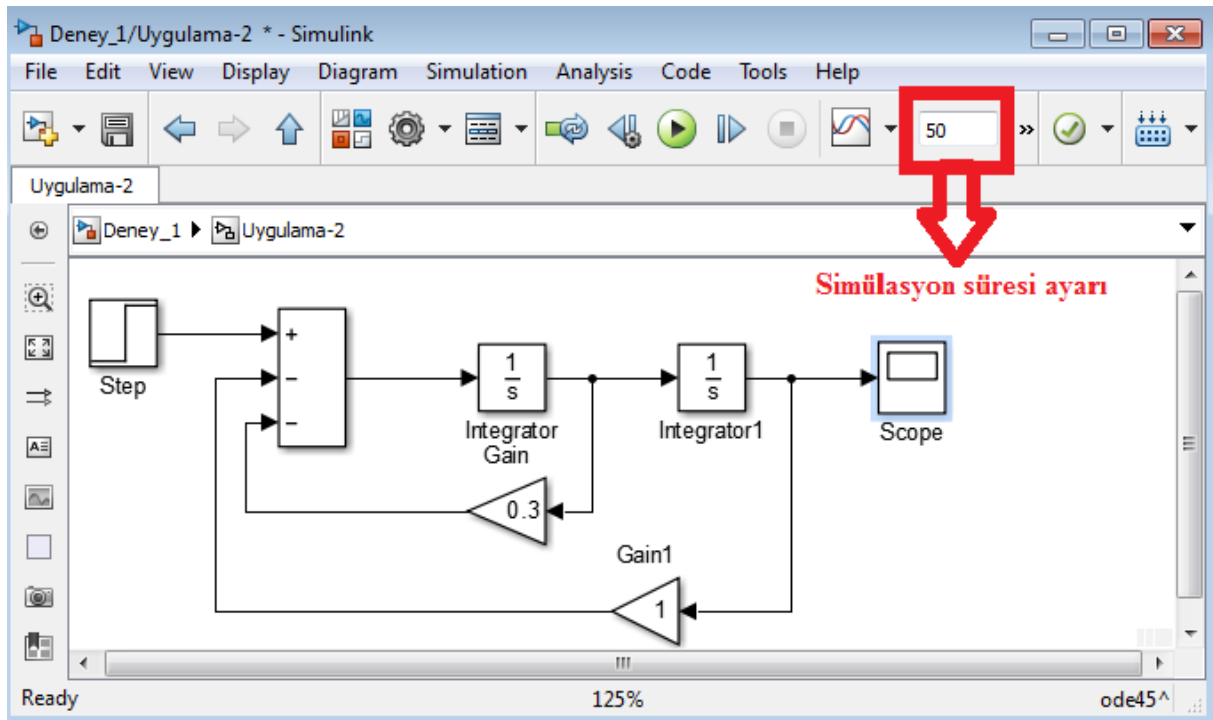
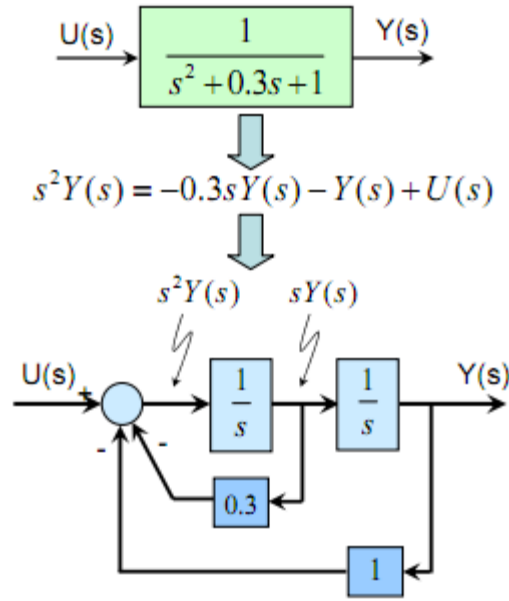


5. **Adım:** Celcius (Santigrat) sıcaklık derecesini, Fahrenheit sıcaklık derecesine dönüştüren sistemi modelleyiniz. Kütüphaneden elemanları “**Simulink penceresine**” taşımak için “**sürükle-brak**” yöntemini kullanınız. Bloğu döndürmek için “**Ctrl +R**” kullanılır.

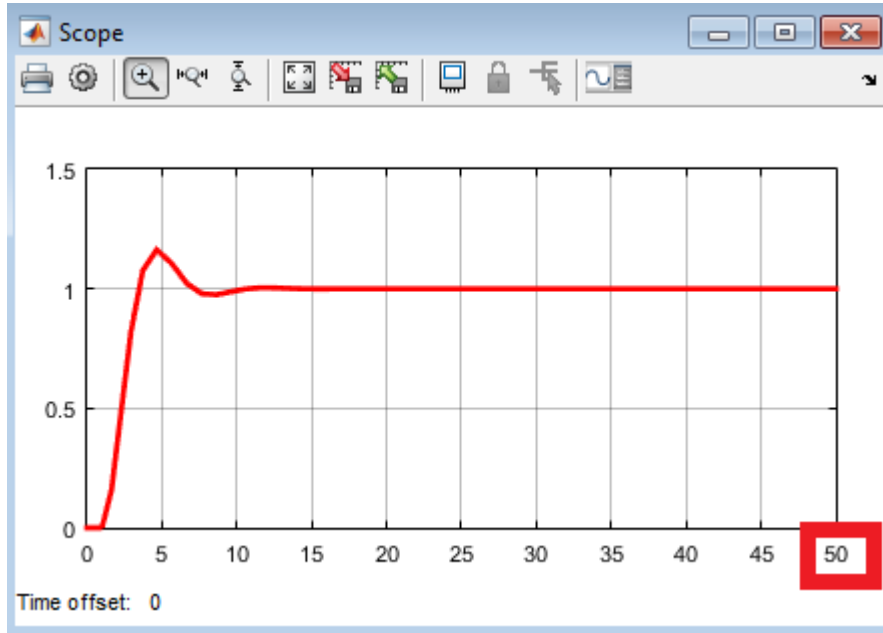
$$F = 1.8 * C + 32 \quad ({}^{\circ}C \rightarrow F \text{ dönüşüm formülü})$$



6. **Adım:** Alttaki sistemi Simulink'te modelleyiniz.



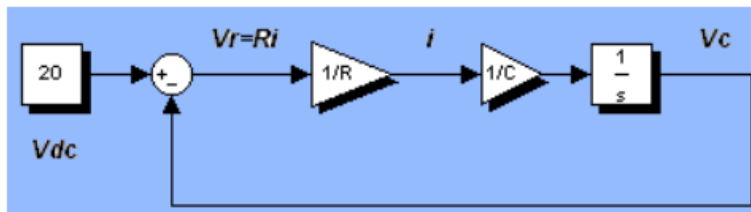
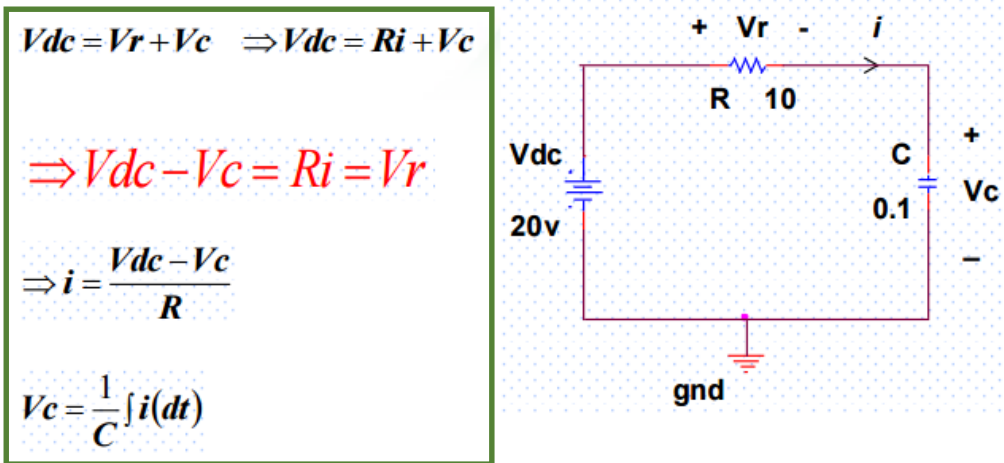
7. **Adım:** Simülasyon süresini **"50 saniye"** olarak belirleyin. Scope bloğu üzerine çift tıklayarak simülasyon çıktısını diğer sayfadaki gibi gözlemleyiniz.



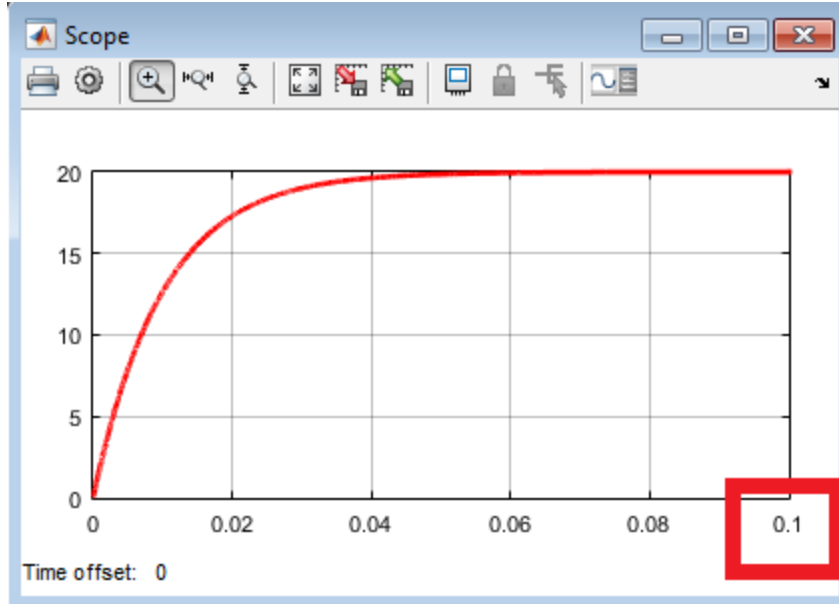
NOT: Deney-2’den itibaren Flash Bellek getiriniz.

1.4 DENEY SONRASI ÇALIŞMA SORULARI

Uygulama-1: Aşağıdaki “kondansatör şarj devresini” Simulink’te modelleyip “Vc gerilimini” gözlemleyiniz.



