

SİMULİNK DERSLERİ

SİMULİNK'E GİRİŞ:

Simulink:

Simulink bize karmaşık sistemleri tasarlama ve simülasyon yapma olanağı vermektedir. Mühendislik sistemlerinde simülasyonun önemi gün geçtikçe artmaktadır. Sistemlerin tasarımında büyük oranda bilgisayar simülasyonlarından faydalanmakta, mümkün olduğunda 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. MATLAB'in tanıtımı için MATLAB'e giriş adlı yazıma bakabilirsiniz. Simulink, MATLAB ile birlikte bütünleşik olarak çalışan bir simülasyon ortamıdır. Sürekli zamanlı ve ayrık zamanlı sistemleri, veya her ikisini de içeren hibrit sistemleri desteklemektedir. İçinde birçok altsistemi blok olarak barındırdığından sürükle-bırak yöntemiyle birçok sistemi bir-kaç dakikada kurarak simüle edebilir, değişik durumlardaki cevabını test edebilirsiniz. Bunun için Simulink bizlere zengin bir blok kütüphanesi sunmaktadır. Electronics Workbench gibi programları kullanarak devre simülasyonu yapmış arkadaşlar Simulink'i kullanmak konusunda pek zorluk çekmeyeceklerdir. Çünkü kullanımı benzer şekilde blokların yerleştirilip birbirine bağlanması şeklindedir.

Öncelikle Simulink ile ilgili temel kavramları tanıyalım:

Simulink Kütüphanesi:

Simulink'i çalıştırdığınızda karşınıza Simulink kütüphanesi gelecektir. Simülasyon yaparken kullanacağımız bloklar kategorilere ayrılmış biçimde burada bulunmaktadır.

Blok Diyagramlar

Her bir blok sürekli zamanda ya da ayrık zamanda çıkış veren temel bir dinamik sistemi ifade eder. Hatlar blokların giriş ve çıkışları arasındaki bağlantıları gösterir. Blok diyagramdaki her bir blok belirli bir tip bloğun örneğidir. Bloğun tipi, bloğun giriş ve çıkışları, durumları ve zaman arasındaki bağıntıyı belirler.

Bloklar

Bloklar Simulink'in nasıl simüle edileceğini bildiği temel dinamik sistemleri temsil eder. Blokları programlama dillerinde kullanılan fonksiyonlara benzetebiliriz. Her bir dinamik sistem için yazılmış hazır fonksiyonlardır. Bize kalan sadece bu hazır fonksiyonları kullanarak sistemimizi kurmaktır. Bir blok şu bileşenlerin birinden veya birkaçından oluşur: Giriş kümesi, durum kümesi ve çıkış kümesi.

Durumlar:

Bloklar durumlara sahip olabilirler. Durum, bloğun çıkışını belirleyen ve şimdiki değeri, bloğun önceki durumları ve/veya girişlerinin fonksiyonu olan bir değişkendir. Duruma sahip olan bir blok şimdiki durumunu belirlemek için geçmiş değerlerini kaydetmek zorundadır.....Duruma sahip olan bloklara, hafızalı blok denilir. Çünkü bu bloklar o anki değerlerini belirlemek için geçmiş değerlerini kaydetmek zorundadırlar.

Simulink İntegral alıcı (integrator) bloğu duruma sahip bloklara bir örnektir. İntegrator bloğu simülasyonun başlangıcından o anki zamana kadar giriş sinyalinin integralini çıkış olarak verir. O anki zaman adımındaki integral değeri, integrator bloğunun geçmişteki

giriş değerlerine bağlıdır. Dolayısıyla integral, integrator bloğunun durumudur. Durumlu bloklara bir başka örnek de Simulink Hafıza (memory) bloğudur. Hafıza bloğu girişlerindeki değerleri o anda kaydedip ileriki bir zamanda çıkışına verir. Hafıza bloğunun durumları önceki giriş değerleridir.

Simulink Kazanç bloğu (Gain) durumsuz bloklara bir örnektir. Kazanç bloğu girişindeki değeri kazanç adı verilen bir sabitle çarparak çıkışına verir. Kazanç bloğunun çıkışı tamamiyle o anki giriş değeri ve sabit olan kazanç ile belirlenir. Dolayısıyla Kazanç bloğunun durumu yoktur. Diğer bazı durumsuz bloklar Toplam (Sum) ve Çarpım (product) bloklarıdır. Bu blokların çıkışları tamamiyle girişlerinin bir fonksiyonudur. (İlki için toplam, ikincisi için çarpımdır) Dolayısıyla bu blokların durumları yoktur.

Blok Parametreleri:

Birçok standart bloğun anahtar özellikleri parametrik hale getirilmiştir. Örneğin, Kazanç Bloğunun (Gain) kazancı bir parametredir. Her bir parametrik blok size blok parametrelerini belirleyebileceğiniz bir diyalog kutusu sunar. Blok parametrelerini belirlemek için MATLAB ifadeleri kullanabilirsiniz. Simulink bu ifadeleri simülasyonu çalıştırmadan önce hesaplar. Parametrelerin değerlerini simülasyon esnasında değiştirebilirsiniz. Bu parametrenin en uygun değerini interaktif bir şekilde belirlemenize olanak sunar.

Parametrik bloklar etkili bir biçimde benzer blok ailelerini temsil ederler. Örneğin, bir model oluştururken modeldeki her bir Kazanç (Gain) bloğunun kazanç değerlerini ayrı ayrı belirleyerek her bir Kazanç Bloğunun farklı davranmasını sağlayabilirsiniz. Blokların parametrik hale getirilmesi, her bir standart bloğun bir blok ailesini temsil etmesini sağlayarak, Simulink'in modelleme gücünü artırmaktadır.

Değiştirilebilir Parametreler:

Birçok blok parametresi değiştirilebilirdir. Simülasyon yapılırken değeri değiştirilebilen parametreler değiştirilebilir parametrelerdir. Örneğin Kazanç bloğunun kazanç parametresi değiştirilebilir parametredir. Simülasyon çalışırken bloğun kazancını değiştirebilirsiniz. Bir parametre değiştirilebilir değilse ve simülasyon çalışıyorsa Simulink parametreyi ayarlayan diyalog kutusunu engeller. Simulink belirledikleriniz dışında bütün parametreleri değiştirilemez olarak belirlemenize izin verir. Bu büyük modellerin çalıştırılmasını hızlandırır ve modelinizden daha hızlı bir şekilde kod üretilmesini sağlar...

Altsistemler:

Simulink bize, kompleks sistemleri, blok diyagramları ile temsil edilen birbirine bağlı altsistemler şeklinde modellemenize izin verir. Altsistemleri Simulink altsistem(subsystem) bloğuyla ve model editörüyle oluşturabilirsiniz. Alt sistemleri ana sistemlere istediğiniz derinliğe kadar gömerek hiyerarşik modeller yaratabilirsiniz. Bir geçiş durumu olduğunda, bir tetikleme veya yetkilendirme girişi geldiğinde çalıştırılan şarta bağlı çalışan alt sistemler oluşturabilirsiniz.

Sinyaller:

Simulink sinyal terimini blokların çıkış değerlerini belirtmekte kullanır. Simulink size sinyal ismi, veri tipi (örn: 8-bit, 16-bit veya 32-bit tamsayı), numerik tip (Reel veya kompleks), ve boyutluluk (tek boyutlu veya 2-boyutlu dizi) gibi sinyal özelliklerini belirlemenize olanak verir. Birçok blok herhangi bir veri veya numerik tipte ve boyutta çıkışı kabul edebilir. Diğerleri de kabul edebildikleri sinyal özellikleri ile ilgili kısıtlamalar taşırlar.

Veri Tipleri:

Veri tipi, verinin bilgisayardaki temsiline verilen addır. Simulink MATLAB'te

desteklenen int8,double ve boolean gibi herhangi bir dahili veri tipini kullanabilir.Bunlara ek olarak Simulink kendine özgü iki veri tipi daha tanımlar:

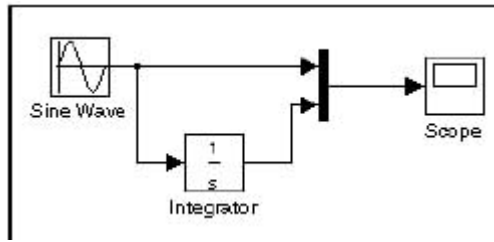
- Simulink.Parameter
- Simulink.Signal

Bu Simulink 'e özgü veri tipleri diğer genel veri tipleri ile tutulamayan Simulink'e özgü bazı bilgilerin tutulmasında kullanılırlar.Simulink size Simulink *veri objeleri* denilen Simulink veri tiplerini kullanarak parametre değerleri ve sinyal olarak kullanılmak üzere yeni veri tipleri yaratmanıza imkan verir.Her iki Simulink veri tipini de kullanarak kendi modellerinize özgü bilgileri saklayabilen yeni ver tipleri yaratabilirsiniz.

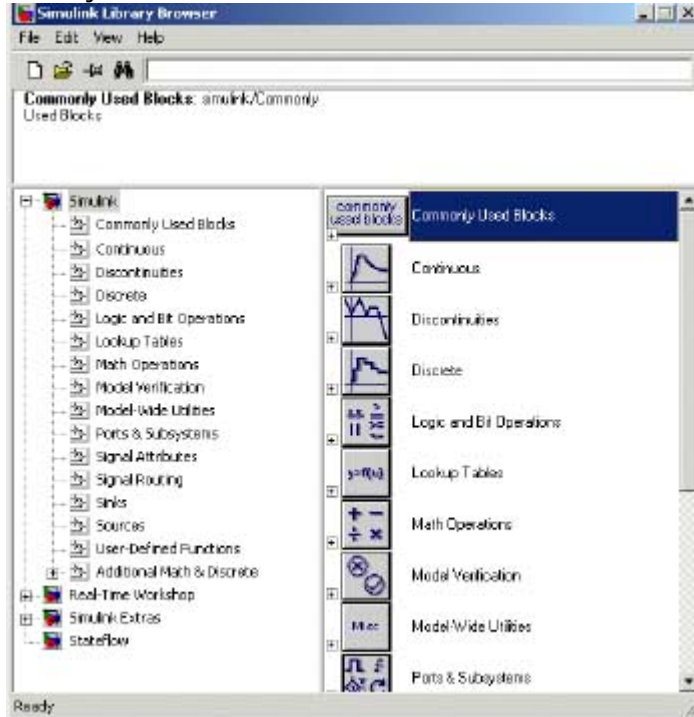
SİMULİNK KULLANIMI:

Model Oluşturmak :

Bu örnekte Simulink'te bir modeli nasıl oluşturup simule edeceğimizi göreceğiz. Bu modelde bir sinüs dalgasını ve integralini birleştirerek sonucu grafik olarak bize gösteren scope bloğunda görüntüleyeceğiz.Oluşturacağımız blok diyagram şu şekilde görünecek;



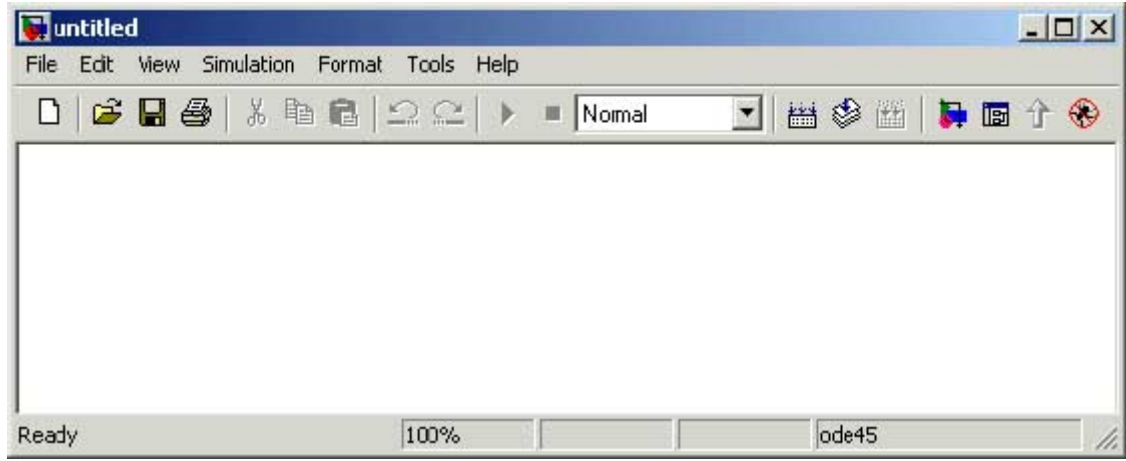
Simulink'i çalıştırmak için MATLAB komut satırında simulink yazıyoruz.Karşımıza Simulink Kütüphanesi çıkacaktır:



Yeni bir model oluřturmak için Kütüphane Gezgini araççubuğundaki new file (Yeni dosya) butonuna tıklayın.



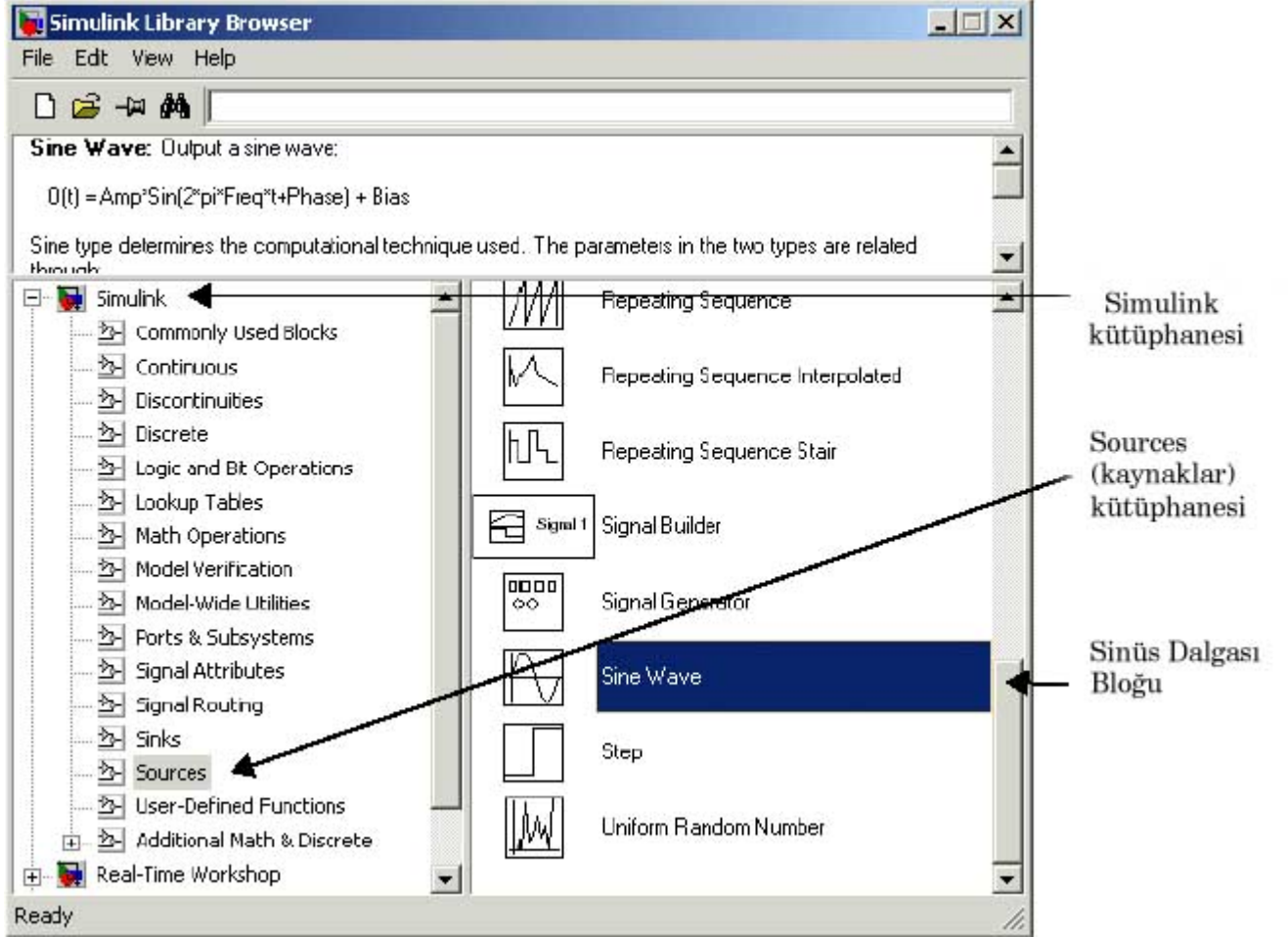
Simulink Yeni bir model penceresi açacaktır:



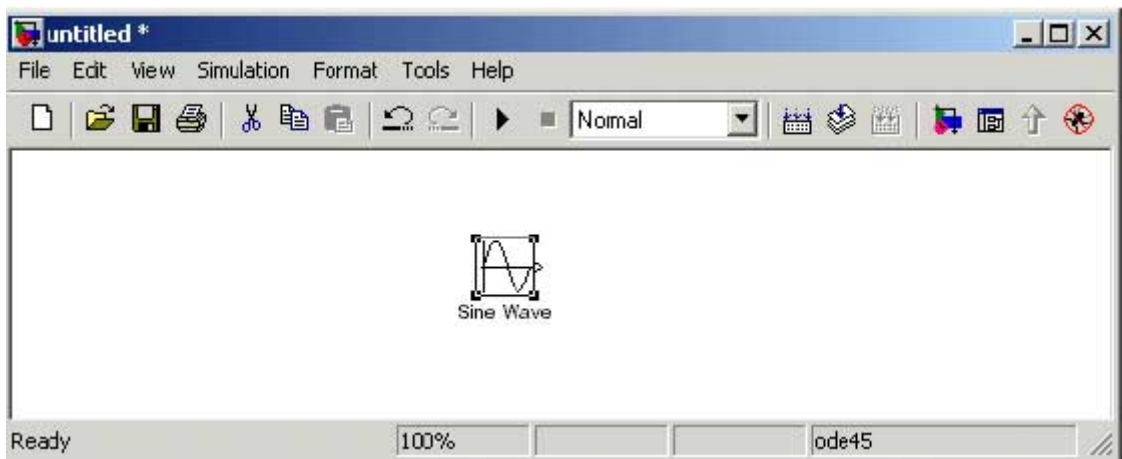
Yukarıda resmini gördüğünüz modeli oluřturmak için řu bloklara ihtiyacımız olacak.Bu blokları kütüphaneden kendi modelimize kopyalamamız gerekecek.

- Sources kütüphanesinden (the Sine Wave bloğı)
 - Sinks kütüphanesinden (the Scope bloğı)
- Continuous kütüphanesinden (the Integrator bloğı)
 - Signal Routing kütüphanesinden (the Mux

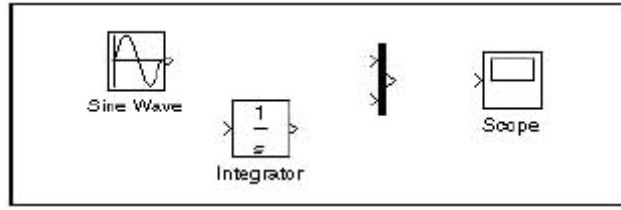
bloğu)



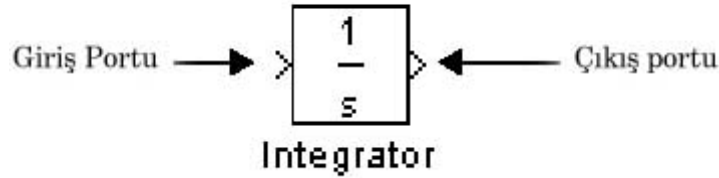
İlk olarak Sinüs dalgası bloğunu modelimize kopyalamak için, kütüphanedeki kütüphane ağacı altındaki sources bölümüne tıklayarak içeriğinin sol tarafta görüntülenmesini sağlayın. Soldaki bölümden Sine Wave bloğuna tıklayarak kendi modelinize sürükleyin.



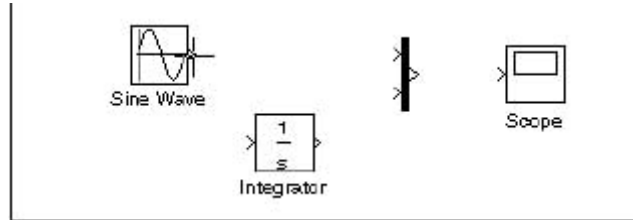
Yukarıdaki listede verdiğimiz blokları da ilgili kütüphanelerden modelinize kopyalayın. Bütün blokları modelinize attığınızda şu şekilde görünecektir:



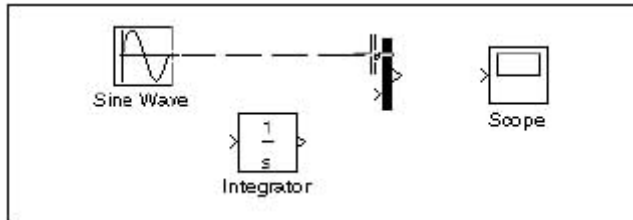
Bloklara baktığınızda Sine Wave bloğunun sağında bir ve Mux bloğunun sol ve sağ yanlarında iki tane olmak üzere > işareti göreceksiniz. Bloktan dışarıya doğru olan > işareti bloğun çıkış portunu, bloğa doğru olan > sembolü de bloğun giriş portunu belirtmektedir. Sinyaller blokların çıkış portlarında çıkıp giriş portlarına doğru giderler. Bloklar birbirine bağlandığında port sembolü kaybolur



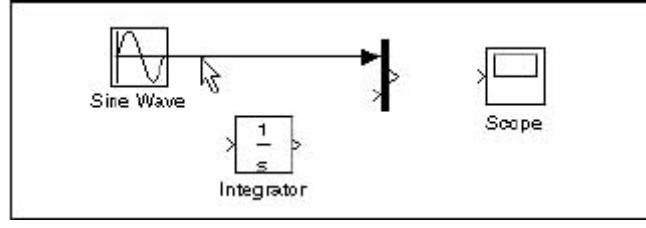
Şimdi sıra geldi blokları bağlamaya. İlk olarak Sine Wave bloğunu Mux bloğuna bağlayalım. Fare işaretçisini Sine Wave bloğunun çıkış portunun üzerine getirdiğinizde + işareti aldığınız göreceksiniz.



