

Model Öngörölü Denetimin Bir Sistemde DSP ile Gerçeklenmesi

Serdar Ergen

**YÜKSEK LİSANS TEZİ**

Elektrik Elektronik Mühendisliđi Anabilim Dalı

Ađustos 2009

An Implementation of Model Predictive Control with DSP

Serdar Ergen

**MASTER OF SCIENCE THESIS**

Department of Electrical & Electronics Engineering

August 2009

Model Öngörölü Denetimin Bir Sistemde DSP ile Gerçeklenmesi

Serdar Ergen

Eskişehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliği Uyarınca  
Elektrik Elektronik Mühendisliği Anabilim Dalı  
Kontrol ve Kumanda Bilim Dalında  
YÜKSEK LİSANS TEZİ  
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Abdurrahman Karamancıođlu

Ađustos 2009

## ONAY

Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Serdar Ergen'in YÜKSEK LİSANS tezi olarak hazırladığı “**Model Öngörülü Denetimin Bir Sistemde DSP ile Gerçeklenmesi**” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

**Danışman** : Prof. Dr. Abdurrahman Karamancıoğlu

**İkinci Danışman** : -

**Yüksek Lisans Tez Savunma Jürisi:**

**İmza**

Üye : Prof. Dr. Abdurrahman Karamancıoğlu .....

Üye : Prof.Dr. Hasan Hüseyin Erkaya .....

Üye : Doç.Dr. Osman Parlaktuna .....

Üye : Y. Doç. Dr. Bünyamin Tamyürek .....

Üye : Y. Doç. Dr. Ahmet Yazıcı .....

<p>Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.</p> <p style="text-align: right;">Prof. Dr. Nimetullah BURNAK Enstitü Müdürü</p>
---

## ÖZET

Bu tez çalışmasında Model Öngörülü Kontrol uygulaması Sayısal İşaret İşleyici Kart (DSP) kullanılarak gerçekleştirilmiştir.

Öncelikle takometreli bir DC motordan oluşan tek giriş ve tek çıkışlı süreç matematiksel olarak modellenmiş ve bu süreç için bir Model Öngörülü Denetleyici MATLAB, Simulink ve Model Predictive Control Toolbox kullanılarak oluşturulmuştur. Oluşturulan Simulink modeli Real Time Workshop (RTW) ile C koduna dönüştürülerek TI CCS (Code Composer Studio) Sayısal İşaret İşleyici Kart yazılım geliştirme ortamına aktarılıp diğer destek dosyalarıyla birlikte derlenmiş ve bilgisayarın paralel portuna bağlı Sayısal İşaret İşleyici Kart'a yüklenmiştir. Yüklenen program çalıştırılarak Model Öngörülü Kontrol uygulaması DSP üzerinde başarılı olarak gerçekleştirilmiştir.

**Anahtar Kelimeler:** Model Öngörülü Kontrol, Sayısal İşaret İşleyici Kart (DSP), MATLAB, Simulink, Real Time Workshop, Code Composer Studio.

## SUMMARY

In this study, an implementation of model predictive control with Digital Signal Processor was studied.

A model predictive controller is implemented by using Model Predictive Control Toolbox of MATLAB and relevant design commands are converted to C code with Real-Time Workshop using softwares MATLAB and Simulink. Compilation of the code with the support files was done with Code Composer Studio, and transferred to Digital Signal Processor from the parallel port, respectively.

The controller above is used in velocity control of a DC motor for validation of its functioning. It is shown that implementation of model predictive control with Digital Signal Processor functions successfully.

**Keywords:** Model Predictive Control, Digital Signal Processor, MATLAB, Simulink, Model Predictive Control Toolbox, Code Composer Studio.

## TEŐEKKÜR

Eskiőehir Osmangazi Üniversitesi'nde lisans programına baŐladığım günden itibaren gerek derslerimde ve gerekse tez çalışmamda, bana danışmanlık ederek, beni yönlendiren ve her türlü olanağı sağlayan danışmanım Prof. Dr. Abdurrahman Karamancıođlu'na, bu uzun süreçte desteklerini benden hiçbir zaman esirgemeyen aileme ve yakınlarıma sonsuz saygı ve teşekkürlerimi sunarım.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	<b>v</b>
<b>SUMMARY</b> .....	<b>vi</b>
<b>TEŞEKKÜR</b> .....	<b>vii</b>
<b>ŞEKİLLER DİZİNİ</b> .....	<b>xi</b>
<b>ÇİZELGELER DİZİNİ</b> .....	<b>xiii</b>
<b>SİMGELER VE KISALTMALAR DİZİNİ</b> .....	<b>xiv</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. MODEL ÖNGÖRÜLÜ KONTROL</b> .....	<b>3</b>
2.1 MPC Stratejisi .....	4
2.2 MPC Yapısı .....	5
2.5 Model Öngörülü Kontrolün Üstünlükleri .....	6
2.6 Modele Öngörülü Kontrolün Eksik Yönleri .....	7
<b>3. KULLANILAN ARAÇLAR</b> .....	<b>8</b>
3.1 Donanımsal Araçlar .....	8
3.1.1 Sayısal İşaretleyici Kart (DSP) .....	8
3.1.2 Digiac 1750 .....	10
3.2 Yazılımsal Araçlar .....	11
3.2.1 Matlab .....	11



## İÇİNDEKİLER (devam)

	<b><u>Sayfa</u></b>
3.2.2 Model Predictive Control Toolbox .....	11
3.2.3 Simulink .....	12
3.2.4 Guide .....	12
3.2.5 Real-Time Workshop .....	13
3.2.6 Embeded Target for TI C2000 DSP .....	14
3.2.7 Sdconfig .....	15
3.2.8 Code Composer Studio (CCS) .....	15
<b>4. MODELLEME.....</b>	<b>16</b>
4.1 Genel Sistem Tanımı .....	16
4.2 Motor Dinamiği .....	16
4.3 Süreç Modelinin Tanımlanması .....	17
4.4 Model Öngörümlü Kontrol Oluşturulması.....	17
<b>5. UYGULAMA .....</b>	<b>24</b>
5.1 Elektriksel Bağlantı .....	24
5.2 DSP Kartın Bağlantı Ayarlarının Yapılması.....	26
5.3 Uygulamanın Genel Yapısı .....	29
5.4 Sistemin Modellenmesi .....	31
5.5 Kullanıcı Ara yüz Tasarımı .....	40

## İÇİNDEKİLER (devam)

	<b><u>Sayfa</u></b>
5.6 Gerçek Zamanlı Uygulama .....	41
5.7 Uygulamanın Çalıştırılması.....	43
<b>6. SONUÇ VE TARTIŞMA .....</b>	<b>46</b>
<b>7. KAYNAKLAR .....</b>	<b>47</b>
<b>EKLER .....</b>	<b>48</b>

## ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 MPC Stratejisi.....	4
2.2 MPC Yapısı.....	6
3.1 TMS320F2812 DSP.....	8
3.2 TMS320F2812 DSP Blok Şeması .....	9
3.3 Digiac 1750.....	10
3.4 Guide.....	13
3.5 Embedded Target for TI C2000 DSP .....	14
4.1 Süreç Modeli Yükleme .....	18
4.2 MPC Tasarım Aracı .....	18
4.3 Sinyal Özelliklerinin Tanımlanması .....	19
4.4 Model Öngörülü Denetleyicinin Özellikler Penceresi.....	20
4.5 Constraints (Kısıtlamalar) Özelliklerini Tanımlanması.....	21
4.6 Süreç Giriş Grafiği.....	22
4.7 Süreç Çıkış Grafiği .....	22
4.8 MPC Dışa Aktarımı .....	23
5.1 Elektriksel Bağlantı Şeması.....	24
5.2 DAC .....	25
5.3 Elektriksel Bağlantı Özellikleri .....	26
5.4 SDconfig .....	26
5.5 SDconfig Bağlantı Ayarları .....	27
5.6 SDconfig Emu Ayarları .....	27

5.7	Sdconfig Data Log Ayarları.....	28
5.8	Sdconfig Bağlantı Test.....	28
5.9	Uygulamanın Genel Yapısı.....	30
5.10	Sistem Modeli .....	31
5.11	Hız Ayar Alt Modeli .....	31
5.12	MPC Blok Parametreleri.....	32
5.13	PWM Blok Timer Özellikleri .....	33
5.14	PWM Blok ADC Kontrol Özellikleri .....	33
5.15	RTDX Veri Gönderim alt modeli .....	34
5.16	Hız Kontrol Alt Modeli.....	35
5.17	ADC Blok Özellikleri .....	36
5.18	ADC Blok Giriş Özellikleri .....	36
5.19	RTDX Veri Gönderim Alt Modeli.....	37
5.20	Hız Ölçüm Alt Modeli .....	37
5.21	Uygulama Modeli .....	38
5.22	F2812 eZdsp Özellikler Penceresi .....	39
5.23	Kullanıcı Arayüzü.....	40
5.24	RTW Hedef Kart Seçimi.....	41
5.25	Solver Özellikleri.....	42
5.26	1000 rpm'den 1200 rpm'e Yükselme Arayüzü .....	43
5.26	1200 rpm'den 1800 rpm'e Yükselme Arayüzü .....	44
5.26	1800 rpm'den 1155 rpm'e Alçalma Arayüzü.....	45

**ÇİZELGELER DİZİNİ****Çizelge****Sayfa**

Tablo 1	Motor Değerleri .....	16
---------	-----------------------	----

**SİMGELER VE KISALTMALAR DİZİNİ**

<b><u>Simgeler</u></b>	<b><u>Açıklama</u></b>
DSP	Sayısal İşaret İşleyici Kart
MPC	Model Öngörülü Kontrol
RTW	Real-Time Workshop
RTDX	Real-Time Data Exchange
CCS	Code Composer Studio
DAC	Dijital Analog Dönüştürücü
ADC	Analog Dijital Dönüştürücü
PWM	Pulse Width Modulation

## BÖLÜM 1

### GİRİŞ

Bu tez çalışmasında model öngörülü denetim (Model Predictive Control - MPC) algoritması Sayısal İşaretleyici (DSP) kartını temel alan donanım kullanılarak doğru akım motor hız kontrolü amaçlı olarak kullanılmaktadır. Bu tezin ana katkısı DSP kart donanımının uygun yazılımlarla birlikte bu amaçla kullanılabileceğini göstermektedir.

Model öngörülü denetim, ele alınan sürecin öngörülü davranışını denetim girişlerini uygun seçerek optimize etme olarak tanımlanabilir (J.B.Rawlings). Öngörü için sürecin modeli kullanılmakta olduğu için model MPC denetleyicinin vazgeçilmez bir bileşenidir. Modellerde oluşabilecek hatalar dolayısıyla geri beslemelerden yararlanılarak öngörü daha güvenli hale getirilmektedir.

Model Öngörülü denetimle ilgili teori lineer modeller için olgunlaşmıştır. Lineer olmayan modeller için literatürde kısıtlı sayıda sonuç bulunmaktadır.

Sisteme ilişkin lineer model durum uzayı formunda olabileceği gibi tek giriş tek çıkışlı sistemler için transfer fonksiyonu formunda da olabilmektedir. Durum uzayı formülasyonu çok giriş çok çıkışlı sistemler için uygun bir arka plan sunmaktadır.

Model öngörülü denetimin çalışma felsefesi aşağıda açıklanmaktadır (S.Kahvecioğlu ve diğ., 2008):

“Model öngörülü denetim, ufku süresince kullanacağı denetim girdileri dizisinin tamamını, denetlenen sistemin kısıtlarını dikkate alan bir optimizasyon kriterine göre hesaplar. Bu şekilde hesaplanan dizinin ilk elemanını sisteme denetim girdisi olarak uyguladıktan sonra yeni oluşan bilgileri dikkate alarak, önündeki ufuk için aynı hesaplamayı tekrar yapar. Bu süreç denetim hedefleri gerçekleşinceye kadar bu şekilde tekrarlanır. Diğer bir ifade ile model öngörülü denetim, tasarımcıların sistem kısıtlarını

dikkate alan optimal denetim girdilerini uzun bir zaman aralığı için hesaplamasına imkan tanıyan, ancak bu hesaplamaların tamamını kullanmak yerine, sadece yeni bilgiler gelinceye kadar gerekli kısmını kullanan, ardından da bu hesaplamayı yeniden yapan bir denetim yaklaşımıdır.”

Bu çalışmada kullanacağımız MPC Toolbox çeşitli sistem model formatlarını kabul etmekte olup biz deneyimizin doğası gereği transfer fonksiyonu modeli kullanmaktayız. Uygun arayüz kullanılarak model tanıtımı ve hedef belirlenmesinin ardından uzman olmayan bir mühendisin dahi bu yöntemle Sayısal İşaret İşleyici Kart kullanarak MPC sistem denetimini sıradan bir mühendislik işi olarak yapabileceğini bu çalışmamızda göstermekteyiz.

Yapılan tez çalışmasının organizasyonu aşağıdaki gibidir;

Bölüm 2’de Model Öngörülü Kontrol yapısı ve kavramı anlatılmıştır. Artıları ve eksileri belirtilmiştir.

Bölüm 3’de ise Model Öngörülü Kontrol uygulamasının Sayısal İşaretleyici Kart kullanılarak gerçekleştirilmesine yardımcı olan donanımsal ve yazılımsal araçlara değinilmiştir.

Bölüm 4’te tek giriş ve tek çıkışlı bir DC motorun hız kontrolünün modellenmesi ve Model Öngörülü Denetleyicinin oluşturulması anlatılmıştır.

Bölüm 5’te üretilen modelin Sayısal İşaretleyici kartı üzerinde gerçekleştirilme basamakları donanımsal ve yazılımsal olarak anlatılmıştır.

Bölüm 6’da ise bu çalışmanın tartışma ve sonuçlarına değinilmiştir.



## BÖLÜM 2

### MODEL ÖNGÖRÜLÜ KONTROL

Model Öngörülü Kontrol 1970 ve 1980 yılları arasında petrol rafinelerinde karşılaşılan çeşitli kontrol problemlerine alternatif çözüm oluşturmak için geliştirilmiştir.

Hem iyi bir başarıml göstermeleri hem de anlaşılmalarnn, formüle edilmelerinin ve ayarlanmalarının kolay olmasından dolayı, gelecekteki sistem çıkışının öngörülmesine dayalı denetim yöntemleri 1970'lerin sonundan itibaren büyük ilgi görmüştür (Demircioglu ve diğ., 1991).

Model Öngörülü Kontrol özel bir kontrol şekli değildir; kontrol metotlarının birçoğundan daha fazla alanda kullanılabilen belirli genel bir fikir etrafında geliştirilmiştir. Bu tasarım yöntemleri uygulamalı olarak aynı yapıya sahip doğrusal denetleyicilere yol gösterir ve yeterli derecede serbestlik sunar. Model Öngörülü Kontrol algoritmaları sadece kullanılan sistem modeli, gürültü modelleri ve minimize edilecek olan ölçütler bakımından farklılık gösterirler (Camacho ve Bordons, 2004).

Model Öngörülü Kontrol çalışma ilkeleri genel olarak şu şekilde özetlenebilir:

- Belirli bir zaman aralığında sistem çıkışı öngörülür.
- Gelecekteki istenen sistem çıkışının bilindiği varsayımıyla, bu çıkışla öngörülen gelecek çıkışı arasındaki farkı minimum yapacak şekilde bir gelecek giriş dizisi seçilir.
- Seçilen giriş dizisinin ilki sisteme uygulanır ve bu adımlar bir sonraki örnekleme anında tekrarlanır (Camacho ve Bordons, 2004).

İlk olarak güç reaktörlerindeki ve petrol rafinerilerindeki kontrol gereksinimlerine bağlı olarak özel geliştirilmiş olan bu yöntemler, günümüzde kimya,



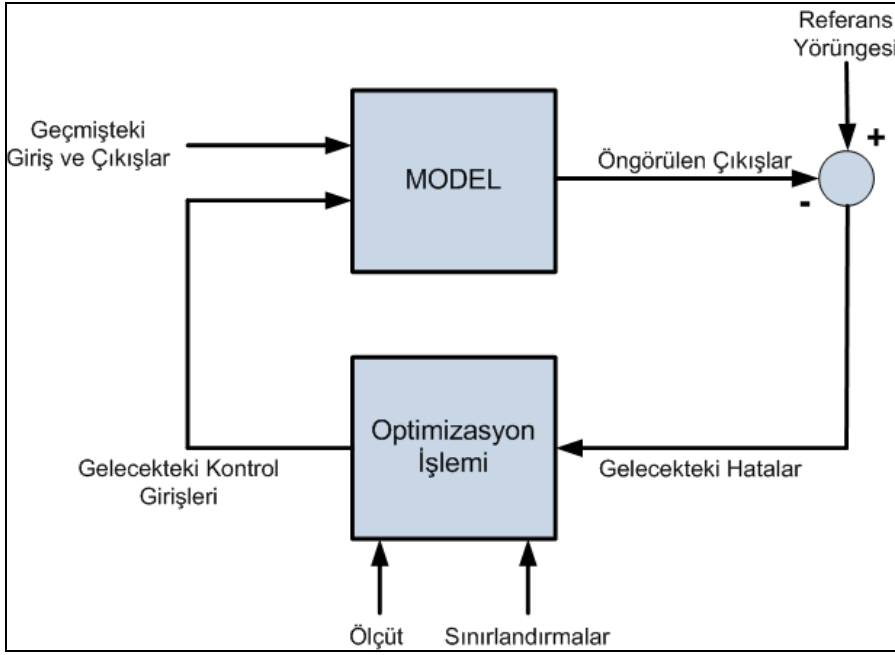
- Gelecek kontrol sinyali, süreci referans yörüngeye  $w(t+k)$  mümkün olduğunca yakın tutmak için uygun bir ölçütün optimizasyonu ile hesaplanır. Bu ölçüt genellikle öngörülü referans yörünge ve öngörülü çıkış sinyali arasındaki hataların karesel bir fonksiyonu şeklindedir. Kontrolün amacı pek çok durumda nesnel fonksiyonda yer alır. Model doğrusalsa, kısıtlama yoksa ve ölçüt karesel hatadan oluşursa, kesin bir çözüm elde edilir.
- Kontrol sinyali  $u(t|t)$ , hesaplanan gelecek kontrol sinyalleri alınmadığı halde, sürece gönderilir; çünkü gelecek örnekleme anında  $y(t+1)$  zaten bilinmektedir ve ilk adım bu yeni değerle tekrarlanır ve tüm diziler yenilenerek alınır. Böylece  $u(t+1|t+1)$  kayan ufuk stratejisi kullanılarak hesaplanabilir (Camacho ve Bordons, 2004).

## 2.2 MPC Yapısı

Model Öngörülü Kontrolün temel yapısı Şekil 2.2'de verilmiştir (Camacho ve Bordons, 2004). Model önceki sistem giriş ve çıkışları ile öngörülen kontrol girişlerine bağlı olarak sistemin gelecekteki yanıtlarını öngörmekte kullanılır. Bu kontrol girişleri optimizasyon işlemi ile elde edilir.

MATLAB, Simulink ve MPC Toolbox Şekil 2.2'deki yapıyı gerçekleyen bir yazılımdır. Bu yazılıma sistem modeli, kısıtlamalar ve optimizasyon ölçütü tanımlanarak, uygun ayar ve konfigürasyon altında model öngörülü denetim işlemi gerçekleştirilir.

Bu tezin amaçlarından biri sıradan bir Elektrik-Elektronik mühendisinin de bu dokümantasyon sayesinde model öngörülü denetim DSP kart kullanılarak gerçekleştirebilir olmasıdır.



Şekil 2.2 MPC Yapısı

### 2.3 Model Öngörülü Kontrolün Üstünlükleri

- Kavramlar sezgisel olduğu ve aynı zamanda parametre ayarları diğer yöntemlere göre daha kolay olduğu için kontrol bilgisi sınırlı olan kişiler tarafından bile kullanılabilir.
- Basit dinamiğe sahip olan proseslerden tutun çok daha karmaşık dinamiklere sahip olan çeşitli proseslerin kontrolünde de kullanılabilir.
- Çok değişkenli sistemlerin kontrolünde kullanılabilir.
- Ölü zaman etkisini giderme özelliğine sahiptir.
- Uygulaması kolay doğrusal bir kontrol kuralı elde edilir.
- Geleceğe ilişkin giriş değeri bilindiğinde oldukça yararlı ve kullanışlıdır.
- Belirli temel ilkeler üzerine kurulmuş geliştirilmeye tamamen açık bir yöntemdir.

## 2.4 Modele Öngörülü Kontrolün Eksik Yönleri

- Sistem dinamiğinin değişmediği durumlarda kontrolörün elde edilme işlemi önceden yapılabilir ancak uyarlamalı kontrol yapılıyorsa tüm hesap işlemleri her örnekleme zamanında tekrarlanır.
- En önemli sorun ise sisteme ilişkin uygun modelin elde edilmesidir. Algoritma sistem modelinden yararlanılarak geleceğe ilişkin sistem yanıtının elde edilmesine dayandığından dolayı gerçek sistemle yanıt arasındaki farklılık arttıkça istenilen sistem yanıtını elde etmekte sorunlar yaşanır.

## BÖLÜM 3

### KULLANILAN ARAÇLAR

Bu tezde kullanılan araçları donanımsal ve yazılımsal araçlar olarak sınıflandırmaktayız. Bu bölümde her iki sınıfı da incelemekteyiz.

#### 3.1 Donanımsal Araçlar

##### 3.1.1 Sayısal İşaretleyici Kart (DSP)

Sayısal işaret işleyiciler DSP olarak bilinmektedirler ve bundan sonra sayısal işaret işleyiciler kısaca DSP olarak belirtilecektir.

Günümüzde DSP'ler video, müzik, haberleşme ve ölçme tekniğinde yaygın olarak kullanılmaktadır. Elektrik makinelerinin hız, moment ya da konum denetiminde yüksek hızlı ve doğruluklu işlem başarımının elde edilmesinde ve daha karmaşık denetim yöntemlerinin uygulanmasında DSP'ler önemli yer tutmaktadır.

Her türlü konvektör ve motor denetimi için üretilmiş hızlı ve yetenekli DSP'lerden birisi olan TMS320F2812 DSP kartı bu çalışmada kullanılmıştır.

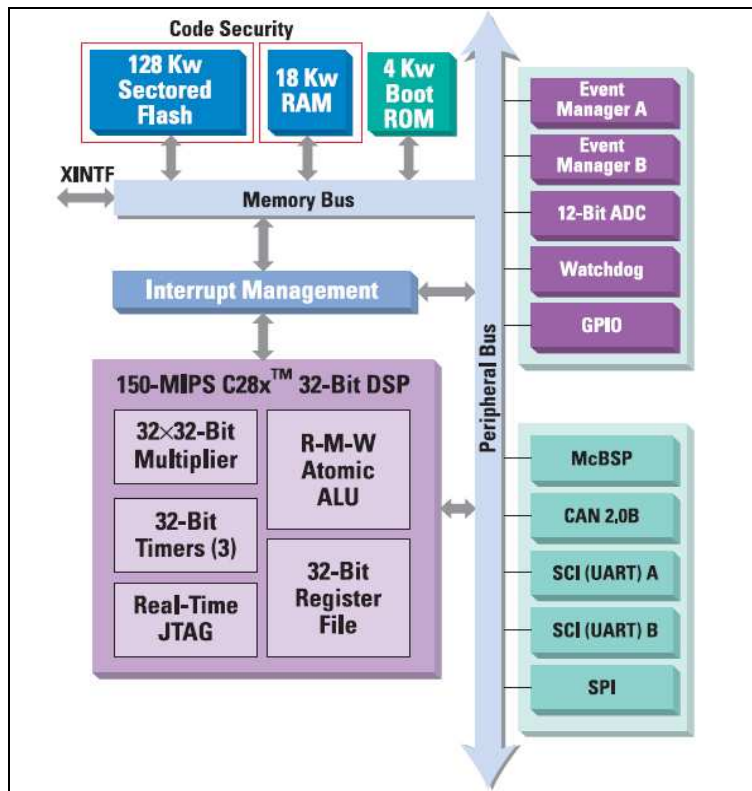


Şekil 3.1 TMS320F2812 DSP

Spectrum Digital tarafından deneysel kullanımlar için geliştirilmiş kartı olan eZdsp TMS320F2812 DSK, bilgisayarların paralel portu üzerinden Code Composer Studio (CCS) ara yüzü ile programlanabilmektedir. TMS320F2812 DSP, sabit noktalı işlem yapan 32-bit 150Mhz bir işlemci olup, 16-kanal 6.67ns çözünürlüklü ölü zaman ayarlı programlanabilir.

PWM çıkışı, 16 kanal 12-bit 80ns dönüşüm zamanlı A/D çevirici, 4 adet sayısal yakalama girişi ve 4 adet kare dalga kodlayıcı girişi, 16-bit 7 port programlanabilir sayısal giriş-çıkış, 18K word RAM ve 128K word Flash EEROM ve C/C++ programlama desteğine sahiptir.

İşlemci sabit noktalı aritmetik işlem yapmasına rağmen IQmath kütüphane desteği ile kayan noktalı aritmetik işlem kolaylığında ve ona yakın doğrulukta işlem yapabilmektedir. Kartın Blok Şeması Şekil 3.2'deki gibidir.