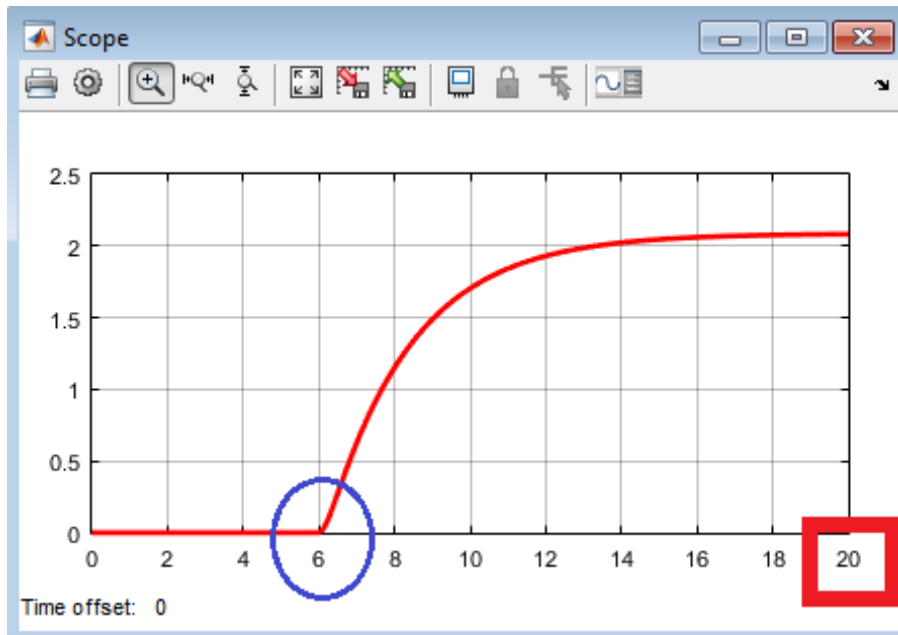
Sonuç-1: “Vc geriliminin deęişim grafięi” dięer sayfadaki gibi olmalıdır. Simülasyon süresi “0.1 s” olarak alınmıştır.



Uygulama-2: Aşağıdaki diferansiyel denklem takımı verilen sistemi Simulink'te modelleyiniz. Giriş için birim basamak sinyali kullanınız. **Çıkış sinyalini (y)** gözlemleyiniz.

$$\dot{x}_1 = x_2 \quad ; \quad \dot{x}_2 = -2,3x_1 - 5,6x_2 + 3u \quad ; \quad y = 1,6x_1$$

Sonuç-2: Simülasyon süresi “20 s” olarak alınmıştır. Neden “6. Saniyeye” kadar çıkış sinyali sıfır kalıp bu zamandan sonra deęişim meydana gelmiştir?



DENEY NO: 2

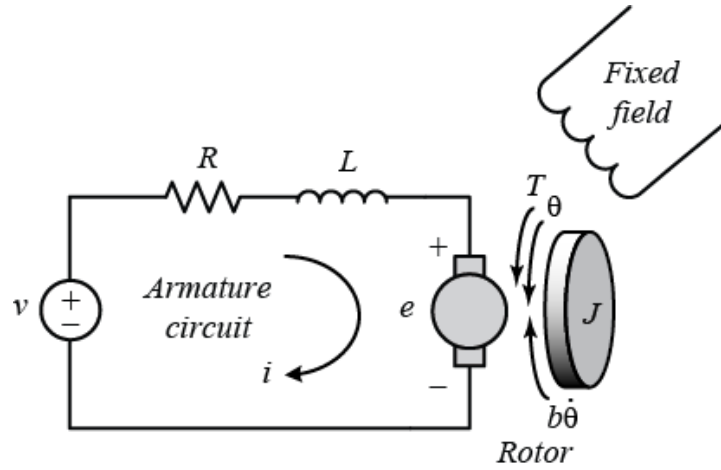
DENEY ADI: DC Motorun Modellenmesi

2.1 DENEYİN AMACI

- i. DC motorun elektriksel ve mekanik denklemlerini çıkartmak
- ii. Simulink ortamında DC motoru modellemek

2.2 TEORİK BİLGİLER

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 ve elde ettiğimiz model sayesinde gerekli simülasyonları gerçekleştirebiliriz. Motorun ürettiği moment (Tork, T), endüvi (armatür) akımı (i) ile K_t sabiti ile çarpımına eşittir. Ters emk kuvveti (e) ile açısal hız (ω) arasında da şu ilişki vardır:

$$T = K_t i \quad (1)$$

$$e = K_b \omega \quad (2)$$

Burada K_t tork sabiti, K_b ise zık emk sabitidir.

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

$$v = Ri + L \frac{di}{dt} + e \quad (3)$$

$$T = J \frac{d\omega}{dt} + B\omega \quad (4)$$

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

$$v = Ri + L \frac{di}{dt} + K_b \omega \quad (5)$$

$$K_t i = J \frac{d\omega}{dt} + B \omega \quad (6)$$

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

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

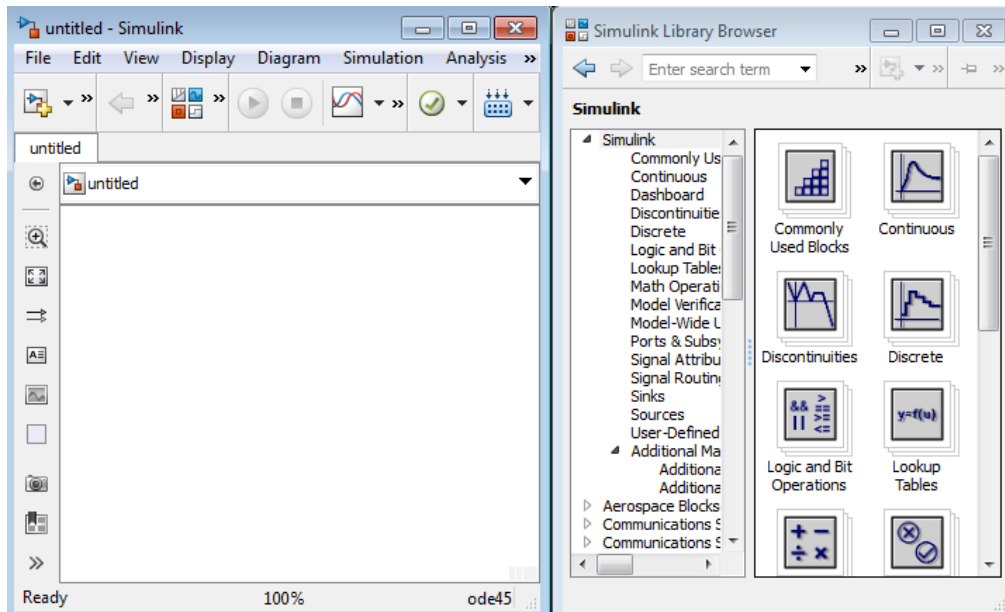
$$\begin{aligned} \frac{di}{dt} &= \frac{v}{L} - \frac{Ri}{L} - \frac{K_b \omega}{L} \\ \frac{d\omega}{dt} &= \frac{K_t i}{J} - \frac{B \omega}{J} \end{aligned}$$

$$\begin{bmatrix} di \\ d\omega \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_t}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ \omega \end{bmatrix}$$

2.3 DENEYİN YAPILIŞI

1. Adım: “Simulink çalışma penceresi” ile “kütüphane penceresi” yan yana gelecek şekilde bilgisayar ekranını ayarlayınız.



2. Adım: Elde edilen durum denklemlerinden faydalanarak sistemin blok şemasını alta çiziniz.

$$\frac{di}{dt} = \frac{v}{L} - \frac{Ri}{L} - \frac{K_b\omega}{L}$$
$$\frac{d\omega}{dt} = \frac{K_t i}{J} - \frac{B\omega}{J}$$

Sistemin Blok Şeması


3. Adım: “2. Adımda” çizdiğiniz blok şemayı Simulink ortamında oluşturunuz.


NOT-1: Kazanç (Gain) değerlerini harf şeklinde blok içerisine yazınız.

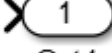
NOT-2: Giriş gerilimi (v) için herhangi bir şey yapmayınız.

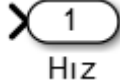
4. Adım: Çizdiğiniz blok şemanın tümünü seçiniz. Sağ tıklayınız → “**Create Subsystem from Selection**” seçiniz. Altteki blok yapı meydana gelecektir.



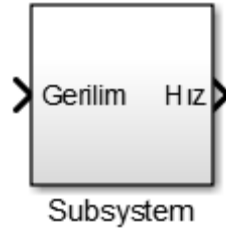
5. Adım: Elde ettiğiniz blok içerisine girerek  In1 blok ismini değiştirebilirsiniz. Yani, 4. Adımda oluşturduğunuz Subsystem'e (alt-sistem) dışarıdan giriş sinyali uygulayabilirsiniz.

Örnek: Gerilim → 

6. Adım: Subsystem'den çıkış sinyali alabilmek için içerisine  Out1 çıkış bloğunu ekleyin ve hız (ω) çıkışına bağlayınız. Bunun da ismini değiştirebilirsiniz.

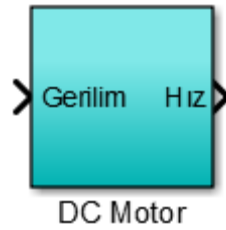
Örnek: Hız → 

7. Adım: Üstteki adımları yapınca bloğunuz alttaki gibi görünecektir.

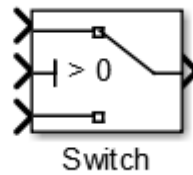


8. Adım: Arzu ederseniz bloğunuzun ismini ve rengini değiştirebilirsiniz. (Bloğun rengini değiştirmek için: Fareye sağ tıkla → Format → Background color → Renk seç)

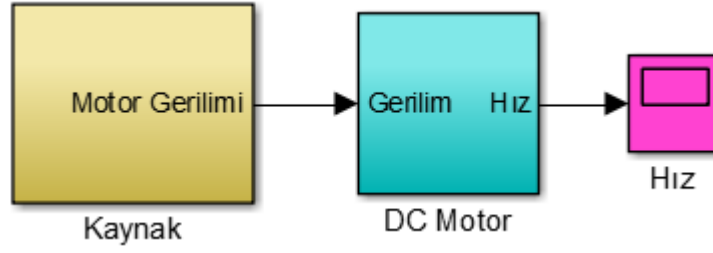
Örnek:



9. Adım: Motora 0-5 s aralığında 10 V, 5-15 s aralığında 30 V ve 15. s'den sonra 0 V uygulanacak şekilde Motor kaynak gerilimini ayarlayınız. (Altındaki blokları kullanınız.)