Fareye tıklayın ve basılı tutarak imleci Mux bloğunun giriş portuna sürükleyin. Fareyle sürüklerken sinyal kesik kesik görünecektir. İmleci Mux bloğuna yaklaştığınızda imlec şeklinin çift-çizgili + işaretine dönüştüğünü göreceksiniz.



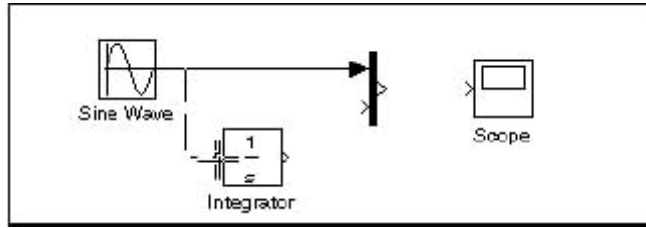
Fareyi bıraktığınızda bloklar bağlanmış olacaktır. Sinyal düz çizgi ile görülecektir.



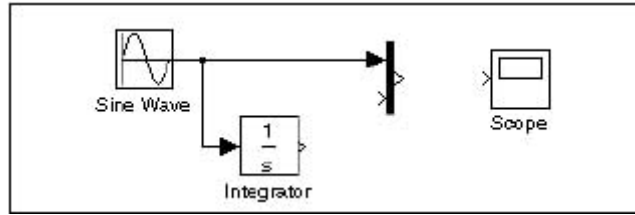
İlk başta vermiş olduğumuz modele baktığınızda genellikle bütün bağlantıların bir bloğun çıkışından diğerine doğru olduğunu görürsünüz: Sadece Sine Wave bloğundan çıkan sinyal iki ayrı bloğa bağlanmıştır. Bu tür bağlantılara dal bağlantı denilmektedir. Sine Wave bloğundan çıkan bilgi aynı anda hem Integrator hem de Mux bloğuna girmektedir.

Dal bağlantılarının yapılmasında bazı değişik adımlar izlenmektedir. Varolan bir bağlantıdan dal bağlantısı yapmak için şu adımları izleyin:

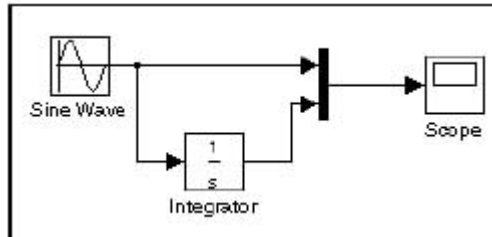
1 İmleci Sine Wave ve Mux bloklarının arasındaki bağlantının üzerine getirin.



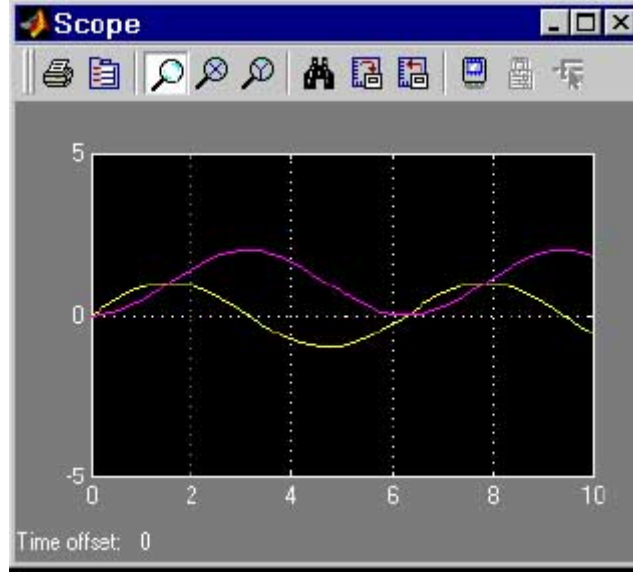
2 Ctrl butonunu basılı tutarak (veya sağ fare butonunu basılı tutarak) imleci Integrator bloğunun giriş portuna sürükleyip bırakın.



Böylelikle modelimiz tamamlandı:



Şimdi model pencremizdeki  butonuna tıklayarak similasyonu çalıştıralım. Ardından Scope bloğuna çift tıklayarak similasyon çıktısını görebiliriz.



Grafikte bir sinüs dalgasını ve integralini görmekteyiz.Scope bloğu ile aynı anda iki sinyali de görüntülemek için MUX (tekleyici) kullanarak sinyalleri tek hattan scope'a bağladık.Umarım bu yazı Simulink için iyi bir başlangıç dökümanı olur.Her türlü görüş ve önerilerinizi bekliyorum

SİMULİNK KULLANIMI:

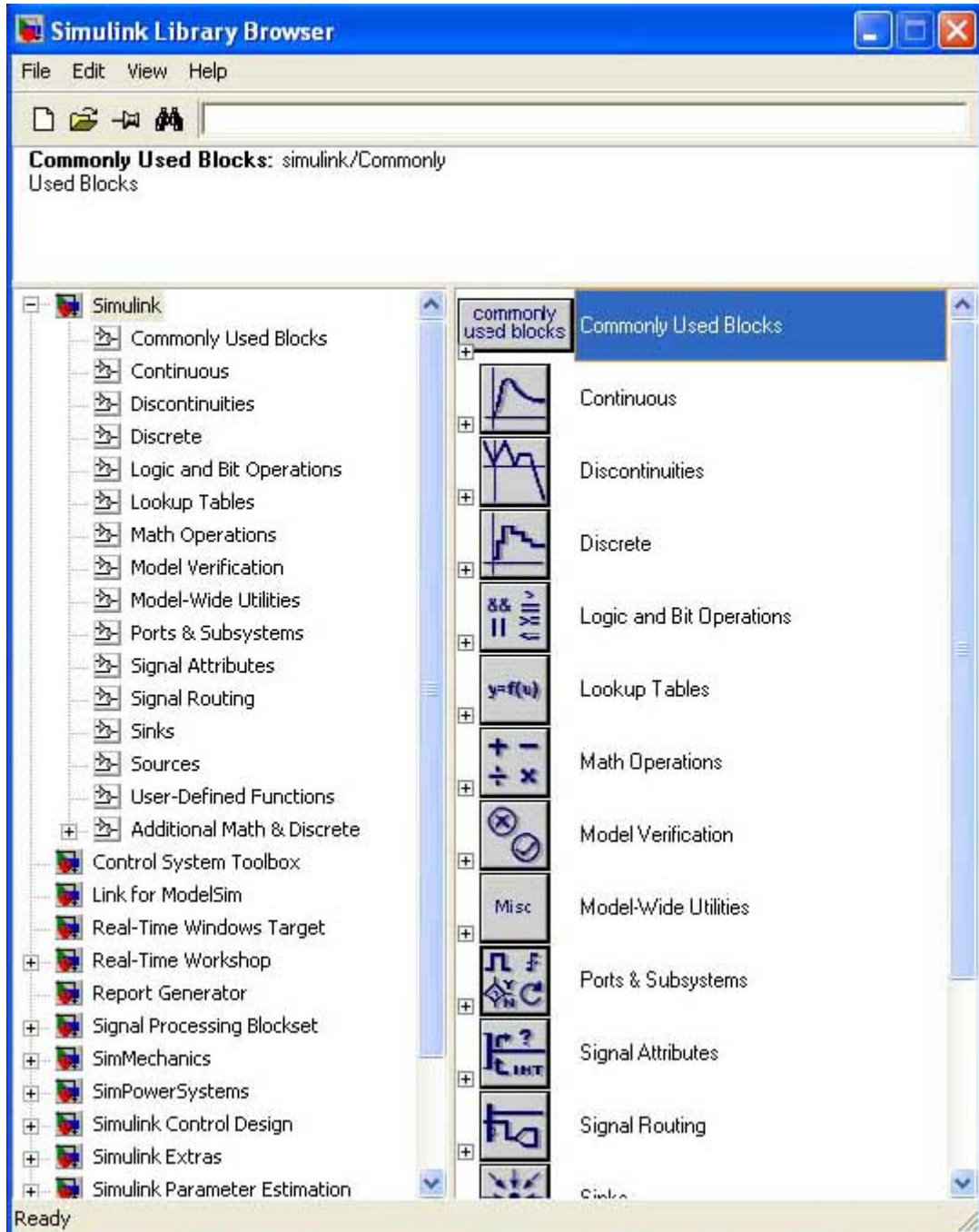
Model oluşturmak 2(Basit bir oransal denetleyici tasarımı)

Bu yazımızda Simulink'i kullanarak basit bir otomatik kontrol sisteminin simülasyonunu yapacağız.Böylelikle temel Simulink kullanımını da görmüş olacağız.

Simulink'i çalıştırmak için öncelikle MATLAB'i çalıştıralım.Simulink'i MATLAB araç çubuklarındaki Simulink simgesine tıklayarak ya da komut satırına *simulink* yazarak çalıştırabiliriz.

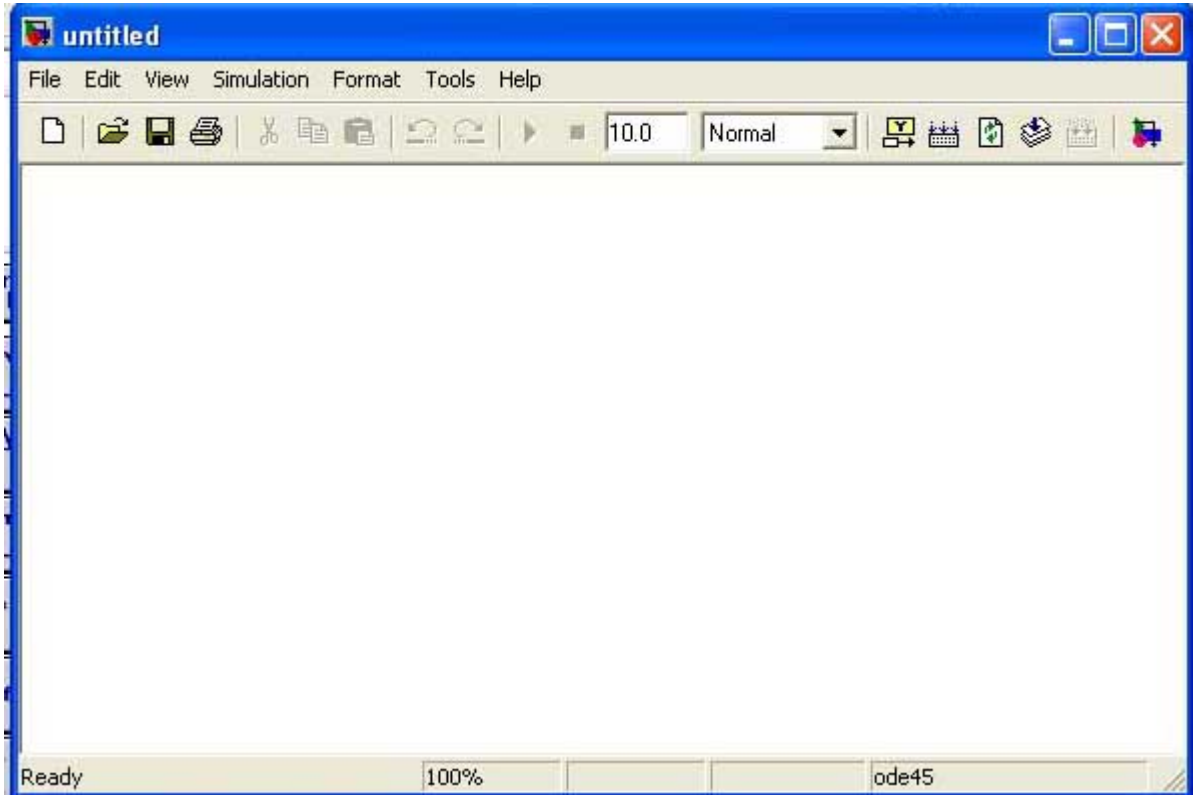


Simulink'i çalıştırdığımızda karşımıza Simulink Kütüphanesi gelecektir.Simulink blokları bu kütüphanede kategorilere göre ayrılmış olarak bulunmaktadır.

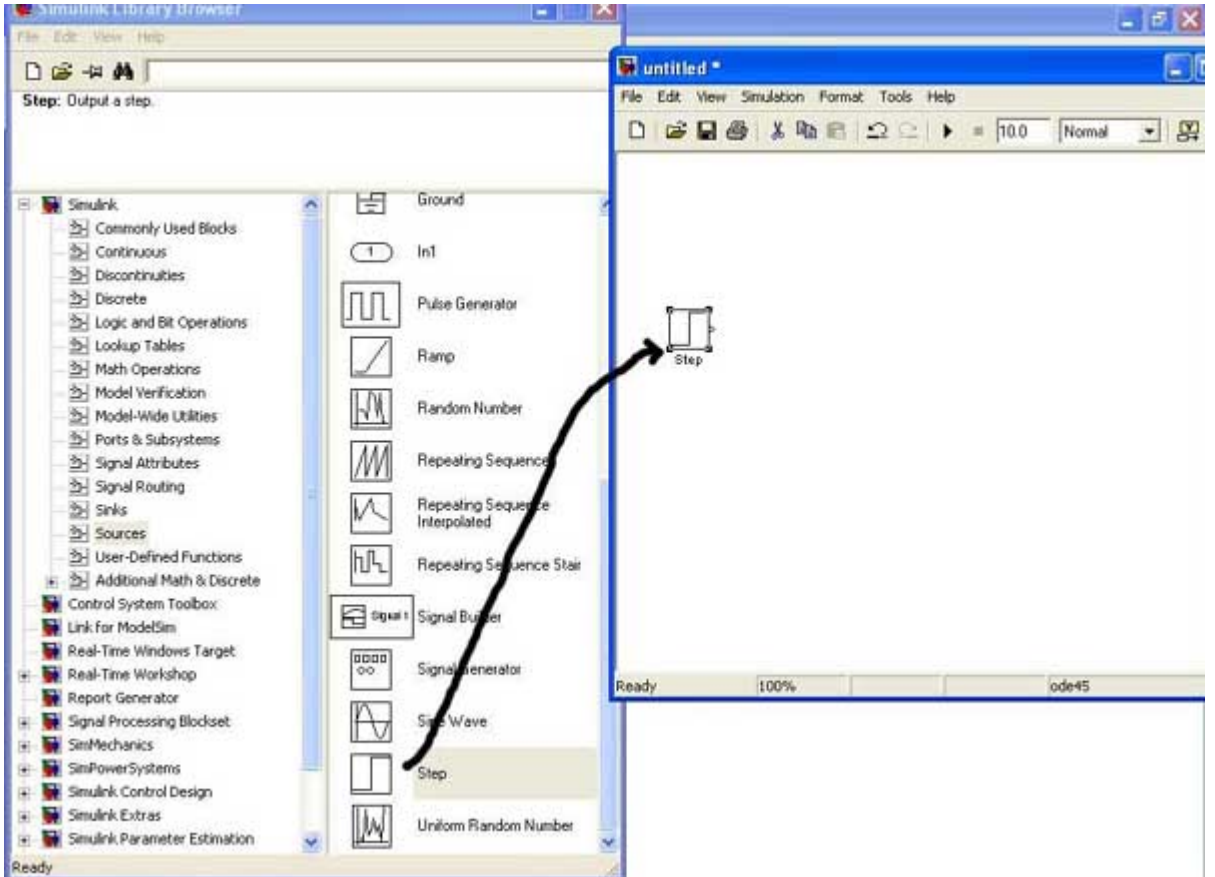


Simulink Kütüphanesi

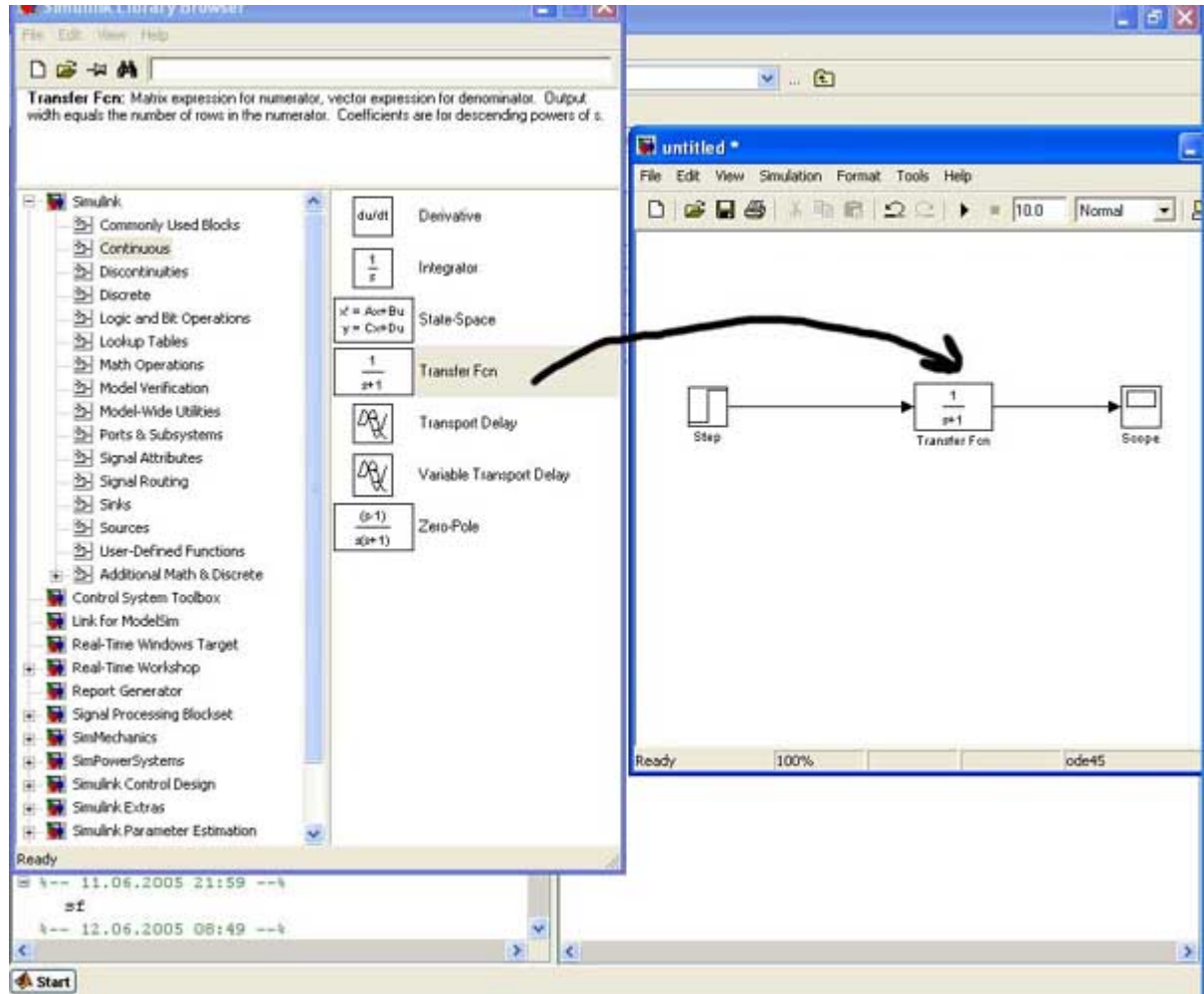
Simulink Kütüphanesinin file menüsünden New/model seçeneğine tıklayarak yeni bir Simulink ortamı açalım. Simulink otomatik olarak untitled* adında bir sayfa açacaktır. İsmi yanındaki * işareti henüz Simulink sayfamızı kaydetmediğimizi belirtiyor.