Şekil 3.2 TMS320F2812 DSP Blok Şeması (Spectrum Digital Inc., 2004)

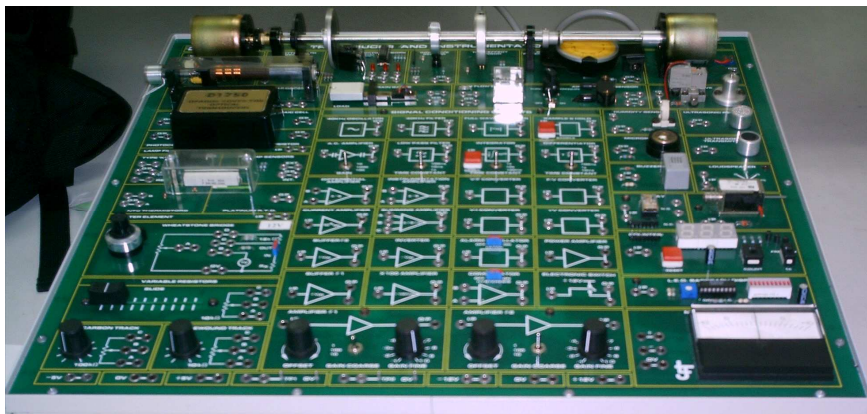
### eZdsp TMS320F2812 İşlemcisinin Seçilme Nedeni

TMS320F2812 işlemcisinin tez çalışmasında kullanılması aşağıdaki nedenlerden dolayı düşünülmüştür:

- Motor kontrol sistemlerine için geliştirilmiş özel bir işlemcidir.
- 150MIPS işlem basarımı
- 18K Word RAM, 128K Word Flash EEPROM
- 30MHz Clock
- 6 çift tümleyenli, 4'ü bağımsız toplam 16 PWM
- Programlanabilir ölü zaman mantığı
- 12-bit 16 kanal ADC
- 56-pin Sayısal Giriş/Çıkış
- Gerçek zamanlı analiz, grafik ara yüz, değişken ve kayıtçı gözlemi
- CCS, C/C++ , MATLAB desteği

#### 3.1.2 Digiac 1750

Bu çalışmamızda, kartımızın performansını Digiac 1750 deney seti üzerinde göstereceğiz. Digiac 1750 klasik kontrol sistemleri eğitiminde kullanılan deney setidir. Şekil 3.3'te Digiac 1750'nin genel görünümü gösterilmiştir.



Şekil 3.3 Digiac 1750



## **3.2 Yazılımsal Araçlar**

### **3.2.1 MATLAB**

MATLAB; ilk defa 1985’de C.B Moler tarafından matematik ve özellikle de matris esaslı matematik ortamında kullanılmak üzere geliştirilmiş etkileşimli bir paket programlama dilidir. MATLAB tüm mühendislik alanında, sayısal hesaplamalar, veri çözümlenmesi ve grafik işlemlerinde kolaylıkla kullanılabilen bir programlama dilidir. FORTRAN ve C dili gibi yüksek seviyeden programlama dilleri ile yapılabilen hesaplamaların pek çoğunu MATLAB ile daha kullanıcı dostu tarzda yapmak mümkündür ( Yüksel, I., 2004).

MATLAB’ın kullanım yerleri aşağıdaki başlıklarla ifade edilebilir:

- Denklemlerin çözümünü, doğrusal ve doğrusal olmayan diferansiyel denklemlerin çözümünü, integral hesabı gibi sayısal hesaplamalar,
- Veri çözümlenme işlemleri,
- İstatistiksel hesaplamalar ve çözümlenmeler,
- Grafik çizimi ve çözümlenmeler,
- Bilgisayar destekli denetim sistemi tasarımı.

### **3.2.2 Model Predictive Control Toolbox**

Model Predictive Control Toolbox tanımlanmış sistem modelleri veya lineer Simulink modeller üzerinden tasarım, analiz ve simülasyon yapmak için kullanılır. Model Predictive Kontrol araçları çoklu giriş ve çıkış içeren kontrol sistemlerinin optimizasyonunu sağlar ve belirli zaman aralıklarındaki her adımda maliyet hesaplayarak minimize eder (Bemporad ve diğ., 2009).

### **Temel Özellikler**

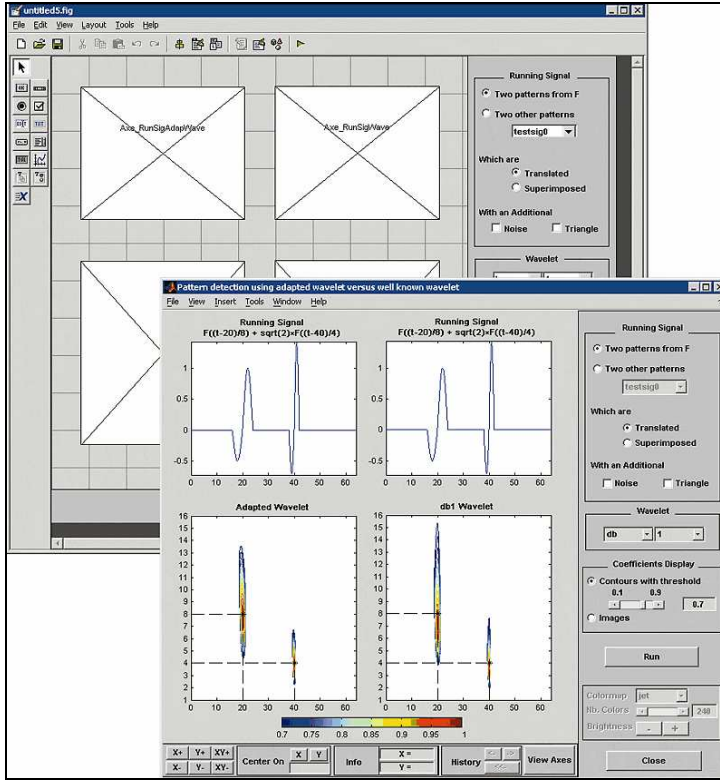
- Model öngörülü kontrollerin modellenmesi ve simülasyonu için grafik kullanıcı ara yüzü ve MATLAB komutları içerir.
- Lineer Simulink modelden veya deneysel verilerden model tanımlanabilir.
- Simulink blokları model öngörülü kontrollerin tasarımı ve simülasyonu için kullanılabilir.
- Ölçülen veya ölçülemeyen tüm dış etkenler ve kısıtlamaları kullanır.
- Uygulamayı dağıtım için C kodu oluşturur.

### **3.2.3 Simulink**

Bir MATLAB ara yüzü olan Simulink dinamik sistemlerin modellenmesi, benzetimi ve analizi için kullanılan bir yazılım paketidir. Simulink'te bir modelleme işlemi için sürükle bırak mantığı ile taşınan bloklar kullanılır. MATLAB kodu yazmak yerine, işlem blokları birbirine bağlanarak model diyagramları oluşturulur. Blok simgeleri, sistemin girişlerini, sistemin parçalarını veya sistemin çıkışlarını gösterir. Simulink, sayısal işaret işlemede ve birçok diğer alanda gerçek dünya sisteminin benzetimini yapabilmektedir. Akademik ve endüstriyel ortamlarda yaygın biçimde kullanılmaktadır.

### **3.2.4 Guide**

GUIDE, kullanıcı ara yüzü tasarımının yapıldığı MATLAB alt programıdır. Şekil 3.4'te örnek bir ara yüz gösterilmiştir.



Şekil 3.4 Bir Guide Penceresi

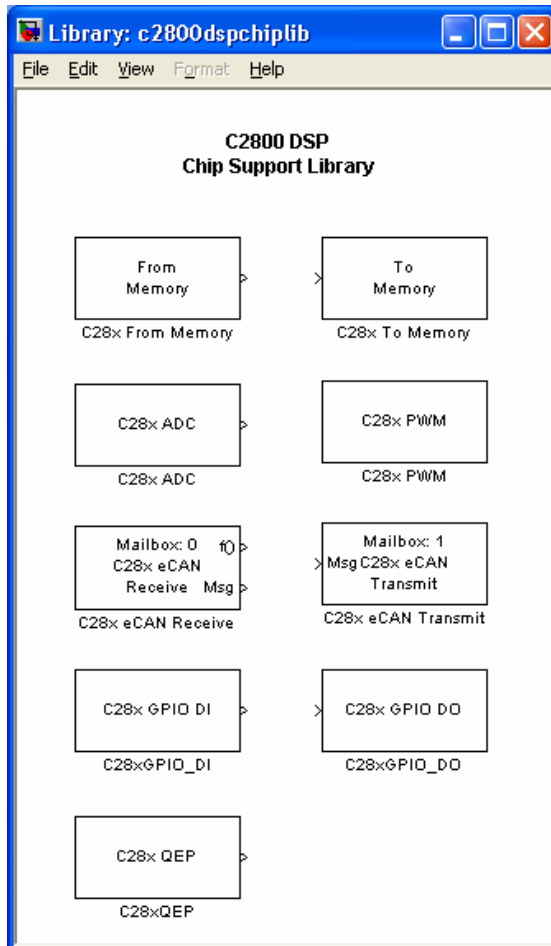
### 3.2.5 Real-Time Workshop

Real – Time Workshop, Simulink için optimum ve uyarlanabilir ANSI C kodu üreten MATLAB alt programıdır. Real–Time Workshop tüm Simulink özelliklerini desteklemekle birlikte 8-bit, 16-bit ve 32-bit tamsayı, virgüllü sayı ve uyarlanabilir (fixed-point) veri tiplerini içermektedir. Real Time Workshop'a eklenen Embedded Coder ürünü, gömülü işlemciler için özel eniyelenmiş kod yaratır. Eniyileme seçenekleri, işlemcinin özelliğine göre; bellek kullanımı, hesaplama hızı, veri türleri ve okunabilirlik gibi özellikleri içerir. Böylece yaratılan kod, elde yazılmış profesyonel kod ile kıyaslanabilir düzeyde verimli, eniyileyebilir ve okunabilir şekle dönüşmüş olur. Simulink'in kod üretme araçları, yaratılan kodu doğrudan doğruya gömülü işlemcinin

belleğine yükleyebilecek araçlar içerir. Bu süreçte, Real Time Workshop ve Embedded Coder, C kodunu seçilen gömülü işlemciye özel üretirler.

### 3.2.6 Embedded Target for TI C2000 DSP

The Embedded Target for TI C2000 DSP, MATLAB ve Simulink ile Texas Instruments C2000 DSP kart ile entegredir. Kullanıcılar için TI C2000 DSP kart üzerinde otomatik kod üretimi gerçekleştirmeyi sağlar. Simulink üzerine bu bloklar eklenerek modelleme yapılabilir. Bu kod blokları modele eklendiğinde MATLAB tarafından gerekli kod otomatik olarak üretilir. Şekil 3.5'te kod blokları gösterilmiştir.



Şekil 3.5 Embedded Target for TI C2000 DSP

### **3.2.7 SDconfig**

SDconfig, kullanıcının ana bilgisayar ile DSP arasındaki bağlantıyı test etmesine ve gerekli ayarları yapmasını sağlayan Windows GUI tabanlı bir yazılımdır. Bu yazılımla donanımın düzgün çalışıp çalışmadığı kolaylıkla kontrol edilebilir.

### **3.2.8 Code Composer Studio**

Code Composer Studio (CCS) programı, TMS320 DSP ailesinin üzerinde yazılım geliştirmek için kullanılan derleyici bir programdır. Bu program sayesinde TMS320 DSP kartı için C programlama dilinde yazılan programlar, TMS320 DSP üzerinde benzetimci (simülatör) biçiminde kullanılabilir. Program, sayısal sinyal işleme temelli uygulamaların tasarım sürecinde doğrudan kart üzerinde olmaksızın, yazılım üzerinden çalışmasına imkân veren bir geliştirme aracıdır. Bu yazılım, TMS320 DSP kartlarının standart algoritmasına sahiptir. Bu nedenle, geliştirme aşamasındaki bir uygulamanın sayısal işaret işleme temelli uygulamalarla birlikte uyumluluğu ve kullanımı bu yazılım üzerinde test edilebilmektedir.

## BÖLÜM 4

### MODELLEME

#### 4.1 Genel Sistem Tanımı

Model Öngörülü Kontrol Algoritması kullanılarak üzerinde test yapacağımız sistem için takometreli DC motor gereklidir. Test için bölümümüz kontrol laboratuvarındaki Digiac 1750 kontrol deney kiti üzerinde bulunan DC motor ve takometre kullanılmaktadır. Tezin katkı alanı dışında olması dolayısı ile DC motor modellemede “Model Öngörümlü Kontrolün ARM Microişlemci ile Gerçeklenmesi” (Ender Kelleci,2009) tezi ile aynı grup çalışması sonuçları kullanılmaktadır.

#### 4.2 Motor Dinamiği

Genel DC motor transfer fonksiyonu

$$G(s) = \frac{w(s)}{V_a(s)} = \frac{\frac{1}{K_b}}{(s\tau_1 + 1)(s\tau_2 + 1)}$$

şeklindedir.  $w$  motor shaft hızı ( $rpm$ ),  $V_a$  uygulanan giriş gerilimi ( $V$ ),  $K_b$  back emf sabiti ( $V / rpm$ ),  $\tau_1$  mekanik zaman sabiti ve  $\tau_2$  elektriksel zaman sabitidir.

Tablo 1 Motor Değerleri

Parametreler	
Sembol	Değer (SI Units)
$K_b$	10/2400 v/rpm
$\tau_1$	0.2 s
$\tau_2$	19.6 ms

Tabloda belirtilen deęerleri girdikten sonra sistemin transfer fonksiyonu;

$$G(s) = \frac{0.24}{(0.0196s + 1)(0.2s + 1)} = \frac{51.02}{(s + 51.02)(s + 5)}$$

şeklinde olmaktadır.

### 4.3 Süreç Modelinin Tanımlanması

Motor transfer fonksiyonunu kullanarak MATLAB'ta süreç modelini aşağıdaki MATLAB kodu ile tanımlanır.

$$Kb=10/2400 ;$$

$$tm=0.2;$$

$$te=19.6/1000;$$

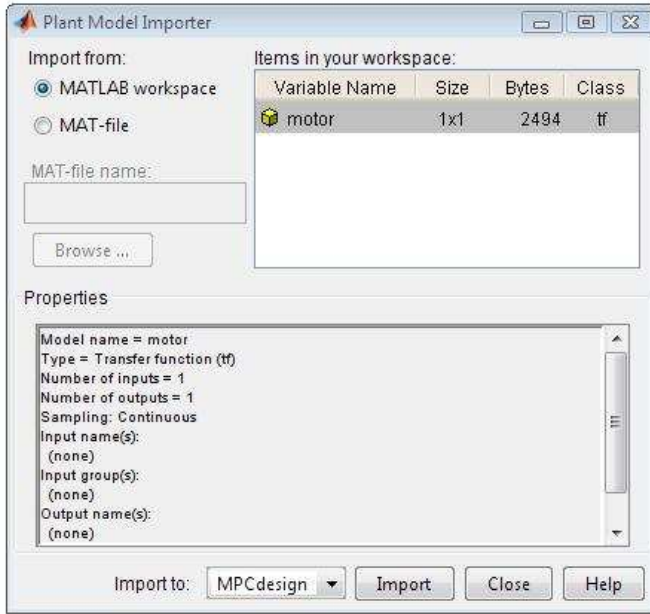
$$num=1/Kb;$$

$$den=[(tm*te) (tm+te) 1];$$

$$motor=tf(num,den);$$

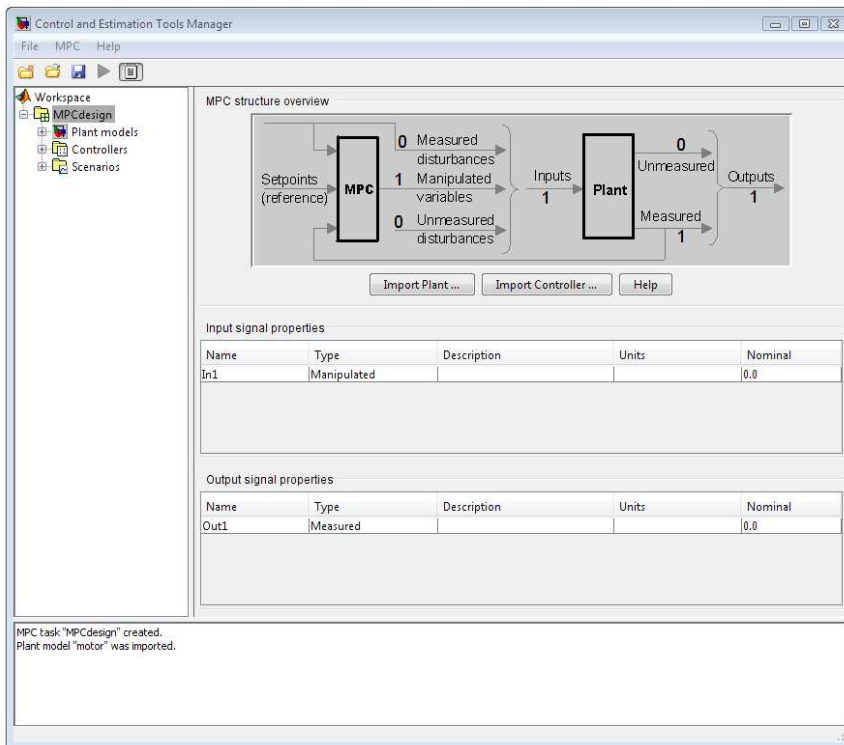
### 4.4 Model Öngörümlü Kontrol Oluşturulması

DC Motor sistemini doğrusal olarak modellediğimiz için Model Predictive Control Toolbox tasarım aracı doğrudan kullanılabilir. Bu nedenle MATLAB'ta Model Predictive Control Toolbox ta oluşturulan sürecin modeli Şekil 4.1'deki gibi yüklenir.



Şekil 4.1 Süreç Modeli Yükleme

Model yükledikten sonra Şekil 4.2’deki gibi tasarım aracının ana penceresinde tablolar görülebilir.

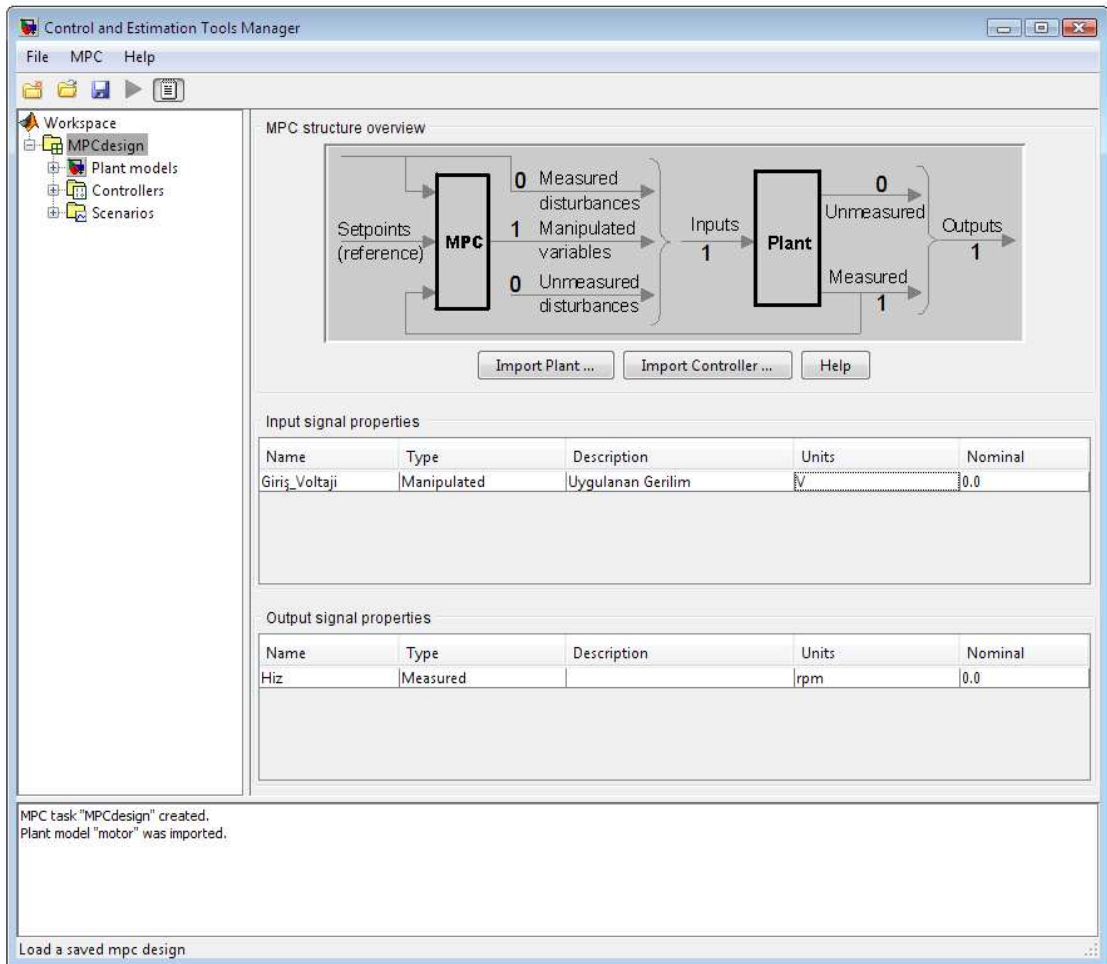


Şekil 4.2 MPC Tasarım Aracı



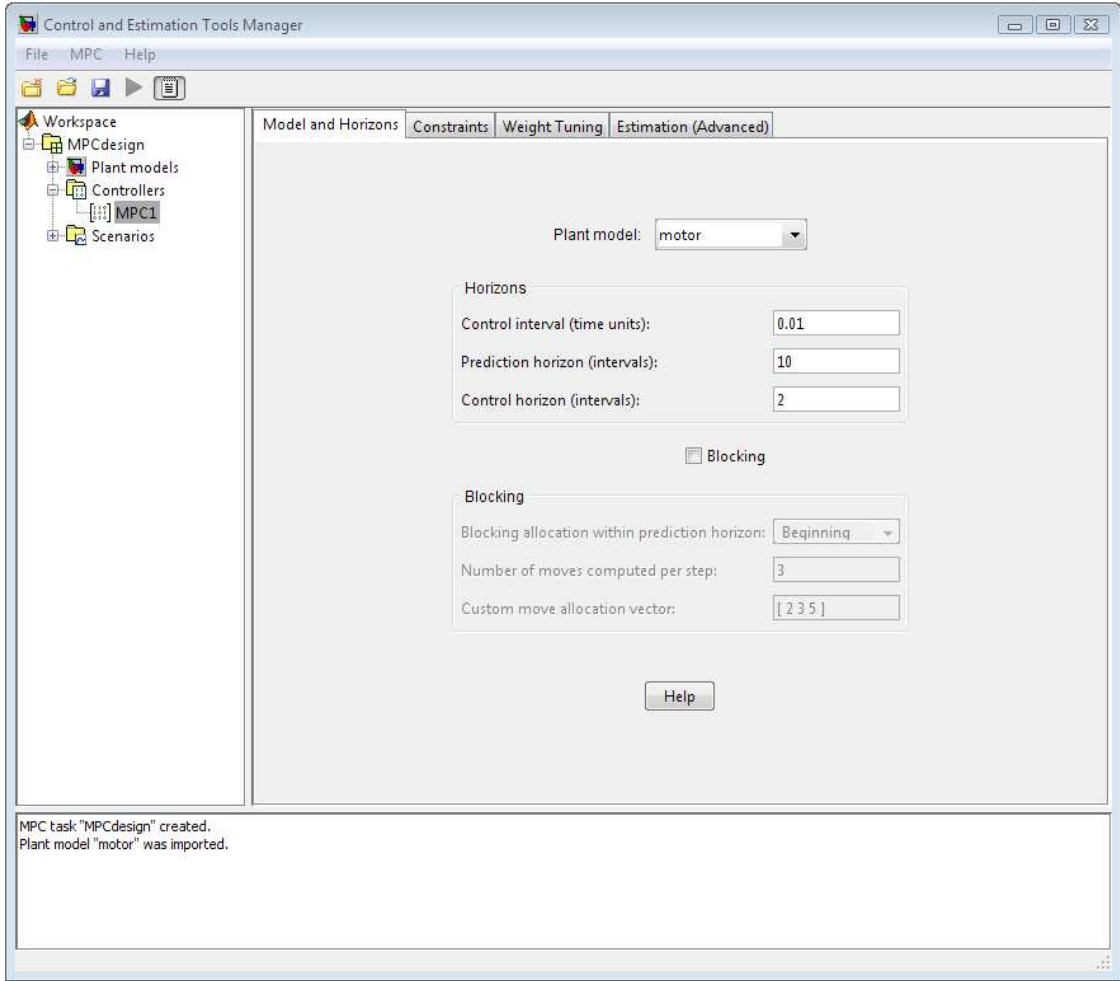
Varsayılan olarak giriş sinyali adı “In1” tipi ise “manipulated” (değiştirilebilir)” ve çıkış sinyali adı “Out1” tipi ise “measured” (ölçülebilir) olarak görülmektedir.

In1, Out1 şeklinde isimlendirilen sinyalleri “Giris\_Voltaji” ,”Hiz” olarak adlandırılır. Description (Açıklama) ve Units (Birim) sütunları varsayılan olarak sıfır olacaktır. Bütün değişiklikleri yaptıktan sonra sistem Şekil 4.3’teki gibi olacaktır.