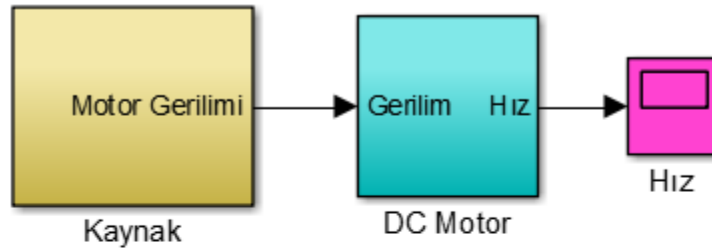
10. Adım: Motor kaynak bloğu ile Motor modeli alttaki gibi görünecektir.



11. Adım: Kazanç (Gain) bloklarına harf şeklinde parametreler girildiği için bunlara değer atamak gerekir. Yani, bu değerlerin “**Command Window (Komut Penceresi)**” kısmında koşturulması gerekir. Bunun için alttaki değerleri kullanınız.

NOT: Oluşturulan model içerisinde bu değerlerin saklanabilmesi için modelin boş bir yerine çift tıklanır ve bu değerler yazılır. Model çalıştırılmak istendiğinde buradaki değerler kopyalanarak “**Command Window (Komut Penceresi)**” kısmına yapıştırılır ve koşturulur. Altta modelin son hali görülmektedir.

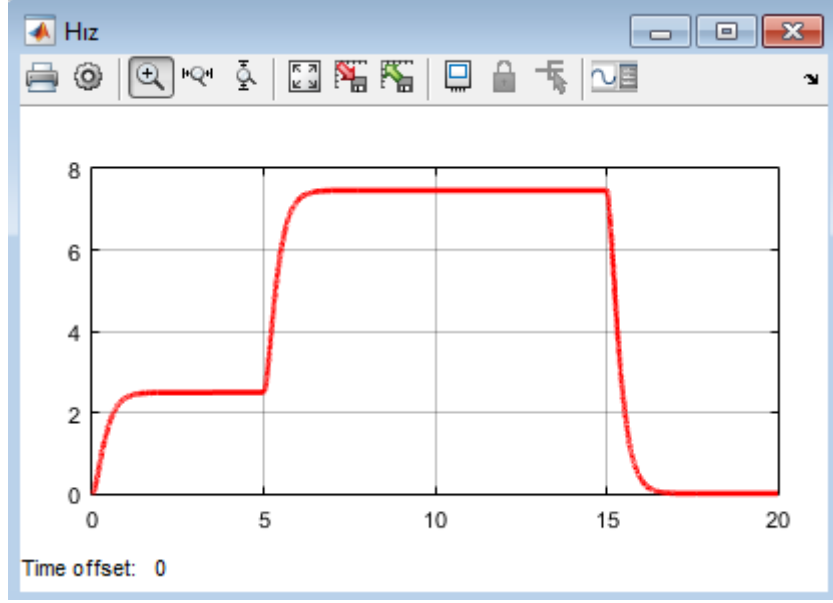
R=2 Ω	L=0,5 H	J=0,01 kg.m ²	B=0,06 N.m.s	K _b =0,02 V/rad/sec	K _t =0,03 N.m/A
-------	---------	--------------------------	--------------	--------------------------------	----------------------------



```
R = 2;  
L = 0.5;  
J = 0.01;  
B = 0.06;  
Kb = 0.02;  
Kt = 0.03;
```

12. Adım: Modeli koşturunuz ve **motorun hız [rad/s]–zaman [s] değişim grafiğini** diğer sayfadaki gibi elde ediniz. DC motor, açık çevrim (open loop) olarak kontrol edilmektedir.

13. Adım: Bundan sonraki deneyde (Deney-3), DC motorun kapalı çevrim (closed-loop) hız kontrolü gerçekleştirileceği için modelinizi “**Flash Belleğinize**” kaydediniz.



2.4 DENEY SONRASI ÇALIŞMA SORULARI

Uygulama-1: Deneyde DC motorun modellenmesini, durum uzay denklemlerinden faydalanarak yaptık. Aynı DC motoru Laplace dönüşümü uygulayarak Simulink'te modelleyiniz. Bunun için alttaki adımları uygulayabilirsiniz.

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

$$V(s) = RI(s) + LsI(s) + K_bW(s) \quad (2.1)$$

$$K_tI(s) = JsW(s) + BW(s) \quad (2.2)$$

- Adım:** (4.1) nolu denklemden $I(s)$, (4.2) nolu denklemden ise $W(s)$ çekilir. (Durum uzayda olduğu gibi yine türevli kısımlar (durum değişkenleri) yalnız bırakılır.)

$$I(s) = \frac{V(s) - K_bW(s)}{R + sL} \quad (2.3)$$

$$W(s) = \frac{K_tI(s)}{B + sJ} \quad (2.4)$$

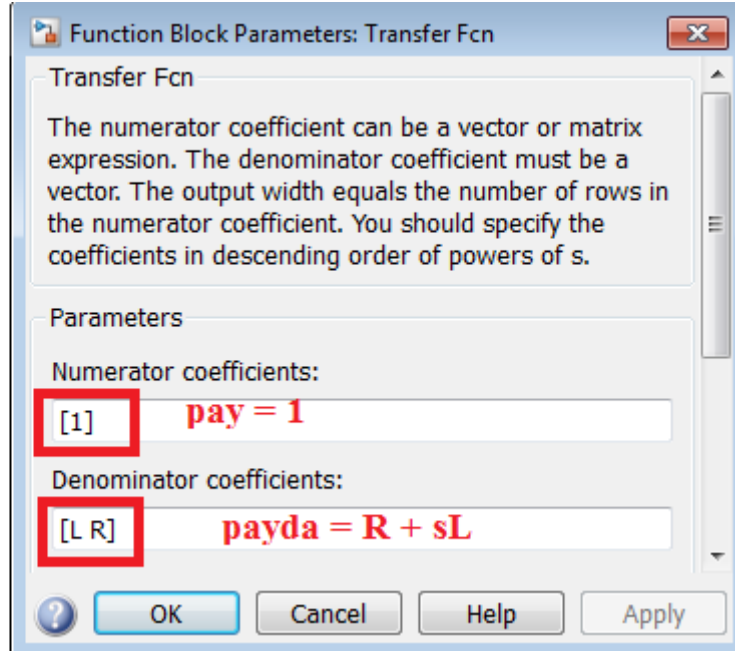
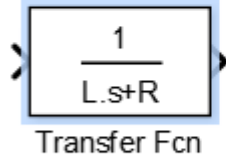
- Adım:** Sistemin girişi $V(s)$; çıkışı $W(s)$ olacak şekilde blok şeması çizilir.

Sistemin Blok Şeması

4. **Adım:** $(R + sL)$ ve $(B + sJ)$ fonksiyonları transfer fonksiyonu şeklinde yazılır. Bunun için kütüphaneden alttaki blok seçilir.

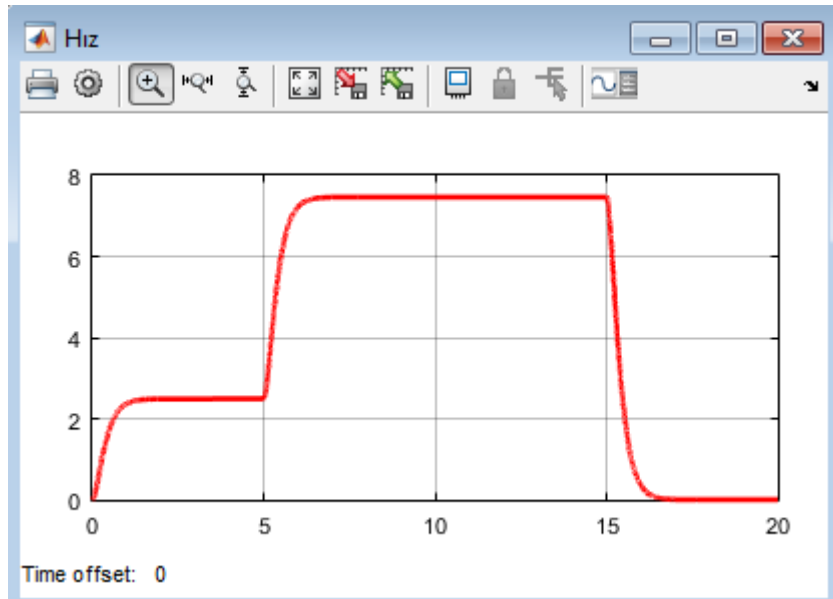


Bu bloğun parametreleri şu şekilde ayarlanır. **Örnek:** $(R + sL)$



5. **Adım:** Deneyde yapılan tüm adımlar takip edilir. Kaynak gerilimi ve DC motor parametreleri aynı şekilde düzenlenir.

Sonuç-1: "Hızın değişim grafiği" deneyde yaptığımız ile birebir aynı çıkar.



DENEY NO: 3

DENEY ADI: DC Motorun Hız Kontrolü

3.1 DENEYİN AMACI

- i. DC motorun hızını kontrol etmek
- ii. P-I-D kontrol etkilerini incelemek

3.2 TEORİK BİLGİLER

Bu deneyde, bir fiziksel sistem verildiğinde, bu sistemi kontrol etmek için temelde hangi adımların izlenmesi gerektiğinin kavranması amaçlanmaktadır. Bir kontrol sistemi oluşturulurken fiziksel sistemin dışında kontrol amacı ile eklenmesi gereken elemanların neler olabileceği Simulink ortamında gösterilecektir. Deney-2’de matematiksel modeli çıkarılarak modellenen DC motorun farklı kontrolcüler ile hız kontrolü gerçekleştirilecektir.

Kapalı çevrimin açık çevrime göre en önemli farkı, dış etkenlere ve iç değişimlere karşı hassasiyetin az oluşudur. Dolayısıyla kapalı çevrim sisteminde çok hassas olmayan ve kusurlu denebilecek bileşenler ile hassas kontrol gerçekleştirebilirken, açık çevrimde bu mümkün olmamaktadır.

Şekil 3.1’den hatırlanacağı gibi, bir kapalı çevrim denetim sisteminde geri besleme elemanı üzerinden alınan çıkış değeri, referans bir değer ile karşılaştırılır ve elde edilen hata sinyalinin yapısına ve kontrol edilen değişkene uygun bir kontrol sinyali üretilir. Bu kontrol sinyali uygun bir kontrol elemanı (P-I-D) ile denetlenerek sistemin kararlı çalışması sağlanır. Sürekli olarak geri besleme yapıldığından kontrol elemanı hatayı sıfırlamaya çalışır. Bu deneyde, kontrol elemanı olarak P, PI, PD veya PID kullanarak sisteme verdikleri tepkiler incelenecek ve DC motorun hızı kontrol edilecektir. Bu kontrol elemanlarının işleyişi tek tek açıklanmıştır.