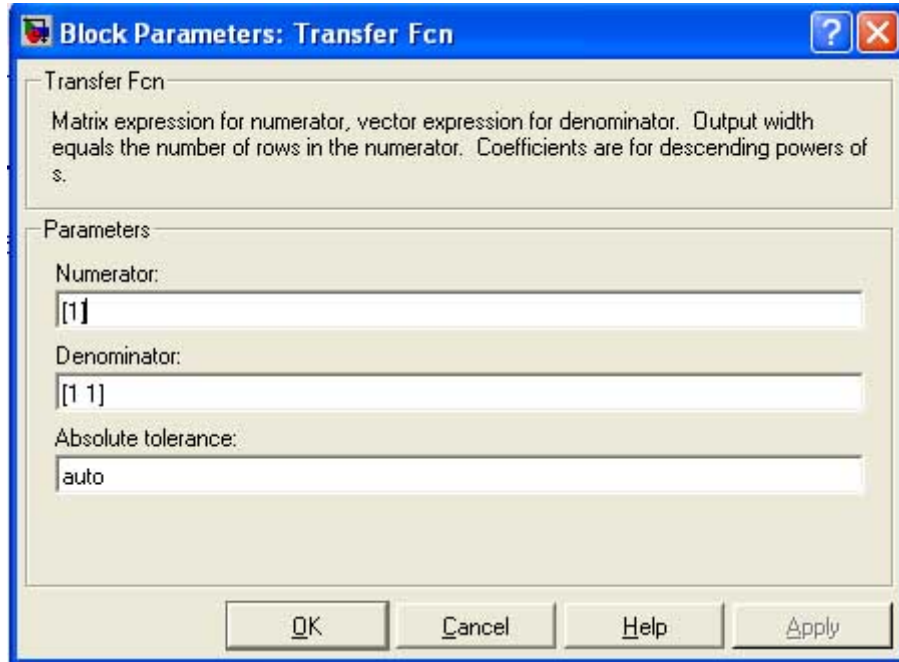
Şimdi kütüphanemizdeki **sources** bölümünde bulunan step adlı bloğa tıklayıp Simulink sayfamıza sürükleyip bırakalım.Bunu sistemin adım fonksiyonu cevabını bulmakta kullanacağız.



Şimdi de **continuous** kategorisinden *Transfer fcn* bloğunu **,sinks** bölümünden de *scope* bloğunu Simulink sayfamıza sürükleyip bırakalım.Sıra geldi bağlantıları yapmaya... Farenin imleci blokların uçlarındaki çıkıntıya getirdiğinizde + işareti şeklini aldığını göreceksiniz.Step fonksiyonun çıkışına fareyi götürüp tıklayarak ortaya çıkan yolu transfer fonksiyonunun girişine sürükleyip iliştirin.Önce kesik kesik görünen çizginin keskin bir hal aldığını göreceksiniz.Aynı işlemi transfer fonksiyonu ile scope bloğu için de uygulayın.Resimde görülen sistemi elde edeceksiniz.



Transfer fonksiyonu bloğuyla istediğiniz transfer fonksiyonunu Pay polinomu/Payda polinomu şeklinde oluşturabilirsiniz:Transfer bloğuna çift tıkladığınızda transfer bloğunun parametre diyalog kutusu açılacaktır:



MATLAB'de genel olarak polinom tanımlama şu şekilde olmaktadır:

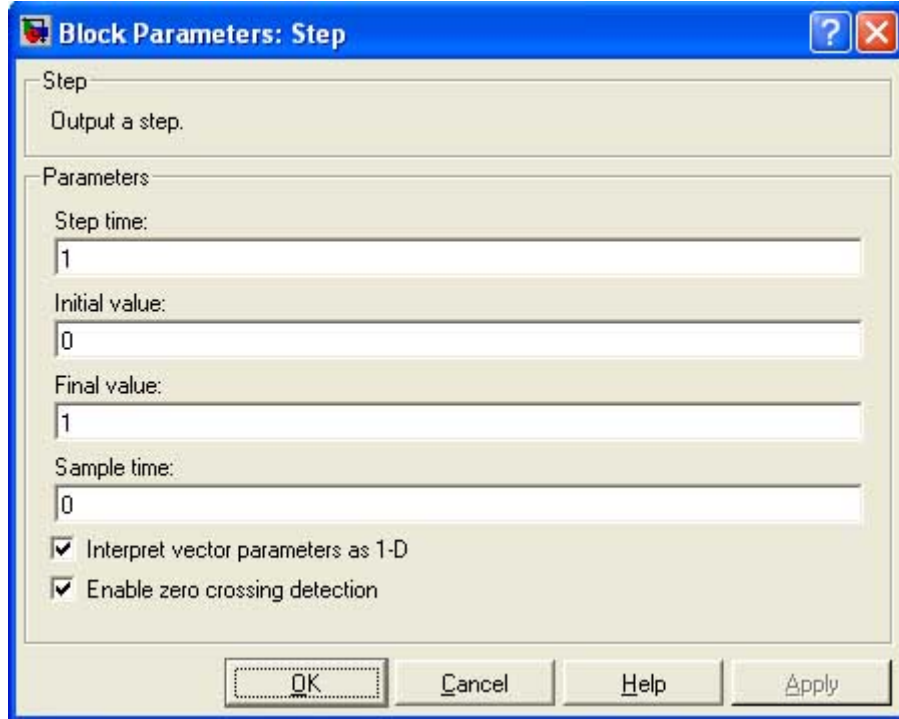
Örn: $s^3 + 3s^2 + s + 4 \implies [1 \ 3 \ 1 \ 4]$

$s^5 + s^2 \implies [1 \ 0 \ 0 \ 1 \ 0]$

Gördüğünüz gibi polinomları katsayılarını bir vektör biçiminde yazarak tanımlıyoruz. Olmayan terimlerin katsayılarına 0 yazıyoruz.

Transfer fonksiyonu bloğunun num ve den parametreleri ,transfer fonksiyonun pay ve payda polinomlarıdır. Bu polinomlara istediğiniz değerleri yukarıdaki biçimde yazarak istediğiniz transfer fonksiyonunu elde edebilirsiniz. Biz bu örneğimizde 1. dereceden bir sistemle çalışacağımız için bu değerleri olduğu gibi bırakıyoruz.

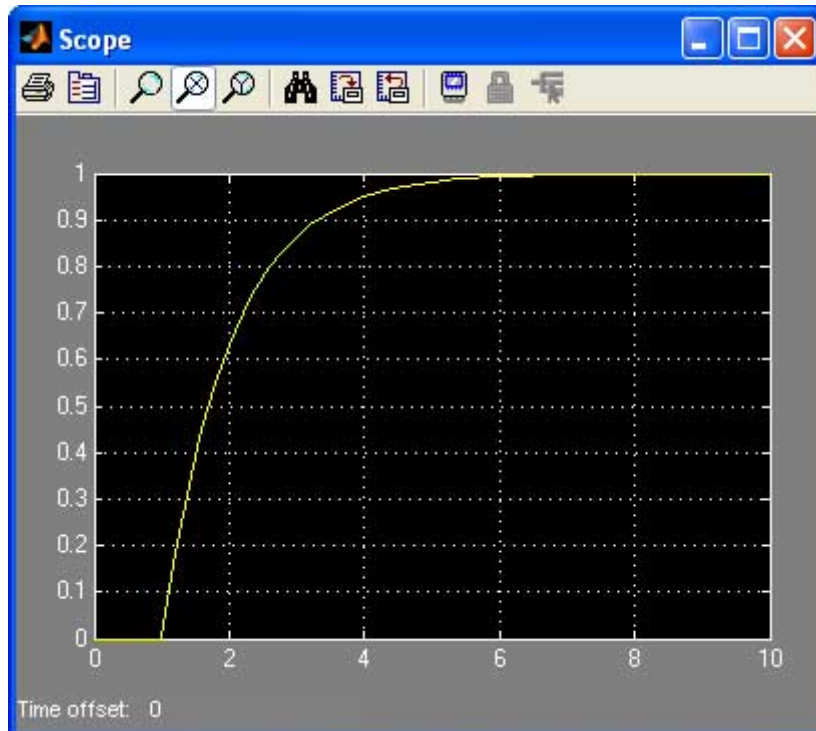
Step fonksiyonu bloğuna tıklayarak parametre ayarlarını yapacağımız diyalog kutusunu açalım. Step fonksiyonunun temel parametreleri şu şekildedir:



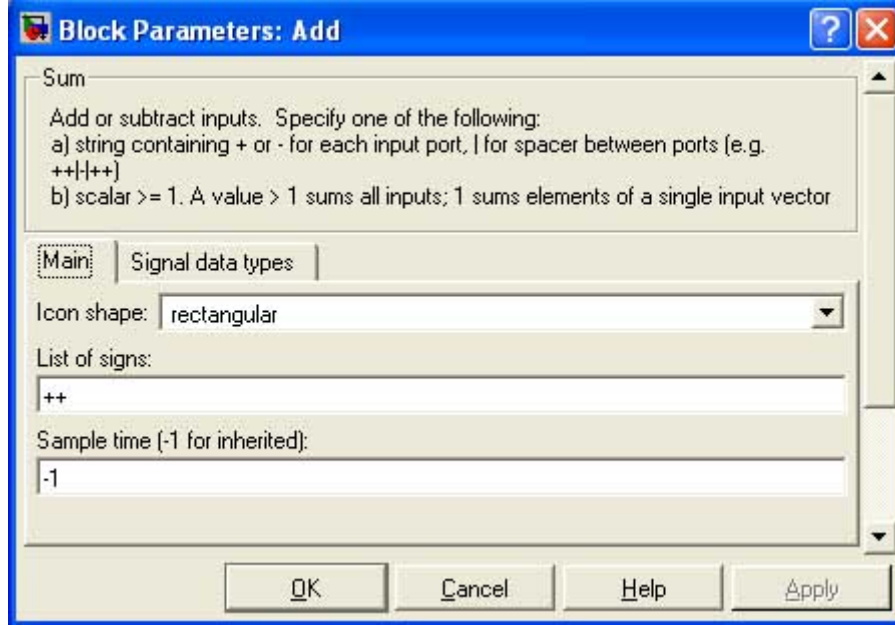
The image shows the 'Block Parameters: Step' dialog box in MATLAB/Simulink. It has a blue title bar with a question mark and a close button. The dialog is divided into two main sections: 'Step' and 'Parameters'. The 'Step' section contains the text 'Output a step.'. The 'Parameters' section contains four input fields: 'Step time:' with the value '1', 'Initial value:' with the value '0', 'Final value:' with the value '1', and 'Sample time:' with the value '0'. Below these fields are two checked checkboxes: 'Interpret vector parameters as 1-D' and 'Enable zero crossing detection'. At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Help', and 'Apply'.

Step time(Adım zamanı - saniye):Step fonksiyonun değer değiştireceği zaman
Initial value(Başlangıç değeri):Step fonksiyonunun başlangıç değeri
Final Value(Son Değer):Step fonksiyonun adım zamanında alacağı son değer

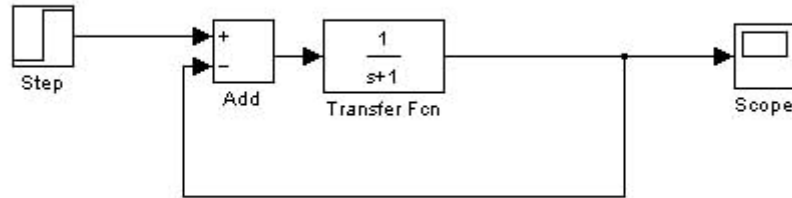
Bu değerlerden istediğinizi değiştirerek değişik denemeler yapabilirsiniz.Şimdi başlangıç değerini 0,son değeri de 1 alarak 1.dereceden sistemimizin cevabını scope'da gözlemeyelim.Simulasyonu çalıştırmak için Simulink araççubuğundan similasyonu başlatma butonuna ► tıklayın.



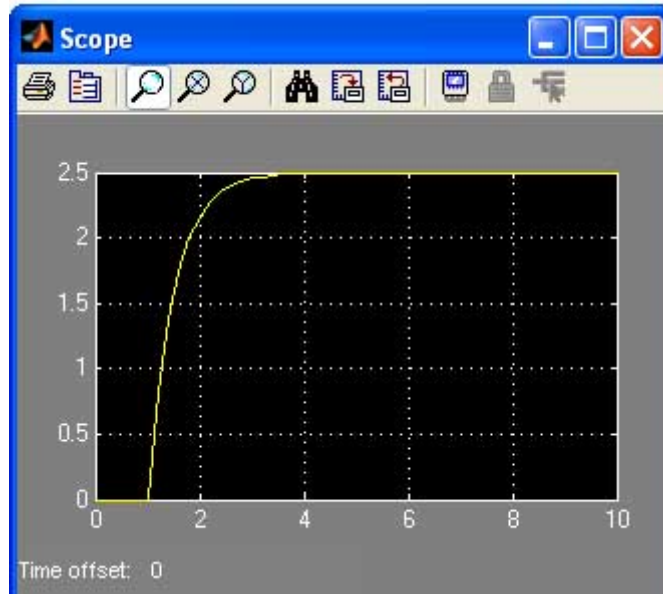
Resimde sistemimizin açık çevrim adım fonksiyonu cevabı görülmektedir. Gördüğünüz gibi sistemimiz 6.saniyeye yakın bir zamanda 1 değerini almaktadır. Şimdi sisteme geribesleme ekleyerek yeniden gözlemleyelim. Sistemin çıkışından alacağımız sinyali girişten çıkararak sisteme gireceğiz. Sisteme geri besleme eklemek için step fonksiyonundan sonra bir toplama bloğu kullanmamız lazım. Bunun için Simulink kütüphanesinden **Math Operations** bölümünden Sum (toplama) bloğunu sürükleyip Step fonksiyonunun yanına bırakın. Ardından toplam bloğuna çift tıklayarak parametre penceresini açın. Orada bulunan list of signs adlı bölümdeki iki ++ işaretinin sondakini silerek eksi yapın (+-)



Önceki bağlantıları koparmak için sinyallerin üzerine tıklayıp seçili hale getirin ve delete tuşuyla onları silin. Step fonksiyonunu sum bloğunun + girişine bağlıyoruz. Eksi girişin üzerinde fareyle tıklayıp çıkışa doğru sürükleyerek scope'a giden sinyal üzerine yapıştırıyoruz. Sistemimiz şekildeki gibi görülmelidir:

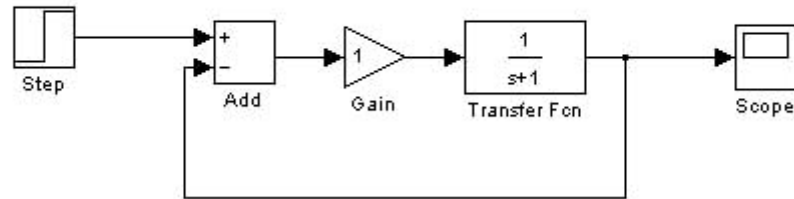


Bu sistemin cevabını görmek için similasyonu çalıştırın ve scope çift tıklayın:

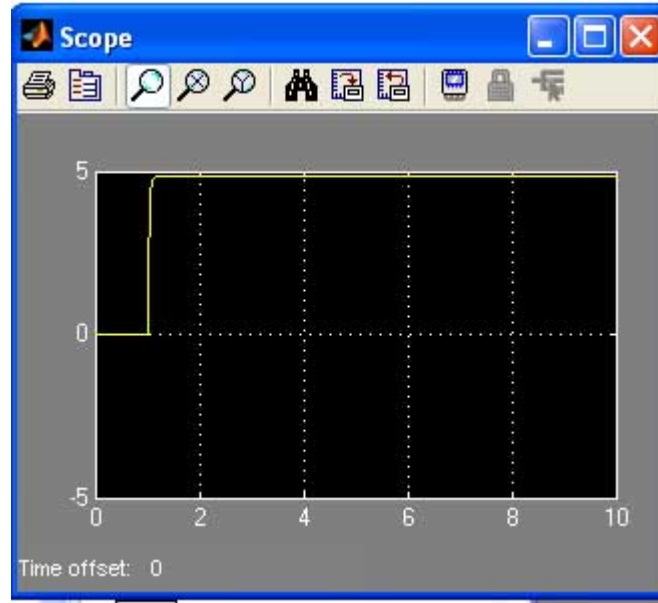


Görüldüğü gibi sistemimizin daha hızlı bir şekilde giriş tepki göstermiştir.Bu geribesleme kullanmanın bir avantajıdır.Ancak görüldüğü gibi sistem kararlı duruma geçtiğinde girişte vermiş olduğumuz 5 değerine değil,2.5 değerine oturmuştur.İstediğimiz 5 değerine oturması için sisteme bir de kazanç bloğu ekleyelim.

Kazanc için Simulink kütüphanesinden **Math Operations** bölümünden Gain bloğuna alarak modelimize ekleyelim.Gain bloğunu Sum bloğu ile transfer fonksiyonu arasına ekleyin.Gain bloğunun yaptığı işlem sadece girişte aldığı değeri içinde bulunan gain(kazanc) katsayısı ile çarpmaktır.Sistemimizin son hali aşağıda görülmektedir:



Gain bloğuna çift tıklayarak blok parametreleri diyalog kutusunu açın.Buradaki gain bölümündeki sayıyı 30 yapın*.Ardından diyalog kutusuna OK diyerek simülasyonu çalıştırın.Sistemimizin cevabı şu şekilde olacaktır:



Görüldüğü gibi sistemimiz girişte verilen 5 değerine hızlı bir şekilde oturmuştur. Burada yapmış olduğumuz basit P (proportional- oransal) kontrol sistemimizdir. Aynı zamanda Simulink'in kullanımını da görmüş olduk. Siz de kendi kendinize değişik sistemlerin cevaplarını izlemek için değişik denemeler yapabilirsiniz. Hepinize kolay gelsin

Simulink İle Fiziksel Bir Sistemin Modellenmesi:

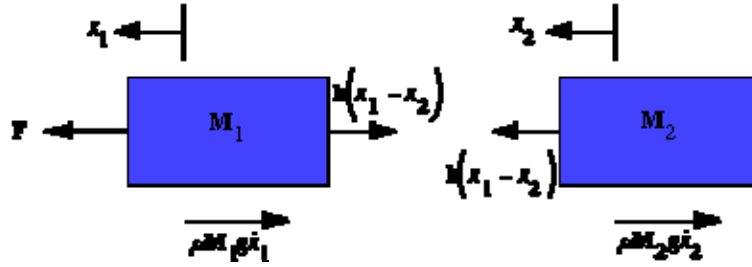
Bu yazıda Simulink ile fiziksel bir sistemin modellenmesini göreceğiz.Örnek olarak bir lokomotif ve bu lokomotif tarafından çekilen bir vagonu modelleyeceğiz.Trenin bir yönde hareket ettiğini varsayacağız.Amacımız trenin yumuşak kalkış ve duruş yapması ve sabit hızda gidebilmesini sağlayacak kontrol sistemini incelemek.



Lokomotifin ve vagonun kütlesine sırasıyla M_1 ve M_2 olsun.İkisi de birbirine k sabitine sahip bir yay ile tutturulmuştur. F , motor tarafından oluşturulan itme gücünü," μ " ise dönme sürtünme katsayısını temsil etmektedir.