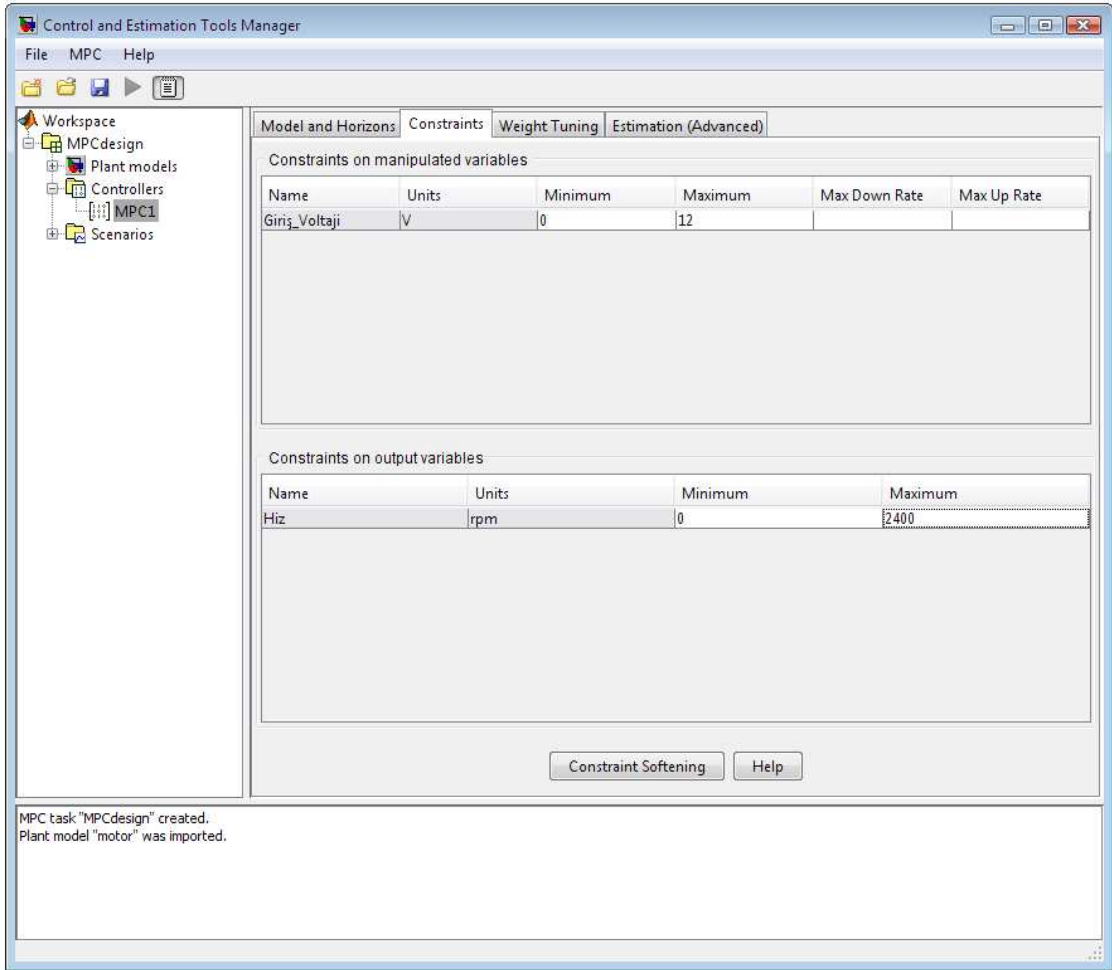
Şekil 4.3 Sinyal Özelliklerinin Tanımlanması

Sinyal özellikleri tanımlandıktan sonra Şekil 4.4’teki gibi Model Öngörülü Denetleyicinin özellikler penceresini açılıp “Control Interval” (Kontrol Aralığı) değeri 0,01 saniye, Prediction Horizon (Tahmin Ufku) değeri “10” ve Control Horizon (Kontrol Ufku) değeri “2” olarak değiştirilir.



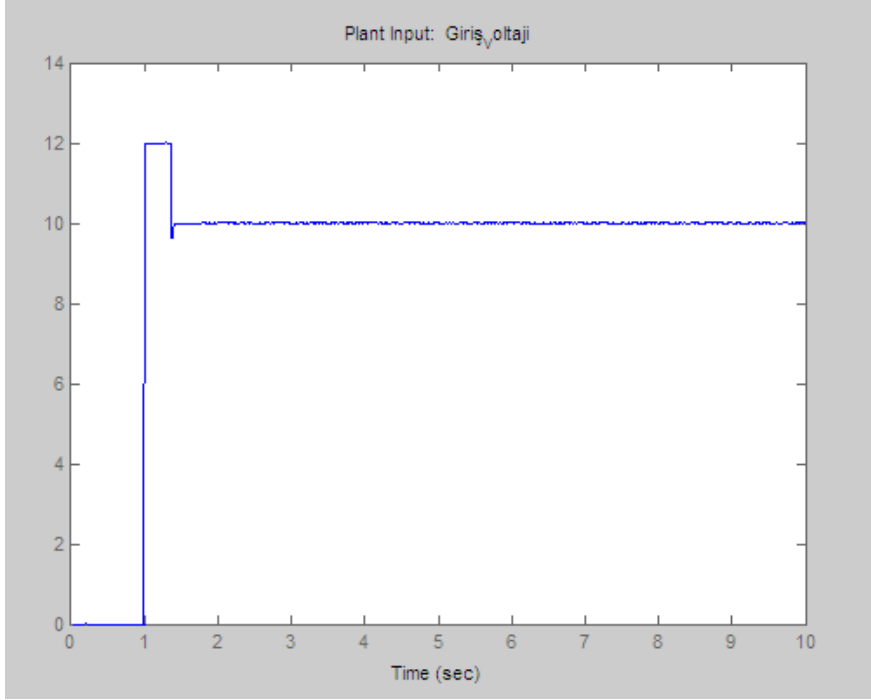
Şekil 4.4 Model Öngörülü Denetleyicinin Özellikler Penceresi

Model Öngörülü Denetleyicinin özellikleri belirledikten sonra Constraints (Kısıtlamalar) özelliklerini değiştirmek için Constraints sekmesinde giriş voltajı için 0-12 volt ve çıkış hızını 0-2400 rpm olarak Şekil 4.5'teki gibi sınırlandırmalar yapılır.

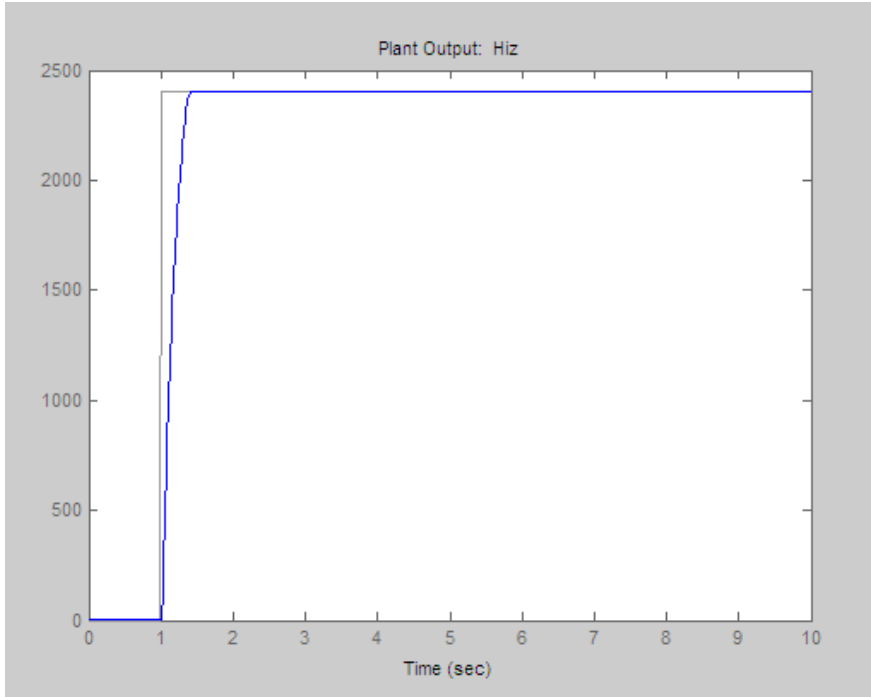


Şekil 4.5 Constraints (Kısıtlamalar) Özelliklerini Tanımlanması

Scenarios (Senaryo) sekmesinde oluşturulan süreç step, ramp, continuous vb. giriş sinyalleri için simüle edilebilir. Örnek olarak 2400 rpm hız için step seçilip simüle edildiğinde Şekil 4.6 ve 4.7’de gösterilen giriş ve çıkış grafikleri elde edilir. Bu grafikler de mavi çizgiler çıkış sinyallerini gri çizgiler ise setpoint değerlerini ifade etmektedir.

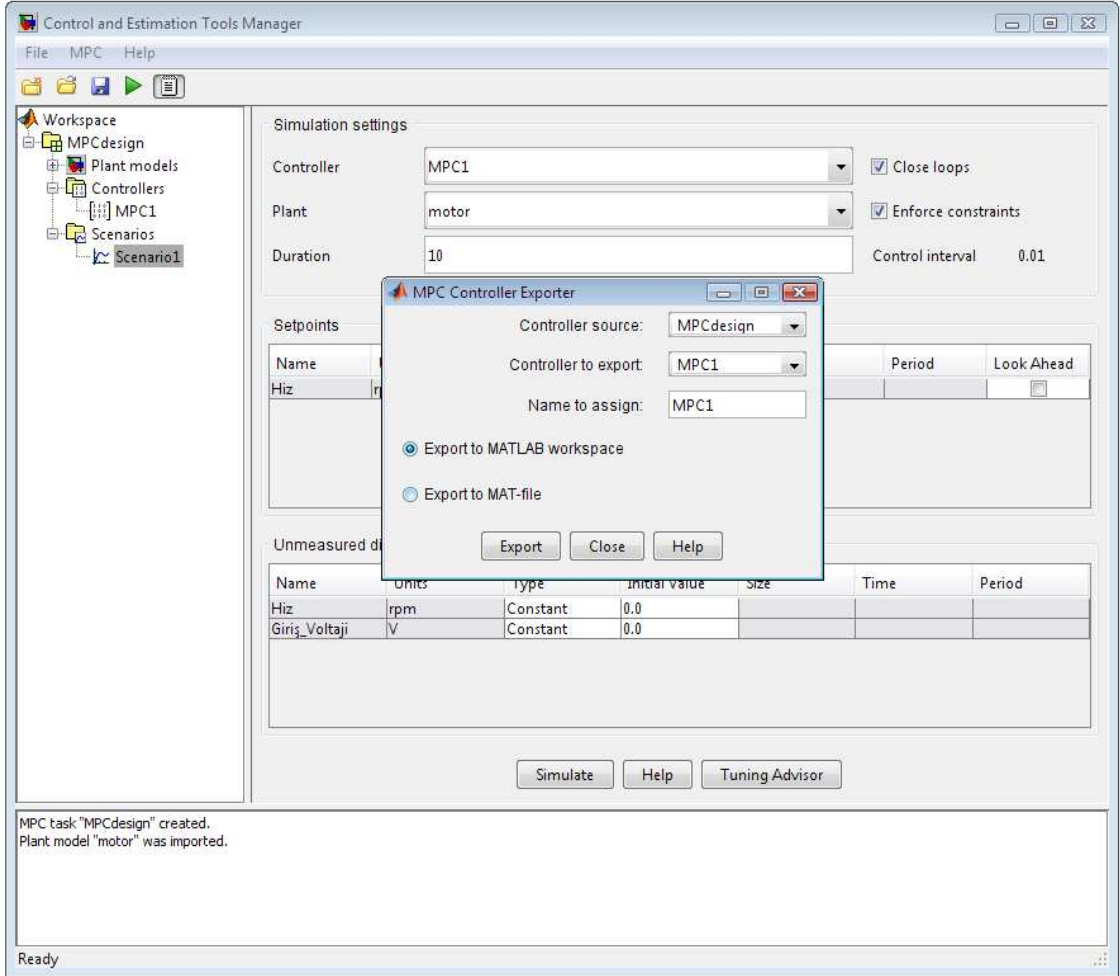


Şekil 4.6 Süreç Giriş Grafiği



Şekil 4.7 Süreç Çıkış Grafiği

Sonuç olarak oluşturulmuş olan Model Öngörülü Denetleyici Şekil 4.8’de görüldüğü gibi çalışma alanına aktarılır.



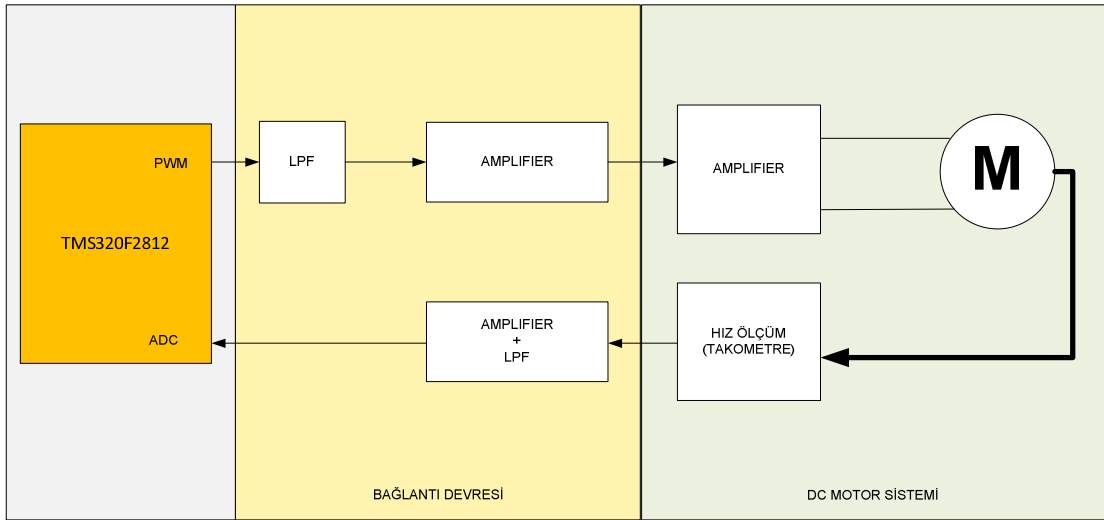
Şekil 4.8 MPC Dışa Aktarımı

## BÖLÜM 5

### UYGULAMA

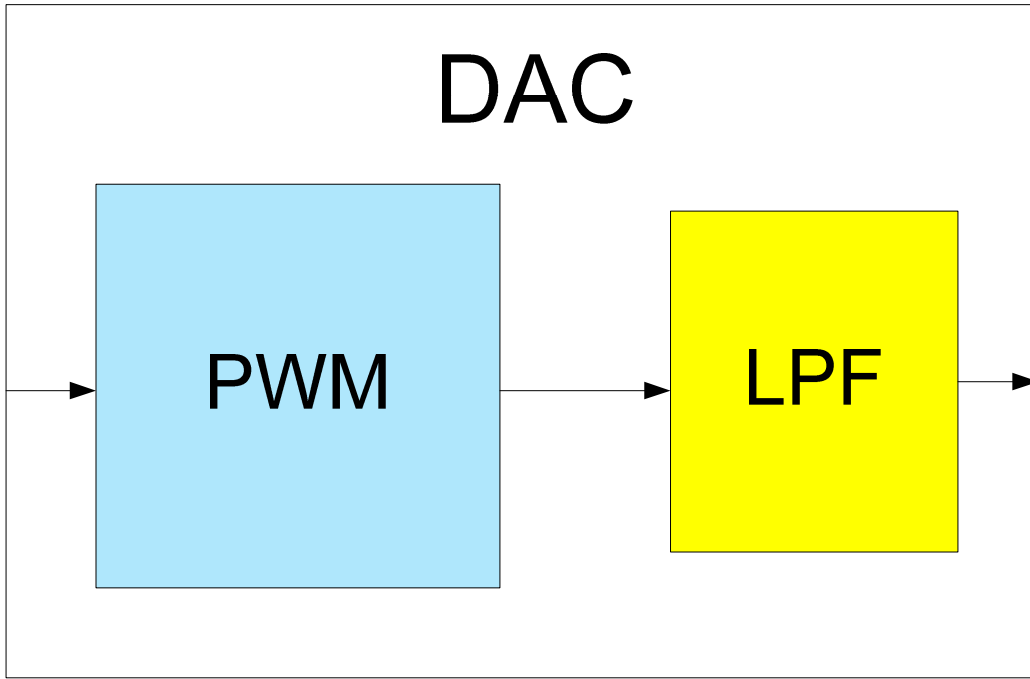
#### 5.1 Elektriksel Bağlantı

Uygulama Şekil 5.1'deki gibi TMS320F2812 DSP kartı, Bağlantı Devresi ve DC Motor olmak üzere üç bölümden oluşmaktadır.



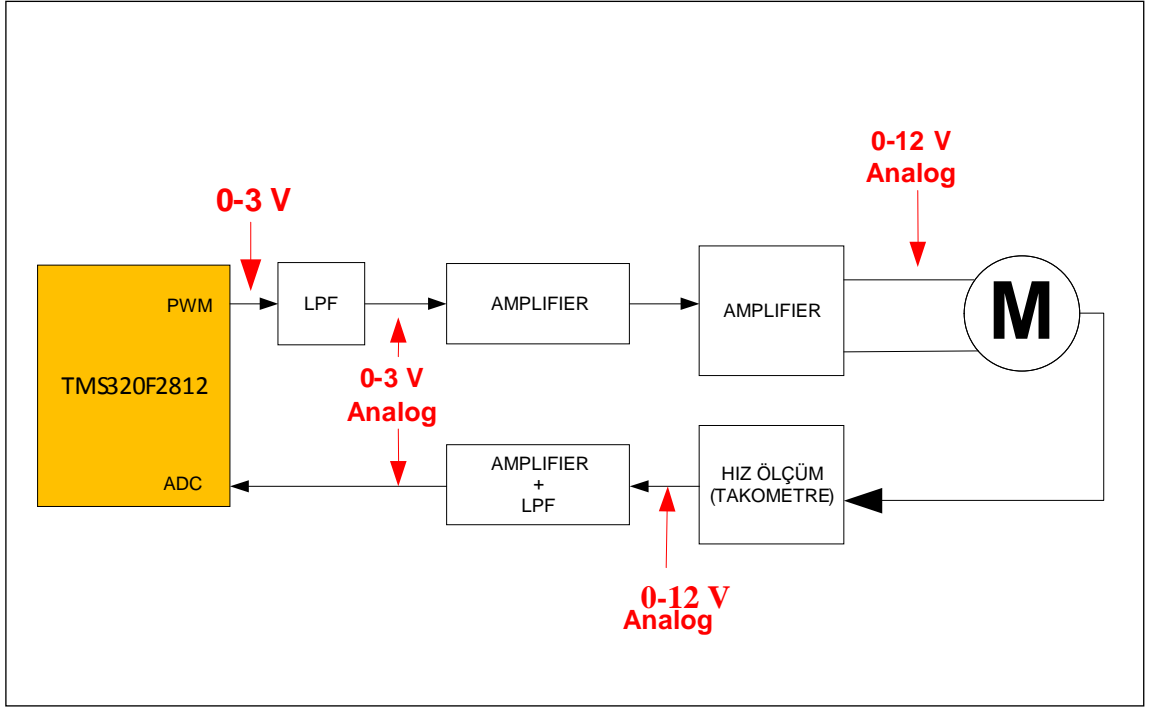
Şekil 5.6 Elektriksel Bağlantı Şeması

Şekil 5.2'de de görüldüğü gibi DSP kartın PWM çıkışı ile Alçak Geçiren Süzgeç (Low Pass Filter- LPF) birlikte Dijital-Analog Çevirici işlevi yapmakta ve motoru sürmektedir.



Şekil 5.27 DAC

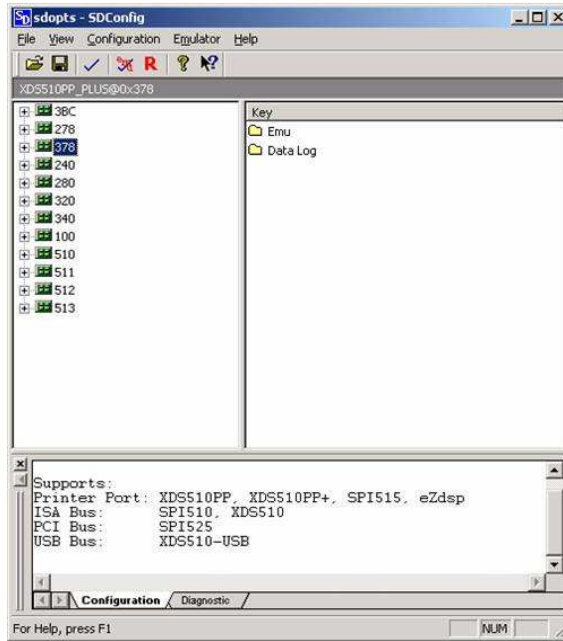
Alçak geçiren süzgeçten çıkan gerilim yükseltici (Amplifier) de yükseltilerek motor için gerekli olan 0-12 V arasında gerilimi üretmektedir. Motor sisteminde takometreden gelen hız değeri analog 0-12 V arasında olacaktır. Takometreden ölçülen gerilim yükseltici ve alçak geçiren süzgeç ile 0-3 V değerlerine düşürülerek DSP kartın ADC girişine bağlantısı yapılmaktadır. Sonuç olarak sistem Şekil 5.3'teki gibi oluşmaktadır.



Şekil 5.3 Elektriksel Bağlantı Özellikleri

## 5.2 DSP Kartın Bağlantı Ayarlarının Yapılması

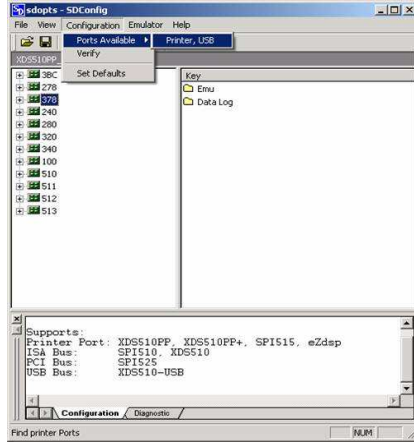
DSP kartın bağlantı ayarları SDconfig programı ile yapılmaktadır. SDconfig programı çalıştırıldığında Şekil 5.4'teki ekran açılır.



Şekil 5.4 SDconfig

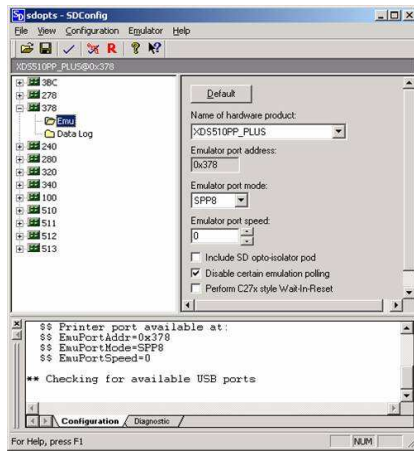


Bilgisayar ile DSP arasındaki bağlantı bilgisayarın paralel portundan yapıldığı için Configuration menüsünün altında Ports Available menüsü seçilir ve burada Printer, USB butonu Şekil 5.5'teki gibi tıklanarak portların uygun olup olmadığını kontrol edilir.



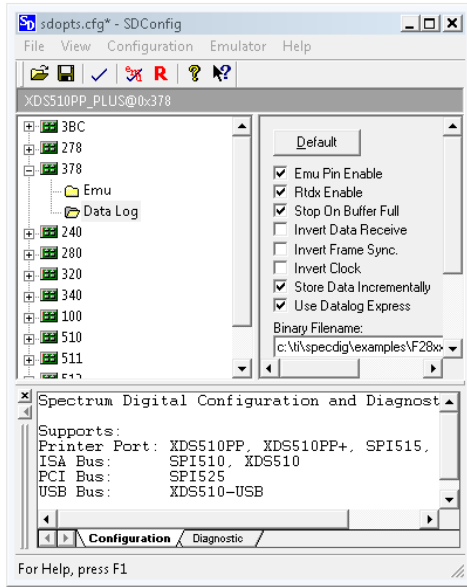
Şekil 5.5 SDconfig Bağlantı Ayarları

Şekil 5.6'da da görüldüğü gibi sol menüde Emu ve Data Log olmak üzere iki kısım bulunmaktadır. Emu emulasyon yapılandırma seçeneklerini, Data Log ise veri giriş seçeneklerini ayarlamakta kullanılır. Emu tıkladığında mevcut emulasyon ayarlarını ekrana getirir.



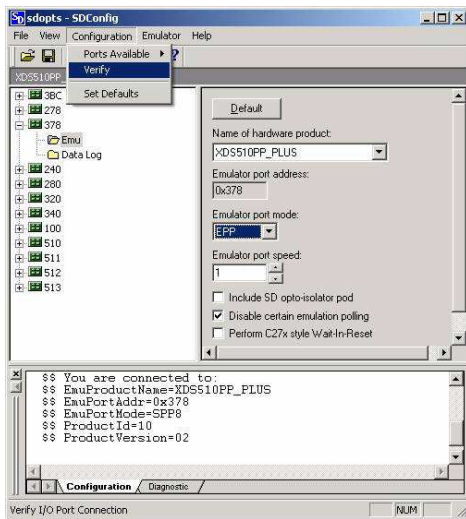
Şekil 5.6 SDconfig Emu Ayarları

Bilgisayar ve DSP arasındaki veri alış verişi RTDX üzerinden yapıldığı için Şekil 5.7'deki gibi Data Log tıklanılarak veri giriş seçenekleri açılır ve RTDX seçeneği etkinleştirilir.



Şekil 5.7 SDconfig Data Log Ayarları

Configuration menüsünde Verify seçeneği Şekil 5.8'deki gibi tıklanılarak yapılmış olan değişikliklerden sonra bilgisayar ve DSP arasındaki bağlantı kontrol edilir ve SDConfig yazılımı kapatılır.