Şekil 3.1 Kapalı çevrim kontrol sistemi

Kontrolcülere geçmeden önce sistemin birim basamak girişine karşı gösterdiği geçici durumun analizinin hatırlanması gerekir. Şekilden de görüldüğü gibi sistemin girişine birim basamak sinyali uygulanmış ve sistem çıkışı geçici durumda salınım yapmıştır. Burada;

t_r : yükselme zamanını (rising time, yani çıkış genliğinin girişe ulaşana kadar geçen süre)

t_p : maksimuma ulaşma zamanını (peak time)

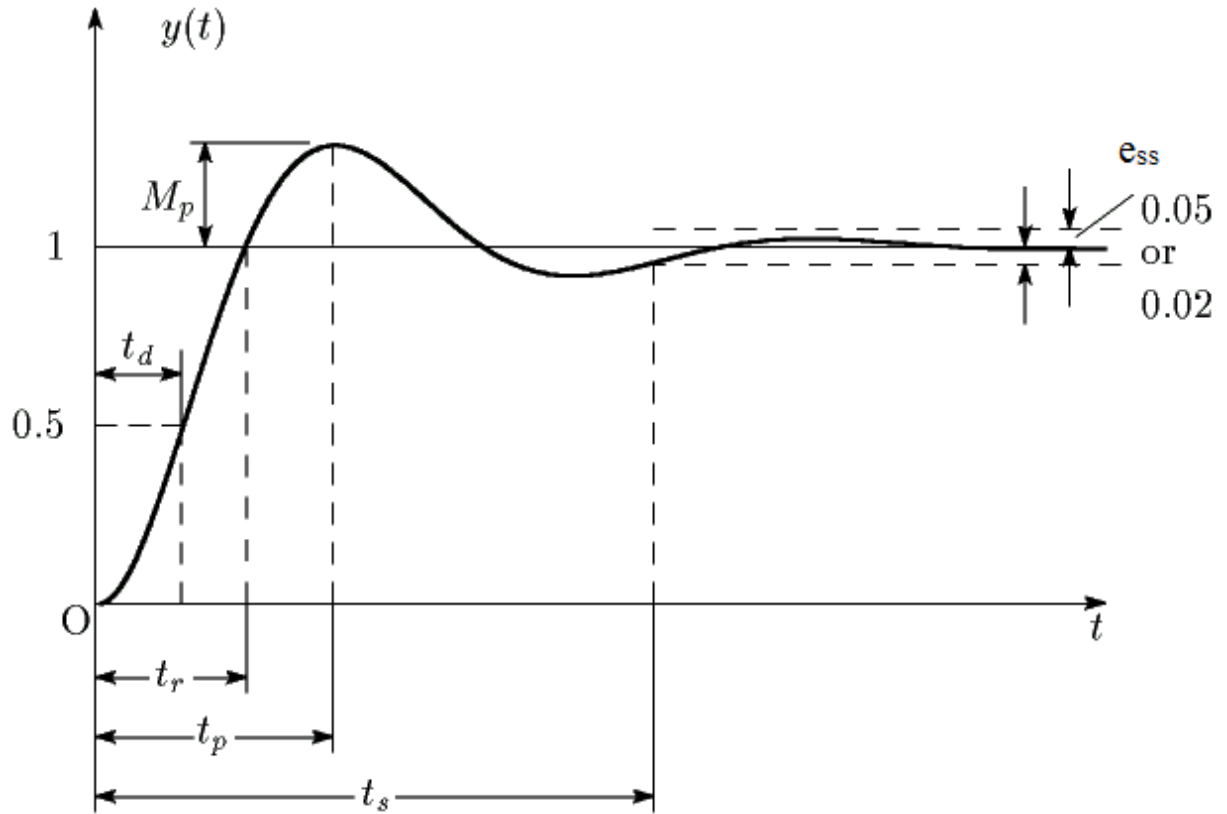
t_s : yerleşme zamanını (settling time)

t_d : gecikme zamanını (delay time, yani çıkış genliğinin girişin yarısına ulaşana kadar geçen süre)

M_p : Maksimum aşmayı

e_{ss} : sürekli durum hatasını (steady-state error)

ifade etmektedir.



A. Oransal (Proportional, P) Kontrolcü

Oransal kontrol algoritması en basit kontrol etkisini temsil eder. Kontrolcü çıkışı o andaki hata değeri ile bir kazanç değerinin (K_p) çarpımıyla hesaplanır. Bu kazanç, kontrolcü çıkışının sistem hatasına hassasiyetini gösterir. Kazanç büyüdükçe kontrolcü reaksiyonu artar. Kontrolcü çıkışı $m(t)$, hata $e(t)$ olmak üzere kontrolcünün tepkisi şu şekilde ifade edilir:

$$m(t) = K_p \cdot e(t)$$

K_P değeriindeki artışın, yükselme zamanını ve sürekli durum hatasını azalttığı, buna karşılık maksimum aşma değerini arttırdığı unutulmamalıdır. Yükselme zamanının azaltılması ile sistem tepkisi hızlanmış olur.

B. İntegral (Integral, I) Kontrolcü

Genellikle integral denetleyiciler, sadece kapalı-çevrim motor hız kontrol sistemi gibi birinci dereceden sistemlerde kullanılır. Yüksek dereceden sistemler genellikle kararsız olduğu için, I denetleyici tek başına nadir olarak kullanılır.

Integral kontrol etkisi oransal kontrol etkisine benzer şekilde hesaplanır, fakat burada anlık hata değeri yerine sistem çalışmaya başladığı andan itibaren olan tüm anlardaki hataların toplamı bir kazanç ile çarpılır. Sistem cevabının referans değerine ulaşması geciktikçe integral kontrol etkisi de artacaktır. Integral kontrol etkisi şu şekilde ifade edilir:

$$m(t) = K_I \int_0^t e(\tau) d(\tau)$$

I denetleyicisiz bir sistemde, basamak giriş için sürekli durum hatası, e_{ss}, sıfır değildir. I denetleyici ile sürekli durum hatası, sıfırdır. K_I'nin artması, tepkeyi hızlandırır ve aşma miktarını artırır.

C. Türevsel (Derivative, D) Kontrolcü

D denetleyicisi, genellikle tek başına kullanılmamasına rağmen, kontrol sistemleri üzerinde D denetleyicinin etkilerini anlamak, bir PD veya PID denetleyiciyi anlamaya yardımcı olacaktır.

Türevsel kontrol etkisi, sistemdeki hatanın değişimine orantılı olarak belirlenir. Türev işlemi sistem çıktısının hesaplandığı andan bir sonraki anda alacağı değere ilişkin bir veri üretir. Dolayısı ile türev kontrol etkisi sisteme bir öngörü kazandırır.

Türev değeri sistemdeki dalgalanmalar ile büyüyeceğinden bu durumdaki türevsel kontrol etkisi de yüksek olacaktır. Dolayısı ile sistem çıktısı daha az dalgalı bir davranış gösterecektir. Türevsel kontrol etkisi şu şekilde ifade edilir:

$$m(t) = K_D \frac{de(t)}{dt}$$

D. PI Kontrolcü

Integral kontrol etkisinin etkisi, oransal kontrol etkisi ile beraber kullanıldığında daha etkin bir biçimde belirgin olur. Integral etkisi daha önce de bahsedildiği gibi kontrolcünün devreye

girmesinden belirli bir süre sonra etkin hale gelmekteydi. Yine oransal kontrol etkisinin hata küçüldükçe zayıfladığı bilindiğine göre bu iki etkinin tamamlayıcı olarak kullanılabilceği kolaylıkla öngörülebilir. PI kontrol etkisi şu şekilde ifade edilir:

$$m(t) = K_P e(t) + K_I \int_0^t e(\tau) d(\tau)$$

Oransal etki hızlı bir şekilde sistem cevabını giriş değerine yaklaştırdıktan sonra etkisi kaybedecek bu noktadan sonra integral etki devreye girip kalıcı hatayı yok edecek ve sistemi yavaşta olsa tam referans değerine taşıyacaktır.

İntegral etkinin kullanım amacı sistemin değişen talepleri üzerinde yeterli bir kontrolör etkisi sağlamaktır. Eğer sistemden gelen bir talep yalnız başına P etkisi ile karşılanabiliyorsa I etkisinin kullanılması gereksizdir. Buna karşılık, sistemden oldukça sık aralıklarda yüksek miktarda talepler ortaya çıkıyorsa, yalnızca P etkisine sahip bir kontrolör bu talepleri karşılayamaz.

E. PD Kontrolcü

Türevsel kontrolün bu geçici rejim sırasında oluşan salınımları azaltma etkisi oransal kontrolün sebep olduğu aşım değerindeki artışı azaltacaktır. Bu sebep ile oransal kontrolün türevsel kontrol etkisi ile beraber kullanımı göreceli olarak daha iyi bir kontrolcü yapısı oluşturur. PD kontrol etkisi şu şekilde ifade edilir:

$$m(t) = K_P e(t) + K_D \frac{de(t)}{dt}$$

PD kontrolcünün, aşma değerini azalttığı, sistemin reaksiyon hızını arttırdığı ve sürekli duruma geçiş süresini (t_s) azalttığı öngörülebilir. Buna karşın, sürekli durum hatası (e_{ss}) göreceli olarak azalsa da sıfırlanmamaktadır.

Türev yöntemi sadece hata değiştiğinde kontrolör çıkışına katkıda bulunur. Bu sebeple bu yöntem her zaman oransal yöntemle ve bazen de bunlara ilave olarak integral yöntemiyle beraber kullanılır. Türev denetim yöntemi tek başına kullanılamaz.

F. PID Kontrolcü

P kontrolcünün yükselme zamanını azaltmaya etkisi vardır ve sürekli durum hatasını azaltır, ama asla tamamen yok etmez.

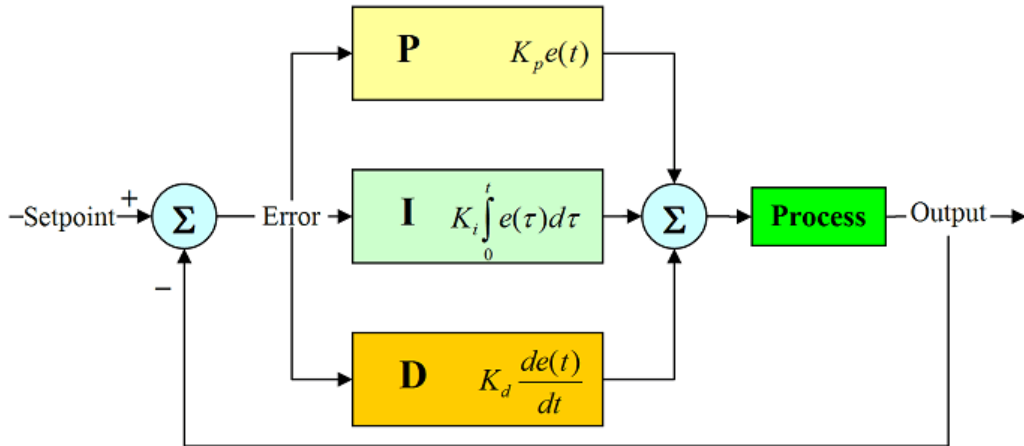
I kontrolcünün karalı hal hatasının sıfırlanmasına etkisi vardır ancak bu geçici tepkinin daha kötü olmasına sebep olur.