SİSTEM MODELİ VE DENKLEMLERİ:

Sistemimize etkiyen kuvvetler aşağıdaki şekilde gibidir:



Newton kanunundan bildiğimiz gibi bir kütleye etki eden kuvvetlerin toplamı ,kütle ve ivme çarpımına eşittir.Bu durumda M_1 kütlesine etki eden kuvvetler , M_2 kütlesinin çekmesi,sürtünme ve motorun sağladığı itme kuvvetleridir. M_2 kütlesine de M_1 'in çekmesi ve sürtünme etki etmektedir.Modelimizi aşağıdaki denklemlere göre kuracağız:

$$\text{Toplam}(M_1_etkiyen_Kuvvet)=M_1 \cdot x_1''$$

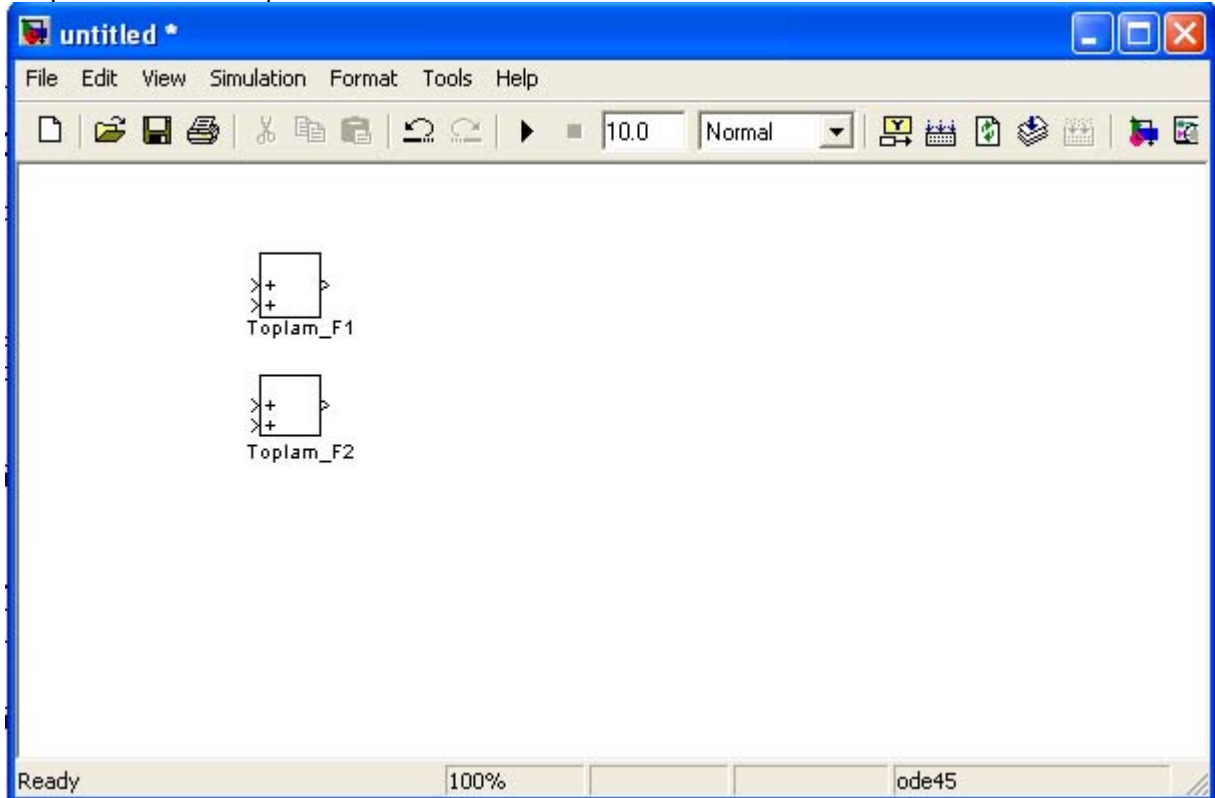
$$\text{Toplam}(M_2_etkiyen_Kuvvet)=M_2 \cdot x_2''$$

(x_1'' ve x_2'' , x_1 ve x_2 yerdeğiřtirmelerinin 2.türevi(ivme))

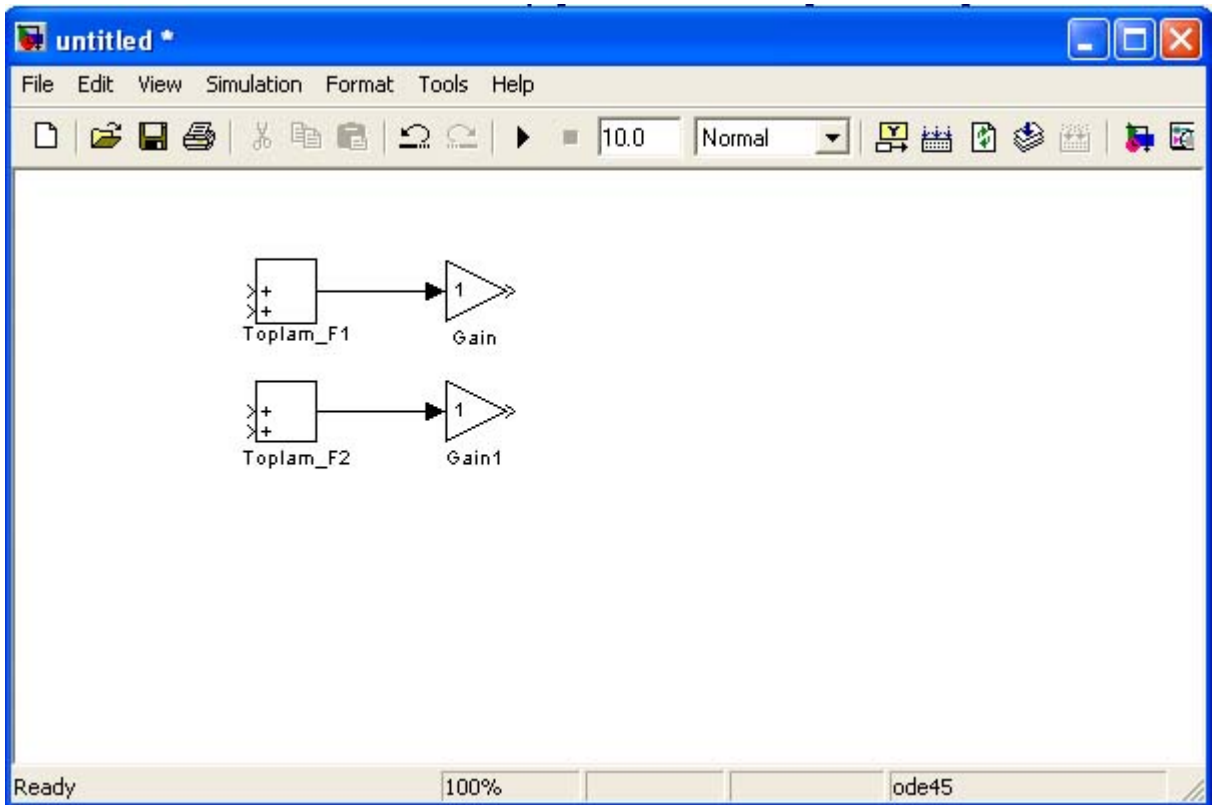
MODELİN OLUŐTURULMASI:

Őimdi denklemlerin görsel olarak oluőturulmasına geçebiliriz.Öncelikle her iki kütle için toplam $F=Ma$ ya da $a=(1/M) \cdot F$ formölünü oluőturacağız.İlk olarak yeni bir model penceresi açın ve Simulink kütüphanesinin Commonly Used Blocks bölümünde iki adet Sum(toplam) bloęunu modele sürükleyin.Bu bölümde kullanacağımız blokların birçoęunu Commonly Used Block bölümünde bulabileceksiniz.Blokların üzerine çift tıklayarak shape bölümlerinden rectangular'ı seçip şekillerini dörtgen haline getirin.Bu bloklara sırasıyla

"Toplam F1" ve "Toplam F2" adını verin.

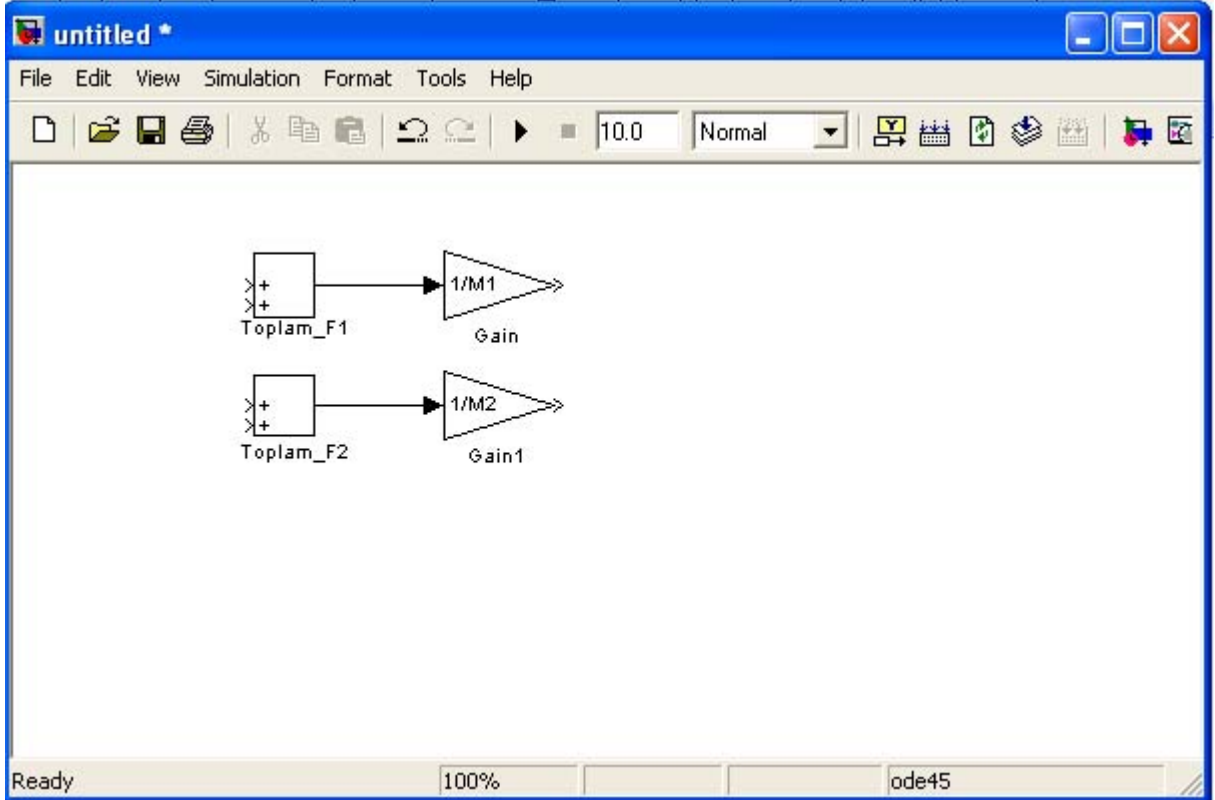


Bu iki Sum bloğunun çıkışları bize kütlelere etki eden kuvvetleri verecektir. Bu kuvvetleri de $1/M$ ile çarptığımızda ivmeyi elde edeceğiz. Modelinize iki adet Gain (kazanç) ekleyerek bunları Sum bloklarının çıkışlarına bağlayın.

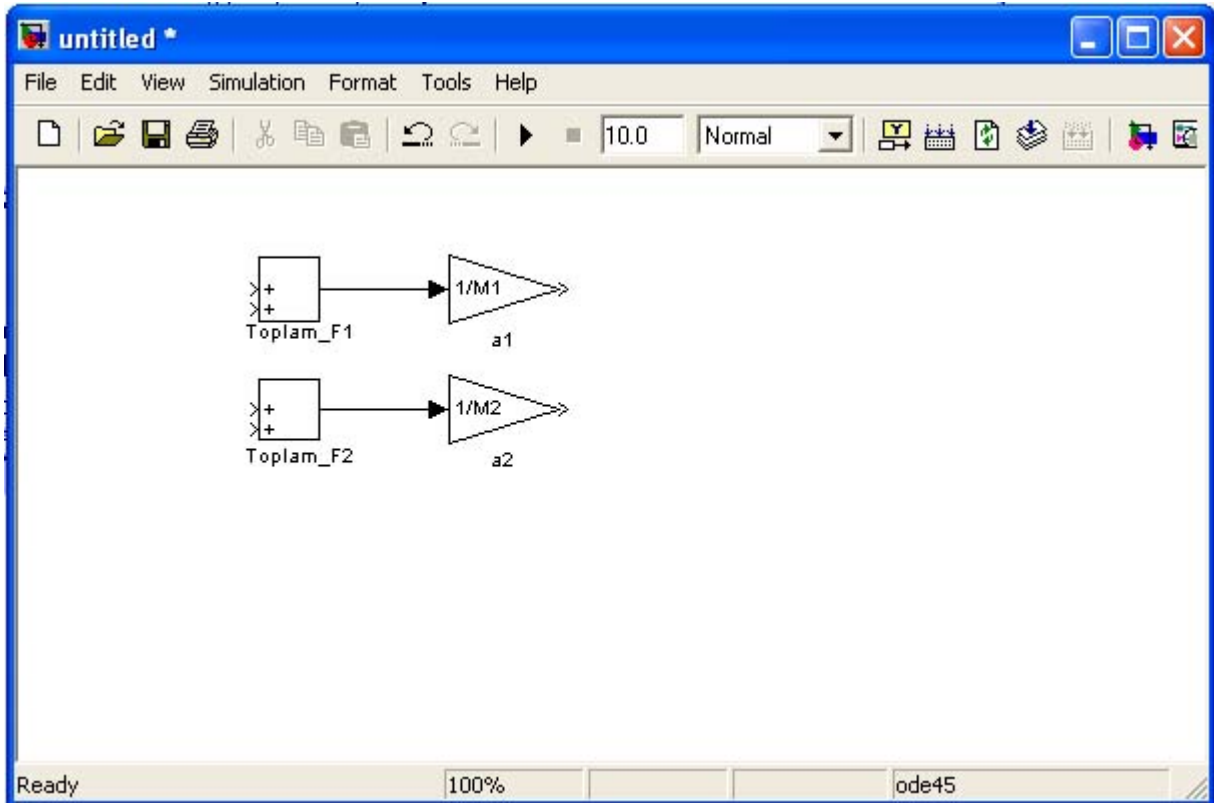


Bu Gain bloklarının gain değerleri sırasıyla $1/M1$ ve $1/M2$ olmalıdır. Bu $M1$ ve $M2$ değişkenlerinin değerlerini MATLAB ortamından alacağız. Şimdilik sadece isimlerini kullanıyoruz. Gain bloklarına çift tıklayarak Gain bölümlerine ilgili değerleri yazın.

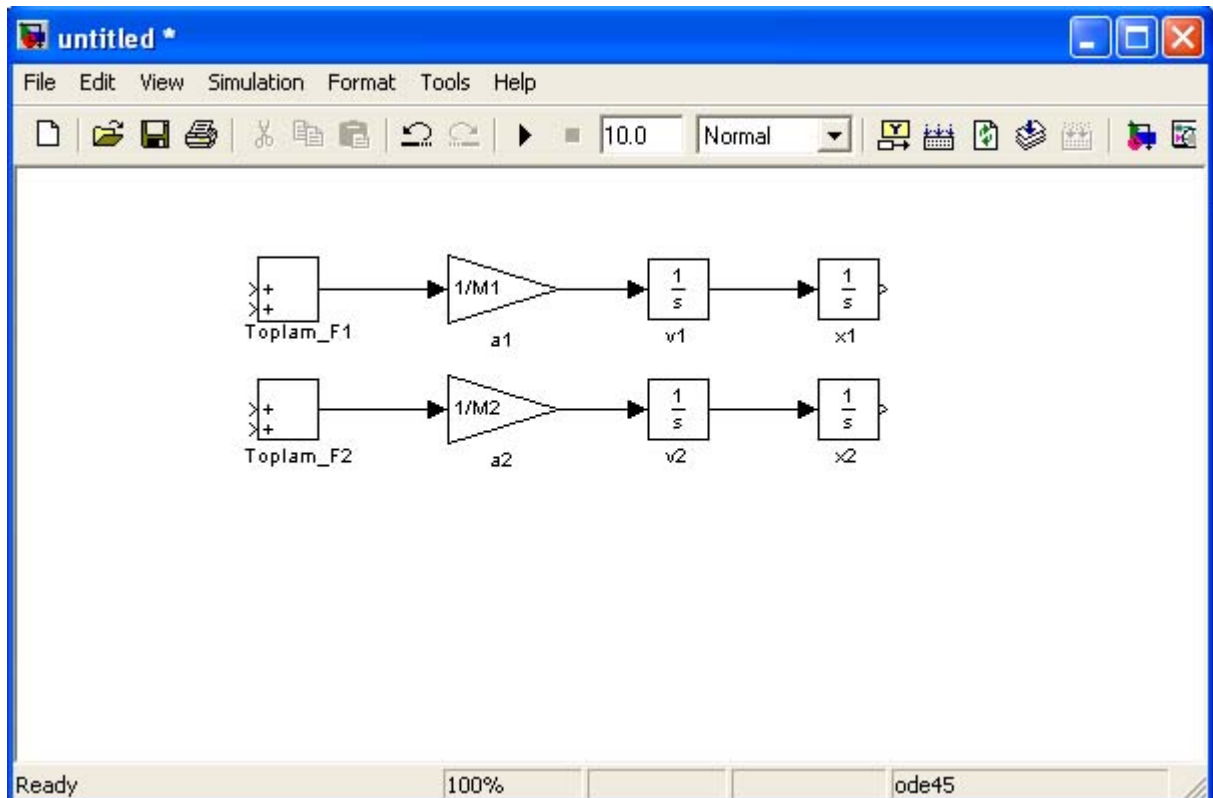
Şimdi Gain bloklarının içinde $-K-$ değerini göreceksiniz. Bu blokların üçgen şekillerinin içlerindeki $1/M1$ ve $1/M2$ değerlerini gösterecek kadar büyük olmadıklarını belirtmektedir. Blokların köşelerine tıklayıp sürükleyerek şekillerinin büyümesini sağlayın.



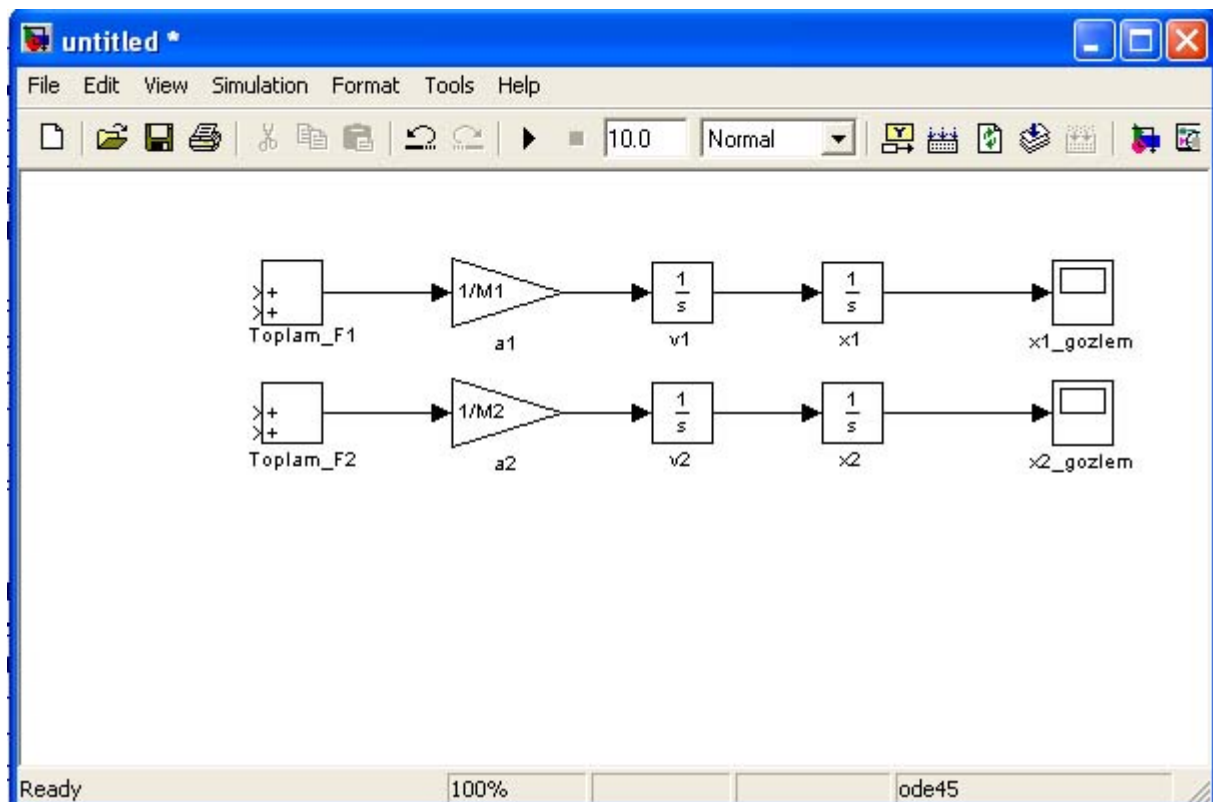
Gain bloklarının büyüklüklerini ayarladıktan sonra Sum bloklarıyla aynı hizaya getirin ve a1,a2 olarak adlandırın.



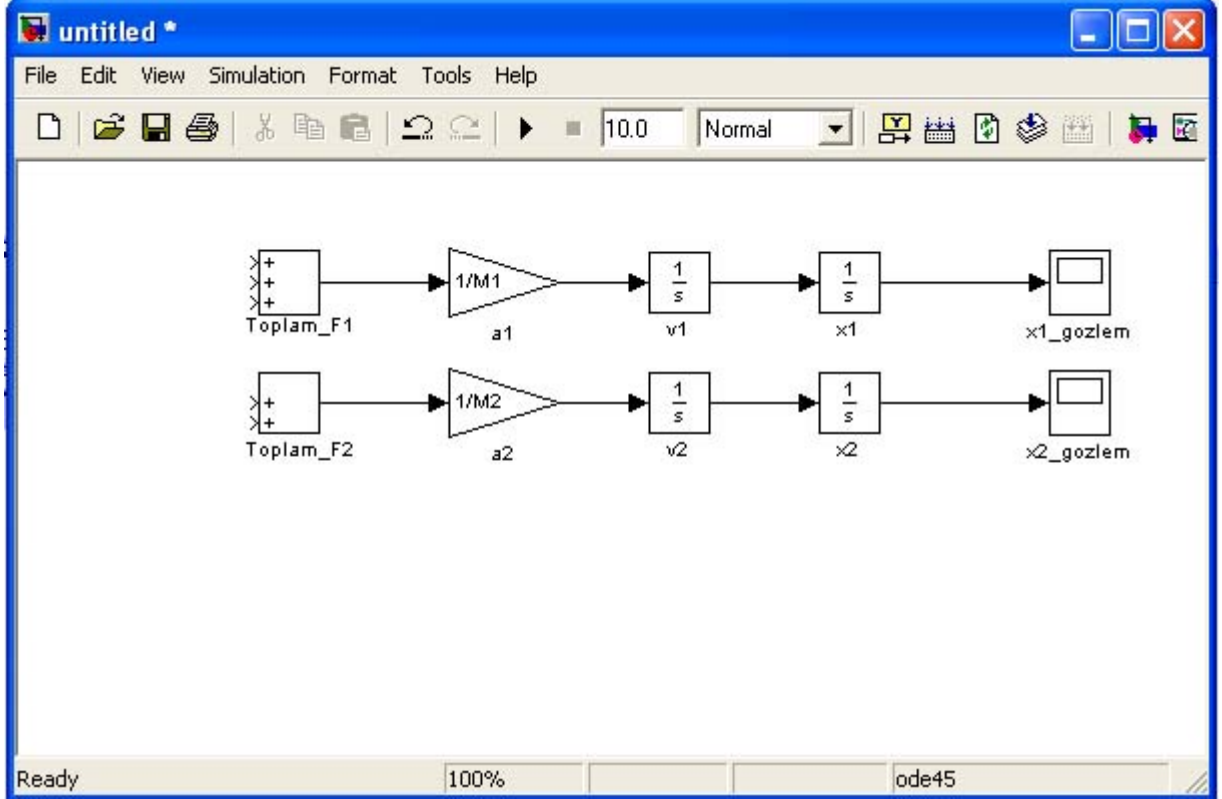
Gain bloklarının çıkışları bizlere kütlelerin ivmelerini verecektir. Bize hem kütlelerin ivmeleri hem de konumları gerektiğinden ivmelerin integrallerini alarak önce hızları, sonra hızların integralini alarak konum bilgilerini elde edeceğiz. Her bir kütle için iki adet olmak üzere dört adet integral bloğunu modele ekleyelim. Blokların çıkışlarını şekildeki gibi "v1", "x1", "v2", "x2" adlarını verelim. Bu adlar her bir bloğun çıkışının ne olduğunu belirtmektedir.