Şekil 5.8 SDconfig Bağlantı Test

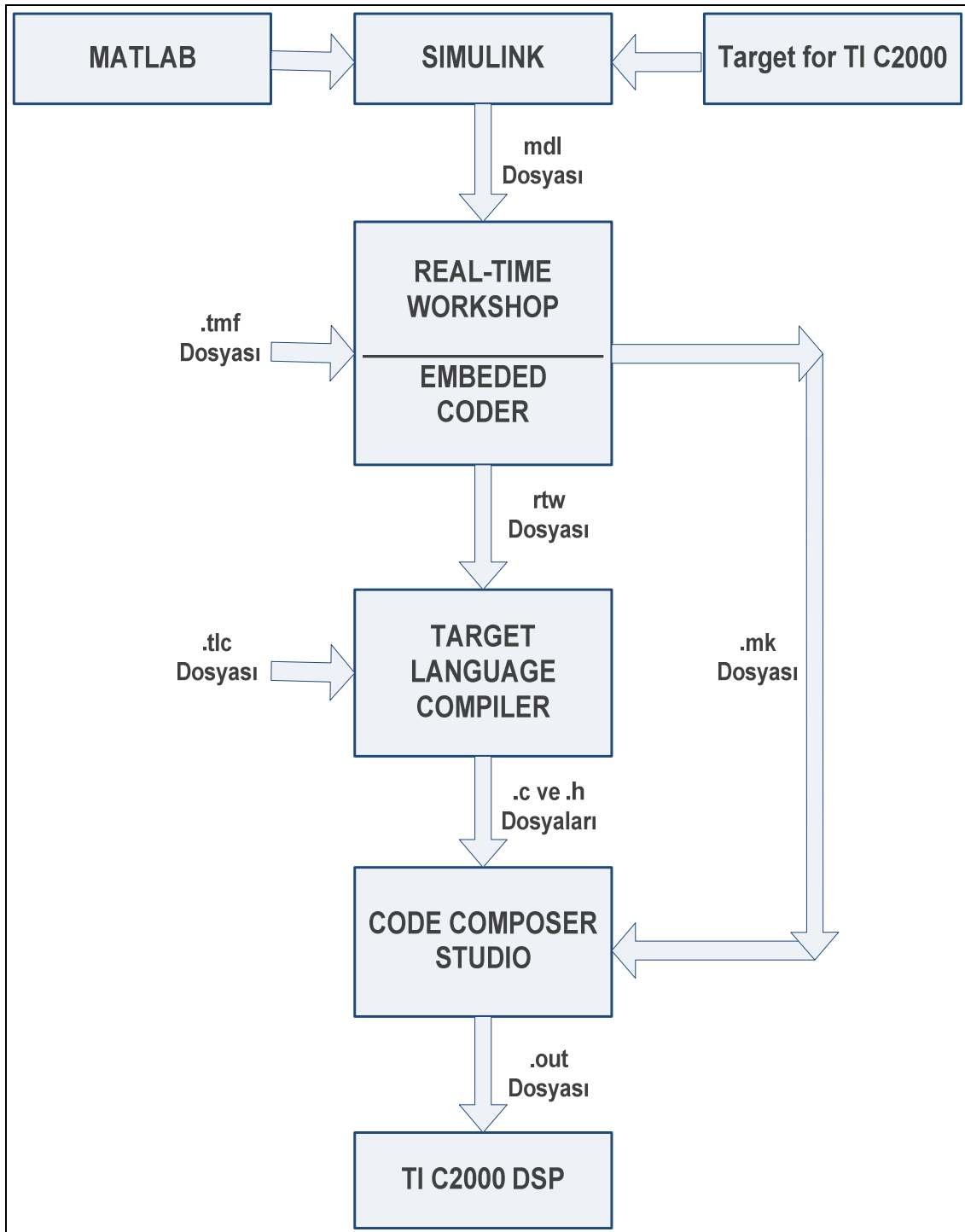
### 5.3 Uygulamamın Genel Yapısı

Yeniden kullanılabilir ve bakım aşamasını kolaylařtıran yazılımların geliştirilmesi ve yazılım geliřtirmede üretkenliđin artırılması, yazılım mühendisliđi arařtırmacılarının öncelikli hedeflerini oluřturmaktadır. Model tabanlı yazılım geliřtirme, bu hedeflere ulařılabilmesi için, yazılım geliřtirmenin model geliřtirme ve model dönüşümleri ile gerçekleştirilmesi fikrine dayanmaktadır.

MATLAB/Simulink, denetim algoritmalarının blok diyagramlarla gösterilebilen benzetimi için model tabanlı bir geliřtirme ortamıdır. RTW (Real Time Workshop), Simulink ortamında oluřturulan grafiksel blok diyagramından C kodlarını otomatik olarak üretmektedir. Bu kodlar TI CCS (Code Composer Studio) DSP yazılım geliřtirme ortamına aktarılarak diđer destek dosyalarıyla birlikte derlenmekte ve bilgisayarın paralel portuna bađlı DSP'nin üzerindeki emülatör aracılıđı ile yüklenmektedir.

Tüm tasarım işlemleri tam otomatik, esnek ve oldukça kolay bir şekilde yapılabilmektedir. Ayrıca, herhangi bir programlama dili içine girmeden DSP blođu ve tasarım parametreleri gerçek zamanda deđiřtirilebilmektedir. Grafiksel blok diyagramların kullanımı, DSP üzerinden denetlenen sistem tasarımını, modellenmesini ve modifikasyonunu oldukça kolaylařtırmaktadır.

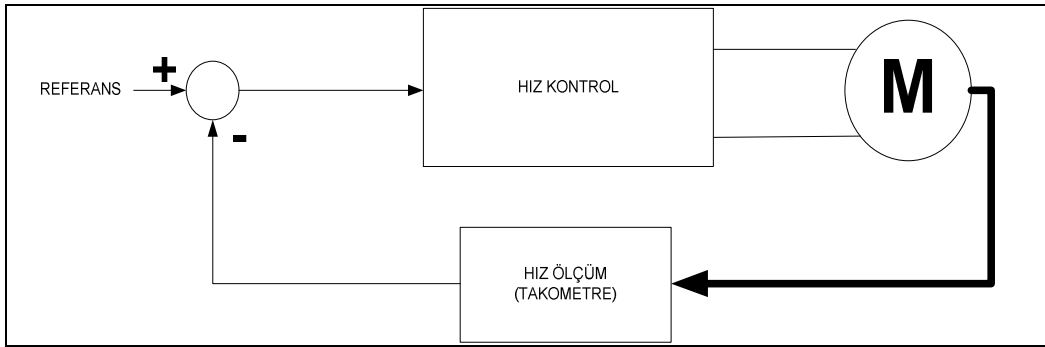
Blok diyagramlarla verilen DSP algoritmaları, Real Time Workshop, Link for Code Composer Studio Development Tools ve Embedded Target for TI C2000 DSP alt yazılımları ile Şekil 5.9'da görüldüđü gibi hedef DSP işlemcileri için koda çevrilmekte ve gerçek zamanda uygulanması ve tasarım dođrulaması çok hızlı bir şekilde yapılabilmektedir.



Şekil 5.9 Uygulamanın Genel Yapısı (Larsson, V., 2008)

#### 5.4 Sistemin Modellenmesi

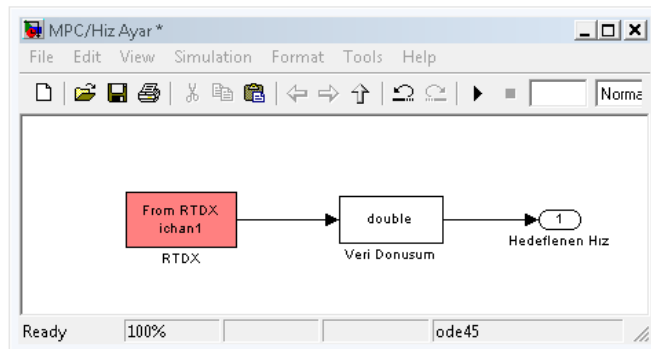
Sistem modeli hız ayar, hız kontrol ve hız ölçüm alt modellerinden oluşmaktadır. Motor hedef hızı kullanıcı arayüzü kullanılarak kullanıcı tarafından belirlenir ve sistemden alınan veriler grafiksel olarak gösterilir. Denetleyici takometreden alınan ve motora verilen gerilim değerlerini denetleyerek istenilen hızı korur. Kontrol döngüsü Şekil 5.10’da gösterilmiştir.



Şekil 5.10 Sistem Modeli

- **Hız Ayar Alt Modeli:**

Bu model Şekil 5.11’de gösterildiği gibi “from RTDX”, “Data Type Conversion” ve “out” bloklarından meydana gelmektedir. Bu alt model de kullanıcı arayüzüyle belirlenen hız RTDX ile hız kontrol bloğuna iletilir.”ichan1” diye isimlendirilen kanaldan veri gönderimi yapılır.



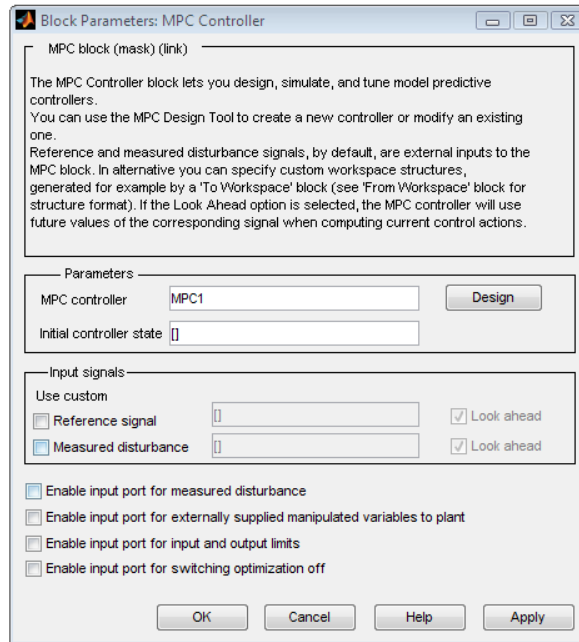
Şekil 5.11 Hız Ayar Alt Modeli

- **Hız Kontrol Alt Modeli:**

Model ölçülen hız ile istenilen hızı karşılaştırır ve motora buna göre gerilim uygular.

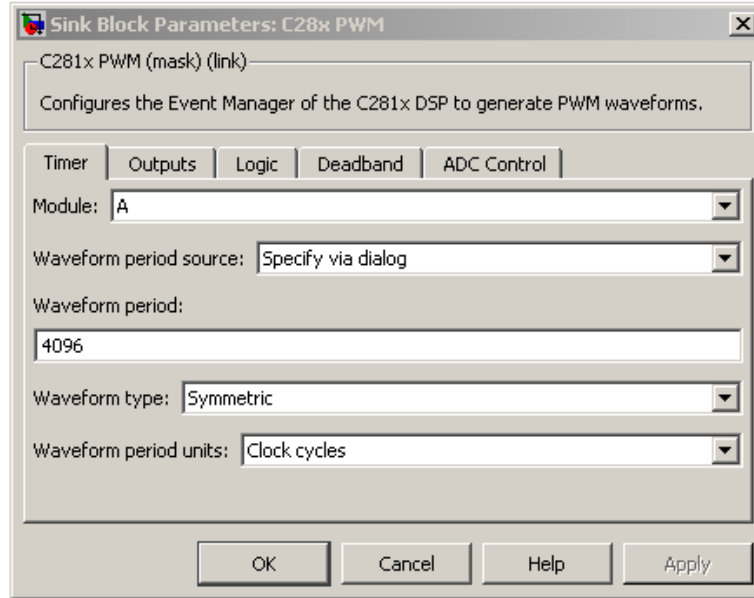
Hız Kontrol Alt Modeli “MPC Controller”, iki adet “Data Type Conversion”, iki adet “in”, ”F281x PWM” bloklarından ve RTDX veri gönderim alt modelinden meydana gelmektedir.

Öncelikle MPC bloğunu eklenir. Blok çift tıklanarak Şekil 5.12’deki gibi özellikler penceresi açılıp daha önceden oluşturulan süreç discrete hale çevrilip gerekli düzenlemeler yapılarak MATLAB çalışma alanında MPC denetleyicisi oluşturulur. Oluşturulan denetleyicinin ismi MPC bloğuna yazılıp “ok” butonuna basılarak Model Öngörülü Denetleyici eklenir.



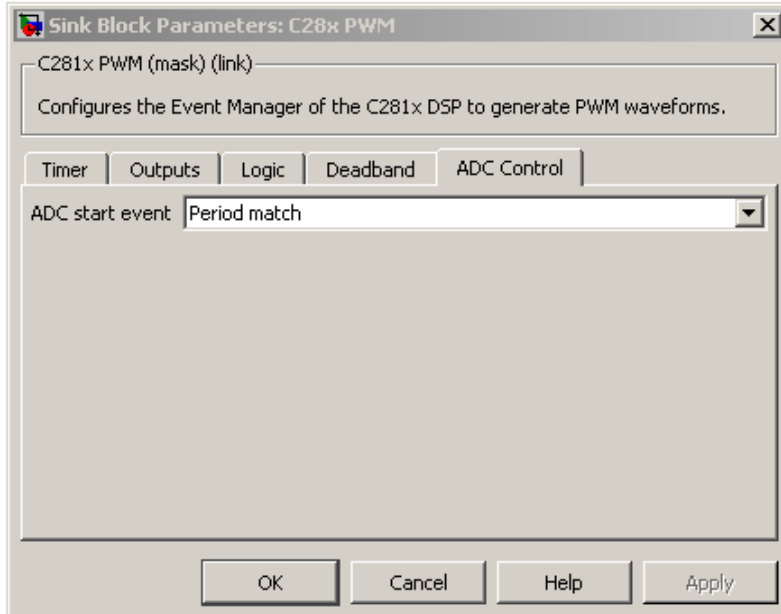
Şekil 5.12 MPC Blok Parametreleri

MPC denetleyicisi ekledikten sonra PWM bloğu eklenir. Eklenen blok çift tıklanarak özellikler penceresi açılıp Şekil 5.13’teki gibi değişiklikler yapılır.



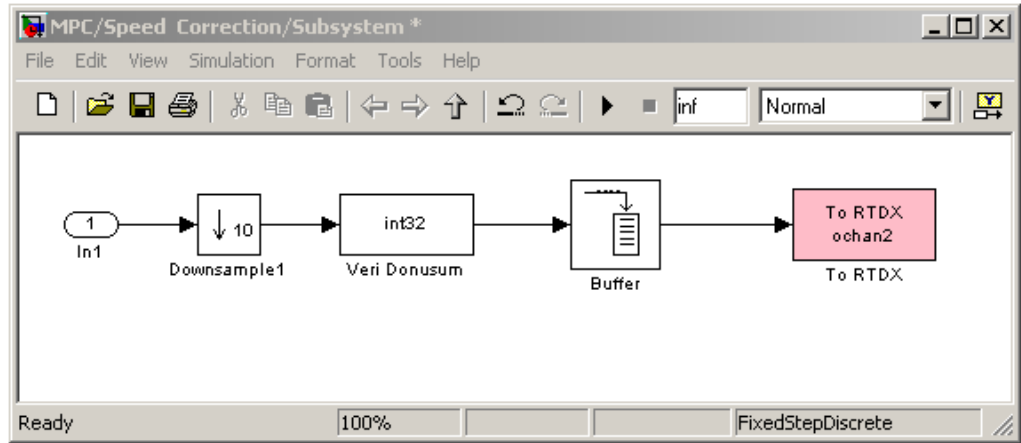
Şekil 5.13 PWM Blok Özellikleri

“ADC Control” sekmesinde Şekil 5.14’teki gibi “ADC start event” özelliği  
 “Period match” olarak değiştirilir.



Şekil 5.14 PWM Blok ADC Kontrol Özellikleri

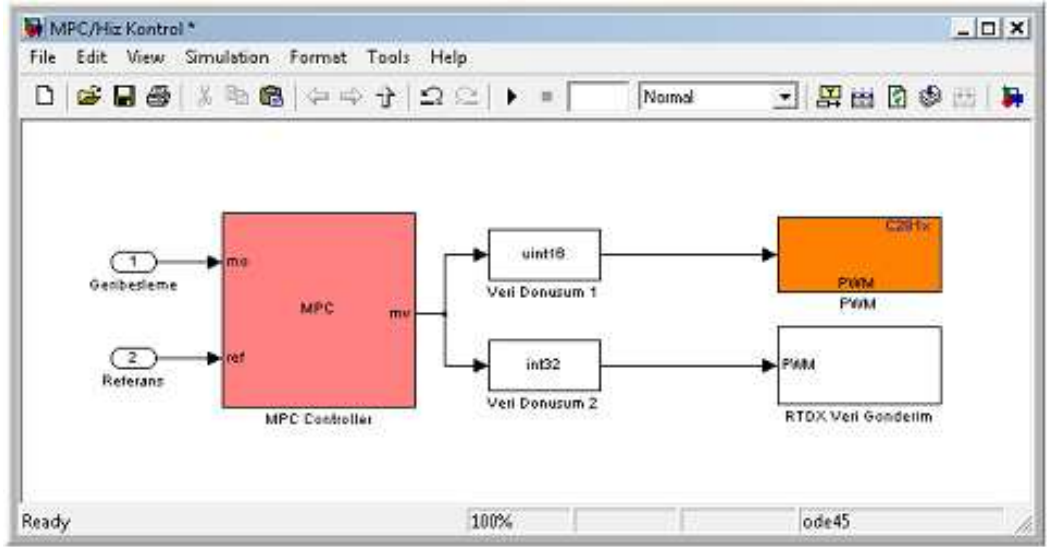
MPC Denetleyicisi için geribesleme ve referans olmak üzere iki adet giriş bloğu eklenir bağlantıları yapılır. Buradaki birinci giriş Hız Ölçüm Alt Modelinden gelen veri olacak diğer giriş ise Hız Ayar Alt modelinden gelen istenilen hız değeri olacaktır. MPC Denetleyicisinden çıkan veriyi uint16 ya dönüştürüp PWM ile bağlantısı yapılır. Kullanıcı arayüzünde PWM değerlerini görmek için RTDX Veri Gönderim alt modeli ile bağlantı yapılır. RTDX Veri Gönderim alt modeli Şekil 5.15'teki gibi "input", "Downsample", "Data Type Conversion", "Buffer" ve "to RTDX" bloklarından meydana gelmektedir.



Şekil 5.15 RTDX Veri Gönderim Alt Modeli

Tüm bağlantılar yapıldıktan sonra Hız Kontrol Alt Modeli Şekil 5.16'daki gibi olacaktır.



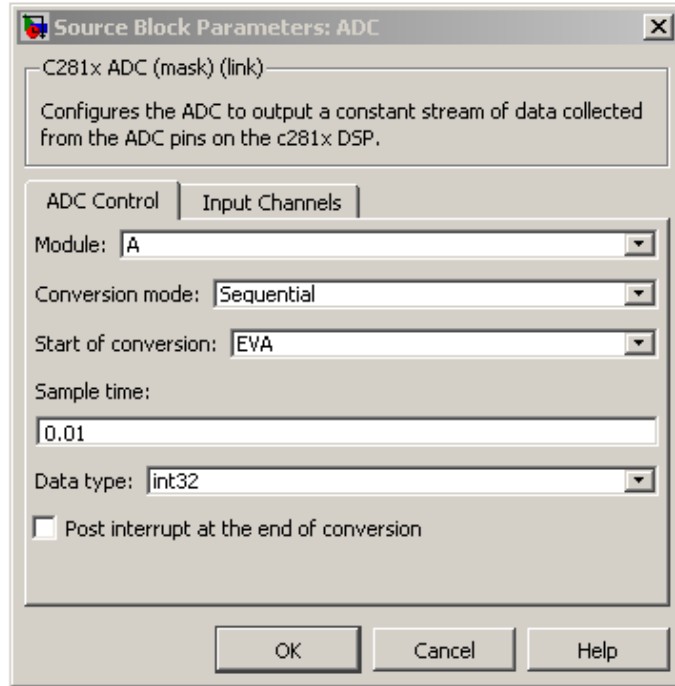


Şekil 5.16 Hız Kontrol Alt Modeli

- **Hız Ölçüm Alt Modeli:**

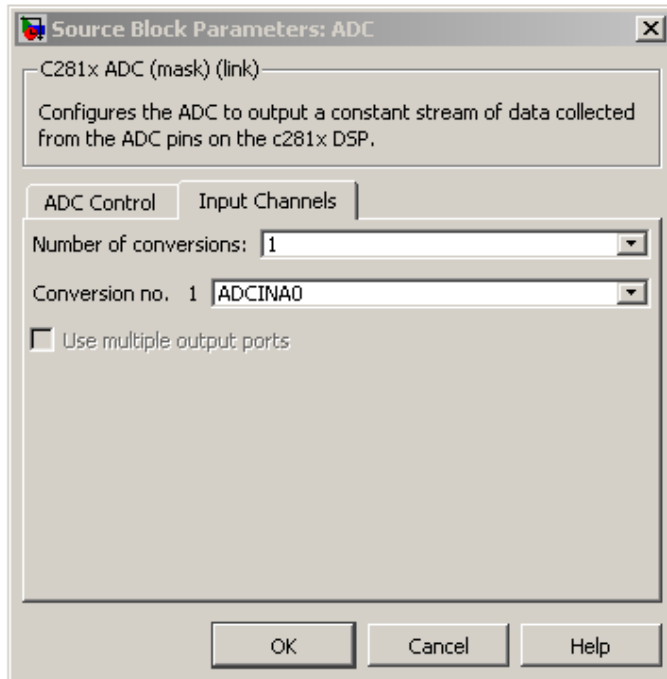
Bu blokta takometreden gelen hız ölçüm değerleri alınarak MPC Denetleyicisi için bir çıkış oluşturulur. Ayrıca RTDX Veri Gönderim alt modeli ile bilgisayara ölçülen hız değerleri gönderilir.

Hız Ölçüm Alt Modeli “ADC”, ”out” ve “Data Type Conversion” blokları ile RTDX Veri Gönderim alt modelden oluşmaktadır. Öncelikle bloklar modele eklenir. Blokları modele ekledikten sonra “ADC” bloğu çift tıklanarak özellikler penceresi açılıp Şekil 5.17’deki gibi “ADC Control” sekmesindeki seçenekler düzenlenir.



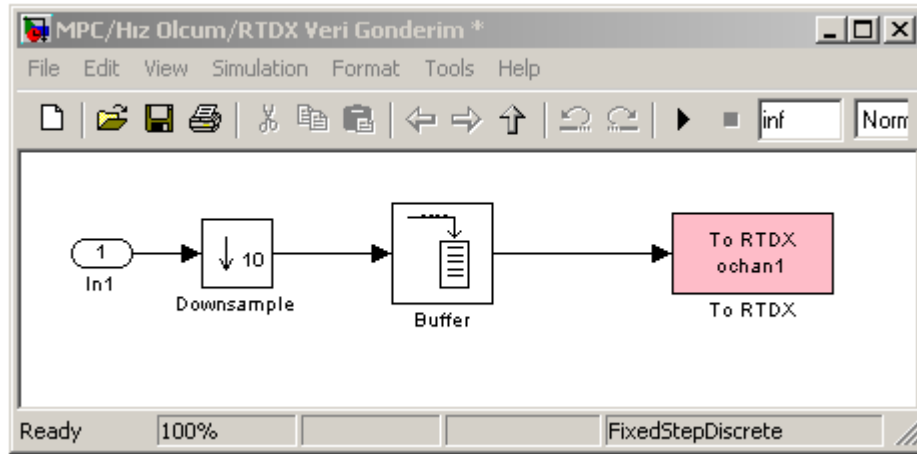
Şekil 5.17 ADC Blok Özellikleri

“ADC Control” sekmesindeki değişikliklerden sonra “Input Channels” sekmesinde DSP kartta kullanacak ADC kanalı Şekil 5.18’deki gibi seçilir.



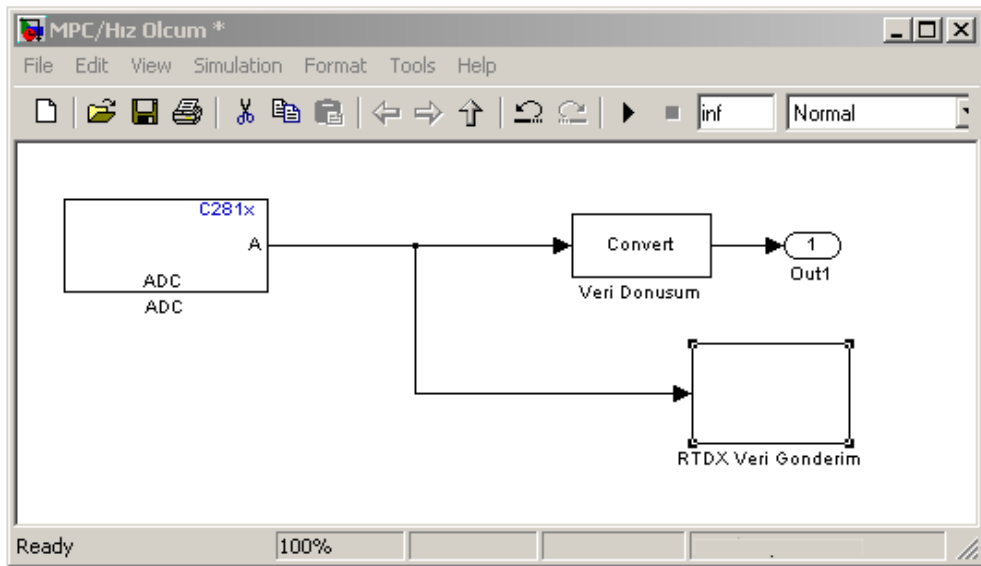
Şekil 5.18 ADC Blok Giriş Özellikleri

ADC bloğunun ayarları yaptıktan sonra RTDX Veri Gönderim alt modeli eklenir. Bu model Şekil 5.19'daki gibi "in","Downsample","Buffer" ve "to RTDX" bloklarından meydana gelmektedir.