D kontrolcünü sistemin kararlılığının artmasında etkisi vardır, aşmayı azaltır ve geçici rejim davranışını düzeltir.

Tüm bu durumlar tablo halinde verilmiştir.

Kontrolör	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
P	Azalır	Artar	Küçük değişim	Azalır
I	Azalır	Artar	Artar	Sıfırlanır
D	Küçük değişim	Azalır	Azalır	Küçük değişim

Bu üç etkinin bir arada olması durumunda yeni kontrol etkisi sistemin hem düzenli rejim hem de geçici rejim davranışını iyileştirecektir. PID kontrolcülü kapalı çevrim kontrol sisteminin blok şeması aşağıda verilmiştir.



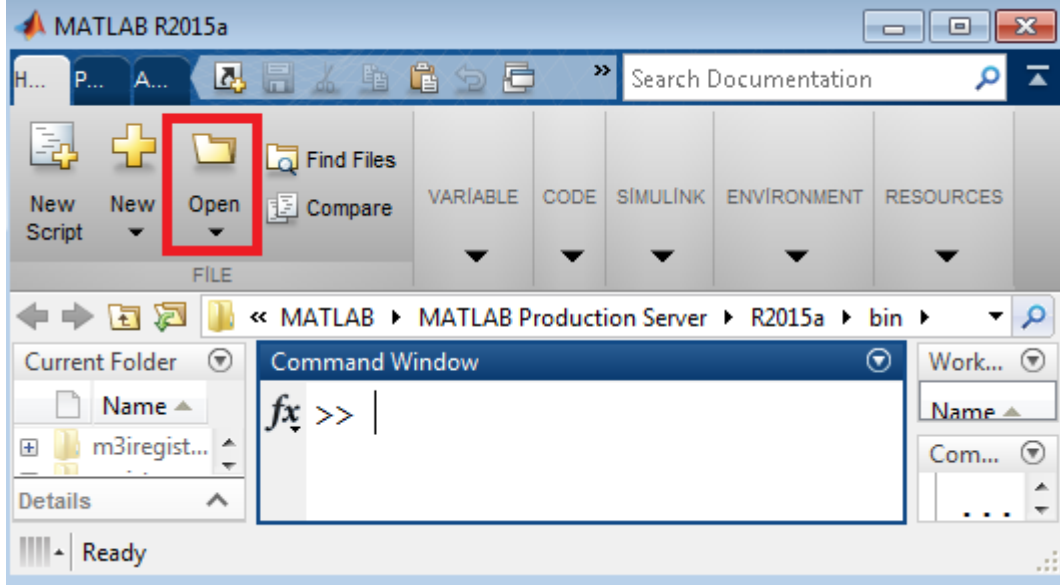
PID kontrolcü etkisi altındaki sistem cevabının, PI ve PD kontrolcülerin olumlu etkilerini bir arada gösterdiği söylenebilir. Sistem hem hızlanmış, hem daha kararlı hale gelmiş hem de girdiyi hatasız bir şekilde takip edebilme yeteneğine kavuşmuştur. PID yöntemi bir veya iki denetim yönteminin hatayı kabul edilebilir limitler içerisinde tutamadığı hızlı ve büyük yük değişimleri olan sistemlerde (process) kullanılır.

PID kontrol etkisi şu şekilde ifade edilir:

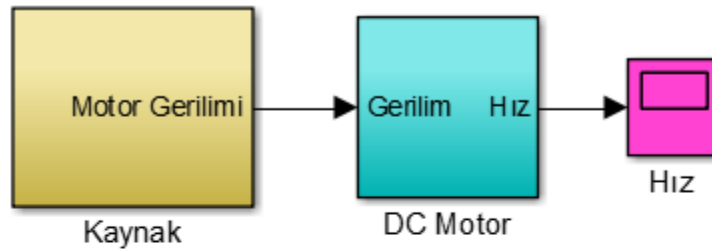
$$m(t) = K_p e(t) + K_D \frac{de(t)}{dt} + K_I \int_0^t e(\tau) d(\tau)$$

3.3 DENEYİN YAPILIŞI

1. Adım: Deney-2’de modellediğimiz DC motorun modelini açınız. Bunun için MATLAB ana penceresinde “Open” bölümünü kullanabilirsiniz.

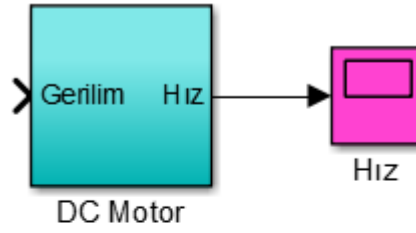


2. Adım: Modeliniz alttaki gibi görünecektir.



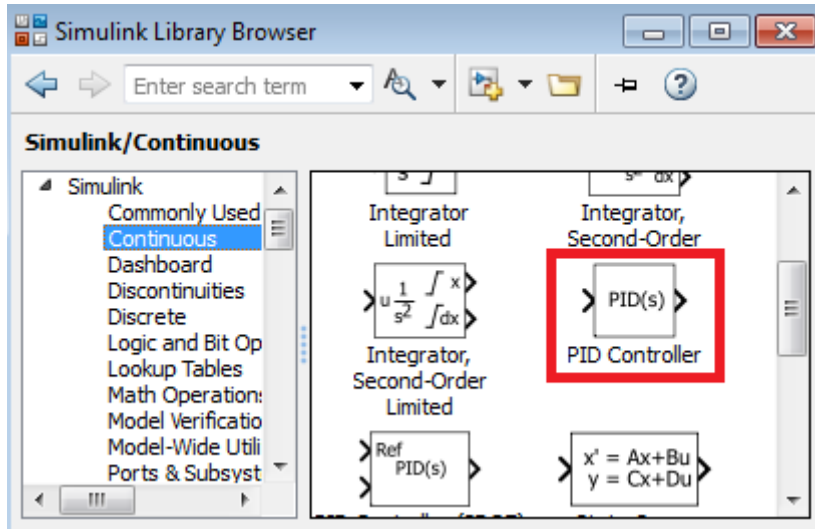
```
R = 2;  
L = 0.5;  
J = 0.01;  
B = 0.06;  
Kb = 0.02;  
Kt = 0.03;
```

3. Adım: DC Motorun kapalı çevrim kontrolü yapılacağı için “Kaynak” alt-bloğunu siliniz.



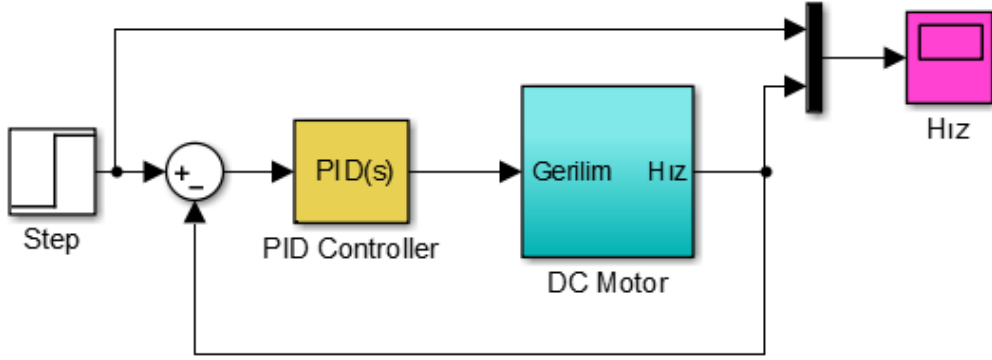
```
R = 2;  
L = 0.5;  
J = 0.01;  
B = 0.06;  
Kb = 0.02;  
Kt = 0.03;
```

4. Adım: Kütüphaneden “PID Controller” bloğunu seçerek modele ekleyiniz.



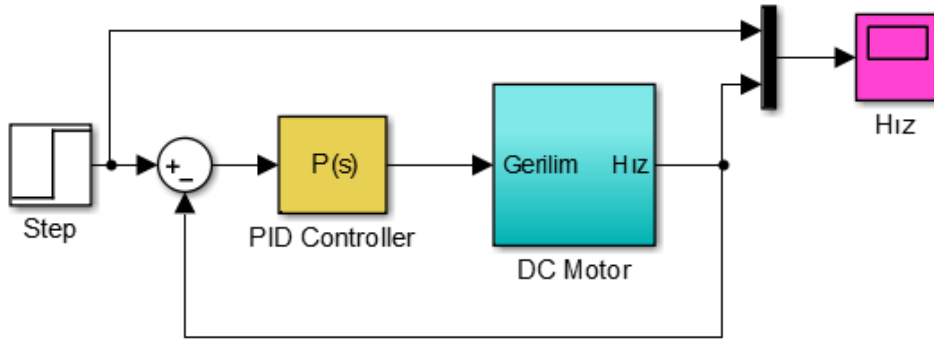
5. Adım: Son olarak DC motor sistemine test (kontrol) sinyali uygulamak için modele “Step” ve “Sum” bloklarını ekleyiniz. “Step” bloğunu direkt kullanınız. Hem girişi hem de çıkışı Scope ile gözlemleyiniz. Modeliniz altta da olduğu gibi kapalı çevrim kontrol sistemi haline gelecektir.

NOT: Step bloğu birim basamak olduğu için referans hız 1 rad/s olarak ayarlanmıştır. Yani, bizim çıkışta olmasını istediğimiz hız 1 rad/s'dir. Girişe farklı genlikte sinyaller uygulayarak da hız ayarı yapılabilir. Şekilden de görüleceği üzere kontrolcünün çıkışı, DC motora uygulayacağımız gerilimdir. Referans hızı değiştirdiğimizde kontrolcünün çıkışındaki gerilimin genliği de değişecektir.