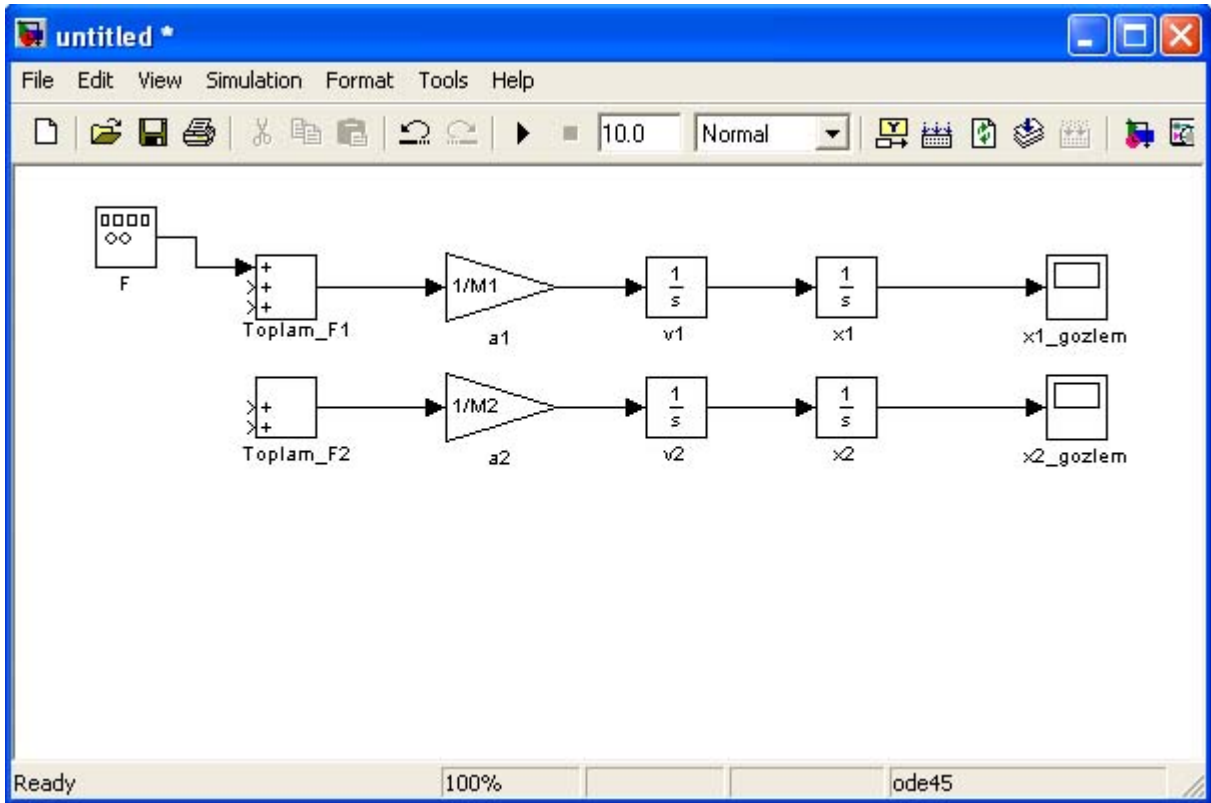
Şimdi de x1 ve x2 bloklarının çıkışlarına scope bağlayalım.(commonly used blocks kütüphanesinden)



Şimdi kütlelere etkiyen kuvvetleri ekleyebiliriz.İlk olarak Sum bloklarının giriş sayılarını, bütün kuvvetleri bağlayabilmemiz için üçe çıkarmamız gerekmekte.M1'e etkiyen üç kuvvet olduğuna göre ilk Sum bloğunun diyalog kutusunu açarak ilgili bölümü +++ haline getirelim:



M1 kütleline etkiyen ilk kuvvet motorun sağladığı F giriş kuvveti.Motoru simule etmek için Sinyal Generatörünü kullanacağız.Kütüphaneden Signal Generator bloğunu alarak(Sources kütüphanesinden) Sum-F1 bloğunun en üstteki girişine bağlayalım.Genetörü F diye isimlendirelim:

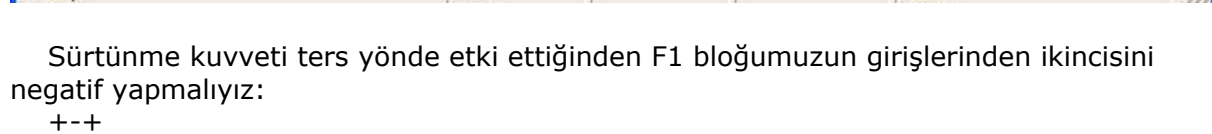


M1' etkiyen ikinci kuvvetimiz srtnme kuvveti.Bu kuvvetin forml şkilde:

$$F_{\text{surtunme}_1} = \mu * g * M1 * v1$$

Bu kuvveti oluřturabilmek iin hız sinyalinin bir dalı olarak bunu bir gain bloęuna baęlıyoruz.İlk olarak bir Gain bloęunu modelimizde uygun yere ekleyelim.Ardından v1 integrator bloęunun ıkıřından bir dalı olarak bunu Gain bloęumuzun giriřine baęlayalım.Gain bloęunun ıkıřını da F1 bloęumuzun ikinci giriřine baęlayalım.Ardından Gain bloęunun diyalog kutusundan Gain deęerine ařaęıdaki ifadeyi yazalım:

$$\mu * g * M1$$

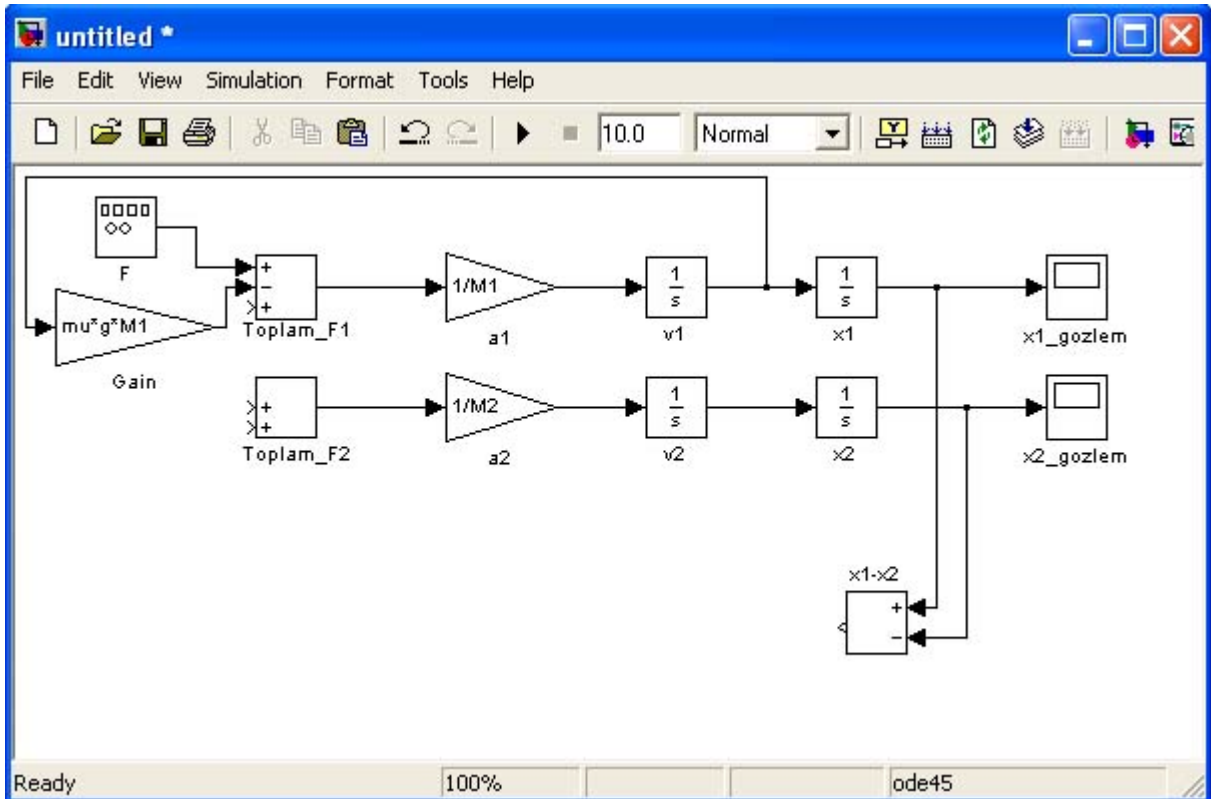


M1'e etkiyen son kuvvet de kütleler arasındaki çekme kuvveti:
 $k \cdot (x_1 - x_2)$

İlk olarak $(x_1 - x_2)$ ifadesini oluşturmamız gerekir. Modelimizin en altına bir Sum bloğu ekleyelim ve giriş listesini şu şekilde getirelim (veya direkt olarak commonly used blocks kütüphanesinden subtract bloğunu ekleyebiliriz):

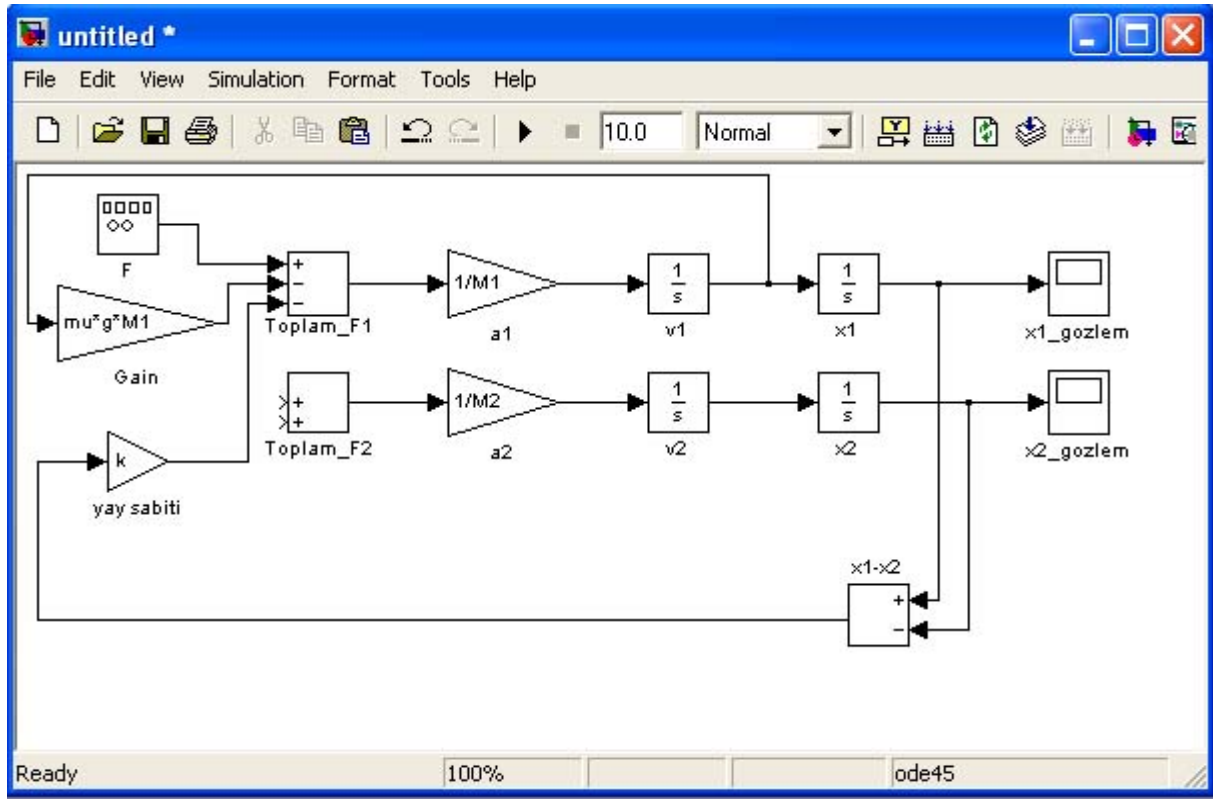
Toplama işlemi sağ taraftan sol tarafa doğru geldiğinden Sum bloğuna sağ tıklayıp açılan menüden format altmenüsünden Flip seçeneğine tıklayarak yönünü değiştirelim. Modelimizi aşağıdaki gibi görünmelidir:

Toplama işlemi sağ taraftan sol tarafa doğru geldiğinden Sum bloğuna sağ tıklayıp açılan menüden format altmenüsünden Flip seçeneğine tıklayarak yönünü değiştirelim. Modelimizi aşağıdaki gibi görünmelidir:

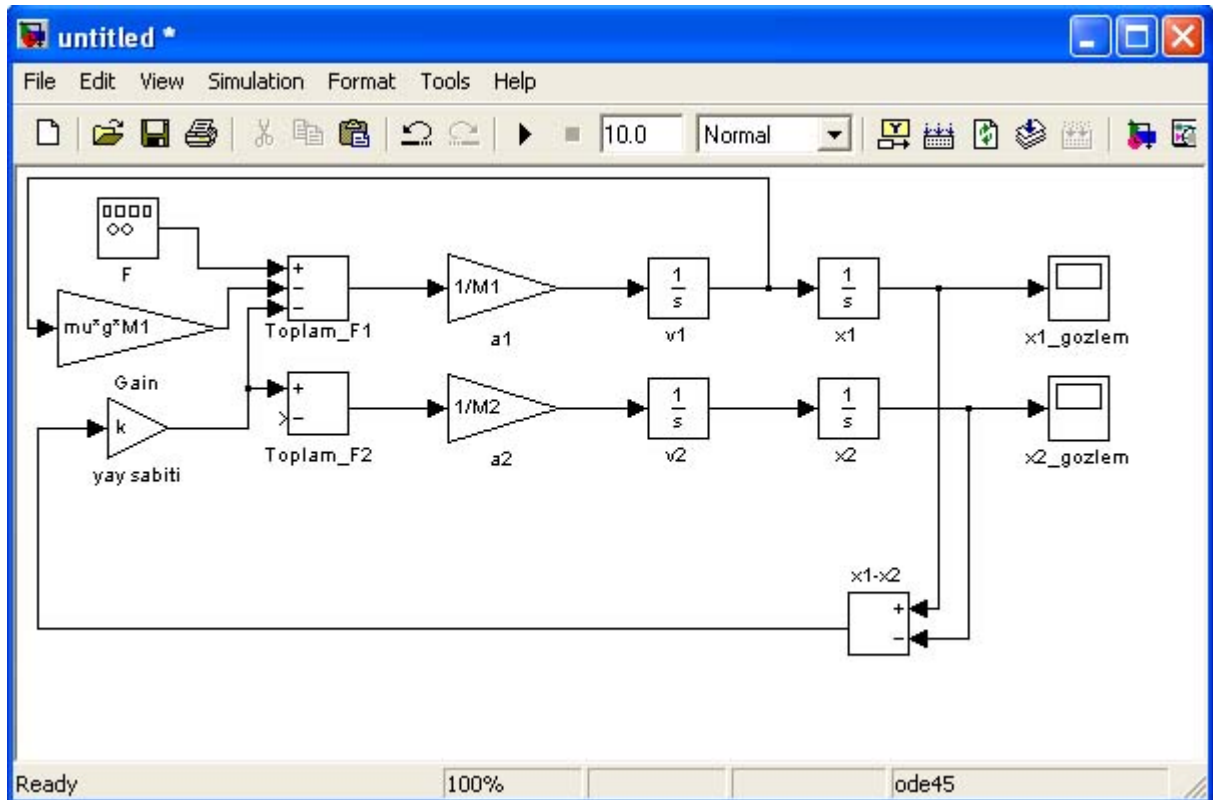


x2 sinyalinden bir dal alarak $(x1-x2)$ Sum bloğunun negatif girişine, x1 sinyalinden dal alarak onu da aynı bloğun pozitif girişine bağlayın*. Bu bağlantıları gerçekleştirdiğinizde bazı hatların üst üste geldiklerini göreceksiniz. Çizgilerin üst üste gelmesi bir sorun oluşturmuyor. Birbirleriyle bağlantılı olan hatlarda ufak bir bağlantı noktası görülür.

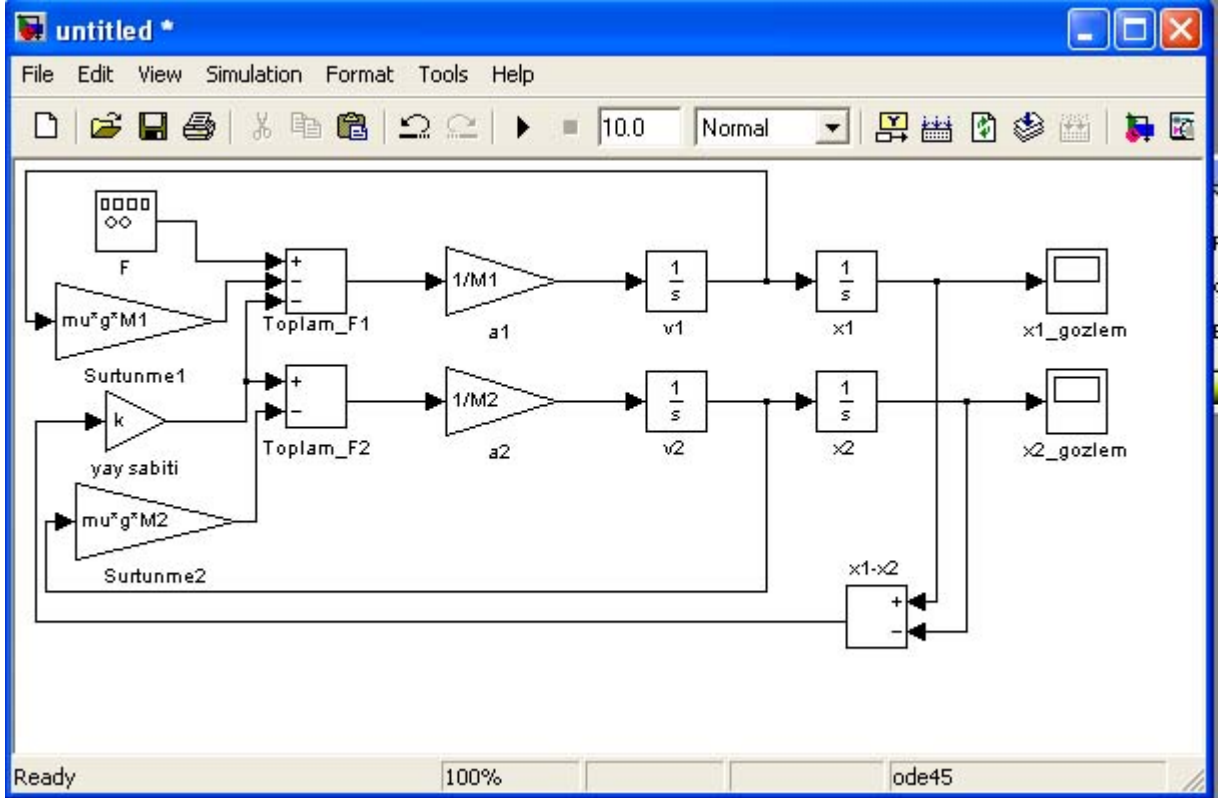
Şimdi konum farkını yay sabiti ile çarpılarak yay gücünü hesaplayabiliriz. Toplam F bloklarının sol tarafına bir Gain bloğu ekleyin. Ardından gain değerini k olarak belirleyin ve bloğu "yay" olarak adlandırın. $(x1-x2)$ bloğunun çıkışını bu yay bloğuna bağlayın, yay bloğunun çıkışını da Toplam F1 bloğunun üçüncü girişine bağlayın. Toplam F1 bloğunun üçüncü girişinin işaretini negatif yapın (+-- haline getirin).