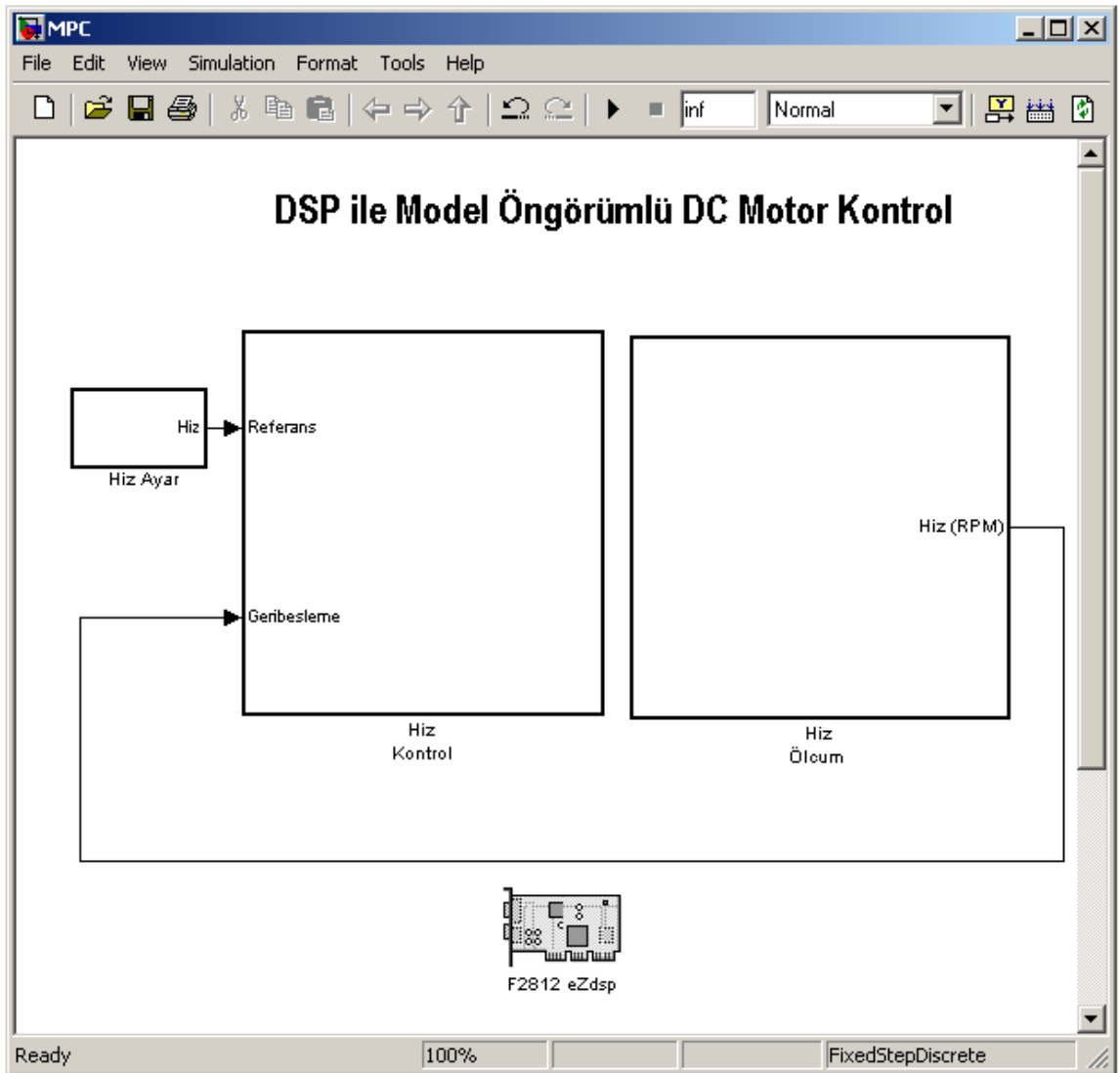
Şekil 5.19 RTDX Veri Gönderim Alt Modeli

Veri Gönderim alt modeli ekledikten sonra modeller arasındaki gerekli bağlantılar yapılır. Bağlantılar yapıldıktan sonra model Şekil 5.20'deki gibi oluşmaktadır.



Şekil 5.20 Hız Ölçüm Alt Modeli

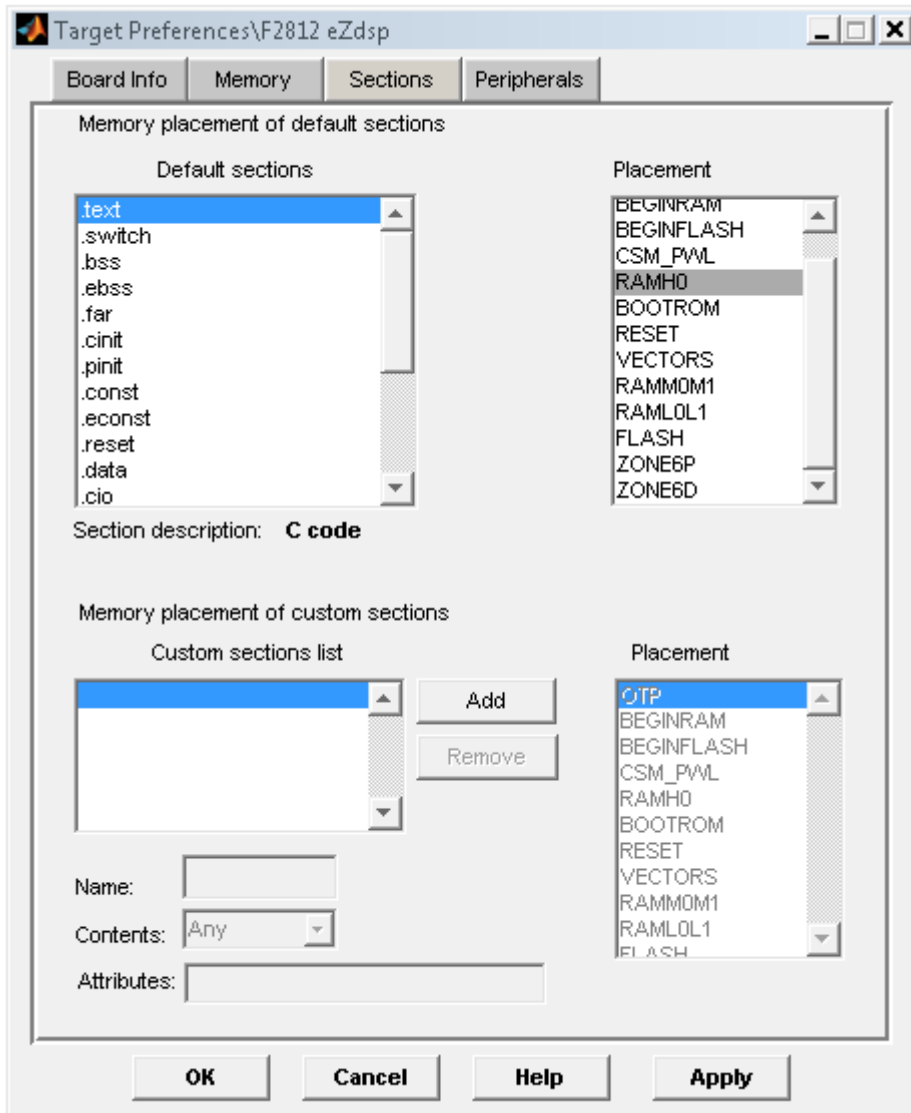
Alt modeller oluşturulduktan sonra birbirleriyle olan bağlantıları yapılır. Hız Ayar alt modelinin çıkışı ile Hız Kontrol alt modelinin referans girişi arasındaki bağlantı yapılır. Hız Ölçüm alt modelinin çıkışı ile de Hız Kontrol alt modelinin geri besleme girişi bağlantısı yapılarak model oluşturulur. Model oluşturulduktan sonra “Target Support Package TC2/C2000 Target Preferences” tan “F2812 eZdsp” bloğu eklenir ve Şekil 5.21’deki gibi model tamamlanır.



Şekil 5.21 Uygulama Modeli

Model tamamladıktan sonra DSP kartın hafıza haritasını düzenlemek için “F2812 eZdsp” bloğu çift tıklanılarak özellikler penceresi açılıp Şekil 5.22’deki gibi açılan pencerede aşağıdaki değişiklikler yapıp kaydedilir.

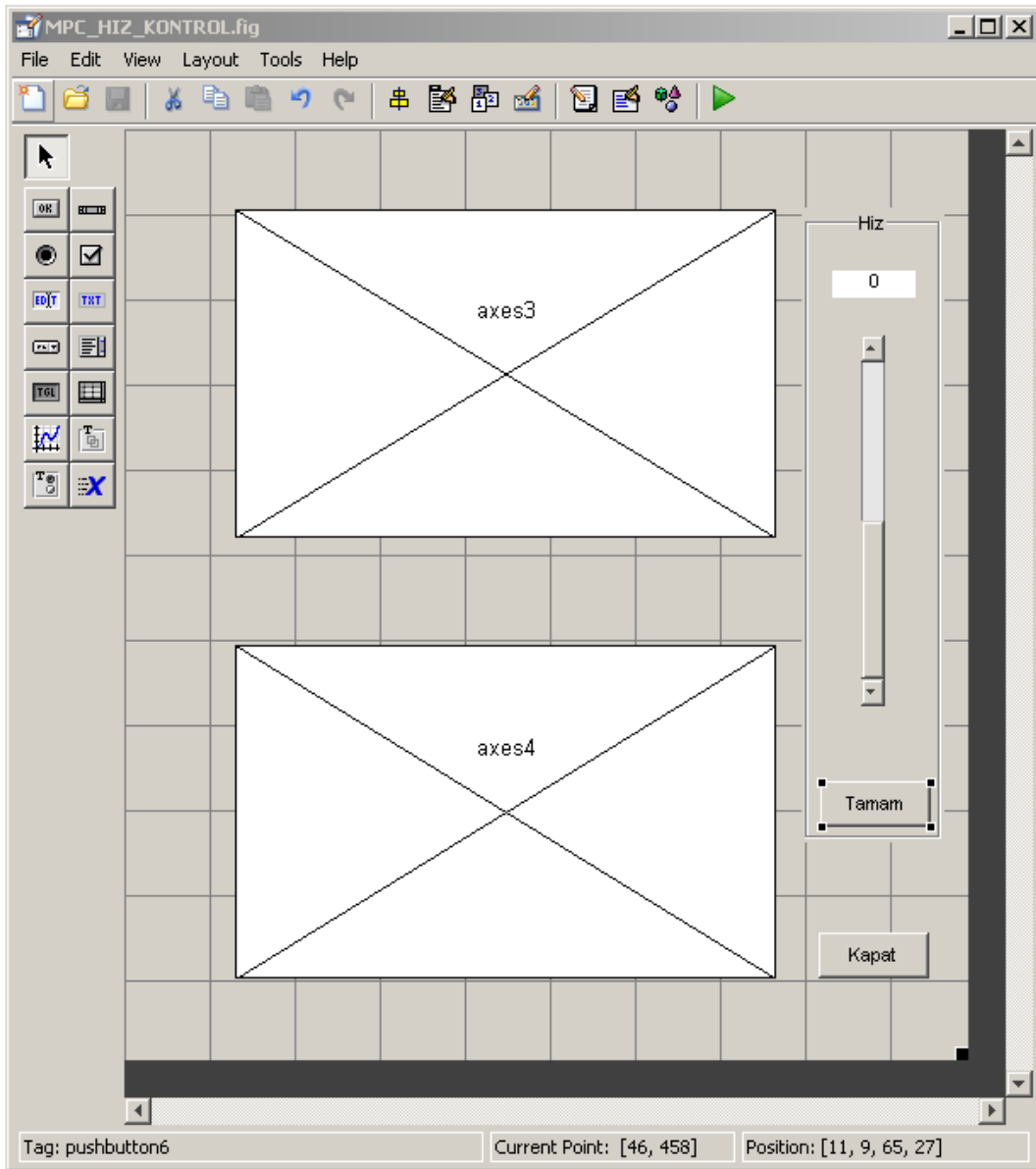
```
.text      ZONE6P
.data      ZONE6D
.rtdx_text ZONE6P
.rtdx_data ZONE6D
```



Şekil 5.22 F2812 eZdsp Özellikler Penceresi

## 5.6 Kullanıcı Arayüzü Tasarımı

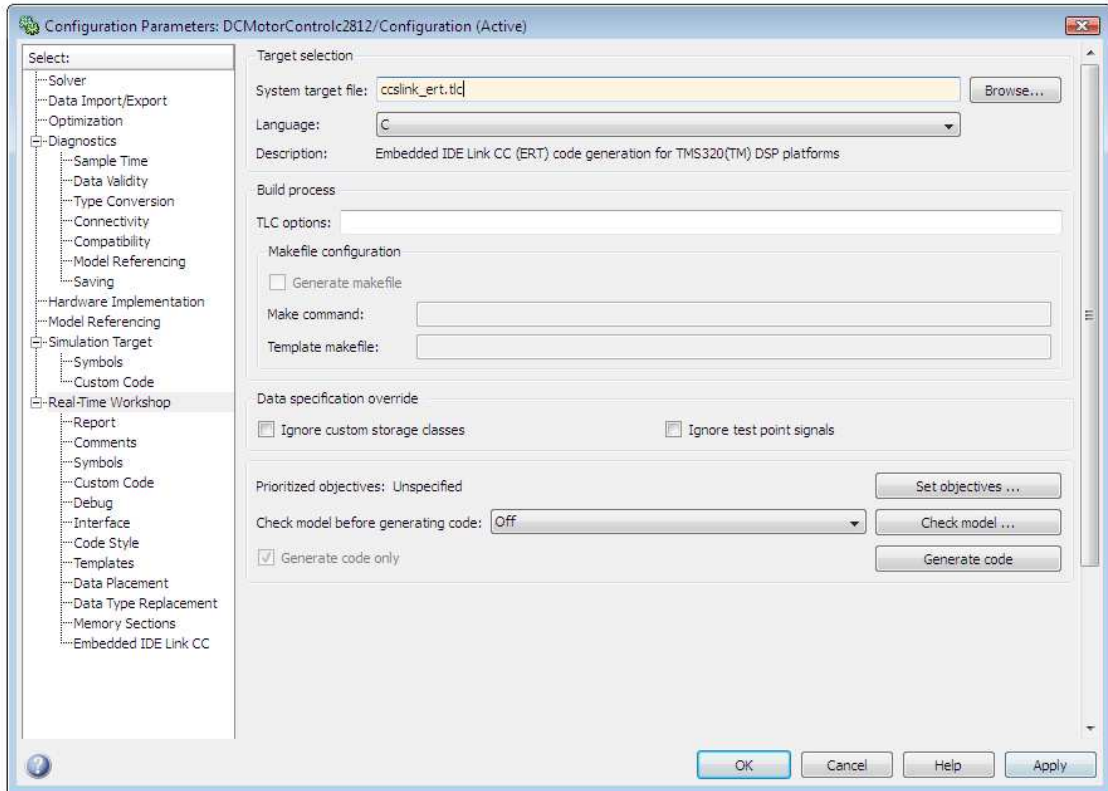
Şekil 5.23'te gösterildiği gibi kullanıcı arayüzü PWM ve motor hızını gösterecek iki grafikten ve hızı ayarlamak için bir sürgüden oluşmaktadır. DSP kartı ve bilgisayar arasındaki bağlantıları sağlayacak gerekli kodlamalar yapıldıktan sonra kullanıcı arayüzü tasarımı tamamlanır.



Şekil 5.23 Kullanıcı Arayüzü

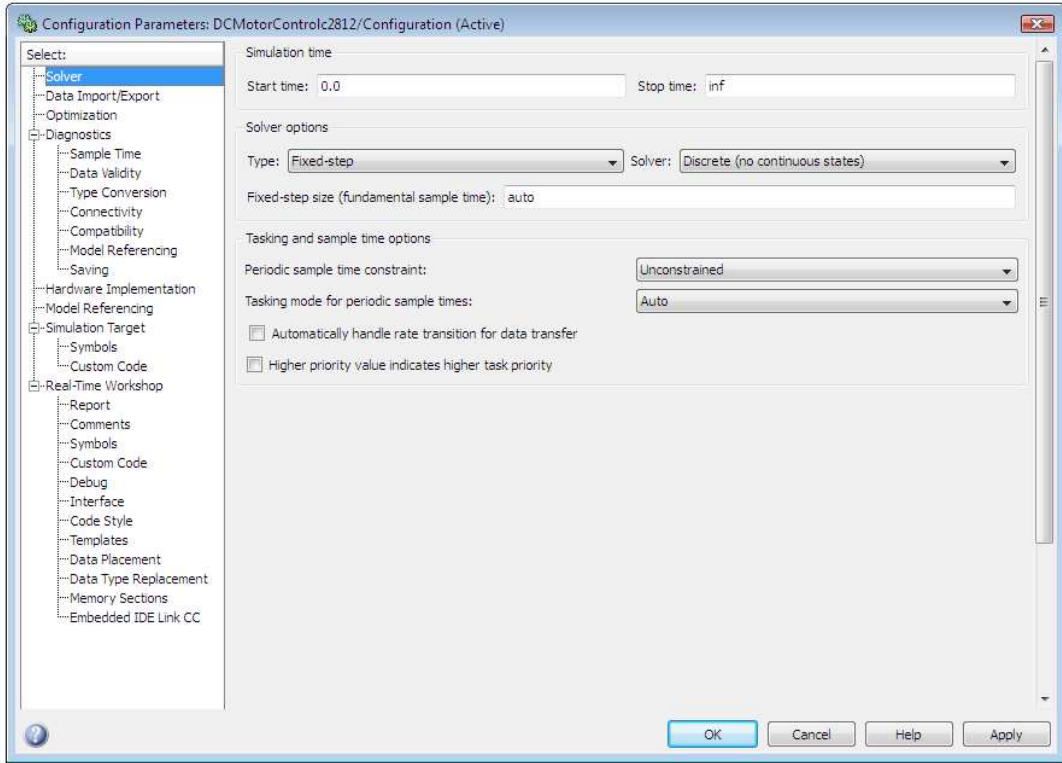
## 5.7 Gerçek Zamanlı Uygulama

Model oluşturulduktan sonra Simulink'te Tools menüsünden Real-Time Workshop - Options tıklanarak Şekil 5.24'teki özellikler penceresi açılır. Açılan pencerede “Target File” özelliği “TMS320 DSP” olarak seçilir.



Şekil 5.24 RTW Hedef Kart Seçimi

Sol menüden Solver'ı tıklanarak Şekil 5.25'teki gibi “Type” özelliği “Fixed-step” ve “Solver” özelliği “Discrete” olarak değiştirilir.



Şekil 5.25 Solver Özellikleri

Sol menüden Real-Time Workshop bölümünde “Interface” seçilip burada “Target Function Library” özelliği “TI C28X” olarak seçilip “Support” bölümünde “non-inlined-S-function” seçilerek Real-Time Workshop ayarları yapılır.

Real-Time Workshop üzerinde gerekli düzenlemeleri yapıldıktan sonra “Simulink > Tools > Real-Time Workshop > Build Model” tıklanılarak oluşturulmuş olan model Real-Time Workshop, Embedded Coder ve Target Language Coder yardımı ile TMS320F2812 DSP kartı için koda dönüştürülür. Üretilen bu kod Code Composer Studio programı ile TMS320F2812 DSP kartı arasında bağlantı sağlanıp yüklenerek Model Öngörülü hız kontrol denetimi DSP kart üzerinde gerçekleştirilmiş olur.

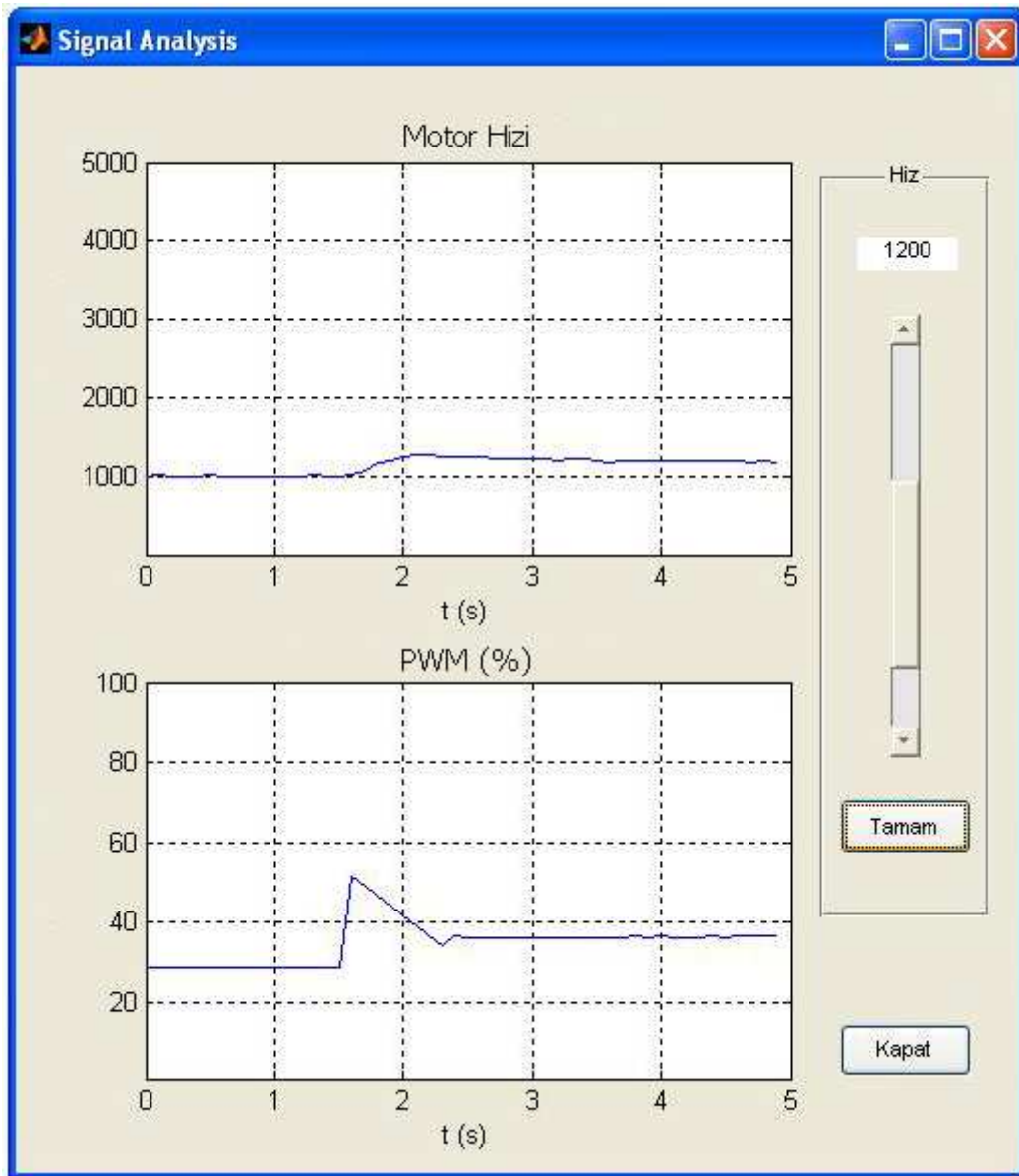


### 5.8 Uygulamamın Çalıştırılması

Oluşturulan program çalıştırıldığında kullanıcı arayüzü açılır. Bu arayüz de motor hızı ve pwm değerleri görülebilir ve slider kullanılarak motor hızı ayarlanabilir.