R = 2;
L = 0.5;
J = 0.01;
B = 0.06;
Kb = 0.02;
Kt = 0.03;

6. Adım: DC Motor sadece **P kontrolcü** ile denetlenecek şekilde “**PID Controller**” bloğunu ayarlayınız.

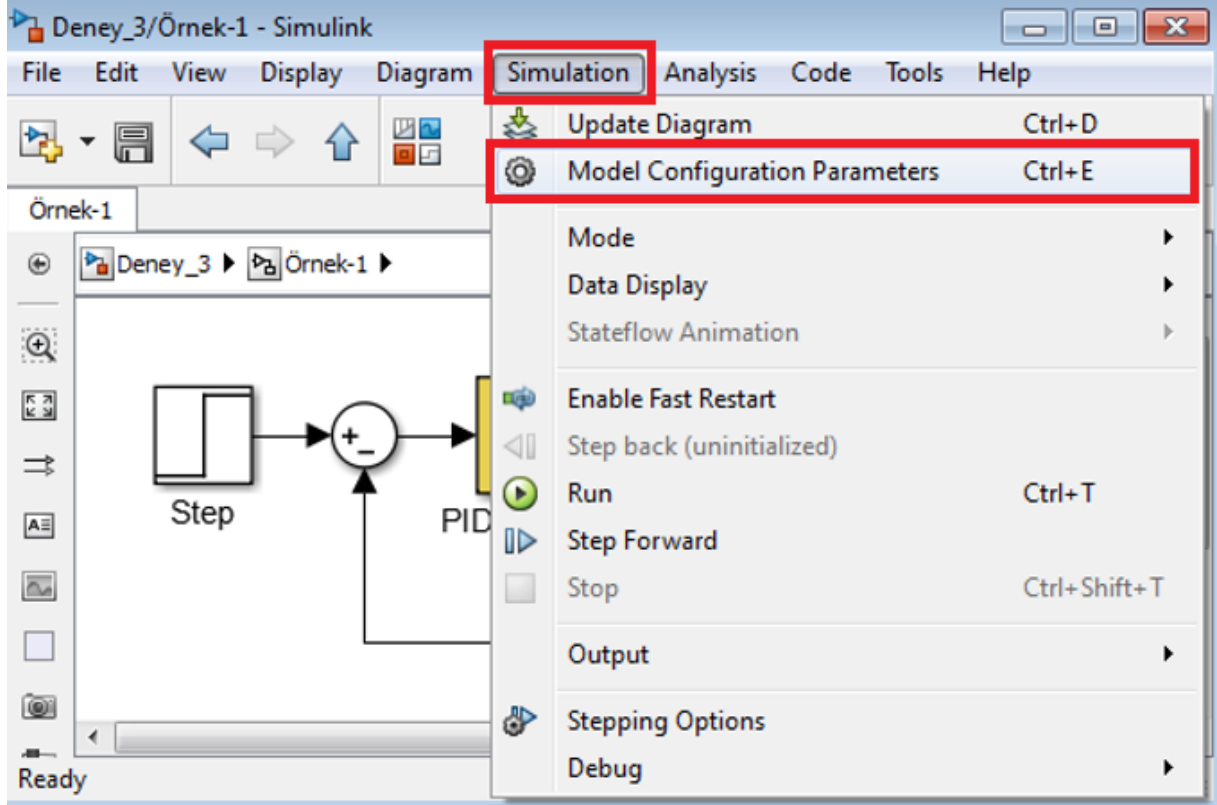


R = 2;
L = 0.5;
J = 0.01;
B = 0.06;
Kb = 0.02;
Kt = 0.03;

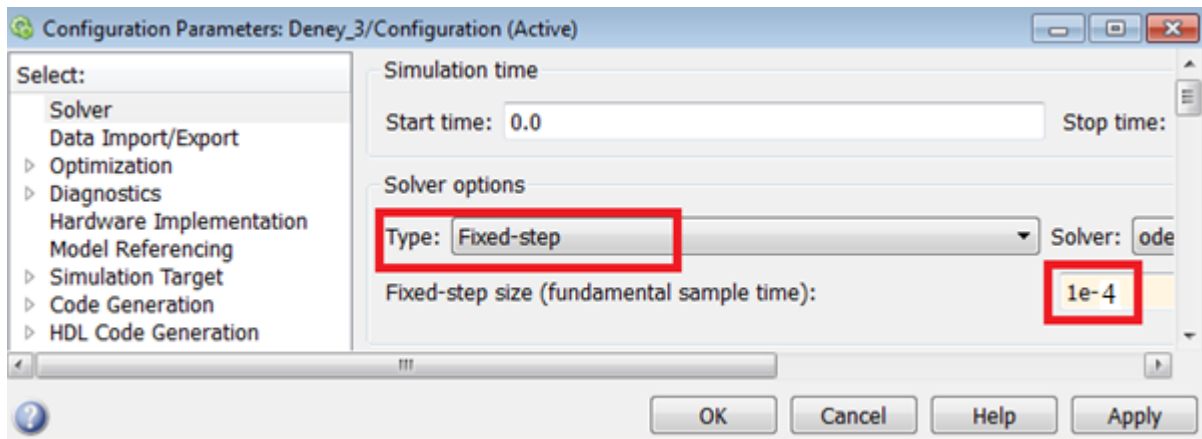
7. Adım: DC motor parametrelerini “**Komut Penceresinde**” koşturunuz. P kazancının (K_P) alttaki değerleri için tabloyu doldurunuz.

P Kontrolcü				
K_P	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
10				
100				
1000				

NOT: Çıkış sinyalini yakınlaştırdığınız zaman keskin dönüşlerin olduğu görülür. Bu çözünürlük ile alakalı bir durumdur. Yani, belirli sayıda örnek nokta ile grafik çizdirildiği için bu nokta sayısının arttırılması grafiğin daha hassas bir şekilde çizdirilmesini sağlayacaktır. Altta ayarlamalar yapılmalıdır.



NOT: “Fixed-step size”, $1e-4$ (10^{-4} saniye) yapıldığı zaman, 10^{-4} s aralıklarla örnekleme yapılarak grafik çizdirilir. Örnekleme zamanı azaltıldığı zaman daha hassas sinyal çizdirilir, fakat bu gerçekte simülasyonun koşturulma zamanını uzatır.



8. Adım: Bir önceki adımda bulduğunuz değerlere göre değerlendirme yapınız. (**Artar-Azalırlı Küçük değişim**)

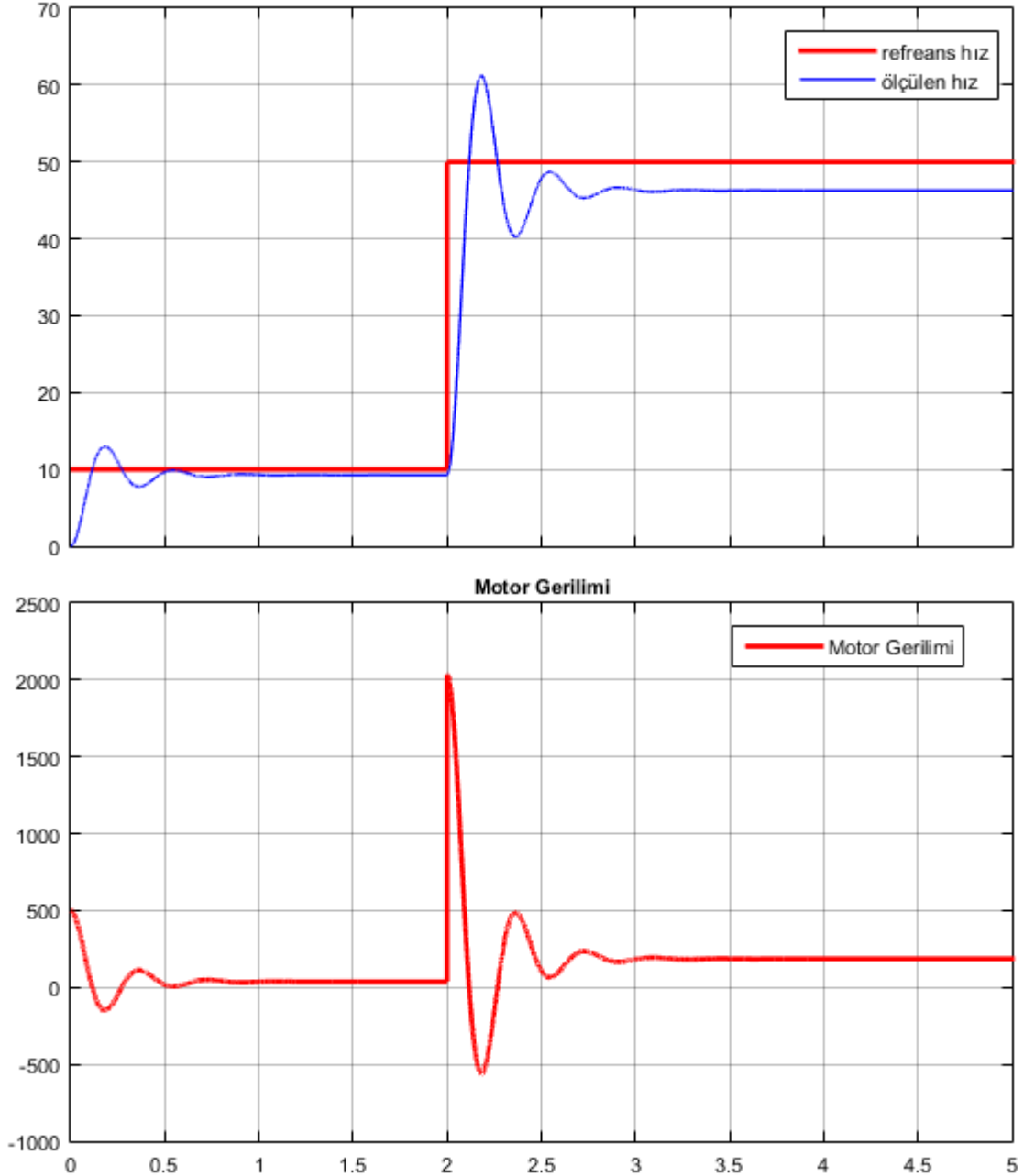
P Kontrolcü				
K_P	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
Artıkça				

9. Adım: Bundan sonraki deneyde (Deney-4), DC motor hız kontrolü için farklı kontrolcüler ile denetim yapılacağı için modelinizi “**Flash Belleğinize**” kaydediniz.

3.4 DENEY SONRASI ÇALIŞMA SORULARI

Uygulama-1: Deneyde yaptığımız uygulamayı göz önüne alarak; K_P kazancını 50 aldığımız zaman referans hıza göre ölçülen hız ne olur? Motor kaynak gerilimini de ölçünüz. Referans hız değiştiği zaman kaynak geriliminin de kontrolcü tarafından otomatikman değiştirildiğini gözlemleyiniz.