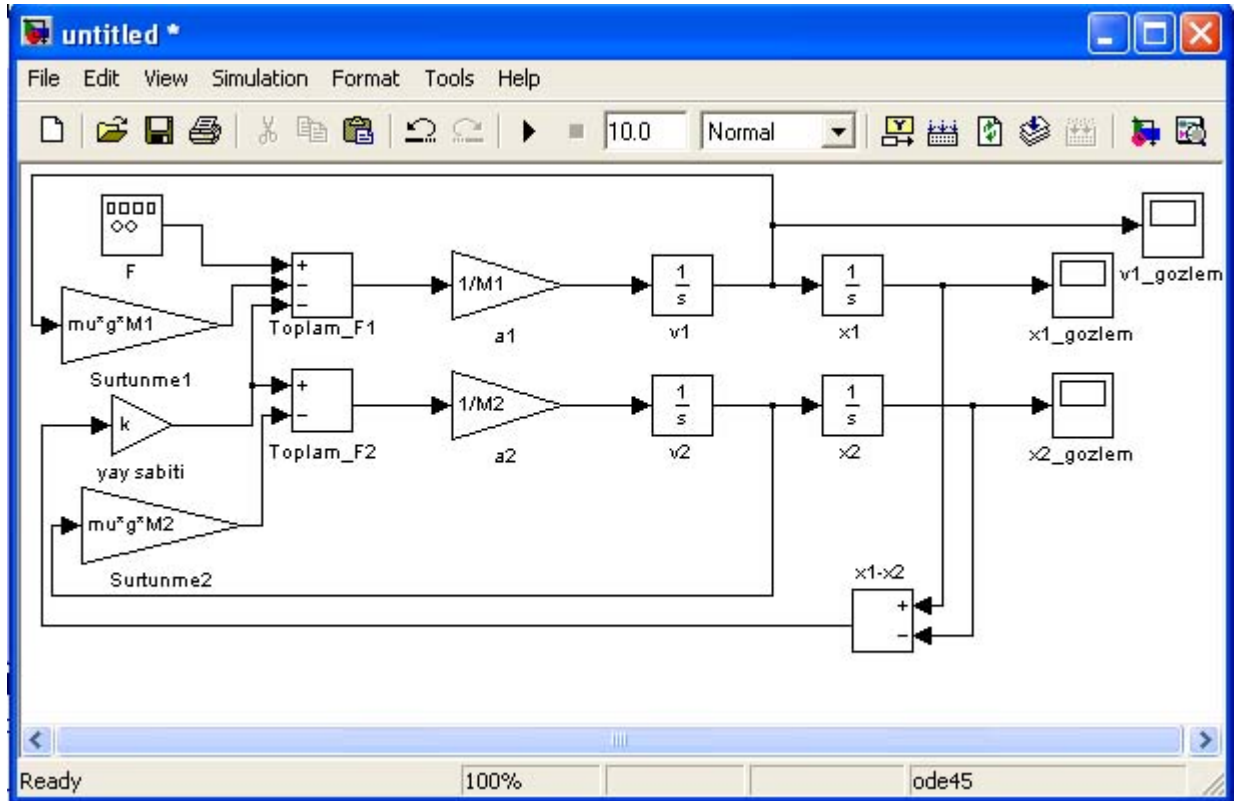
Şimdi M2'ye etkiyen kuvvetleri oluşturabiliriz. İlk etkiyen kuvvet olarak yay kuvvetini alacağız. Yay kuvveti M1'e etkiyen kuvvetle aynı olduğundan daha önce oluşturduğumuz kuvvetten bir dal alarak bunu Toplam F2 bloğuna negatif işaretle bağlayacağız.



M2'ye etki eden son kuvvet olan srtnme kuvvetini aynen M1'de olduėu gibi oluřturacaėız.v2 ıkıřından aldığımızı dalı $\mu \cdot g \cdot M2$ ile arparak Toplam F2 bloėuna negatif iřaretili olarak gireceėız.Bu iřlemler bittiėinde modelimiz řı řekilde grnecektir:



Modelimizi tamamlamıř bulunuyoruz.řimdi modelimize uygun giriřler verip ıkıřı gzlemleyeceėız.Sistemimizin giriři motor tarafından saėlanan F kuvveti olacak.Bu F kuvvetini de Sinyal Generatr ile modelleyeceėız.Sistemimizin ıkıřı da motorun hızı olacak.Bunun iin Sinks veya Commonly Used Blocks blok ktphanesinden bir Scope bloėunu alarak "v1" integrator bloėunun ıkıřına baėlayın ve bu scope'u "v1 gzlem" olarak adlandırın.



Modelimizi tamamladık.Şimdi modeli uygun bir isimle kaydedebilirsiniz.

MODELİMİZİ ÇALIŞTIRALIM:

Modelimizi çalıştırmadan önce modelimizde kullandığımız değişkenlerin değerlerini vermeliyiz.Modelimizin parametreleri aşağıdaki gibi olduğunu farzediyoruz:

- $M1 = 1 \text{ kg}$
- $M2 = 0.5 \text{ kg}$
- $k = 1 \text{ N/sec}$
- $F = 1 \text{ N}$
- $u = 0.002 \text{ sec/m}$
- $g = 9.8 \text{ m/s}^2$

MATLAB komut satırında edit tren.m komutunu vererek yeni bir MATLAB m-dosyası oluşturun ve içine şu satırları yazarak kaydedin:

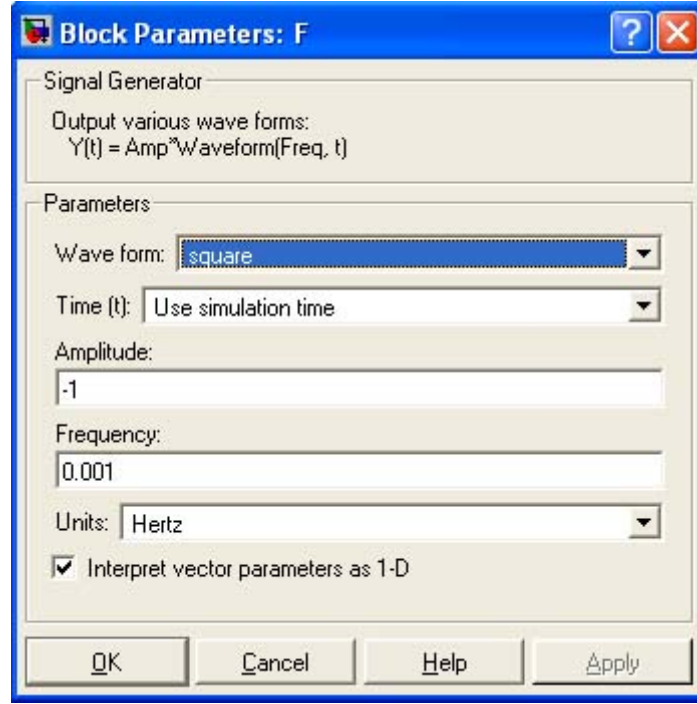
```
M1=1;
M2=0.5;
k=1;
F=1;
mu=0.002;
g=9.8;
```

Şimdi MATLAB komut satırında tren yazdığınızda bu değişkenler belleğe yerleşecek ve Simulink bu değişkenleri tanıyacaktır.

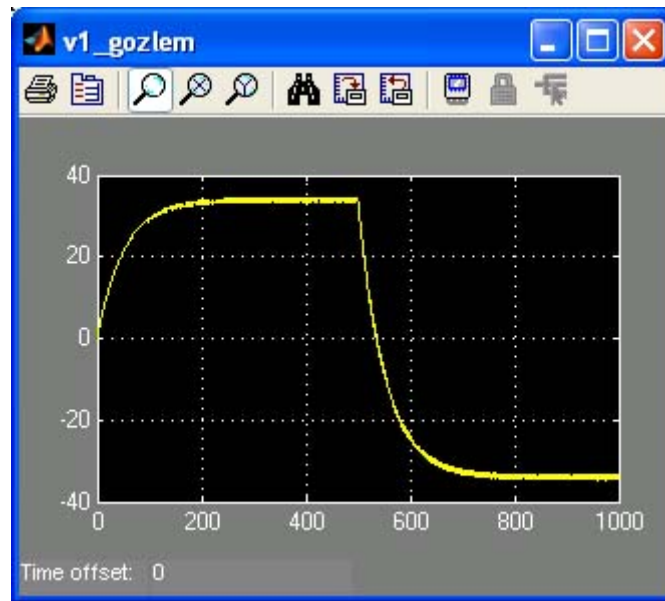
Motorumuzun girişini oluşturalım.Fonksiyon Generatörüne (F blok) çift tıklayın.Square Wave'(kara dalga)i seçerek frequency değerini 0.001 Hz ,amplitude değerini de -1 olarak

ayarlayın.

Şimdi de similasyonumuzun zaman ayarını yapacağız.0.001 Hz'lik karedalgayı görebilmek için similasyonun 1000 saniye sürmesi gerekmektedir.Bunu için araç çubuğundaki similasyon stop time bölümüne 1000 yazın.

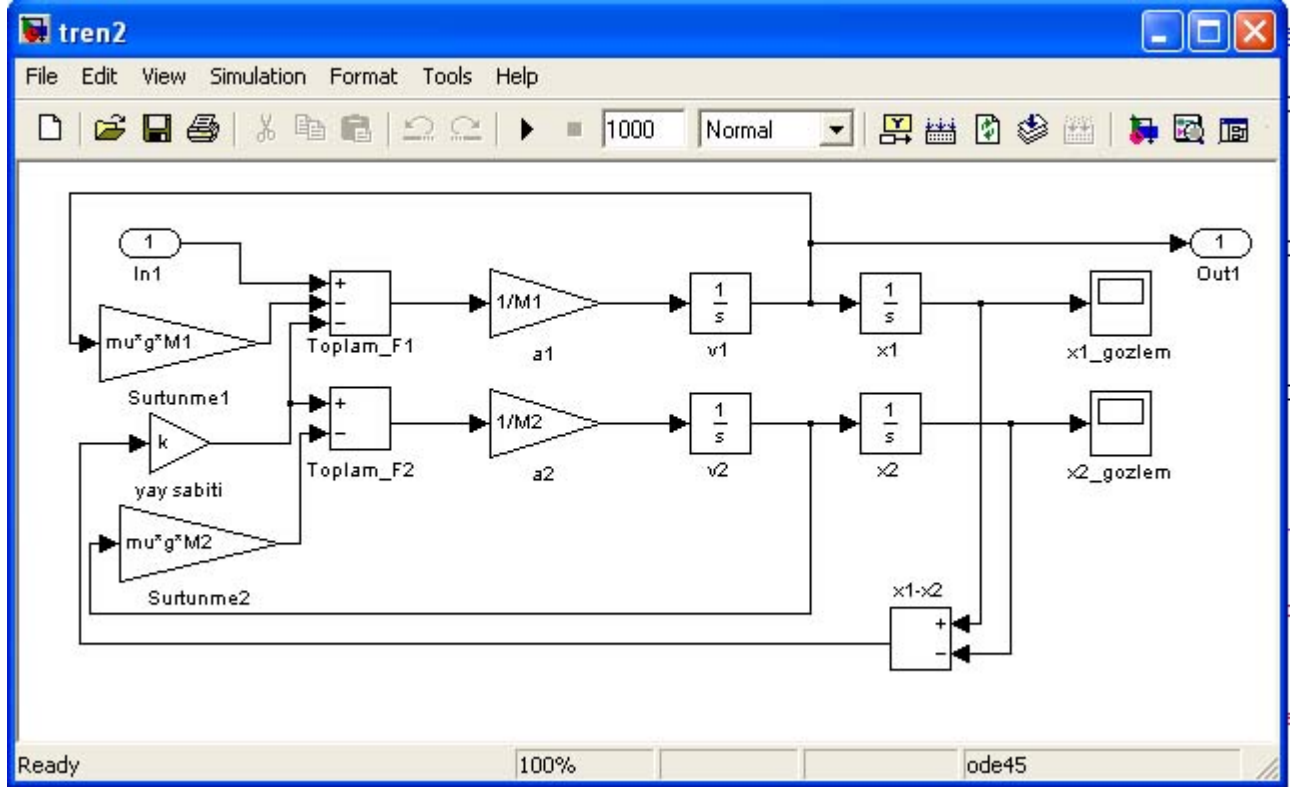


Similasyonu çalıştırın ve v1 çıkışına bağladığımız scope bloğunu açın.Burada trenin hızını göreceğiz.Gördüğümüz kare dalga işareti trenimizin önce ileri sonra da geri gittiğini belirtmektedir.



MATLAB MODELİNİN ELDE EDİLMESİ

Şimdi de modelimizi MATLAB'de kullanılabilecek hale getireceğiz. Başka bir deyişle modelimizi durum-uzayı ve transfer fonksiyonu gösterimine dönüştüreceğiz. Bunun için modelimizdeki giriş ve çıkışları belirtmemiz lazım. Simulink kütüphanesindeki Ports & Subsystems bölümünden In ve Out bloklarını alıp modelimize ekleyelim. Modelimizdeki Fonksiyon Generatörünü(F) ve V1 çıkışına bağlı olan scope bloğunu kaldıralım ve Toplam_F1 bloğunun birinci girişine In bloğunu ,V1 çıkışına da Out bloğunu bağlayalım. Modelimizi farklı bir isimle kaydedelim.



Yeni modelimizi tren2.mdl olarak kaydettiğimizi farzederek MATLAB'e şu komutu veriyoruz:

```
[A,B,C,D]=linmod('tren2')
```

MATLAB'de aşağıdaki değerleri göreceksiniz:

A =

```
-0.0196      0      1.0000     -1.0000
      0     -0.0196     -2.0000      2.0000
      0      1.0000      0          0
      1.0000      0      0          0
```

B =

```
1
0
0
0
```

C =

1 0 0 0

D =

0

Modelimizin transfer fonksiyonunu elde etmek için şu komutu vermeliyiz:

[num,den]=ss2tf(A,B,C,D)

MATLAB bize transfer fonksiyonumuzun pay ve payda polinomu değerlerini verecektir(num=pay,den=payda)

num =

0 1.0000 0.0196 2.0000 0.0000

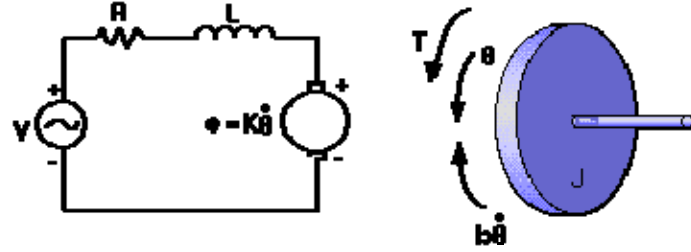
den =

1.0000 0.0392 3.0004 0.0588 0.0000

İşte bu kadar.Bu yazı umarım Simulink öğrenmeye çalışan arkadaşlara yardımcı olmuştur.Başka bir yazıda görüşmek üzere...

DC Motor Hız Kontrolü:

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.

Sisteme ait sabitler şu şekilde verilmiş olsun:

- * Rotor atalet momenti: $(J) = 0.01 \text{ kg.m}^2/\text{s}^2$
- * Mekanik sistem damping oranı: $(b) = 0.1 \text{ Nms}$
- * EMK sabiti $(K=K_e=K_t) = 0.01 \text{ Nm/Amp}$
- * Elektriksel direnç $(R) = 1 \text{ ohm}$
- * Elektriksel endüktans $(L) = 0.5 \text{ H}$
- * Giriş voltajı (V)
- * Çıkış (teta): Şaftın pozisyonu

Motorun ürettiği Tork, T , armatür akımı, i , ve K_t sabiti ile çarpımına eşittir. Ters emk kuvveti, e , ile açısal hız arasında da şu ilişki vardır:

$$T = K_t i$$
$$e = K_e \frac{d\theta}{dt}$$

SI birim sisteminde, K_t (Armatür sabiti) ile K_e motor sabiti eşittir.

DENKLEMLERİN ELDE EDİLMESİ:

Sistem, rotorun atalet momentine etkiyen bütün torkların toplanması ve ivmenin integrali ile hızın elde edilmesiyle modellenenecektir. Ayrıca armatür akımına Kirchhoff yasaları uygulanacaktır. Newton ve Kirchhoff kanunlarını sisteme uyguladığımızda aşağıdaki denklemleri elde ederiz:

$$J \frac{d^2\theta}{dt^2} = T - b \frac{d\theta}{dt} \implies \frac{d^2\theta}{dt^2} = \frac{1}{J} (K_t i - b \frac{d\theta}{dt})$$

$$L \frac{di}{dt} = -Ri + V - e \implies \frac{di}{dt} = \frac{1}{L} (-Ri + V - K_e \frac{d\theta}{dt})$$

İvmenin integralini alarak hızı, akımın türevinin integralini alarak da akımı elde ederiz:

$$\int \frac{d^2\theta}{dt^2} = \frac{d\theta}{dt}$$

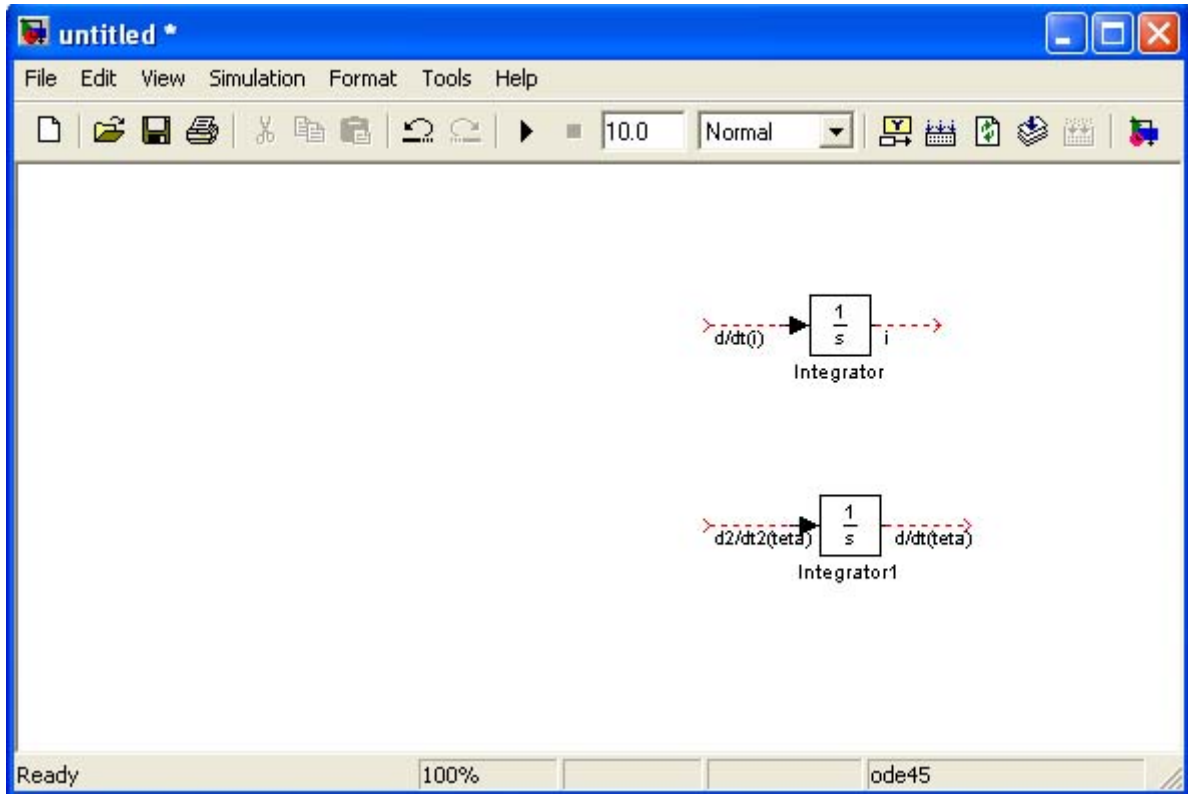
$$\int \frac{di}{dt} = i$$

Artık sisteme ait denklemlerimiz elde ettiğimize göre modelimizi Simulink'te oluşturmaya geçebiliriz.

SİSTEMİN MODELLENMESİ:

Sistemimizde akımın türevinin integralini alarak akımı, dönme açısının ikinci türevinin integralini alarak, açının birinci türevini yani açısal hızı elde edeceğiz.

- Öncelik boş bir Simulink sayfası açın.
- Ardından iki adet integral bloğunu (continuous kütüphanesinden) yerleştirin.
- Integral bloklarının giriş ve çıkışlarını şekildeki gibi isimlendirin. Bunun için integrallerin giriş ve çıkışlarını uzatın, ardından da sinyallerin hemen üzerlerindeki boşluğa tıklayın.



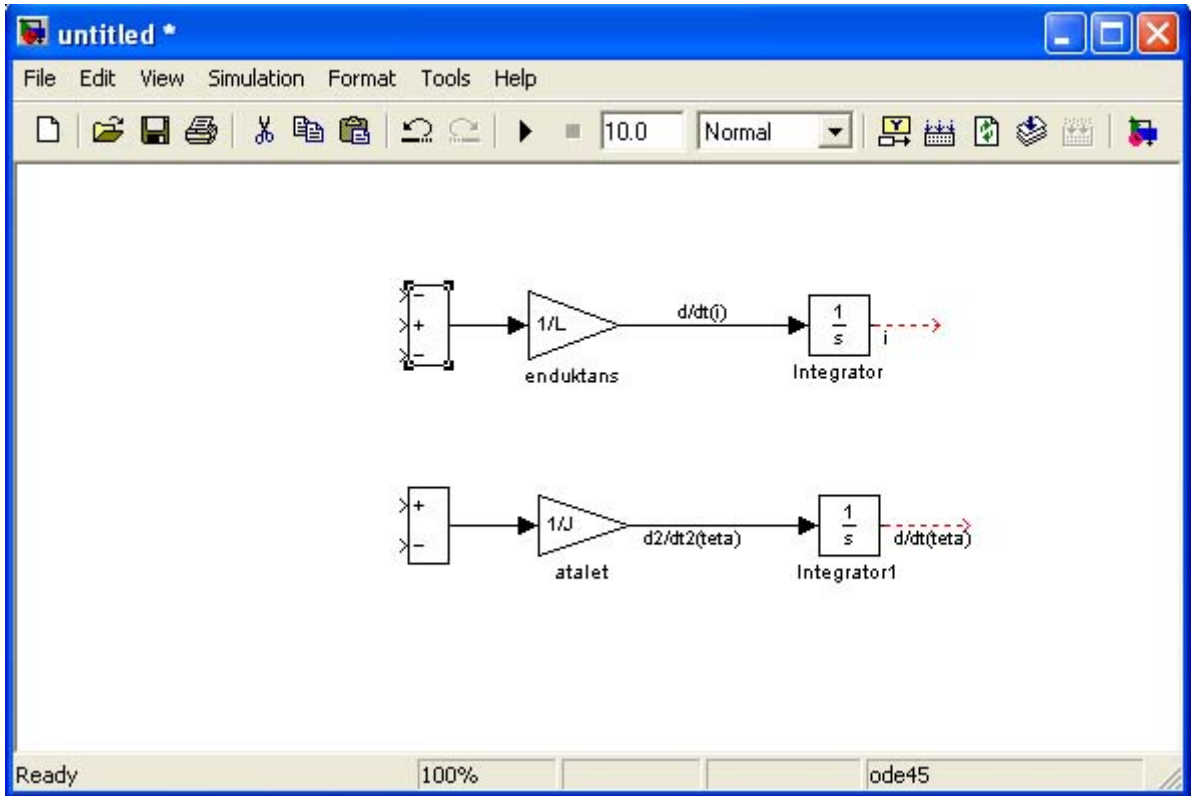
Şimdi denklemlerimizi modelleyeceğiz:

$$J \frac{d^2\theta}{dt^2} = T - b \frac{d\theta}{dt} \implies \frac{d^2\theta}{dt^2} = \frac{1}{J} (K_t i - b \frac{d\theta}{dt})$$

$$L \frac{di}{dt} = -Ri + V - e \implies \frac{di}{dt} = \frac{1}{L} (-Ri + V - K_e \frac{d\theta}{dt})$$

Açısal ivme biri pozitif biri negatif olmak üzere iki terimin toplamının $1/J$ ile çarpımına eşit. Benzer şekilde akımın türevi de bir pozitif, 2 negatif terimin toplamının $1/L$ ile çarpımına eşit.