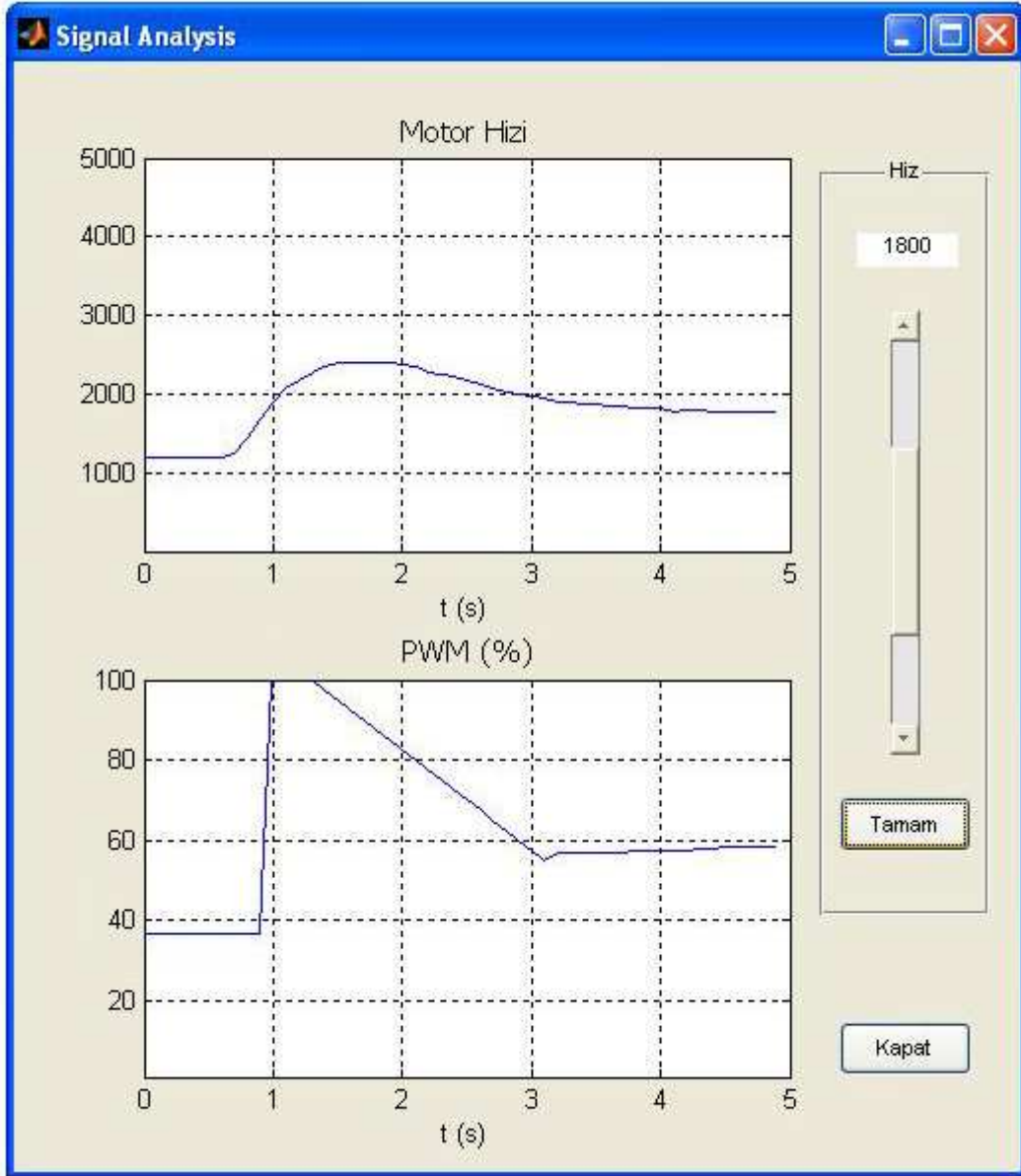
Aşağıda çeşitli hız değerleri için örnekler görülmektedir.

Şekil 5.26'da sonuçları görüldüğü gibi motor hızı 1000 rpm'den 1200 rpm'e çıkarılmıştır.



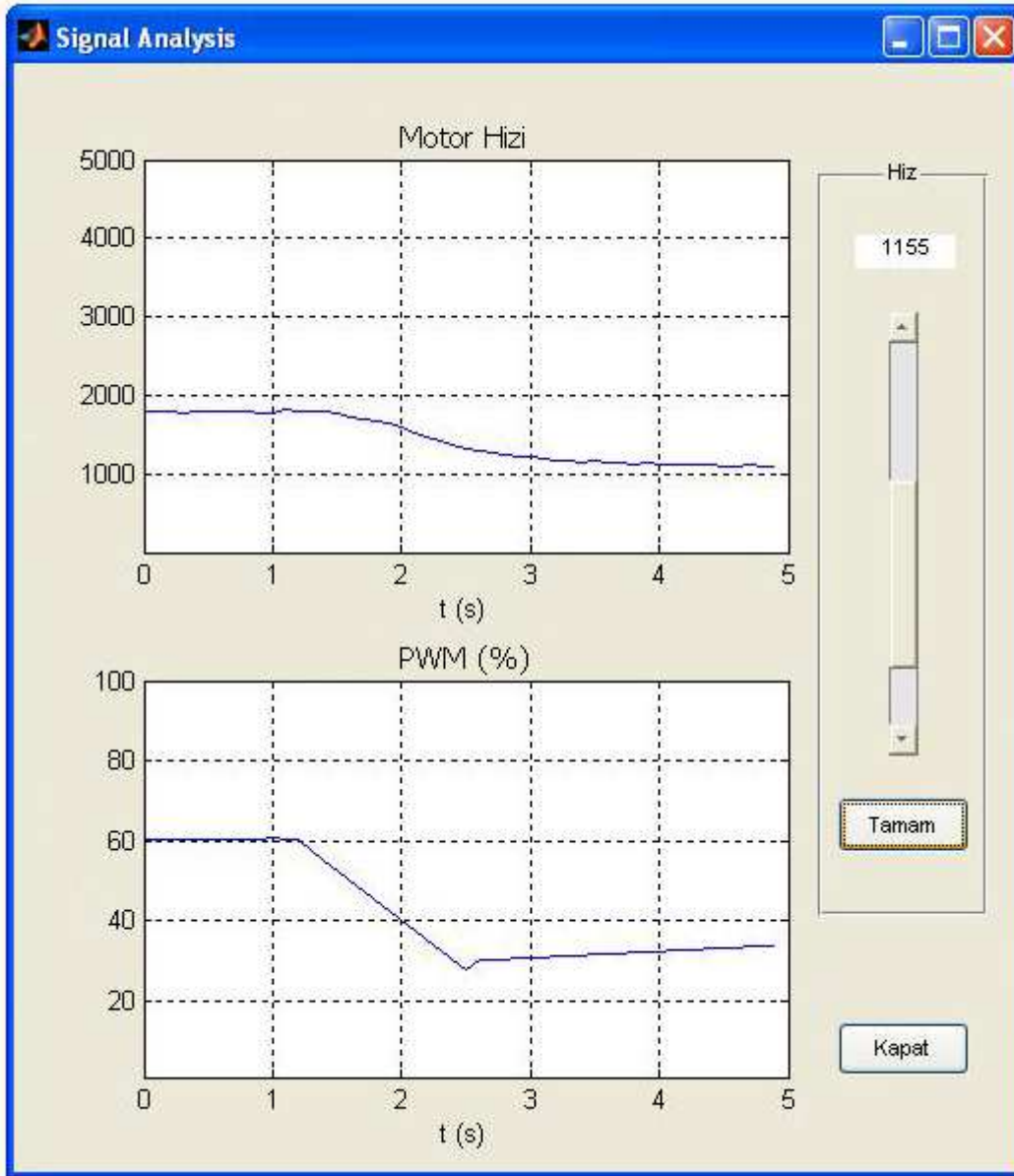
Şekil 5.26 1000 rpm'den 1200 rpm'e Yükselme Arayüzü

Şekil 5.27’de sonuçları görüldüğü gibi motor hızı 1200 rpm’den 1800 rpm’e çıkarılmıştır.



Şekil 5.27 1200 rpm’den 1800 rpm’e Yükselme Arayüzü

Şekil 5.28’de sonuçları görüldüğü gibi motor hızı 1800 rpm’den 1200 rpm’e düşürülmüştür.



Şekil 5.28 1800 rpm’den 1155 rpm’e Alçalma Arayüzü

## BÖLÜM 6

### SONUÇ VE TARTIŞMA

Yapılan çalışma Model Öngörülü Denetleyicinin Sayısal İşaretleyici Kart (DSP) kullanılarak gerçekleşmesidir. Bu amaçla bir hız kontrol uygulaması yapılmıştır.

Yapılan çalışmada öncelikle Model öngörülü Kontrol incelenmiş üstünlükleri ve eksik yönleri belirlenmiştir. Daha sonra hız kontrolü yapılacak sistem belirlenmiş süreç modeli çıkartılmış ve hız kontrol uygulaması için en uygun DSP kart seçilerek gerçek zamanlı kontrol edilebilen bir uygulama gerçekleştirilmiştir. Gerçekleştirilen bu uygulama model tabanlı yazılım geliştirme araçları kullanılarak esnek, tekrar kullanılabilir ve geliştirmeye açık olarak tasarlanmıştır. Tüm işlemleri izleyebilme olanağı sağlayan bu uygulamada kullanıcı arayüzü ile sürecin kontrolü gerçek zamanlı olarak sağlanmıştır.

## KAYNAKLAR DİZİNİ

- Bemporad A, Manfred Morari, N. Lawrence Ricker, 2005, Model Predictive Control Toolbox™ User's Guide, The MathWorks, Massachusetts
- Camacho, F.E, ve Bordons, C., 2004, Model Predictive Control, Springer, New York
- Demircioğlu, H., Gawthrop, P.J., 1991, Continuous-time Generalized Predictive Control (CGPC), Automatica
- Kahvecioğlu, S., Karamancıoğlu, A., Yazıcı, A., 2008, Enerji Kontrolüne Dayalı Doğrusal Olmayan Model Öngörümlü Denetim ile Ters Sarkacın Yukarı Kaldırılması, ELECO, Bursa
- Kelleci, E., 2009, Model Öngörümlü Kontrolün ARM Microişlemci ile Gerçeklenmesi, Eskişehir Osmangazi Üniversitesi Fen Bilimleri Ens. Yüksek Lisans Tezi, Eskişehir
- Larsson, V., 2008, Development of Real Time Test Platform for Motor Drive Algorithms, Lulea Teknoloji Üniversitesi Yüksek Lisans Tezi, İsveç
- Rawlings, J.B., Muske, K.R., 1993, Model Predictive Control With Linear-models, AIChE
- Spectrum Digital Inc. , 2004, eZdsp F2812 Technical Reference, Texas
- Texas Ins., 2007, TMS320F2812 Digital Signal Processors Data Manual
- The MathWorks, 2003, Target Support Package TC2 User's Guide, The MathWorks Inc. , Massachusetts
- Yüksel, İ., 2004, Matlab ile Mühendislik Sistemlerinin Analizi ve Çözümü, Nobel Yayın Dağıtım, Ankara
- Wang, L., 2009, Model Predictive Control System Design and Implementation Using MATLAB, Springer, New York

**EKLER**

Ek.1 Model Öngörülü Kontrol Yazılımı

**Ek.1. Model Öngörülü Kontrol Yazılımı**

```
/*  
 * File: MPCMotorControl_main.c  
 */  
  
#include "MPCMotorControl.h"  
  
#include "rtwtypes.h"  
  
#include "rt_nonfinite.h"  
  
#include "MPCMotorControl_private.h"  
  
#include "c2000_main.h"  
  
#include "DSP281x_Device.h"  
  
#include "DSP281x_Examples.h"  
  
#include <stdlib.h>  
  
#include <stdio.h>  
  
void init_board(void);  
  
void enable_interrupts(void);  
  
void config_schedulerTimer(void);  
  
void disable_interrupts(void);  
  
volatile int IsrOverrun = 0;  
  
boolean_T isRateRunning[2] = { 0, 0 };  
  
boolean_T need2runFlags[2] = { 0, 0 };  
  
void rt_OneStep(void)  
{  
    boolean_T eventFlags[2];  
  
    asm(" SETC INTM");
```

```
if (isRateRunning[0]++) {  
    IsrOverrun = 1;  
    isRateRunning[0]--;  
    asm(" CLRC INTM");  
    return;  
}  
MPCMotorControl_SetEventsForThisBaseStep(eventFlags);  
asm(" CLRC INTM");  
MPCMotorControl_step(0);  
isRateRunning[0]--;  
if (eventFlags[1]) {  
    if (need2runFlags[1]++) {  
        IsrOverrun = 1;  
        need2runFlags[1]--;  
        return;  
    }  
}  
if (need2runFlags[1]) {  
    if (isRateRunning[1]) {  
        return;  
    }  
    asm(" SETC INTM");  
    isRateRunning[1]++;
```



```
    asm(" CLRC INTM");

MPCMotorControl_step(1);

    need2runFlags[1]--;

    isRateRunning[1]--;

}

}

void main(void)

{

    volatile boolean_T noErr;

    init_board();

    MPCMotorControl_initialize(1);

    config_schedulerTimer();

    noErr =

        (rtmGetErrorStatus(MPCMotorControl_M) == (NULL)) && !rtmGetStopRequested

        (MPCMotorControl_M);

    enable_interrupts();

    while (noErr ) {

        noErr =

            (rtmGetErrorStatus(MPCMotorControl_M) == (NULL)) &&

            !rtmGetStopRequested(MPCMotorControl_M);

    }

    MPCMotorControl_terminate();

    disable_interrupts();

}
```

```

/*
 * File: MPCMotorControl.c
 */

#include "MPCMotorControl.h"

#include "MPCMotorControl_private.h"

RTDX_CreateInputChannel(ichan1);

RTDX_CreateOutputChannel(ochan2);

RTDX_CreateOutputChannel(ochan1);

BlockIO_MPCMotorControl MPCMotorControl_B;

D_Work_MPCMotorControl MPCMotorControl_DWork;

RT_MODEL_MPCMotorControl MPCMotorControl_M_;

RT_MODEL_MPCMotorControl *MPCMotorControl_M = &MPCMotorControl_M_;

static void rate_monotonic_scheduler(void);

real_T MPCMotorControl_RGND = 0.0;

void MPCMotorControl_SetEventsForThisBaseStep(boolean_T *eventFlags)
{
    eventFlags[1] = ((boolean_T)rtmStepTask(MPCMotorControl_M, 1));
}

static void rate_monotonic_scheduler(void)
{
    MPCMotorControl_M->Timing.RateInteraction.TID0_1 =
        (MPCMotorControl_M->Timing.TaskCounters.TID[1] == 0);

    MPCMotorControl_M->Timing.perTaskSampleHits[1] =

```

```

MPCMotorControl_M->Timing.RateInteraction.TID0_1;

if (++MPCMotorControl_M->Timing.TaskCounters.TID[1] == 10) {

    MPCMotorControl_M->Timing.TaskCounters.TID[1] = 0;

}

}

void Set_Speed_Start(void)

{

    {

        RTDX_enableInput(&ichan1);

        MPCMotorControl_B.FromRTDX = (int32_T)

        MPCMotorControl_ConstP.FromRTDX_IC;

    }

}

void Set_Speed(void)

{

    RTDX_readNB( &ichan1, (void*) &MPCMotorControl_B.FromRTDX, 1*sizeof

        (int32_T));

    MPCMotorControl_B.ref = (real_T)MPCMotorControl_B.FromRTDX;

}

void Set_Speed_Term(void)

{

    RTDX_disableInput(&ichan1);

}

void Speed_Correction_Init(void)

```

```

{
    {
        SimStruct *rts = MPCMotorControl_M->childSfunctions[0];

        sfcnInitializeConditions(rts);

        if (ssGetErrorStatus(rts) != (NULL))

            return;
    }

    MPCMotorControl_DWork.Downsampling1_Count = 0L;
    MPCMotorControl_DWork.Downsampling1_Buffer = 0L;
    {
        const int_T bufLenBytes = 49 * sizeof(int32_T);

        byte_T *circBufPtr = (byte_T *)&MPCMotorControl_DWork.Buffer_CircBuff_n

            [0];

        const byte_T *icPtr = (const byte_T *)&MPCMotorControl_ConstP.pooled1;

        int_T i = 1;

        while (i-- > 0) {

            MWDSP_CopyScalarICs(circBufPtr, icPtr, 49, sizeof(int32_T));

            circBufPtr += bufLenBytes;

        }

    }

    *&MPCMotorControl_DWork.Buffer_IN_BUF_PTR_m = (void *) ( (byte_T *)

        &MPCMotorControl_DWork.Buffer_CircBuff_n[0] );

}

```

```

void Speed_Correction_Start(void){

    EALLOW;

    GpioMuxRegs.GPAMUX.all |= 12U;

    config_PWM_A (2048.0,1,0,"INPUT_PORT",0.0,

        1,"INPUT_PORT",0.0,0,"SPECIFY_VIA_DIALOG",0.0,1637,

        0, 0, 0, 4, 12, 2);

    EDIS;

    RTDX_enableOutput(&ochan2);

}

void Speed_CorrectionTID0(void)

{

    uint16_T rtb_DataTypeConversion1;

    {

        real_T tmp;

        uint16_T tmp_0;

        int32_T tmp_1;

        {

            SimStruct *rts = MPCMotorControl_M->childSfunctions[0];

            sfcnOutputs(rts, 0);

        }

        tmp = floor(MPCMotorControl_B.sfunction + 0.5);

        if (tmp < 65536.0) {

            if (tmp >= 0.0) {

                tmp_0 = (uint16_T)tmp;

```

```

    } else {
tmp_0 = 0U;
    }
    } else {
        tmp_0 = MAX_uint16_T;
    }
    rtb_DataTypeConversion1 = tmp_0;
    {
        EvaRegs.CMPR2 = (uint16_T) (rtb_DataTypeConversion1 / 2);
    }
    tmp = floor(MPCMotorControl_B.sfunction + 0.5);
    if (tmp < 2.147483648E+009) {
        if (tmp >= -2.147483648E+009) {
            tmp_1 = (int32_T)tmp;
        } else {
            tmp_1 = MIN_int32_T;
        }
    } else {
        tmp_1 = MAX_int32_T;
    }
    MPCMotorControl_B.DataTypeConversion_c = tmp_1;
    if (MPCMotorControl_M->Timing.RateInteraction.TID0_1) {
        MPCMotorControl_B.Downsampling1 =
            MPCMotorControl_B.DataTypeConversion_c;
    }
}

```

```

    }
}
}
void Speed_CorrectionTID1(void)
{
    {
        const byte_T *u = (const byte_T *)&MPCMotorControl_B.Downsamp1;
        byte_T *y = (byte_T *)MPCMotorControl_B.Buffer_m;
        byte_T *mem = (byte_T *)&MPCMotorControl_DWork.Buffer_CircBuff_n[0];
        int_T uWidth = 1 * sizeof(int32_T);
        int_T yWidth = 50 * sizeof(int32_T);
        int_T memWidth = yWidth - uWidth;
        int_T bpeTimesNSampsAtBot;
        byte_T* bufPtr = (byte_T *)MPCMotorControl_DWork.Buffer_IN_BUF_PTR_m;
        byte_T* topBuf = mem;
        byte_T* endBuf = topBuf + memWidth;
        bpeTimesNSampsAtBot = (endBuf - bufPtr);
        memcpy(y, bufPtr, bpeTimesNSampsAtBot);
        memcpy(y + bpeTimesNSampsAtBot, topBuf, (memWidth-
bpeTimesNSampsAtBot));
        memcpy((y+memWidth), u, uWidth);
        memcpy(bufPtr, u, uWidth);
        bufPtr += uWidth;
        if (bufPtr >= endBuf) {

```

```

    bufPtr -= memWidth;

}

MPCMotorControl_DWork.Buffer_IN_BUF_PTR_m = bufPtr;

}

if (RTDX_isOutputEnabled( &ochan2 )) {

    while (RTDX_writing != NULL) {

        }

        RTDX_write( &ochan2, (void*) MPCMotorControl_B.Buffer_m, 50*sizeof
            (int32_T));

        }

    }

void Speed_Correction_UpdateTID0(void)

{

    {

        SimStruct *rts = MPCMotorControl_M->childSfunctions[0];

        sfcnUpdate(rts, 0);

        if (ssGetErrorStatus(rts) != (NULL))

            return;

        }

    }

void Speed_Correction_Term(void)

{

    {

        SimStruct *rts = MPCMotorControl_M->childSfunctions[0];

```



```

    sfcnTerminate(rts); }

RTDX_disableOutput(&ochan2);

}

void Measure_Speed_Init(void)

{

MPCMotorControl_DWork.Downsampling_Count = 0L;

MPCMotorControl_DWork.Downsampling_Buffer = 0L;

{

    const int_T bufLenBytes = 49 * sizeof(int32_T);

    byte_T *circBufPtr = (byte_T *)&MPCMotorControl_DWork.Buffer_CircBuff[0];

    const byte_T *icPtr = (const byte_T *)&MPCMotorControl_ConstP.pooled1;

    int_T i = 1;

    while (i-- > 0)

        {

            MWDSP_CopyScalarICs(circBufPtr, icPtr, 49, sizeof(int32_T));

            circBufPtr += bufLenBytes;

        }

}

*&MPCMotorControl_DWork.Buffer_IN_BUF_PTR = (void *)((byte_T *)

    &MPCMotorControl_DWork.Buffer_CircBuff[0] );

}

void Measure_Speed_Start(void)

{

    InitAdc();

```

```

    config_ADC_A (0U, 0U, 0U, 0U, 0U);

RTDX_enableOutput(&ochan1);

}

void Measure_SpeedTID0(void)

{

    {

        MPCMotorControl_B.ADC = (AdcRegs.ADCRESULT0) >> 4;

        AdcRegs.ADCTRL2.bit.RST_SEQ1 = 0x1;

        AdcRegs.ADCST.bit.INT_SEQ1_CLR = 1;

    }

    if (MPCMotorControl_M->Timing.RateInteraction.TID0_1) {

        MPCMotorControl_B.Downsampling = MPCMotorControl_B.ADC;

    }

    MPCMotorControl_B.DataTypeConversion = (real_T)MPCMotorControl_B.ADC;

}

void Measure_SpeedTID1(void)

{

    {

        const byte_T *u = (const byte_T *)&MPCMotorControl_B.Downsampling;

        byte_T *y = (byte_T *)&MPCMotorControl_B.Buffer;

        byte_T *mem = (byte_T *)&MPCMotorControl_DWork.Buffer_CircBuff[0];

        int_T uWidth = 1 * sizeof(int32_T);

        int_T yWidth = 50 * sizeof(int32_T);

        int_T memWidth = yWidth - uWidth;

```

```

int_T bpeTimesNSampsAtBot;

byte_T* bufPtr = (byte_T *)MPCMotorControl_DWork.Buffer_IN_BUF_PTR;

byte_T* topBuf = mem;

byte_T* endBuf = topBuf + memWidth;

bpeTimesNSampsAtBot = (endBuf - bufPtr);

memcpy(y, bufPtr, bpeTimesNSampsAtBot);

memcpy(y + bpeTimesNSampsAtBot, topBuf, (memWidth-
bpeTimesNSampsAtBot));

memcpy((y+memWidth), u, uWidth);

memcpy(bufPtr, u, uWidth);

bufPtr += uWidth;

if (bufPtr >= endBuf) {

    bufPtr -= memWidth;

}

MPCMotorControl_DWork.Buffer_IN_BUF_PTR = bufPtr;

}

if (RTDX_isOutputEnabled( &ochan1 )) {

    while (RTDX_writing != NULL) {

    }

    RTDX_write( &ochan1, (void*) MPCMotorControl_B.Buffer, 50*sizeof(int32_T));

}

}

void Measure_Speed_Term(void)

{

```

```

RTDX_disableOutput(&ochan1);  }

void MPCMotorControl_step0(void)
{
    {
        rate_monotonic_scheduler();
    }
    Set_Speed();
    Measure_SpeedTID0();
    Speed_CorrectionTID0();
    Speed_Correction_UpdateTID0();
    MPCMotorControl_M->Timing.t[0] =
        (++)MPCMotorControl_M->Timing.clockTick0) *
        MPCMotorControl_M->Timing.stepSize0;
}

void MPCMotorControl_step1(void)
{
    Measure_SpeedTID1();
    Speed_CorrectionTID1();
}

void MPCMotorControl_step(int_T tid)
{
    switch (tid) {
        case 0 :
            MPCMotorControl_step0();

```

```

    break;

case 1 :

    MPCMotorControl_step1();

    break;

default :

    break;

}

}

void MPCMotorControl_initialize(boolean_T firstTime)
{
    (void)firstTime;

    rt_InitInfAndNaN(sizeof(real_T));

    (void) memset((void *)MPCMotorControl_M,0,
                 sizeof(RT_MODEL_MPCMotorControl));

    rtsiSetSolverName(&MPCMotorControl_M->solverInfo,"FixedStepDiscrete");
    MPCMotorControl_M->solverInfoPtr = (&MPCMotorControl_M->solverInfo);
    {
        int_T *mdlTsMap = MPCMotorControl_M->Timing.sampleTimeTaskIDArray;
        mdlTsMap[0] = 0;
        mdlTsMap[1] = 1;
        MPCMotorControl_M->Timing.sampleTimeTaskIDPtr = (&mdlTsMap[0]);
        MPCMotorControl_M->Timing.sampleTimes =
            (&MPCMotorControl_M->Timing.sampleTimesArray[0]);
        MPCMotorControl_M->Timing.offsetTimes =

```

```

    (&MPCMotorControl_M->Timing.offsetTimesArray[0]);
MPCMotorControl_M->Timing.sampleTimes[0] = (0.01);
MPCMotorControl_M->Timing.sampleTimes[1] = (0.1);
MPCMotorControl_M->Timing.offsetTimes[0] = (0.0);
MPCMotorControl_M->Timing.offsetTimes[1] = (0.0);
}
rtmSetTPtr(MPCMotorControl_M, &MPCMotorControl_M->Timing.tArray[0]);
{
    int_T *mdlSampleHits = MPCMotorControl_M->Timing.sampleHitArray;
    int_T *mdlPerTaskSampleHits =
        MPCMotorControl_M->Timing.perTaskSampleHitsArray;
    MPCMotorControl_M->Timing.perTaskSampleHits =
(&mdlPerTaskSampleHits[0]);
    mdlSampleHits[0] = 1;
    MPCMotorControl_M->Timing.sampleHits = (&mdlSampleHits[0]);
}
rtmSetTFinal(MPCMotorControl_M, -1);
MPCMotorControl_M->Timing.stepSize0 = 0.01;
MPCMotorControl_M->solverInfoPtr = (&MPCMotorControl_M->solverInfo);
MPCMotorControl_M->Timing.stepSize = (0.01);
rtsiSetFixedStepSize(&MPCMotorControl_M->solverInfo, 0.01);
rtsiSetSolverMode(&MPCMotorControl_M->solverInfo,
SOLVER_MODE_MULTITASKING);
(void) memset(((void *) &MPCMotorControl_B),0,