Sonuç-1:



DENEY NO: 4

DENEY ADI: DC Motorun Hız Kontrolü

4.1 DENEYİN AMACI

- i. DC motorun hızını kontrol etmek
- ii. P-I-D kontrol etkilerini incelemek

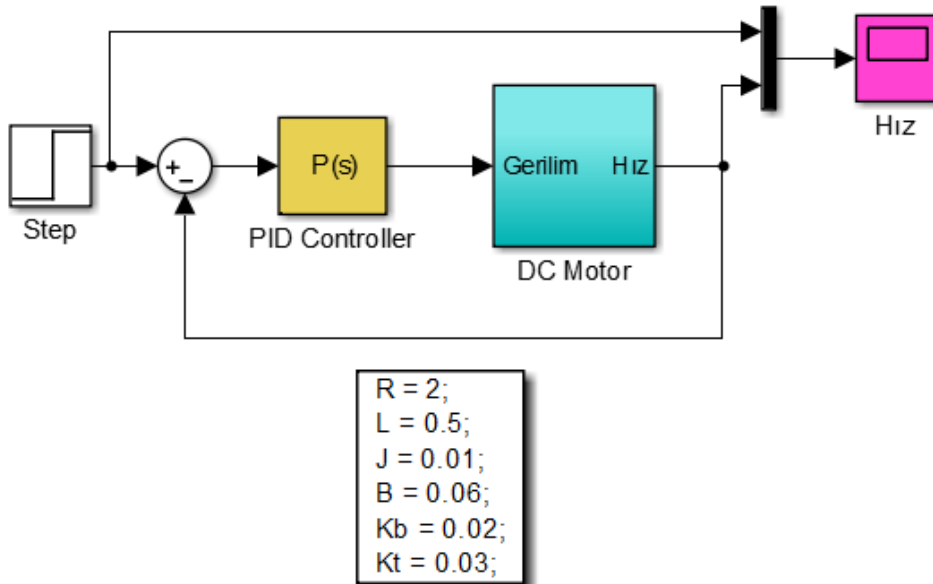
4.2 TEORİK BİLGİLER

Deney-3'te Oransal (P) kontrolcü ile kontrol edilen DC motorun, bu deneyde farklı kontrolcüler (PI, PD ve PID) ile hız kontrolü gerçekleştirilecek ve P-I-D kontrol etkileri incelenecektir.

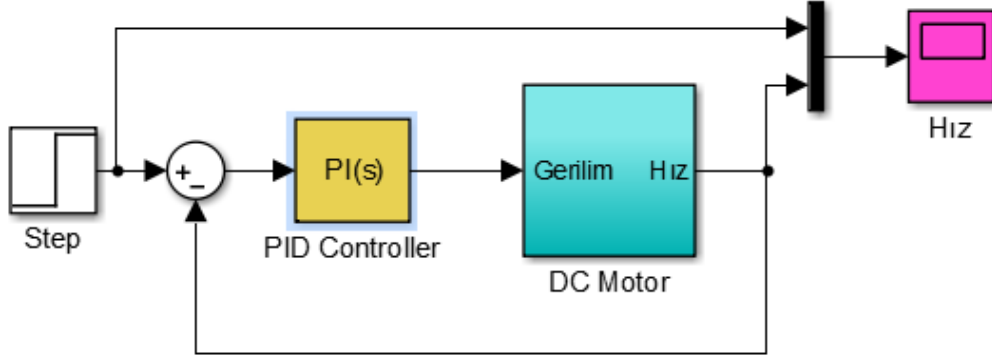
4.3 DENEYİN YAPILIŞI

1. Adım: Deney-3'te modellediğimiz DC motorun modelini açınız. Bunun için MATLAB ana penceresinde “**Open**” bölümünü kullanabilirsiniz.

2. Adım: Modeliniz alttaki gibi görünecektir.



3. Adım: DC Motor **PI kontrolcü** ile denetlenecek şekilde “PID Controller” bloğunu ayarlayınız.



R = 2;
L = 0.5;
J = 0.01;
B = 0.06;
Kb = 0.02;
Kt = 0.03;

❖ Örnekleme zamanını $1e-4$ 'e ayarlayınız. Tüm ölçümlerde y-eksen ölçeği 0.1 rad/s, x-eksen ölçeği 1 saniye (sadece t_r ölçümünde 0,01 saniye) olacak şekilde yakınlaştırma yapınız.

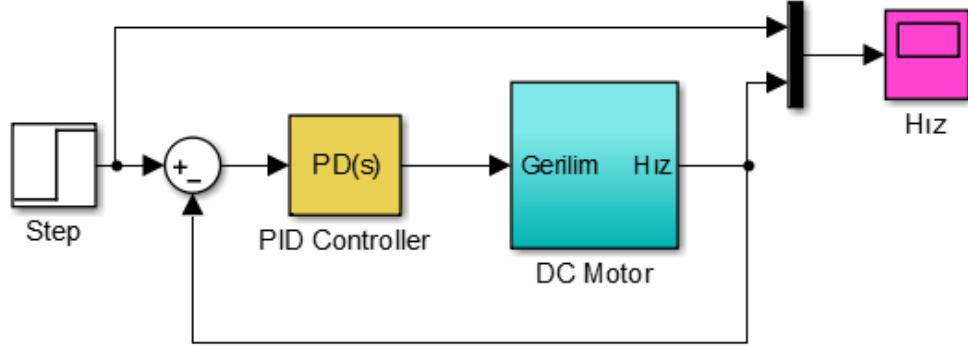
4. Adım: P kazancının (K_P) ve I kazancının (K_I) alttaki değerleri için tabloyu doldurunuz.

PI Kontrolcü				
$K_P - K_I$	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
$K_P = 30$				
$K_I = 100$				
$K_P = 30$				
$K_I = 300$				

5. Adım: Bir önceki adımda bulduğunuz değerlere göre değerlendirme yapınız. (Artar-Azalırlı Küçük değişim)

PI Kontrolcü				
K_I	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
Arttıkça				

6. Adım: DC Motor PD kontrolcü ile denetlenecek şekilde “PID Controller” bloğunu ayarlayınız.



R = 2;
L = 0.5;
J = 0.01;
B = 0.06;
Kb = 0.02;
Kt = 0.03;

❖ *Örnekleme zamanını $1e-4$ 'e ayarlayınız. Tüm ölçümlerde y-eksen ölçeği 0.1 rad/s, x-eksen ölçeği 1 saniye (sadece t_r ölçümünde 0,01 saniye) olacak şekilde yakınlaştırma yapınız.*

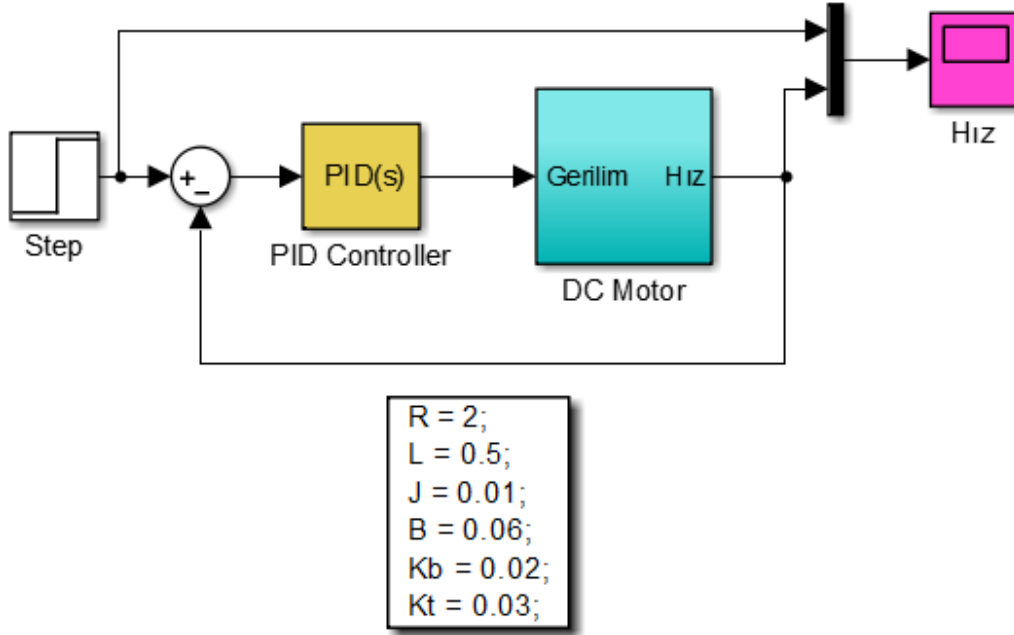
7. Adım: P kazancının (K_P) ve D kazancının (K_D) alttaki değerleri için tabloyu doldurunuz.

PD Kontrolcü				
$K_P - K_D$	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
$K_P = 50$				
$K_D = 0,1$				
$K_P = 50$				
$K_D = 2$				

8. Adım: Bir önceki adımda bulduğunuz değerlere göre değerlendirme yapınız. (**Artar-Azalırlar Küçük değişim**)

PD Kontrolcü				
K_D	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
Arttıkça				

9. Adım: DC Motor **PID kontrolcü** ile denetlenecek şekilde “PID Controller” bloğunu ayarlayınız.



❖ Örnekleme zamanını $1e-4$ 'e ayarlayınız. Tüm ölçümlerde y-ksen ölçeği 0.1 rad/s, x-ksen ölçeği 1 saniye (sadece t_r ölçümünde 0,01 saniye) olacak şekilde yakınlaştırma yapınız.

10. Adım: P kazancının (K_P), I kazancının (K_I) ve D kazancının (K_D) alttaki değerleri için tabloyu doldurunuz. 2. satırda sadece K_D değeri artırılarak maksimum aşma değeri % 6 seviyesine indirilebilir.

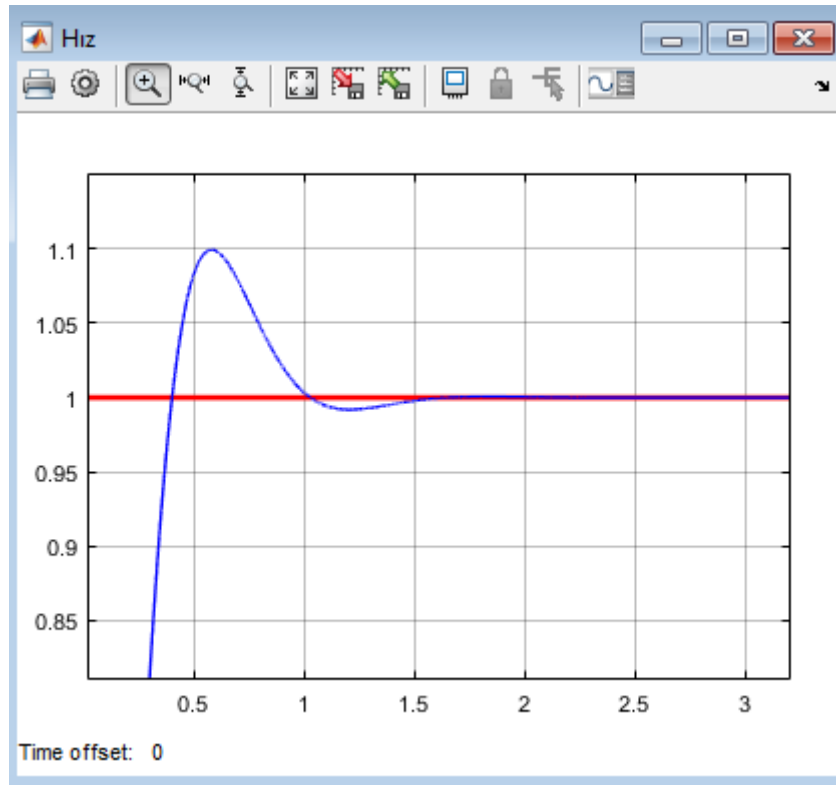
PID Kontrolcü				
K_P - K_I - K_D	Yükselme zamanı t_r	Aşma değeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
$K_P = 50$ $K_I = 300$ $K_D = 2$				
$K_P = 50$ $K_I = 300$ $K_D = 5$				

4.4 DENEY SONRASI ÇALIŞMA SORULARI

Uygulama-1: Deneyde yaptığımız uygulamayı göz önüne alarak; en fazla % 10'luk maksimum aşma ve 2 s'lik oturma süresi için PID kazançlarını ayarlayınız.