- İki adet Gain bloğu ekleyerek (linear blok kütüphanesinden) integratorlerin girişlerine bağlayın.
- Gain bloklarına çift tıklayarak açılan pencerede value bölümlerine sırasıyla $1/J$ ve $1/L$ değerlerini yazın
- Bloklara enduktans ve atalet isimlerini verin.
- İki adet Sum (toplama) bloğu ekleyin. Çıkışlarını da sırasıyla Gain bloklarının girişlerine bağlayın. Şekillerini de daha iyi görünmeleri için "rectangular" biçimine getirin.
- Toplama bloklarının diyalog kutularını açarak işaretlerini sırasıyla "+-" ve "-+-" şekline getirin.

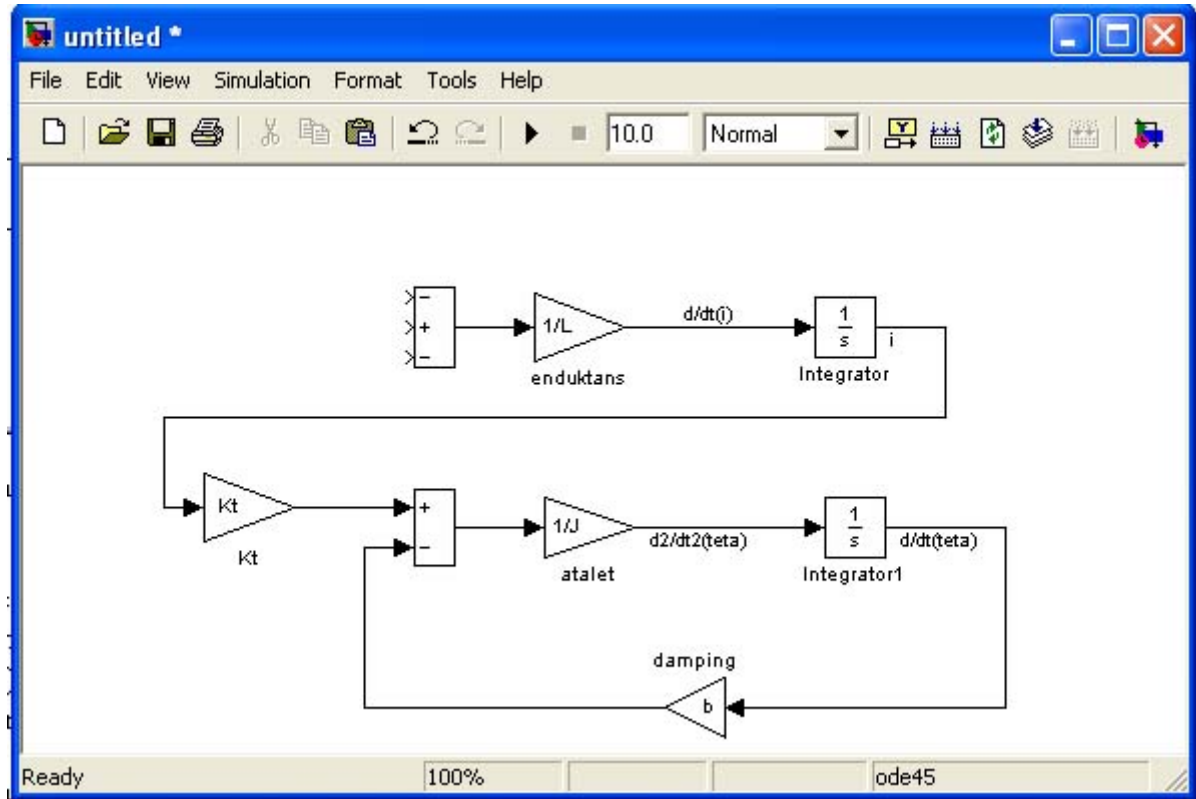


Şimdi Newton denklemiyle ifade edilen tork toplamını elde edeceğiz.İlk olarak tork kaybını elde edeceğiz.

- Atalet bloğunun altına bir adet Gain blok ekleyin.Ardından bloğu seçili hale getirin ve Format menüsünden "Flip" seçeneğini seçerek bloğun yönünü tersine çevirin.
- Gain değerini "b" olarak girin ve bloğu "damping" olarak adlandırın.
- Alttaki integral bloğunun çıkışını alarak damping bloğunun girişine bağlayın
- Damping bloğunun çıkışını alttaki toplam bloğunun negatif girişine bağlayın.

Ardından armatürden gelen torku ekliyoruz:

- Bir gain bloğu ekleyin ve bu bloğu alttaki integral bloğunun pozitif girişine bağlayın.
- Değer olarak "k" girin ve bu sabiti K_t olarak adlandırın.
- Üstteki integratorün çıkışını K_t bloğuna bağlayın.

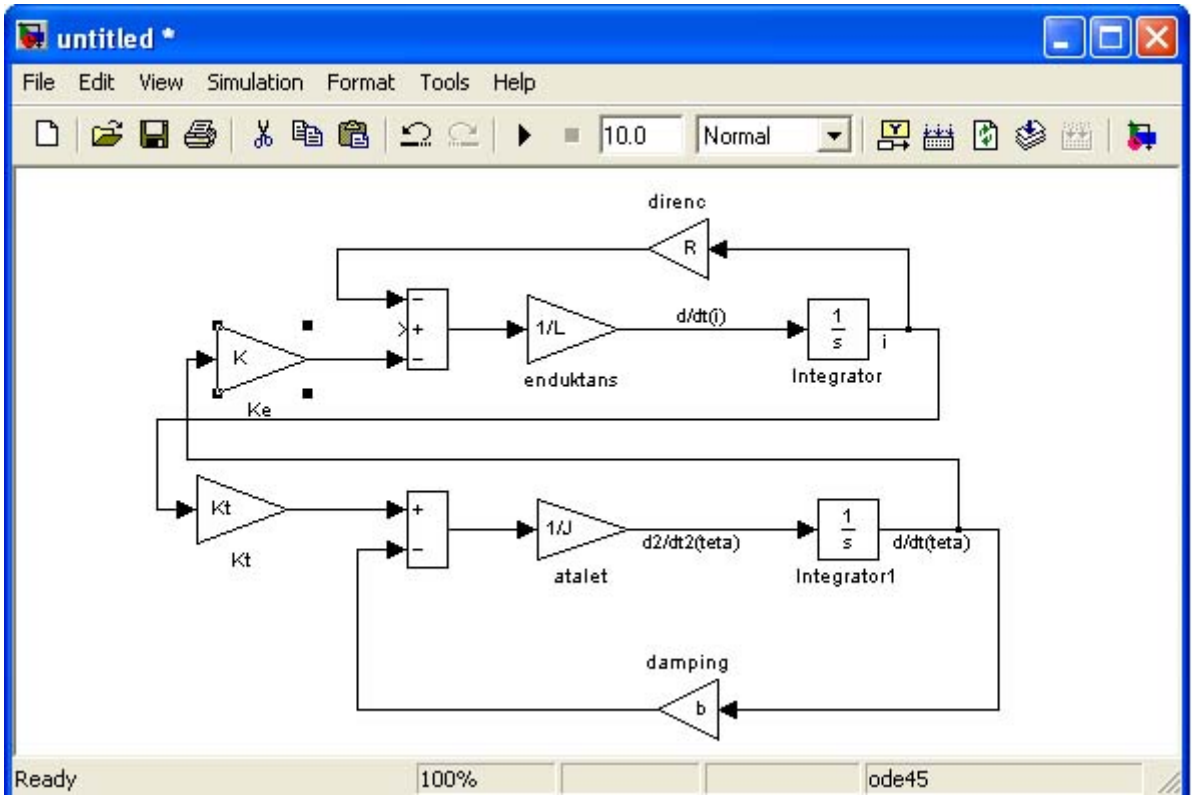


Şimdi de Kirchoff yasası yardımıyla yazdığımız voltja bileşenlerini oluşturacağız.İlk olarak sargı direncindeki voltaj düşümünü ekleyeceğiz.

- Endüktans bloğunun üstüne bir gain bloğu ekleyin ve yönünü ters çevirin.
- Kazanç(gain) değerini "R" olarak verin ve bu bloğu "direnc" adlandırın.
- Akım (i) çıkışından bir dal alarak (Ctrl'ye basılı tutarak çizin) direnc bloğunun girişine bağlayın.
- Direnc bloğunun çıkışını üstteki toplama bloğunun en üstündeki negatif girişe bağlayın.

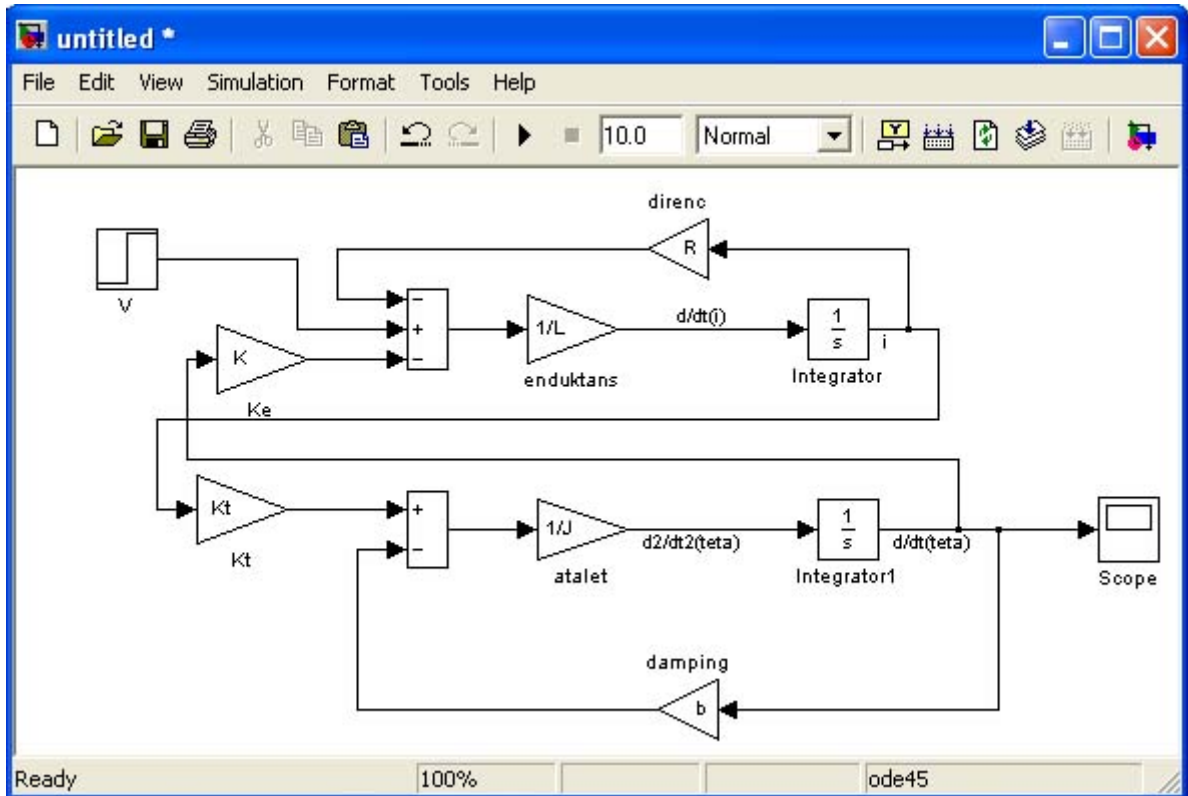
Ters emk 'yı ekleyelim:

- Üstteki toplama bloğunun negatif girişine bağlanmış bir gain bloğu ekleyin.Değerini K olarak belirleyin ve "Ke" ismini verin.
- Açısal hız (d/d(teta) çıkışından bir dal alarak Ke bloğunun girişine bağlayın.



Kirchoff denklemindeki 3. bileşen de kontrol girişi olan V. Bunun için step (adım) girişi uygulayacağız.

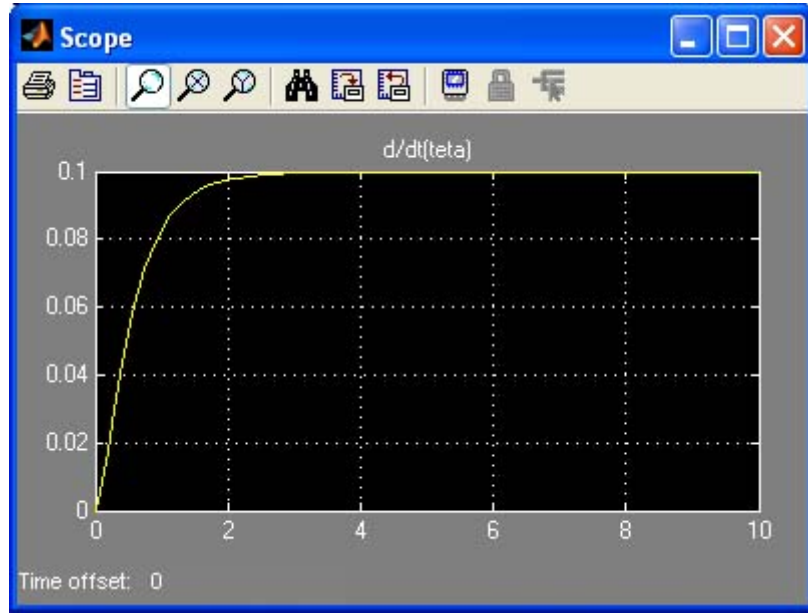
- "Sources " blok kütüphanesinden bir adet Step bloğunu modele ekleyin ve üstteki toplam bloğunun pozitif girişine bağlayın.
- Sistemin çıkışı olan açısal hızı gözlemleyebilmek için "Sinks" blok kütüphanesinden Scope bloğunu modele ekleyin ve d/dt(teta) çıkışına bağlayın.
- Step bloğuna çift tıklayın ve diyalog kutusundaki Step Time değerini "0" yapın.



Modelimizi elde ettik.Şimdi modele "dcmotor" adını vererek kaydedebilirsiniz.Sonra da MATLAB komut satırına aşağıdaki komutları girin:

```
J=0.01;  
b=0.1;  
Kt=0.01;  
K=0.01;  
R=1;  
L=0.5;
```

Ardından Ctrl-T kombinasyonu ile veya Simulation menüsünden start butonuna basarak similasyonu başlatın.Scope bloğuna çift tıklayarak açın ve sağ tıklayarak "auto scale" seçeneğini seçin.Sistemin açık çevrim cevabı aşağıdaki gibi olacaktır:



Buraya kadar bir DC motoru nasıl modelleyebileceğimiz gördük. Bundan sonra motorumuzu kontrol projelerinde kullanabilir, onu kontrol edebiliriz. Bunları da diğer yazılarda göreceğiz.

SİMULINK - SIMPOWER TOOLBOX İLE SÜRÜCÜ SİSTEMLERİN MODELLENMESİ

SimPowerSystems araçları Simulink ile birlikte bulunmaktadır. Simpower toolbox güç sistemlerinin kolaylıkla modellenmesine olanak vermektedir. Simulink'in içerdiği değişik bloklar sayesinde güç sistemlerinin kendi içindeki ilişkileri ve bunun yanında termal, kontrol gibi diğer disiplinlerle etkileşimleri incelenebilmektedir.

Sürücü Sistemler ve Modellenmesi:

Elektrik Sürücüler, elektrik enerjisinin mekanik enerjiye, değişken hızlarda dönüşmesini sağlayan sistemlerdir. Bu nedenler bu sistemlere ayarlı hız sürücüleri (adjustable speed drive (ASD)) de denilmektedir. Bu sürücüler hız kontrolünün yanında akım (tork) düzenlemesi de yaparak motorun güvenliğini de sağlamaktadır. Dolayısıyla sürücünün hız ve torku herhangi bir mekanik yükün sürekli zaman karakteristiğini karşılayabilmektedir. Böylelikle daha yüksek enerji verimi ve enerji tasarrufu ile birlikte düşük enerji maliyetleri elde edilebilmektedir. Bunlara ek olarak, hızlanma ve yavaşlamaların geçici hal süreleri boyunca elektrik sürücü hızlı dinamik cevap ve yumuşak kalkış/duruş gibi özellikler sağlar.

Gittikçe artan sayıda çeşitli uygulamalar değişken hız ve tork ihtiyaçlarına gereksinim duymaktadır. Elektrikli taşıma sistemler, asansörler, bilgisayar disk sürücüleri, makine elemanları hareketin zamana göre değişiminin hassas bir şekilde kontrol edilmesini gerektiren yüksek performans uygulamalarının başta gelen örnekleridir. Pompalar, fanlar, konveyörler ve HVAC olarak bilinen ısıtma, havalandırma, klima sistemleri de orta performanslı uygulamalardır. Bu uygulamalarda da sürücülerin kullanımı büyük enerji tasarrufları anlamına gelmektedir.

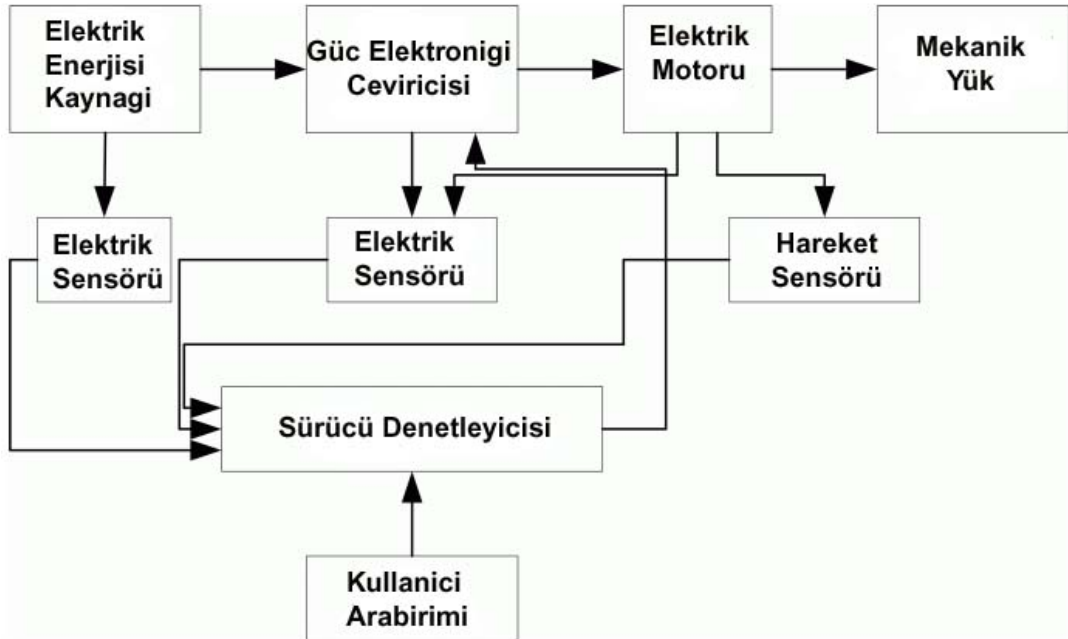
Bu yazıda Simulink'te bulunan SimPowerSytems araçlarından Sürücüler kütüphanesi tanıtılacaktır. Bu kütüphanede DC1 ve DC7 arasında 7 adet DC motor sürücüsü modeli bulunmaktadır. Her bir model yine Simpower Kütüphanesinde bulunan Fırçalı DC motor modelini kullanmaktadır. Bütün elektrik motorlarında olduğu gibi fırçalı DC motor da stator (sabit) ve rotor (hareketli) olmak üzere iki kısımdan oluşmaktadır. DC motorun iki tür sargısı bulunmaktadır: manyetik uyarmayı sağlayan alan sargısı, endüklenen motor akımını taşıyan armatür sargısı. Armatür devresinin zaman sabiti (L/R), alan sargısı devresinden çok daha ufak olduğundan, hızın armatür gerilimini değiştirerek kontrol edilmesi çok daha hızlı bir biçimde gerçekleştirilebilir. Dolayısıyla uyarma sargıları sabit bir DC kaynakla beslenirken, armatür sargıları değişken bir kaynakla

beslenmektedir. Değişken kaynak olarak DC1'den DC4'e kadar olan modeller de faz kontrollü tristörlü çevirici, DC5,DC6,DC7 modellerinde ise transistorlü kıyıcı kullanılmaktadır. Tristörlü çevirici DC1 ve DC2 modellerinde tek fazla AC kaynaktan,DC3 ve DC4 modellerinde ise üç fazlı AC kaynaktan beslenmektedir. Bütün modeller değişik çalışma bölgeleri kombinasyonlarına sahiptirler. Bu kombinasyonlar aşağıdaki tabloda verilmiştir.