```

```

        sizeof(BlockIO_MPCMotorControl));

{
    MPCMotorControl_B.DataTypeConversion = 0.0;
    MPCMotorControl_B.sfunction = 0.0;
    MPCMotorControl_B.ref = 0.0;
}

(void) memset((void *)&MPCMotorControl_DWork, 0,
              sizeof(D_Work_MPCMotorControl));

MPCMotorControl_DWork.sfunction_DSTATE[0] = 0.0;
MPCMotorControl_DWork.sfunction_DSTATE[1] = 0.0;
MPCMotorControl_DWork.sfunction_DSTATE[2] = 0.0;

{
    RTWSfcnInfo *sfcnInfo = &MPCMotorControl_M->NonInlinedSFens.sfcnInfo;
    MPCMotorControl_M->sfcnInfo = (sfcnInfo);

    rtssSetErrorStatusPtr(sfcnInfo, (&rtmGetErrorStatus(MPCMotorControl_M)));
    rtssSetNumRootSampTimesPtr(sfcnInfo,
                                &MPCMotorControl_M->Sizes.numSampTimes);

    rtssSetTPtrPtr(sfcnInfo, &rtmGetTPtr(MPCMotorControl_M));
    rtssSetTStartPtr(sfcnInfo, &rtmGetTStart(MPCMotorControl_M));
    rtssSetTFinalPtr(sfcnInfo, &rtmGetTFinal(MPCMotorControl_M));
    rtssSetTimeOfLastOutputPtr(sfcnInfo, &rtmGetTimeOfLastOutput
                                (MPCMotorControl_M));

    rtssSetStepSizePtr(sfcnInfo, &MPCMotorControl_M->Timing.stepSize);
    rtssSetStopRequestedPtr(sfcnInfo, &rtmGetStopRequested(MPCMotorControl_M));
}

```

```

rtssSetDerivCacheNeedsResetPtr(sfcnInfo,
&MPCMotorControl_M->ModelData.derivCacheNeedsReset);

rtssSetZCCacheNeedsResetPtr(sfcnInfo,
    &MPCMotorControl_M->ModelData.zCCacheNeedsReset);

rtssSetBlkStateChangePtr(sfcnInfo,
    &MPCMotorControl_M->ModelData.blkStateChange);

rtssSetSampleHitsPtr(sfcnInfo, &MPCMotorControl_M->Timing.sampleHits);

rtssSetPerTaskSampleHitsPtr(sfcnInfo,
    &MPCMotorControl_M->Timing.perTaskSampleHits);

rtssSetSimModePtr(sfcnInfo, &MPCMotorControl_M->simMode);

rtssSetSolverInfoPtr(sfcnInfo, &MPCMotorControl_M->solverInfoPtr);
}

MPCMotorControl_M->Sizes.numSFcns = (1);

{
    (void) memset((void *)
        &MPCMotorControl_M->NonInlinedSFcns.childSFunctions[0],0,
        1*sizeof(SimStruct));

MPCMotorControl_M->childSfunctions =
    (&MPCMotorControl_M->NonInlinedSFcns.childSFunctionPtrs[0]);

MPCMotorControl_M->childSfunctions[0] =
    (&MPCMotorControl_M->NonInlinedSFcns.childSFunctions[0]);
{
    SimStruct *rts = MPCMotorControl_M->childSfunctions[0];

    time_T *sfcnPeriod =

```





```

_ssSetNumInputPorts(rts, 9);

ssSetPortInfoForInputs(rts,

    &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.inputPortInfo[0]);

{

    real_T const **sfcnUPtrs = (real_T const **)

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs0;

    sfcnUPtrs[0] = &MPCMotorControl_B.DataTypeConversion;

    ssSetInputPortSignalPtrs(rts, 0, (InputPtrsType)&sfcnUPtrs[0]);

    _ssSetInputPortNumDimensions(rts, 0, 1);

    ssSetInputPortWidth(rts, 0, 1);

}

{

    real_T const **sfcnUPtrs = (real_T const **)

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs1;

    sfcnUPtrs[0] = &MPCMotorControl_B.ref;

    ssSetInputPortSignalPtrs(rts, 1, (InputPtrsType)&sfcnUPtrs[0]);

    _ssSetInputPortNumDimensions(rts, 1, 1);

    ssSetInputPortWidth(rts, 1, 1);

}

{

    real_T const **sfcnUPtrs = (real_T const **)

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs2;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 2, (InputPtrsType)&sfcnUPtrs[0]);

```

```

    _ssSetInputPortNumDimensions(rts, 2, 1);

    ssSetInputPortWidth(rts, 2, 1);

}

{

    real_T const **sfcnUPtrs = (real_T const **)

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs3;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 3, (InputPtrsType)&sfcnUPtrs[0]);

    _ssSetInputPortNumDimensions(rts, 3, 1);

    ssSetInputPortWidth(rts, 3, 1);

}

{

    real_T const **sfcnUPtrs = (real_T const **)

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs4;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 4, (InputPtrsType)&sfcnUPtrs[0]);

    _ssSetInputPortNumDimensions(rts, 4, 1);

    ssSetInputPortWidth(rts, 4, 1);

}

{

    real_T const **sfcnUPtrs = (real_T const **)

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs5;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 5, (InputPtrsType)&sfcnUPtrs[0]);

```

```

    _ssSetInputPortNumDimensions(rts, 5, 1);

    ssSetInputPortWidth(rts, 5, 1);
}

{
    real_T const **sfcnUPtrs = (real_T const **)
        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs6;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 6, (InputPtrsType)&sfcnUPtrs[0]);

    _ssSetInputPortNumDimensions(rts, 6, 1);

    ssSetInputPortWidth(rts, 6, 1);
}

{
    real_T const **sfcnUPtrs = (real_T const **)
        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs7;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 7, (InputPtrsType)&sfcnUPtrs[0]);

    _ssSetInputPortNumDimensions(rts, 7, 1);

    ssSetInputPortWidth(rts, 7, 1);
}

{
    real_T const **sfcnUPtrs = (real_T const **)
        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.UPtrs8;

    sfcnUPtrs[0] = &MPCMotorControl_RGND;

    ssSetInputPortSignalPtrs(rts, 8, (InputPtrsType)&sfcnUPtrs[0]);
}

```

```

    _ssSetInputPortNumDimensions(rts, 8, 1);

    ssSetInputPortWidth(rts, 8, 1);

}

}

{

    ssSetPortInfoForOutputs(rts,

        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.outputPortInfo[0]);

    _ssSetNumOutputPorts(rts, 1);

    {

        _ssSetOutputPortNumDimensions(rts, 0, 1);

        ssSetOutputPortWidth(rts, 0, 1);

        ssSetOutputPortSignal(rts, 0, ((real_T *)

            &MPCMotorControl_B.sfunction));

    }

}

ssSetDiscStates(rts, (real_T *)

    &MPCMotorControl_DWork.sfunction_DSTATE[0]);

ssSetModelName(rts, "sfunction");

ssSetPath(rts, "MPCMotorControl/Hiz Kontrol/MPC Controller/sfunction");

ssSetRTModel(rts, MPCMotorControl_M);

ssSetParentSS(rts, (NULL));

ssSetRootSS(rts, rts);

ssSetVersion(rts, SIMSTRUCT_VERSION_LEVEL2);

{

```

```
mxArray **sfcnParams = (mxArray **)
    &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.params;
ssSetSFcnParamsCount(rts, 65);
ssSetSFcnParamsPtr(rts, &sfcnParams[0]);
ssSetSFcnParam(rts, 0, (mxArray*)
    &MPCMotorControl_P.sfunction_P1_Size[0]);
ssSetSFcnParam(rts, 1, (mxArray*)
    &MPCMotorControl_P.sfunction_P2_Size[0]);
ssSetSFcnParam(rts, 2, (mxArray*)
    &MPCMotorControl_P.sfunction_P3_Size[0]);
ssSetSFcnParam(rts, 3, (mxArray*)
    &MPCMotorControl_P.sfunction_P4_Size[0]);
ssSetSFcnParam(rts, 4, (mxArray*)
    &MPCMotorControl_P.sfunction_P5_Size[0]);
ssSetSFcnParam(rts, 5, (mxArray*)
    &MPCMotorControl_P.sfunction_P6_Size[0]);
ssSetSFcnParam(rts, 6, (mxArray*)
    &MPCMotorControl_P.sfunction_P7_Size[0]);
ssSetSFcnParam(rts, 7, (mxArray*)
    &MPCMotorControl_P.sfunction_P8_Size[0]);
ssSetSFcnParam(rts, 8, (mxArray*)
    &MPCMotorControl_P.sfunction_P9_Size[0]);
ssSetSFcnParam(rts, 9, (mxArray*)
    &MPCMotorControl_P.sfunction_P10_Size[0]);
```

```
ssSetSFcnParam(rts, 10, (mxArray*)
    &MPCMotorControl_P.sfunction_P11_Size[0]);
ssSetSFcnParam(rts, 11, (mxArray*)
    &MPCMotorControl_P.sfunction_P12_Size[0]);
ssSetSFcnParam(rts, 12, (mxArray*)
    &MPCMotorControl_P.sfunction_P13_Size[0]);
ssSetSFcnParam(rts, 13, (mxArray*)
    &MPCMotorControl_P.sfunction_P14_Size[0]);
ssSetSFcnParam(rts, 14, (mxArray*)
    &MPCMotorControl_P.sfunction_P15_Size[0]);
ssSetSFcnParam(rts, 15, (mxArray*)
    &MPCMotorControl_P.sfunction_P16_Size[0]);
ssSetSFcnParam(rts, 16, (mxArray*)
    &MPCMotorControl_P.sfunction_P17_Size[0]);
ssSetSFcnParam(rts, 17, (mxArray*)
    &MPCMotorControl_P.sfunction_P18_Size[0]);
ssSetSFcnParam(rts, 18, (mxArray*)
    &MPCMotorControl_P.sfunction_P19_Size[0]);
ssSetSFcnParam(rts, 19, (mxArray*)
    &MPCMotorControl_P.sfunction_P20_Size[0]);
ssSetSFcnParam(rts, 20, (mxArray*)
    &MPCMotorControl_P.sfunction_P21_Size[0]);
ssSetSFcnParam(rts, 21, (mxArray*)
    &MPCMotorControl_P.sfunction_P22_Size[0]);
```

```
ssSetSFcnParam(rts, 22, (mxArray*)
    &MPCMotorControl_P.sfunction_P23_Size[0]);
ssSetSFcnParam(rts, 23, (mxArray*)
    &MPCMotorControl_P.sfunction_P24_Size[0]);
ssSetSFcnParam(rts, 24, (mxArray*)
    &MPCMotorControl_P.sfunction_P25_Size[0]);
ssSetSFcnParam(rts, 25, (mxArray*)
    &MPCMotorControl_P.sfunction_P26_Size[0]);
ssSetSFcnParam(rts, 26, (mxArray*)
    &MPCMotorControl_P.sfunction_P27_Size[0]);
ssSetSFcnParam(rts, 27, (mxArray*)
    &MPCMotorControl_P.sfunction_P28_Size[0]);
ssSetSFcnParam(rts, 28, (mxArray*)
    &MPCMotorControl_P.sfunction_P29_Size[0]);
ssSetSFcnParam(rts, 29, (mxArray*)
    &MPCMotorControl_P.sfunction_P30_Size[0]);
ssSetSFcnParam(rts, 30, (mxArray*)
    &MPCMotorControl_P.sfunction_P31_Size[0]);
ssSetSFcnParam(rts, 31, (mxArray*)
    &MPCMotorControl_P.sfunction_P32_Size[0]);
ssSetSFcnParam(rts, 32, (mxArray*)
    &MPCMotorControl_P.sfunction_P33_Size[0]);
ssSetSFcnParam(rts, 33, (mxArray*)
    &MPCMotorControl_P.sfunction_P34_Size[0]);
```



```
ssSetSFcnParam(rts, 34, (mxArray*)
    &MPCMotorControl_P.sfunction_P35_Size[0]);
ssSetSFcnParam(rts, 35, (mxArray*)
    &MPCMotorControl_P.sfunction_P36_Size[0]);
ssSetSFcnParam(rts, 36, (mxArray*)
    &MPCMotorControl_P.sfunction_P37_Size[0]);
ssSetSFcnParam(rts, 37, (mxArray*)
    &MPCMotorControl_P.sfunction_P38_Size[0]);
ssSetSFcnParam(rts, 38, (mxArray*)
    &MPCMotorControl_P.sfunction_P39_Size[0]);
ssSetSFcnParam(rts, 39, (mxArray*)
    &MPCMotorControl_P.sfunction_P40_Size[0]);
ssSetSFcnParam(rts, 40, (mxArray*)
    &MPCMotorControl_P.sfunction_P41_Size[0]);
ssSetSFcnParam(rts, 41, (mxArray*)
    &MPCMotorControl_P.sfunction_P42_Size[0]);
ssSetSFcnParam(rts, 42, (mxArray*)
    &MPCMotorControl_P.sfunction_P43_Size[0]);
ssSetSFcnParam(rts, 43, (mxArray*)
    &MPCMotorControl_P.sfunction_P44_Size[0]);
ssSetSFcnParam(rts, 44, (mxArray*)
    &MPCMotorControl_P.sfunction_P45_Size[0]);
ssSetSFcnParam(rts, 45, (mxArray*)
    &MPCMotorControl_P.sfunction_P46_Size[0]);
```

```
ssSetSFcnParam(rts, 46, (mxArray*)
    &MPCMotorControl_P.sfunction_P47_Size[0]);
ssSetSFcnParam(rts, 47, (mxArray*)
    &MPCMotorControl_P.sfunction_P48_Size[0]);
ssSetSFcnParam(rts, 48, (mxArray*)
    &MPCMotorControl_P.sfunction_P49_Size[0]);
ssSetSFcnParam(rts, 49, (mxArray*)
    &MPCMotorControl_P.sfunction_P50_Size[0]);
ssSetSFcnParam(rts, 50, (mxArray*)
    &MPCMotorControl_P.sfunction_P51_Size[0]);
ssSetSFcnParam(rts, 51, (mxArray*)
    &MPCMotorControl_P.sfunction_P52_Size[0]);
ssSetSFcnParam(rts, 52, (mxArray*)
    &MPCMotorControl_P.sfunction_P53_Size[0]);
ssSetSFcnParam(rts, 53, (mxArray*)
    &MPCMotorControl_P.sfunction_P54_Size[0]);
ssSetSFcnParam(rts, 54, (mxArray*)
    &MPCMotorControl_P.sfunction_P55_Size[0]);
ssSetSFcnParam(rts, 55, (mxArray*)
    &MPCMotorControl_P.sfunction_P56_Size[0]);
ssSetSFcnParam(rts, 56, (mxArray*)
    &MPCMotorControl_P.sfunction_P57_Size[0]);
ssSetSFcnParam(rts, 57, (mxArray*)
    &MPCMotorControl_P.sfunction_P58_Size[0]);
```

```

ssSetSFcnParam(rts, 58, (mxArray*)
    &MPCMotorControl_P.sfunction_P59_Size[0]);
ssSetSFcnParam(rts, 59, (mxArray*)
    &MPCMotorControl_P.sfunction_P60_Size[0]);
ssSetSFcnParam(rts, 60, (mxArray*)
    &MPCMotorControl_P.sfunction_P61_Size[0]);
ssSetSFcnParam(rts, 61, (mxArray*)
    &MPCMotorControl_P.sfunction_P62_Size[0]);
ssSetSFcnParam(rts, 62, (mxArray*)
    &MPCMotorControl_P.sfunction_P63_Size[0]);
ssSetSFcnParam(rts, 63, (mxArray*)
    &MPCMotorControl_P.sfunction_P64_Size[0]);
ssSetSFcnParam(rts, 64, (mxArray*)
    &MPCMotorControl_P.sfunction_P65_Size[0]);
}
ssSetPWork(rts, (void **) &MPCMotorControl_DWork.sfunction_PWORK[0]);
{
    struct _ssDWorkRecord *dWorkRecord = (struct _ssDWorkRecord *)
        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.dWork;
    struct _ssDWorkAuxRecord *dWorkAuxRecord = (struct _ssDWorkAuxRecord *)
        &MPCMotorControl_M->NonInlinedSFcns.Sfcn0.dWorkAux;
    ssSetSFcnDWork(rts, dWorkRecord);
    ssSetSFcnDWorkAux(rts, dWorkAuxRecord);
    _ssSetNumDWork(rts, 2);
}

```

```
    ssSetDWorkWidth(rts, 0, 5);

    ssSetDWorkDataType(rts, 0, SS_POINTER);

    ssSetDWorkComplexSignal(rts, 0, 0);

    ssSetDWork(rts, 0, &MPCMotorControl_DWork.sfunction_PWORK[0]);

    ssSetDWorkWidth(rts, 1, 3);

    ssSetDWorkDataType(rts, 1, SS_DOUBLE);

    ssSetDWorkComplexSignal(rts, 1, 0);

    ssSetDWorkUsedAsDState(rts, 1, 1);

    ssSetDWork(rts, 1, &MPCMotorControl_DWork.sfunction_DSTATE[0]);

}

mpc_sfun(rts);

sfcnInitializeSizes(rts);

sfcnInitializeSampleTimes(rts);

ssSetSampleTime(rts, 0, 0.01);

ssSetOffsetTime(rts, 0, 0.0);

sfcnTsMap[0] = 0;

ssSetNumNonsampledZCs(rts, 0);

_ssSetInputPortConnected(rts, 0, 1);

_ssSetInputPortConnected(rts, 1, 1);

_ssSetInputPortConnected(rts, 2, 0);

_ssSetInputPortConnected(rts, 3, 0);

_ssSetInputPortConnected(rts, 4, 0);

_ssSetInputPortConnected(rts, 5, 0);

_ssSetInputPortConnected(rts, 6, 0);
```

```
_ssSetInputPortConnected(rts, 7, 0);
_ssSetInputPortConnected(rts, 8, 0);
_ssSetOutputPortConnected(rts, 0, 1);
_ssSetOutputPortBeingMerged(rts, 0, 0);
ssSetInputPortBufferDstPort(rts, 0, -1);
ssSetInputPortBufferDstPort(rts, 1, -1);
ssSetInputPortBufferDstPort(rts, 2, -1);
ssSetInputPortBufferDstPort(rts, 3, -1);
ssSetInputPortBufferDstPort(rts, 4, -1);
ssSetInputPortBufferDstPort(rts, 5, -1);
ssSetInputPortBufferDstPort(rts, 6, -1);
ssSetInputPortBufferDstPort(rts, 7, -1);
ssSetInputPortBufferDstPort(rts, 8, -1);
}
}
Set_Speed_Start();
Measure_Speed_Start();
Speed_Correction_Start();
Measure_Speed_Init();
Speed_Correction_Init();
}
void MPCMotorControl_terminate(void)
{
Set_Speed_Term();
```

```
Measure_Speed_Term();

Speed_Correction_Term();

}

/*

* File: MPCMotorControl_data.c

*/

#include "MPCMotorControl.h"

#include "MPCMotorControl_private.h"

Parameters_MPCMotorControl MPCMotorControl_P = {

    { 1.0, 1.0 },

    0.01,

    { 2.0, 2.0 },

    { 9.5122942450071402E-001, 0.0, 0.0, 1.0 },

    { 1.0, 2.0 },

    { 6.0, 1.0 },

    { 1.0, 1.0 },

    0.0,

    { 2.0, 1.0 },

    { 9.7541150998571996E-003, 0.0 },

    { 2.0, 1.0 },

    { 0.0, 0.0 },

    { 1.0, 1.0 },

    0.0,

    { 1.0, 1.0 },
```

1.0,  
 { 1.0, 1.0 },  
 1.0,  
 { 1.0, 1.0 },  
 1.0,  
 { 1.0, 1.0 },  
 1.0,  
 { 1.0, 1.0 },  
 2.0,  
 { 1.0, 1.0 },  
 2.0,  
 { 2.0, 1.0 },  
 { 4.0689191366278278E-003, 9.8273047193063109E-003 },  
 { 26.0, 3.0 },  
 { 1.8413035200645220E+000, 1.5273567463906801E+000,  
 1.2287213375450561E+000,  
 9.4465054945329729E-001, 6.7443405717930904E-001, 4.1739617874292101E-  
 001,  
 1.7289418556299019E-001, -5.9683304698829431E-002, -2.8091785691240290E-  
 001,  
 -4.9136267269419243E-001, -1.8413035200645220E+000, -  
 1.5273567463906801E+000,  
 -1.2287213375450561E+000, -9.4465054945329729E-001, -6.7443405717930904E-  
 001,

-4.1739617874292101E-001, -1.7289418556299019E-001, 5.9683304698829431E-002,  
2.8091785691240290E-001, 4.9136267269419243E-001,  
3.1461994949726034E+001,  
-3.8299278310873390E+000, -3.1461994949726034E+001,  
3.8299278310873390E+000,  
-3.1461994949726034E+001, 3.5291922780813373E+001, -  
2.0654488613959558E+000,  
-1.6342075681964987E+000, -1.2239981610454356E+000, -  
8.3379490275635071E-001,  
-4.6262208193572008E-001, -1.0955157319620668E-001, 2.2629948364025765E-001,  
5.4577089115276145E-001, 8.4966149426531601E-001,  
1.1387311777752451E+000,  
2.0654488613959558E+000, 1.6342075681964987E+000,  
1.2239981610454356E+000,  
8.3379490275635071E-001, 4.6262208193572008E-001, 1.0955157319620668E-001,  
-2.2629948364025765E-001, -5.4577089115276145E-001, -8.4966149426531601E-001,  
-1.1387311777752451E+000, -3.5291922780813373E+001,  
5.6473286775422977E+000,  
3.5291922780813373E+001, -5.6473286775422977E+000,  
3.5291922780813373E+001,



-4.0939251458355670E+001, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 -9.999999999999974E-006, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 -9.999999999999974E-006, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 -9.999999999999974E-006, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 -9.999999999999974E-006, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 -9.999999999999974E-006, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 -9.999999999999974E-006, -9.999999999999974E-006, -9.999999999999974E-  
 006,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 },  
 { 3.0, 3.0 },  
 { 3.1461994949726034E+001, -3.5291922780813373E+001, 0.0,  
 -3.5291922780813373E+001, 4.0939251458355670E+001, 0.0, 0.0, 0.0,  
 9.999999999999974E-006 },  
 { 2.0, 2.0 },  
 { 1.1979931827045343E+001, 2.7908501725420862E+000,  
 9.7611768162589438E+000,  
 2.3186869641972456E+000 },  
 { 1.0, 2.0 },

{ 9.5303384164143390E-001, 8.3018899378490607E-001 },  
 { 10.0, 2.0 },  
 { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0 },  
 { 10.0, 2.0 },  
 { -5.8524690599143198E-002, -1.1419509835684852E-001, -1.6715042828993065E-  
 001,  
 -2.1752309630642180E-001, -2.6543906031431419E-001, -3.1101813518193860E-  
 001,  
 -3.5437429233754392E-001, -3.9561594475723288E-001, -4.3484621805387208E-  
 001,  
 -4.7216320834483994E-001, -0.0, -5.8524690599143198E-002,  
 -1.1419509835684852E-001, -1.6715042828993065E-001, -2.1752309630642180E-  
 001,  
 -2.6543906031431419E-001, -3.1101813518193860E-001, -3.5437429233754392E-  
 001,  
 -3.9561594475723288E-001, -4.3484621805387208E-001 },  
 { 11.0, 2.0 },  
 { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 },  
 { 3.0, 1.0 },  
 { -10.0, -10.0, 0.0 },  
 { 26.0, 1.0 },

{ 5.8524690599143203E-001, 1.7271978895599172E+000,  
 2.8134552664677916E+000,  
 3.8467352459635249E+000, 4.8296215662073596E+000,  
 5.7645719549625278E+000,  
 6.6539242751948251E+000, 7.4999023709477681E+000,  
 8.3046216281110503E+000,  
 9.0700942639871194E+000, -5.8524690599143203E-001, -  
 1.7271978895599172E+000,  
 -2.8134552664677916E+000, -3.8467352459635249E+000, -  
 4.8296215662073596E+000,  
 -5.7645719549625278E+000, -6.6539242751948251E+000, -  
 7.4999023709477681E+000,  
 -8.3046216281110503E+000, -9.0700942639871194E+000, 10.0, 20.0, -10.0, -20.0,  
 -10.0, -10.0 },  
 { 26.0, 1.0 },  
 { 4000.0, 4000.0, 4000.0, 4000.0, 4000.0, 4000.0, 4000.0, 4000.0, 4000.0,  
 4000.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, -0.0, 4000.0,  
 4000.0, -0.0, -0.0, 10.0, 10.0 },  
 { 26.0, 2.0 },  
 { -5.7073765470042837E+000, -5.4290245082157567E+000, -  
 5.1642478585503460E+000,  
 -4.9123845184678903E+000, -4.6728046984284282E+000, -  
 4.4449093240903066E+000,

-4.2281285383122800E+000, -4.0219202762138355E+000, -  
 3.8257689097306393E+000,  
 -3.6391839582758001E+000, 5.7073765470042837E+000,  
 5.4290245082157567E+000,  
 5.1642478585503460E+000, 4.9123845184678903E+000,  
 4.6728046984284282E+000,  
 4.4449093240903066E+000, 4.2281285383122800E+000,  
 4.0219202762138355E+000,  
 3.8257689097306393E+000, 3.6391839582758001E+000, 0.0, 0.0, 0.0, 0.0, 0.0,  
 0.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0,  
 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 },  
 { 26.0, 1.0 },  
 { -5.8524690599143198E-002, -1.1419509835684852E-001, -1.6715042828993065E-  
 001,  
 -2.1752309630642180E-001, -2.6543906031431419E-001, -3.1101813518193860E-  
 001,  
 -3.5437429233754392E-001, -3.9561594475723288E-001, -4.3484621805387208E-  
 001,  
 -4.7216320834483994E-001, 5.8524690599143198E-002, 1.1419509835684852E-  
 001,  
 1.6715042828993065E-001, 2.1752309630642180E-001, 2.6543906031431419E-  
 001,  
 3.1101813518193860E-001, 3.5437429233754392E-001, 3.9561594475723288E-  
 001,