K_P	K_I	K_D

Sonuç-1: Sistemin girişine birim basamak sinyali uygulayınız. Basamak zamanını, “0 s” olarak ayarlayınız. Uygun PID katsayılarını ayarladığınız zaman, hız-zaman grafiğiniz alttaki gibi görünecektir.



DENEY NO: 5

DENEY ADI: Ziegler–Nichols Yöntemi ile PID Parametrelerinin Bulunması

5.1 DENEYİN AMACI

- i. Ziegler–Nichols yönteminin incelenmesi
- ii. PID parametrelerinin ince ayarının yapılması

5.2 TEORİK BİLGİLER

Eğer sistemin matematiksel modeli doğrusalsa ve sistem parametreleri biliniyorsa, PID denetleyici parametreleri, kontrol teorisi kullanılarak tasarlanabilir. Maalesef gerçek sistemlerin çoğu doğrusal değildir ve karmaşıktır, bu yüzden matematiksel denklemleri kullanarak bir sistemi tam olarak tanımlamak zordur. Endüstriyel uygulamalarda, PID denetleyiciler, sıklıkla deneysel yöntem kullanılarak tasarlanırlar. Çok bilinen deneysel ayarlama kuralları, Ziegler ve Nichols'a ait olanlardır. Transfer fonksiyonu;

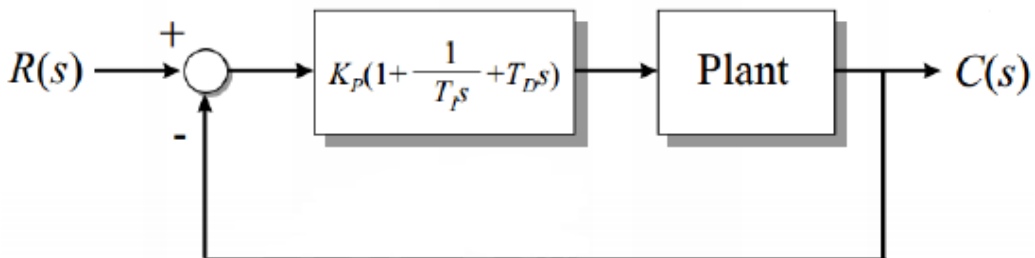
$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

olan bir PID kontrolör sisteminde, Zeigler ve Nichols oransal kazanç K_p , integral zamanı T_i ve diferansiyel zamanı T_d değerlerinin hesaplamasında basit kurallar sunmuşlardır. Bu kurallarda maksimum aşım %25 olarak hedeflenmektedir.

Ziegler ve Nichols iki farklı yöntem önermişlerdir. 1. Yöntem **cevap eğrisi yöntemi** olup sistemin açık çevrimine göre PID parametreleri belirlenir. 2. Yöntem **salınım veya titreşim yöntemi** olup sistemin kapalı çevrimine göre PID parametreleri belirlenir. Bu deneyde 2. Yöntem ele alınacaktır.

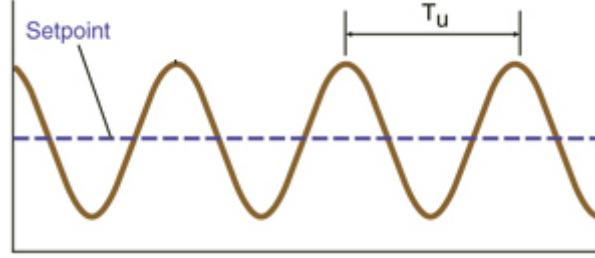
Salınım veya Titreşim Yöntemi

Bu yöntem, geribeslemeli kapalı döngü bir sistemin basamak girişi ile sürekli salınımda test edilmesine dayanır. Bu yöntemde başlangıç olarak integral ve türev etkileri devre dışı bırakılarak denetleyici sadece orantı etki ile çalıştırılır. Yani, $T_i = \infty$, $T_d = 0$ alınır.



Şekil 5.1 PID kontrolör blok diyagramı

Bu yöntemde K_p değeri 0'dan kritik değer olan $K_{p_{max}}$ değerine kadar artırılır ve bu değerde sistem çıkışı sürekli salınıma girer. Böylelikle $K_{p_{max}}$ ve buna karşı düşen osilasyon periyodu (T_u) not edilir.



Şekil 5.2 Salınımın bulunması

Bu işlem gerçekleşikten sonra K_p , T_i ve T_d parametreleri aşağıdaki tabloya göre hesaplanır.

Kontrolör	K_p	T_i	T_d
P	$0,5 * K_{p_{max}}$	∞	0
PI	$0,45 * K_{p_{max}}$	$0,825 * T_u$	0
PD	$0,8 * K_{p_{max}}$	∞	$0,125 * T_u$
PID	$0,6 * K_{p_{max}}$	$0,5 * T_u$	$0,125 * T_u$

NOT: $G_c(s) = K_p + \frac{1}{T_i s} + T_d s$ PID transfer fonksiyonu kullanıldığında zaman tablodaki değerleri kullanamayız.

Örnek: $T(s) = \frac{1}{s^3 + 6s^2 + 8s}$ transfer fonksiyonuna sahip sistemin PID katsayılarını Ziegler-Nichols 2.yöntem ile bulunuz.

Cözüm

$T_i = \infty$, $T_d = 0$ alınarak kapalı çevrim sisteminin transfer fonksiyonu aşağıdaki gibi elde edilir.

$$\frac{C(s)}{R(s)} = \frac{K_p}{s^3 + 6s^2 + 8s + K_p}$$

Sistemin salınımı, Routh kararlılık kriteri kullanılarak bulunabilir. Kapalı çevrim sisteminin karakteristik denklemi

$$s^3 + 6s^2 + 8s + K_p = 0$$

olarak yazılır.

s^3	1	8
s^2	6	K_P
s^1	$\frac{48 - K_P}{6}$	0
s^0	K_P	0

Kararlılık için ikinci sütunda işaret değişimi olmaması gerekir. Bu yüzden, tüm değerler pozitif olmalıdır. Bu durumda, K_P değişim aralığı;

$$0 < K_P < 48$$

olarak elde edilir. Yani, $K_{P_{max}} = 48$ değerinde sistem osilasyon yapmaktadır. Bu değere karşılık gelen karakteristik denklem aşağıdaki gibi elde edilir.

$$s^3 + 6s^2 + 8s + 48 = 0$$

Salınım frekansını bulmak için $s=j\omega$ yazılırsa (sistemin salınımlı yani yarı kararlı çalışmasına karşılık gelen kökler sanal eksen üzerinde yer aldığından) karakteristik denklem,

$$(j\omega)^3 + 6(j\omega)^2 + 8j\omega + 48 = 0$$

$$6(8 - \omega^2) + j\omega(8 - \omega^2) = 0$$

yazılır ve salınım frekansı $\omega = 2\sqrt{2}$ olur. Salınım periyodunu bulmak için;

$$T_u = \frac{2\pi}{\omega} = \frac{2\pi}{2\sqrt{2}} = 2,22s$$

Tabloya göre, PID parametreleri için;

$K_P = 28,8$	$T_i = 1,11s$	$T_d = 0,277s$	$K_i = \frac{1}{T_i} = 0,901$	$K_d = 0,277$
--------------	---------------	----------------	-------------------------------	---------------

hesaplanır.

PID Denetleyicinin İnce Ayarı

Ziegler-Nichols yöntemine göre tasarlanan PID denetleyici bir sisteme eklendiğinde, teoride, kapalı çevrim sistemin basamak tepkisi, yaklaşık 25%'lik maksimum aşmaya sahip olur. Bununla birlikte, bu sonuç tüm sistemler için garanti edilemez. Genel anlamda, çeşitli sistemler için, 10% ile 60% aralığında maksimum aşma elde edilebilir.

Pratikte, PID denetleyici sisteme eklenirken, PID denetleyici parametreleri, Ziegler-Nichols yöntemine göre ayarlanır ve daha sonra istenilen tepke elde edilinceye kadar, PID

parametreleri tekrar tekrar deęiştirilir. Bu açıdan Ziegler-Nichols ayarları günümüzde PID ayarlarında kaba ayarlar olarak görülür.

5.3 DENEYİN YAPILIŞI

1. Adım: Boş Simulink modeli açınız.

2. Adım: 38. ve 39. sayfada transfer fonksiyonu verilen sistemin Ziegler-Nichols yöntemi ile elde edilmiş PID parametrelerini yerine koyarak simülasyon yapınız ve Simulink modelini çiziniz.

3. Adım: Alttaki deęerleri ölçünüz.

t_r (s) y-eksen: 0.1br/kare x-eksen: 0.01s/kare	M_p (br) y-eksen: 0.01br/kare x-eksen: 1s/kare	t_s (s) y-eksen: 0.01br/kare x-eksen: 1s/kare	e_{ss} (br) y-eksen: 0.01br/kare x-eksen: 1s/kare

4. Adım: Alttaki tablodan faydalanarak $t_s < 5s$, $M_p < \%2$ ve $e_{ss} < \%1$ olacak şekilde kontrolcü tasarlayınız. Aşağıya not ediniz.

Kontrolör	Yükselme zamanı t_r	Aşma deęeri M_p	Oturma zamanı t_s	Sürekli Durum Hatası e_{ss}
P	Azalır	Artar	Küçük deęişim	Azalır
I	Azalır	Artar	Artar	Sıfırlanır
D	Küçük deęişim	Azalır	Azalır	Küçük deęişim

K_P	K_i	K_D