Model	Çevirici tipi	Çalışma Bölgeleri
DC1	Tek faz tristörlü çevirici	I-II
DC2	Tek faz tristörlü çevirici	I-II-III-IV
DC3	Üç fazlı tristörlü çevirici	I-II
DC4	Üç fazlı tristörlü çevirici	I-II-III-IV
DC5	Kıyıcı	I
DC6	Kıyıcı	I-II
DC7	Kıyıcı	I-II-III-IV

Elektrik Sürücülerin Bileşenleri:

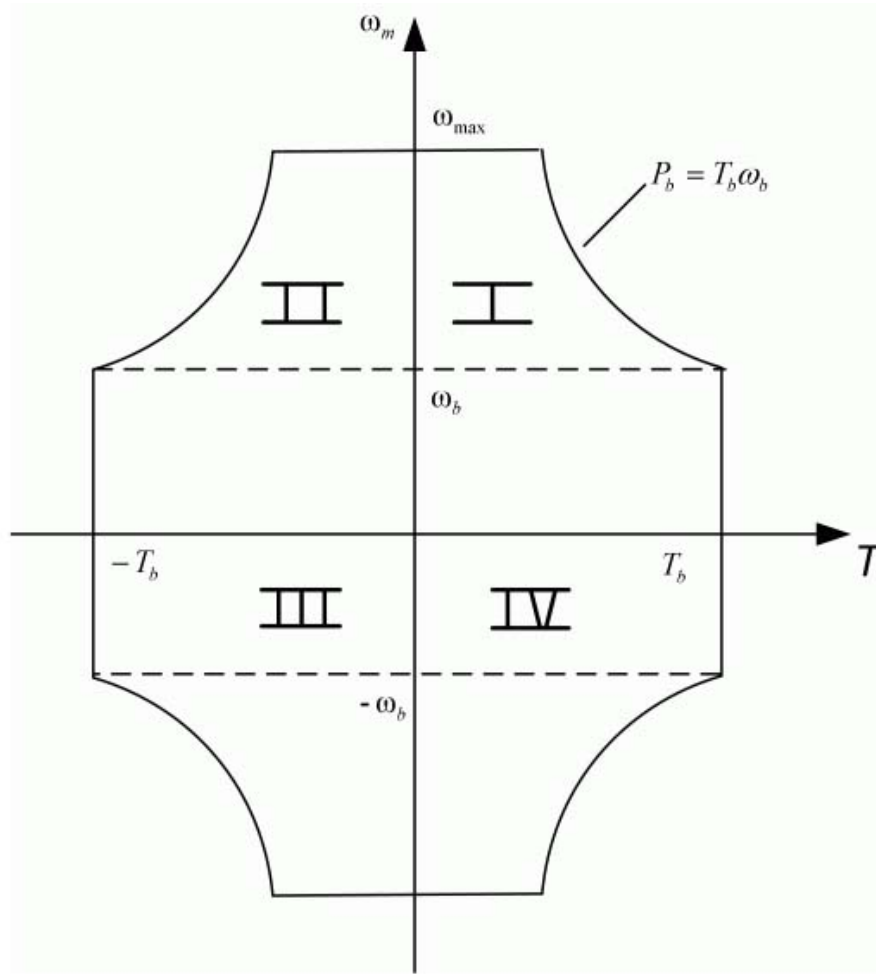
Elektrik sürücülü sistemlerde 3 temel bileşen bulunmaktadır: Elektrik motoru, güç elektroniği çeviricisi ve sürücü denetleyicisi. Ayrıca bu üç bileşen dışındaki birimler de aşağıda gösterilmiştir.



Çok Bölge Çalışma

Her bir sürücü uygulamasında, mekanik yükün sürülmesinde değişik gereksinimler bulunmaktadır. Tork/hız karakteristiklerinin bütün olasılıkları 4 bölgele tork/hız grafiğinde belirtilebilir. Birinci bölgede elektrik tork ve hızın her ikisi de pozitifdir. Bu elektriksel torkun hareketle aynı yönde olduğundan ileri yönde motor çalışmayı belirtir. İkinci bölgede elektriksel torkun işareti negatif ve hızı ise pozitifdir. Bu, elektriksel torkun hareket yönüne ters olduğunu, dolayısıyla ileri yönde frenleme yapıldığını gösterir. Üçüncü bölgede elektriksel tork ve hızın her ikisinin de işareti negatiftir. Ters yönde motor çalışma olmaktadır. Dördüncü ve son bölgede ise elektriksel torkun işareti pozitif, torkunki ise negatiftir. Ters yönde motor çalışma olmaktadır. Sürücünün frenlemesi frenleme kısıcısı (dinamik frenleme) veya iki yönlü güç akışı (rejeneratif frenleme) ile sağlanır.

Aşağıdaki şekilde dört çalışma bölgesi gösterilmektedir. Her bir bölge 0-nominal hız arasında sabit bir tork bölgesine sahiptir. Sıfır ile maksimum hız arasında da azalan tork bölgeleri bulunmaktadır. Bu bölgelere sabit güç bölgesi de denilmektedir.



Çok Bölgele Çalışma

Rejeneratif Frenleme

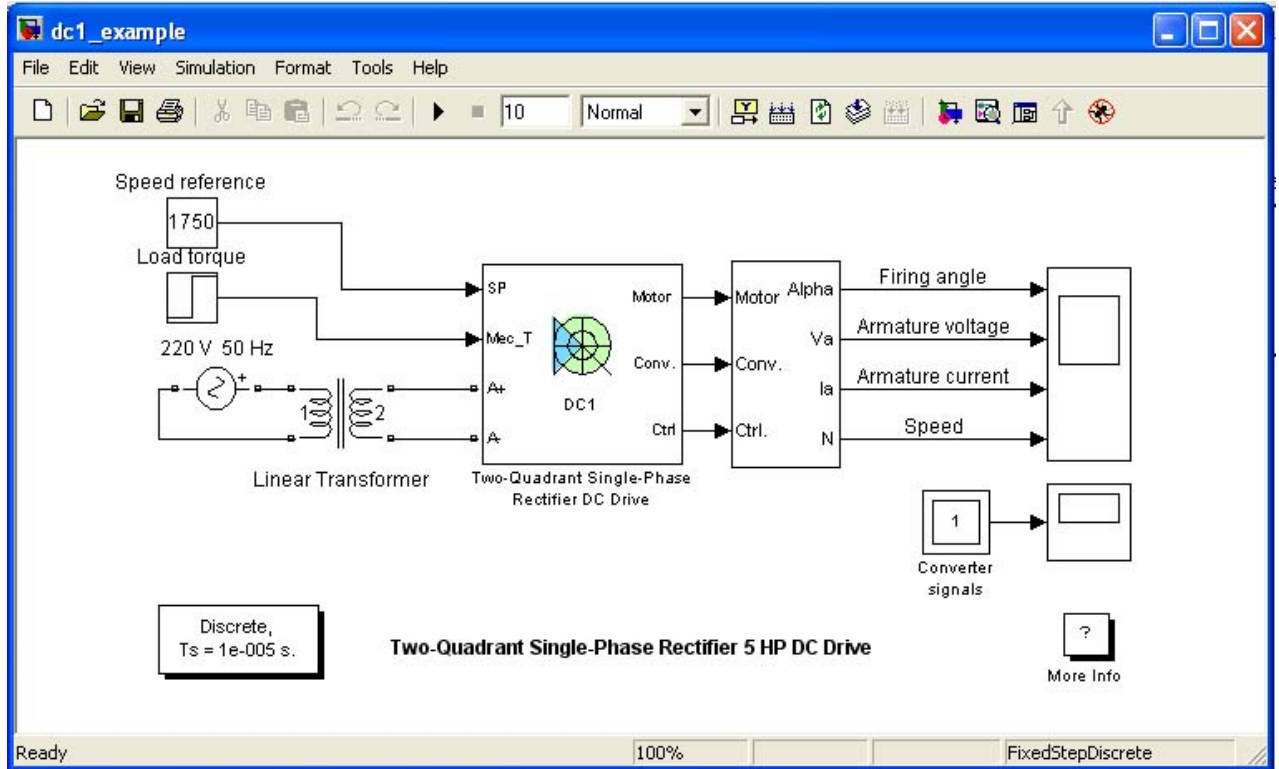
II ve IV nolu çalışma bölgeleri ileri ve ters yönde frenleme bölgeleridir. Rejeneratif frenleme, motor yükünün kinetik enerjisinin elektrik enerjisine çevrilip güç kaynağına geri verilmesi demektir. Bu iki yönlü enerji akışı motorun bağlantılarının ters çevrilmesi veya ikinci bir çevirisi kullanılması ile sağlanabilir.

SİMÜLASYON

Simulink'te SimPowerSystems kütüphanesi içinde bulunan sürücü modelleri ile ilgili birçok demo bulunmaktadır. Bu demolara MATLAB komut satırında "demos" yazılarak ulaşılabilir.

Yukarıda anlatılan sürücülere örnek olması açısından DC1 sürücü modeli ile oluşturulmuş bir model incelenecektir.

Devre SimPowerSystem kütüphanesindeki DC1 bloğu üzerine kurulmuştur. 2-bölgeli tek fazlı doğrultucu sürücünün 5 HP DC motoru sürmesi modellenmiştir. Ayrıca rejeneratif frenlemeyi simule edebilmek için bir frenleme ünitesi eklenmiştir. (4.bölge çalışması)



İki bölgeli çalışan DC motor sürücü sistemin modellenmesi

2-Quadrant Single-Phase Rectifier DC Motor Drive

The DC motor parameters are specified in the DC Machine tab. The converter parameters, smoothing inductance and field voltage values are specified in the Converter tab. The bridge firing unit, speed and current regulator parameters are specified in the Controller tab.

DC Machine | Converter | Controller

Electrical parameters

Mutual inductance (H): 1.234

Armature

Resistance (ohm): 0.78
Inductance (H): 0.016

Field

Resistance (ohm): 150
Inductance (H): 112.5

Mechanical parameters

Inertia (kg*m²): 0.25
Viscous friction coefficient (N-m-s): 0.01
Coulomb friction torque (N-m): 0
Initial speed (rad/s): 0

Parameters file options

Load Save

OK Cancel Help Apply

5 HP DC motorun alan sargıları 150 V sabit bir DC kaynakla beslenmektedir. Armatür gerilimi iki adet PI düzenleyici ile kontrol edilen tek-fazlı doğrultucu ile sağlanmaktadır. Doğrultucu, 220 V AC – 50 Hz bir kaynaktan beslenerek gerilimi yeterli seviyeye yükselten bir lineer transformatörle beslenmektedir.

DC1 bloğunun üzerinde tıklandığında sürücü ve motor parametrelerinin ayarlanabileceği bir diyalog kutusu çıkmaktadır. Buradan motorun karakteristik değerleri, kontrol sisteminin PI, hızlanma yavaşlama eğrileri gibi değerler girilebilmektedir.

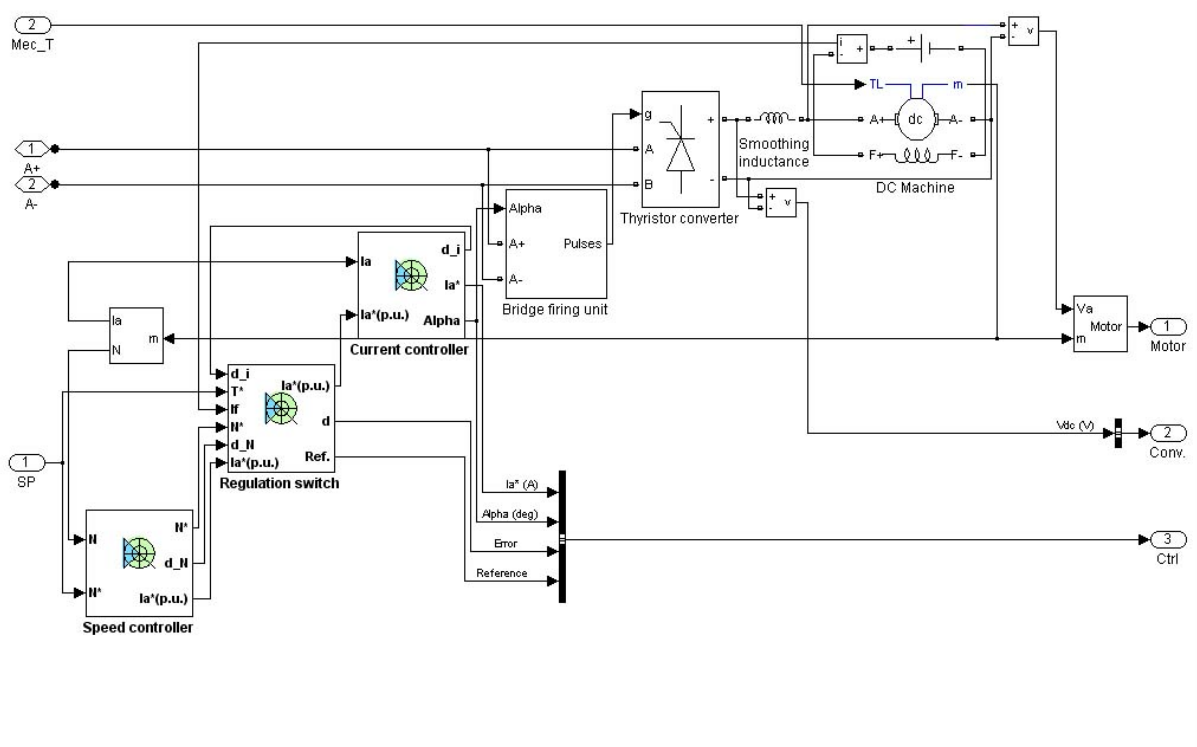
PI düzenleyiciler doğrultucu tristörlerini kontrol etmektedirler. İlk düzenleyici hızı kontrol etmektedir. Hız düzenleyicisi referans armatür akımını çıkış olarak vermektedir (p.u olarak). Bu referans değeri akım düzenleyici tarafından istenilen hızı sağlayacak torkun üretilmesi için kullanılır. Hız referansının değişim hızı, hızlanma ve yavaşlama rampalarını

takip eder.Böylelikle ani referans değişimlerinde armatür akımının aniden artarak sistemi kararsız duruma sokması önlenir. Akım düzenleyicisi uygun tristör açısını hesaplayarak armatür akımını kontrol eder.Doğrultucudan istenilen armatür akımını kontrol sağlayacak gerilim üretilir.

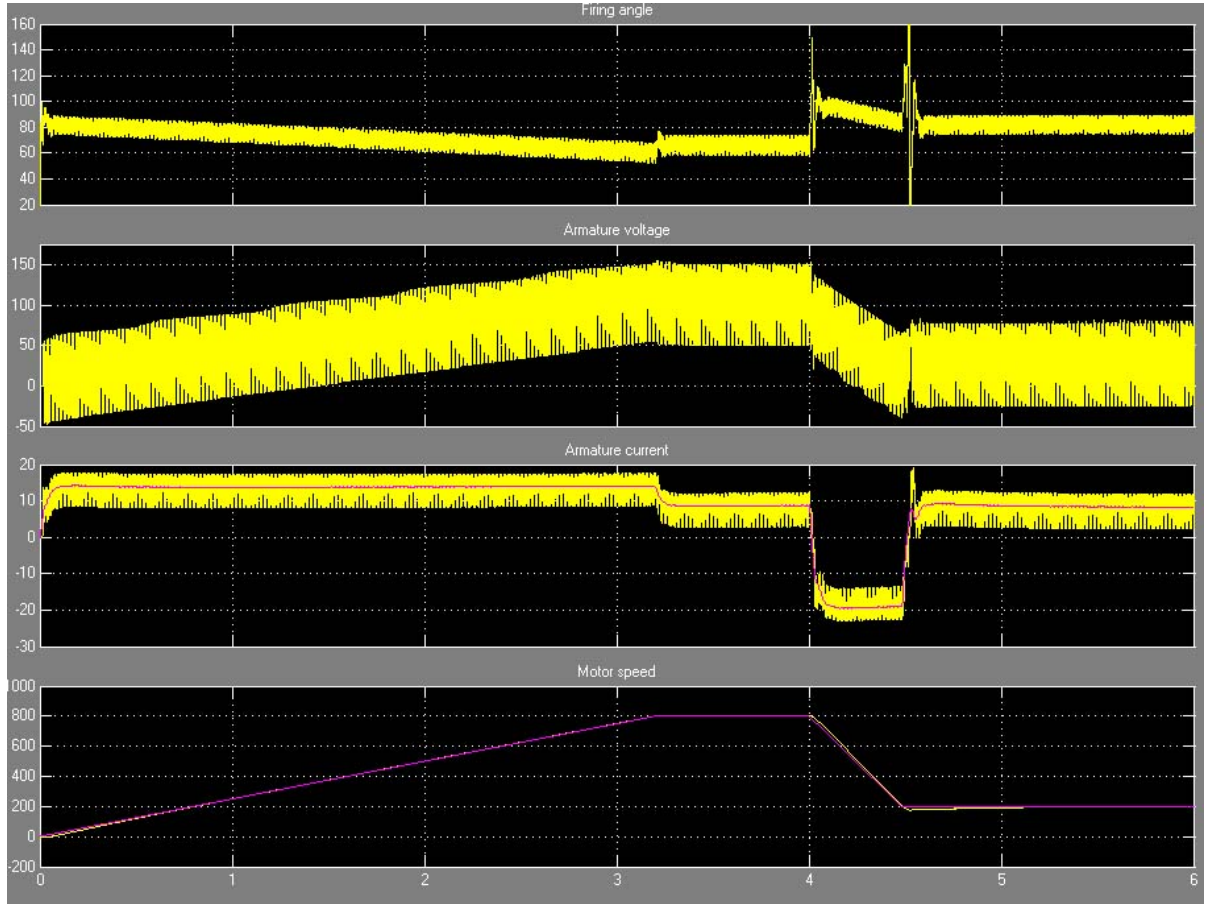
Frenleme ünitesi ,iki duruma sahip (çalışma modu ve frenleme modu) sonlu durum makinesi (FSM) ile kontrol edilir.Sistem frenleme moduna girdiğinde ,armatür anahtarları aktifleştirilir ve armatür akımının terst yönde akması sağlanır.Bu ters yönde elektromanyetik frenleme torku ile hızlı bir yavaşlama sağlar.Ters akımın akması,armatür akımının 0 A değerine gelmesi ile başlatılır.Bu komutasyon esnasında anahtarlarda oluşabilecek tehlikeli arkları önler.

Armatür akımındaki salınımları azaltmak için seri olarak 150 mH 'lik bir filtreleme endüktansı kullanılmıştır.

Simülasyonun Çalışma Sistemi



Similasyon başlatıldığın da motor gerilim,,akım,hız ve doğrultucu ateşleme açıları scope'ta gözlenebilir.Ayrıca hız ve akım referansları da gösterilmiştir.



Başlangıçtaki hız referansı 800 rpm'dir Yük torku ise 10 N.m.Grafikte görülebileceği gibi motor hızı, hızlanma referans eğrisini takip etmektedir ve 3.5 saniye sonra referans değerine oturmaktadır.Armatür akımı da akım referansını takip etmekte ve nominal akımın altında kalmaktadır.Bu etapta ,ortalama ateşleme açısı 90 derecenin altındadır ve doğrultucu doğrultma modunda çalışmaktadır(1.bölge)

$t = 4$ s'de , hız referansı 200 rpm'e düşmekte ve sistem frenleme moduna geçmektedir.Armatür anahtarları akım 0 A değerine ulaştığında aktifleştirilmesi ve armatür akımının ters yönde akması sağlanmaktadır.Motorun yavaşlamasının da ,yavaşlama rampasını takip ettiğine dikkat edilmelidir(-1250 rpm/s).Frenleme esnasında doğrultmaç evirici modunda çalışmaktadır(ikinci bölge)

$t = 4.5$ s'de , motorun hızı referansın biraz altındadır ve armatür akımı tekrar motora hızlanma yönünde akmaktadır.Tristör köprüsü yeniden doğrultucu modda çalışmaktadır ve motor $t=5.5$ s'de 200 rpm değerine gelmektedir.

Komutasyon süresince akımda bir yükselme görülmektedir. Bu armatür gerilimindeki ani değişim sonucu oluşmaktadır. Köprünün çıkış gerilimi , gerilim değilimini aniden takip edememektedir ve bu ani gerilim değişimi akımda bir yükselmeye neden olmaktadır. Ancak bu yükselme çok fazla değildir ve zararlı olmamaktadır.