{ 1.7932228354263401E+001, 1.5702847401118476E+001, 0.0 },  
{ 29.0, 29.0 },  
{ -3.1461994949726034E+001, 3.5291922780813373E+001, -0.0,  
1.8413035200645220E+000, 1.5273567463906801E+000,  
1.2287213375450561E+000,  
9.4465054945329729E-001, 6.7443405717930904E-001, 4.1739617874292101E-  
001,  
1.7289418556299019E-001, -5.9683304698829431E-002, -2.8091785691240290E-  
001,  
-4.9136267269419243E-001, -1.8413035200645220E+000, -  
1.5273567463906801E+000,  
-1.2287213375450561E+000, -9.4465054945329729E-001, -6.7443405717930904E-  
001,  
-4.1739617874292101E-001, -1.7289418556299019E-001, 5.9683304698829431E-  
002,  
2.8091785691240290E-001, 4.9136267269419243E-001,  
3.1461994949726034E+001,  
-3.8299278310873390E+000, -3.1461994949726034E+001,  
3.8299278310873390E+000,  
-3.1461994949726034E+001, 3.5291922780813373E+001,  
3.5291922780813373E+001,  
-4.0939251458355670E+001, -0.0, -2.0654488613959558E+000,  
-1.6342075681964987E+000, -1.2239981610454356E+000, -  
8.3379490275635071E-001,



-9.9999999999999974E-006, -9.9999999999999974E-006, -9.9999999999999974E-006,  
-9.9999999999999974E-006, -9.9999999999999974E-006, -9.9999999999999974E-006,  
-9.9999999999999974E-006, -9.9999999999999974E-006, 0.0, 0.0, 0.0, 0.0, 0.0,  
0.0, 1.8413035200645220E+000, -2.0654488613959558E+000,  
-9.9999999999999974E-006, -1.0777171881088940E-001, -8.9398081017028577E-002,  
-7.1920536112389799E-002, -5.5295381131064844E-002, -3.9481044525943920E-002,  
-2.4437982218194123E-002, -1.0128578716464850E-002, 3.4829469414333820E-003,  
1.6430630659572759E-002, 2.8746848391395678E-002, 1.0775171881088941E-001,  
8.9378081017028585E-002, 7.1900536112389807E-002, 5.5275381131064838E-002,  
3.9461044525943914E-002, 2.4417982218194124E-002, 1.0108578716464851E-002,  
-3.5029469414333820E-003, -1.6450630659572758E-002, -2.8766848391395677E-002,  
-1.8413035200645220E+000, 2.2414534133143410E-001,  
1.8413035200645220E+000,  
-2.2414534133143410E-001, 1.8413035200645220E+000, -  
2.0654488613959558E+000,



1.5273567463906801E+000, -1.6342075681964987E+000, -9.999999999999974E-006,  
-8.9398081017028577E-002, -7.8785161576601551E-002, -6.8689840325011770E-002,  
-5.9086873700712167E-002, -4.9952249285180099E-002, -4.1263125759363209E-002,  
-3.2997775788484833E-002, -2.5135531692389437E-002, -1.7656733765576357E-002,  
-1.0542681117696882E-002, 8.9378081017028585E-002, 7.8765161576601558E-002,  
6.8669840325011777E-002, 5.9066873700712161E-002, 4.9932249285180093E-002,  
4.1243125759363203E-002, 3.2977775788484827E-002, 2.5115531692389437E-002,  
1.7636733765576357E-002, 1.0522681117696882E-002, -1.5273567463906801E+000,  
1.0685082180581873E-001, 1.5273567463906801E+000, -1.0685082180581873E-001,  
1.5273567463906801E+000, -1.6342075681964987E+000, 1.2287213375450561E+000,  
-1.2239981610454356E+000, -9.999999999999974E-006, -7.1920536112389785E-002,  
-6.8689840325011783E-002, -6.5616707430447355E-002, -6.2693452995736546E-002,

-5.9912767362137499E-002, -5.7267697367171383E-002, -5.4751628958095815E-002,  
-5.2358270653326697E-002, -5.0081637810457040E-002, -4.7916037661534787E-002,  
7.1900536112389793E-002, 6.8669840325011791E-002, 6.5596707430447362E-002,  
6.2673452995736553E-002, 5.9892767362137493E-002, 5.7247697367171377E-002,  
5.4731628958095808E-002, 5.2338270653326691E-002, 5.0061637810457034E-002,  
4.7896037661534781E-002, -1.2287213375450561E+000, -4.7231764996200631E-003,  
1.2287213375450561E+000, 4.7231764996200631E-003,  
1.2287213375450561E+000,  
-1.2239981610454356E+000, 9.4465054945329729E-001, -8.3379490275635071E-001,  
-9.9999999999999974E-006, -5.5295381131064907E-002, -5.9086873700712132E-002,  
-6.2693452995736573E-002, -6.6124137342958758E-002, -6.9387505240210631E-002,  
-7.2491716807047218E-002, -7.5444534189297890E-002, -7.8253340968472210E-002,  
-8.0925160624559803E-002, -8.3466674098389745E-002, 5.5275381131064900E-002,

5.9066873700712126E-002, 6.2673452995736581E-002, 6.6104137342958766E-002,  
6.9367505240210639E-002, 7.2471716807047226E-002, 7.5424534189297898E-002,  
7.8233340968472218E-002, 8.0905160624559810E-002, 8.3446674098389753E-002,  
-9.4465054945329729E-001, -1.1085564669694625E-001, 9.4465054945329729E-001,  
1.1085564669694625E-001, 9.4465054945329729E-001, -8.3379490275635071E-001,  
6.7443405717930904E-001, -4.6262208193572008E-001, -9.999999999999974E-006,  
-3.9481044525943969E-002, -4.9952249285180043E-002, -5.9912767362137458E-002,  
-6.9387505240210548E-002, -7.8400154699265243E-002, -8.6973252057427994E-002,  
-9.5128234523622068E-002, -1.0288549380175360E-001, -1.1026442708059341E-001,  
-1.1728348553685343E-001, 3.9461044525943963E-002, 4.9932249285180037E-002,  
5.9892767362137452E-002, 6.9367505240210556E-002, 7.8380154699265250E-002,  
8.6953252057428002E-002, 9.5108234523622076E-002, 1.0286549380175361E-001,

1.1024442708059341E-001, 1.1726348553685344E-001, -6.7443405717930904E-001,  
-2.1181197524358808E-001, 6.7443405717930904E-001, 2.1181197524358808E-001,  
6.7443405717930904E-001, -4.6262208193572008E-001, 4.1739617874292101E-001,  
-1.0955157319620668E-001, -9.999999999999974E-006, -2.4437982218194199E-002,  
-4.1263125759363355E-002, -5.7267697367171626E-002, -7.2491716807047454E-002,  
-8.6973252057428452E-002, -1.0074851449953451E-001, -1.1385194946468576E-001,  
-1.2631632236556950E-001, -1.3817280062683934E-001, -1.4945103161991236E-001,  
2.4417982218194200E-002, 4.1243125759363349E-002, 5.7247697367171620E-002,  
7.2471716807047462E-002, 8.6953252057428460E-002, 1.0072851449953452E-001,  
1.1383194946468576E-001, 1.2629632236556948E-001, 1.3815280062683932E-001,  
1.4943103161991234E-001, -4.1739617874292101E-001, -3.0784460554671522E-001,  
4.1739617874292101E-001, 3.0784460554671522E-001, 4.1739617874292101E-001,

-1.0955157319620668E-001, 1.7289418556299019E-001, 2.2629948364025765E-001,  
-9.9999999999999974E-006, -1.0128578716465043E-002, -3.2997775788485000E-002,  
-5.4751628958096148E-002, -7.5444534189298237E-002, -9.5128234523622610E-002,  
-1.1385194946468576E-001, -1.3166249805258923E-001, -1.4860441593590296E-001,  
-1.6472006673398581E-001, -1.8004974796810072E-001, 1.0108578716465044E-002,  
3.2977775788484993E-002, 5.4731628958096142E-002, 7.5424534189298245E-002,  
9.5108234523622617E-002, 1.1383194946468576E-001, 1.3164249805258921E-001,  
1.4858441593590294E-001, 1.6470006673398579E-001, 1.8002974796810070E-001,  
-1.7289418556299019E-001, -3.9919366920324872E-001, 1.7289418556299019E-001,  
3.9919366920324872E-001, 1.7289418556299019E-001, 2.2629948364025765E-001,  
-5.9683304698829431E-002, 5.4577089115276145E-001, -9.9999999999999974E-006,  
3.4829469414333169E-003, -2.5135531692389513E-002, -5.2358270653326870E-002,

-7.8253340968472293E-002, -1.0288549380175392E-001, -1.2631632236556917E-001,  
-1.4860441593590262E-001, -1.6980550635602937E-001, -1.8997260739515409E-001,  
-2.0915614731044846E-001, -3.5029469414333170E-003, 2.5115531692389514E-002,  
5.2338270653326864E-002, 7.8233340968472301E-002, 1.0286549380175393E-001,  
1.2629632236556915E-001, 1.4858441593590260E-001, 1.6978550635602935E-001,  
1.8995260739515407E-001, 2.0913614731044844E-001, 5.9683304698829431E-002,  
-4.8608758645393202E-001, -5.9683304698829431E-002, 4.8608758645393202E-001,  
-5.9683304698829431E-002, 5.4577089115276145E-001, -2.8091785691240290E-001,  
8.4966149426531601E-001, -9.999999999999974E-006, 1.6430630659572686E-002,  
-1.7656733765576350E-002, -5.0081637810457151E-002, -8.0925160624559775E-002,  
-1.1026442708059354E-001, -1.3817280062683876E-001, -1.6472006673398515E-001,  
-1.8997260739515376E-001, -2.1399356711545803E-001, -2.3684301080615797E-001,

-1.6450630659572685E-002, 1.7636733765576351E-002, 5.0061637810457145E-002,  
8.0905160624559783E-002, 1.1024442708059355E-001, 1.3815280062683874E-001,  
1.6470006673398513E-001, 1.8995260739515374E-001, 2.1397356711545801E-001,  
2.3682301080615795E-001, 2.8091785691240290E-001, -5.6874363735291222E-001,  
-2.8091785691240290E-001, 5.6874363735291222E-001, -2.8091785691240290E-001,  
8.4966149426531601E-001, -4.9136267269419243E-001,  
1.1387311777752451E+000,  
-9.999999999999974E-006, 2.8746848391395585E-002, -1.0542681117696896E-002,  
-4.7916037661534787E-002, -8.3466674098389579E-002, -1.1728348553685337E-001,  
-1.4945103161991147E-001, -1.8004974796809969E-001, -2.0915614731044782E-001,  
-2.3684301080615758E-001, -2.6317957003541143E-001, -2.8766848391395584E-002,  
1.0522681117696896E-002, 4.7896037661534781E-002, 8.3446674098389587E-002,  
1.1726348553685338E-001, 1.4943103161991145E-001, 1.8002974796809967E-001,

2.0913614731044780E-001, 2.3682301080615756E-001, 2.6315957003541141E-001,  
4.9136267269419243E-001, -6.4736850508105115E-001, -4.9136267269419243E-001,  
6.4736850508105115E-001, -4.9136267269419243E-001,  
1.1387311777752451E+000,  
-1.8413035200645220E+000, 2.0654488613959558E+000, -9.999999999999974E-006,  
1.0775171881088941E-001, 8.9378081017028585E-002, 7.1900536112389807E-002,  
5.5275381131064838E-002, 3.9461044525943914E-002, 2.4417982218194124E-002,  
1.0108578716464851E-002, -3.5029469414333820E-003, -1.6450630659572758E-002,  
-2.8766848391395677E-002, -1.0777171881088940E-001, -8.9398081017028577E-002,  
-7.1920536112389799E-002, -5.5295381131064844E-002, -3.9481044525943920E-002,  
-2.4437982218194123E-002, -1.0128578716464850E-002, 3.4829469414333820E-003,  
1.6430630659572759E-002, 2.8746848391395678E-002,  
1.8413035200645220E+000,  
-2.2414534133143410E-001, -1.8413035200645220E+000, 2.2414534133143410E-001,



-1.8413035200645220E+000, 2.0654488613959558E+000, -  
1.5273567463906801E+000,  
1.6342075681964987E+000, -9.999999999999974E-006, 8.9378081017028585E-  
002,  
7.8765161576601558E-002, 6.8669840325011777E-002, 5.9066873700712161E-  
002,  
4.9932249285180093E-002, 4.1243125759363203E-002, 3.2977775788484827E-  
002,  
2.5115531692389437E-002, 1.7636733765576357E-002, 1.0522681117696882E-  
002,  
-8.9398081017028577E-002, -7.8785161576601551E-002, -6.8689840325011770E-  
002,  
-5.9086873700712167E-002, -4.9952249285180099E-002, -4.1263125759363209E-  
002,  
-3.2997775788484833E-002, -2.5135531692389437E-002, -1.7656733765576357E-  
002,  
-1.0542681117696882E-002, 1.5273567463906801E+000, -1.0685082180581873E-  
001,  
-1.5273567463906801E+000, 1.0685082180581873E-001, -  
1.5273567463906801E+000,  
1.6342075681964987E+000, -1.2287213375450561E+000,  
1.2239981610454356E+000,  
-9.999999999999974E-006, 7.1900536112389793E-002, 6.8669840325011791E-  
002,

6.5596707430447362E-002, 6.2673452995736553E-002, 5.9892767362137493E-002,  
5.7247697367171377E-002, 5.4731628958095808E-002, 5.2338270653326691E-002,  
5.0061637810457034E-002, 4.7896037661534781E-002, -7.1920536112389785E-002,  
-6.8689840325011783E-002, -6.5616707430447355E-002, -6.2693452995736546E-002,  
-5.9912767362137499E-002, -5.7267697367171383E-002, -5.4751628958095815E-002,  
-5.2358270653326697E-002, -5.0081637810457040E-002, -4.7916037661534787E-002,  
1.2287213375450561E+000, 4.7231764996200631E-003, -1.2287213375450561E+000,  
-4.7231764996200631E-003, -1.2287213375450561E+000,  
1.2239981610454356E+000, -9.4465054945329729E-001, 8.3379490275635071E-001, -9.999999999999974E-006,  
5.5275381131064900E-002, 5.9066873700712126E-002, 6.2673452995736581E-002,  
6.6104137342958766E-002, 6.9367505240210639E-002, 7.2471716807047226E-002,  
7.5424534189297898E-002, 7.8233340968472218E-002, 8.0905160624559810E-002,

8.3446674098389753E-002, -5.5295381131064907E-002, -5.9086873700712132E-002,  
-6.2693452995736573E-002, -6.6124137342958758E-002, -6.9387505240210631E-002,  
-7.2491716807047218E-002, -7.5444534189297890E-002, -7.8253340968472210E-002,  
-8.0925160624559803E-002, -8.3466674098389745E-002, 9.4465054945329729E-001,  
1.1085564669694625E-001, -9.4465054945329729E-001, -1.1085564669694625E-001,  
-9.4465054945329729E-001, 8.3379490275635071E-001, -6.7443405717930904E-001,  
4.6262208193572008E-001, -9.999999999999974E-006, 3.9461044525943963E-002,  
4.9932249285180037E-002, 5.9892767362137452E-002, 6.9367505240210556E-002,  
7.8380154699265250E-002, 8.6953252057428002E-002, 9.5108234523622076E-002,  
1.0286549380175361E-001, 1.1024442708059341E-001, 1.1726348553685344E-001,  
-3.9481044525943969E-002, -4.9952249285180043E-002, -5.9912767362137458E-002,  
-6.9387505240210548E-002, -7.8400154699265243E-002, -8.6973252057427994E-002,

-9.5128234523622068E-002, -1.0288549380175360E-001, -1.1026442708059341E-001,  
-1.1728348553685343E-001, 6.7443405717930904E-001, 2.1181197524358808E-001,  
-6.7443405717930904E-001, -2.1181197524358808E-001, -6.7443405717930904E-001,  
4.6262208193572008E-001, -4.1739617874292101E-001, 1.0955157319620668E-001,  
-9.999999999999974E-006, 2.4417982218194200E-002, 4.1243125759363349E-002,  
5.7247697367171620E-002, 7.2471716807047462E-002, 8.6953252057428460E-002,  
1.0072851449953452E-001, 1.1383194946468576E-001, 1.2629632236556948E-001,  
1.3815280062683932E-001, 1.4943103161991234E-001, -2.4437982218194199E-002,  
-4.1263125759363355E-002, -5.7267697367171626E-002, -7.2491716807047454E-002,  
-8.6973252057428452E-002, -1.0074851449953451E-001, -1.1385194946468576E-001,  
-1.2631632236556950E-001, -1.3817280062683934E-001, -1.4945103161991236E-001,  
4.1739617874292101E-001, 3.0784460554671522E-001, -4.1739617874292101E-001,

-3.0784460554671522E-001, -4.1739617874292101E-001, 1.0955157319620668E-001,  
-1.7289418556299019E-001, -2.2629948364025765E-001, -9.999999999999974E-006,  
1.0108578716465044E-002, 3.2977775788484993E-002, 5.4731628958096142E-002,  
7.5424534189298245E-002, 9.5108234523622617E-002, 1.1383194946468576E-001,  
1.3164249805258921E-001, 1.4858441593590294E-001, 1.6470006673398579E-001,  
1.8002974796810070E-001, -1.0128578716465043E-002, -3.2997775788485000E-002,  
-5.4751628958096148E-002, -7.5444534189298237E-002, -9.5128234523622610E-002,  
-1.1385194946468576E-001, -1.3166249805258923E-001, -1.4860441593590296E-001,  
-1.6472006673398581E-001, -1.8004974796810072E-001, 1.7289418556299019E-001,  
3.9919366920324872E-001, -1.7289418556299019E-001, -3.9919366920324872E-001,  
-1.7289418556299019E-001, -2.2629948364025765E-001, 5.9683304698829431E-002,  
-5.4577089115276145E-001, -9.999999999999974E-006, -3.5029469414333170E-003,

2.5115531692389514E-002, 5.2338270653326864E-002, 7.8233340968472301E-002,

1.0286549380175393E-001, 1.2629632236556915E-001, 1.4858441593590260E-001,

1.6978550635602935E-001, 1.8995260739515407E-001, 2.0913614731044844E-001,

3.4829469414333169E-003, -2.5135531692389513E-002, -5.2358270653326870E-002,

-7.8253340968472293E-002, -1.0288549380175392E-001, -1.2631632236556917E-001,

-1.4860441593590262E-001, -1.6980550635602937E-001, -1.8997260739515409E-001,

-2.0915614731044846E-001, -5.9683304698829431E-002, 4.8608758645393202E-001,

5.9683304698829431E-002, -4.8608758645393202E-001, 5.9683304698829431E-002,

-5.4577089115276145E-001, 2.8091785691240290E-001, -8.4966149426531601E-001,

-9.999999999999974E-006, -1.6450630659572685E-002, 1.7636733765576351E-002,

5.0061637810457145E-002, 8.0905160624559783E-002, 1.1024442708059355E-001,

1.3815280062683874E-001, 1.6470006673398513E-001, 1.8995260739515374E-001,

2.1397356711545801E-001, 2.3682301080615795E-001, 1.6430630659572686E-002,  
-1.7656733765576350E-002, -5.0081637810457151E-002, -8.0925160624559775E-002,  
-1.1026442708059354E-001, -1.3817280062683876E-001, -1.6472006673398515E-001,  
-1.8997260739515376E-001, -2.1399356711545803E-001, -2.3684301080615797E-001,  
-2.8091785691240290E-001, 5.6874363735291222E-001, 2.8091785691240290E-001,  
-5.6874363735291222E-001, 2.8091785691240290E-001, -8.4966149426531601E-001,  
4.9136267269419243E-001, -1.1387311777752451E+000, -9.9999999999999974E-006,  
-2.8766848391395584E-002, 1.0522681117696896E-002, 4.7896037661534781E-002,  
8.3446674098389587E-002, 1.1726348553685338E-001, 1.4943103161991145E-001,  
1.8002974796809967E-001, 2.0913614731044780E-001, 2.3682301080615756E-001,  
2.6315957003541141E-001, 2.8746848391395585E-002, -1.0542681117696896E-002,  
-4.7916037661534787E-002, -8.3466674098389579E-002, -1.1728348553685337E-001,

-1.4945103161991147E-001, -1.8004974796809969E-001, -2.0915614731044782E-001,  
-2.3684301080615758E-001, -2.6317957003541143E-001, -4.9136267269419243E-001,  
6.4736850508105115E-001, 4.9136267269419243E-001, -6.4736850508105115E-001,  
4.9136267269419243E-001, -1.1387311777752451E+000,  
3.1461994949726034E+001,  
-3.5291922780813373E+001, 0.0, -1.8413035200645220E+000,  
-1.5273567463906801E+000, -1.2287213375450561E+000, -  
9.4465054945329729E-001,  
-6.7443405717930904E-001, -4.1739617874292101E-001, -1.7289418556299019E-001,  
5.9683304698829431E-002, 2.8091785691240290E-001, 4.9136267269419243E-001,  
1.8413035200645220E+000, 1.5273567463906801E+000,  
1.2287213375450561E+000,  
9.4465054945329729E-001, 6.7443405717930904E-001, 4.1739617874292101E-001,  
1.7289418556299019E-001, -5.9683304698829431E-002, -2.8091785691240290E-001,  
-4.9136267269419243E-001, -3.1461994949726034E+001,  
3.8299278310873390E+000,



3.1461994949726034E+001, -3.8299278310873390E+000,  
3.1461994949726034E+001,  
-3.5291922780813373E+001, -3.8299278310873390E+000,  
5.6473286775422977E+000,  
0.0, 2.2414534133143382E-001, 1.0685082180581862E-001,  
-4.7231764996205072E-003, -1.1085564669694659E-001, -2.1181197524358897E-  
001,  
-3.0784460554671433E-001, -3.9919366920324784E-001, -4.8608758645393202E-  
001,  
-5.6874363735291311E-001, -6.4736850508105270E-001, -2.2414534133143382E-  
001,  
-1.0685082180581862E-001, 4.7231764996205072E-003, 1.1085564669694659E-  
001,  
2.1181197524358897E-001, 3.0784460554671433E-001, 3.9919366920324784E-  
001,  
4.8608758645393202E-001, 5.6874363735291311E-001, 6.4736850508105270E-  
001,  
3.8299278310873390E+000, -1.8174008464549587E+000, -  
3.8299278310873390E+000,  
1.8174008464549587E+000, -3.8299278310873390E+000,  
5.6473286775422977E+000,  
-3.1461994949726034E+001, 3.5291922780813373E+001, 0.0,  
1.8413035200645220E+000, 1.5273567463906801E+000,  
1.2287213375450561E+000,

9.4465054945329729E-001, 6.7443405717930904E-001, 4.1739617874292101E-001,  
1.7289418556299019E-001, -5.9683304698829431E-002, -2.8091785691240290E-001,  
-4.9136267269419243E-001, -1.8413035200645220E+000, -1.5273567463906801E+000,  
-1.2287213375450561E+000, -9.4465054945329729E-001, -6.7443405717930904E-001,  
-4.1739617874292101E-001, -1.7289418556299019E-001, 5.9683304698829431E-002,  
2.8091785691240290E-001, 4.9136267269419243E-001, 3.1461994949726034E+001,  
-3.8299278310873390E+000, -3.1461994949726034E+001, 3.8299278310873390E+000,  
-3.1461994949726034E+001, 3.5291922780813373E+001, 3.8299278310873390E+000,  
-5.6473286775422977E+000, 0.0, -2.2414534133143382E-001, -1.0685082180581862E-001,  
4.7231764996205072E-003, 1.1085564669694659E-001,  
2.1181197524358897E-001, 3.0784460554671433E-001, 3.9919366920324784E-001,  
4.8608758645393202E-001, 5.6874363735291311E-001, 6.4736850508105270E-001,

2.2414534133143382E-001, 1.0685082180581862E-001, -4.7231764996205072E-003,  
-1.1085564669694659E-001, -2.1181197524358897E-001, -3.0784460554671433E-001,  
-3.9919366920324784E-001, -4.8608758645393202E-001, -5.6874363735291311E-001,  
-6.4736850508105270E-001, -3.8299278310873390E+000,  
1.8174008464549587E+000,  
3.8299278310873390E+000, -1.8174008464549587E+000,  
3.8299278310873390E+000,  
-5.6473286775422977E+000, -3.1461994949726034E+001,  
3.5291922780813373E+001,  
0.0, 1.8413035200645220E+000, 1.5273567463906801E+000,  
1.2287213375450561E+000, 9.4465054945329729E-001, 6.7443405717930904E-001,  
4.1739617874292101E-001, 1.7289418556299019E-001, -5.9683304698829431E-002,  
-2.8091785691240290E-001, -4.9136267269419243E-001, -  
1.8413035200645220E+000,  
-1.5273567463906801E+000, -1.2287213375450561E+000, -  
9.4465054945329729E-001,  
-6.7443405717930904E-001, -4.1739617874292101E-001, -1.7289418556299019E-001,

5.9683304698829431E-002, 2.8091785691240290E-001, 4.9136267269419243E-  
 001,  
 3.1461994949726034E+001, -3.8299278310873390E+000, -  
 3.1461994949726034E+001,  
 3.8299278310873390E+000, -3.1461994949726034E+001,  
 3.5291922780813373E+001,  
 3.5291922780813373E+001, -4.0939251458355670E+001, 0.0,  
 -2.0654488613959558E+000, -1.6342075681964987E+000, -  
 1.2239981610454356E+000,  
 -8.3379490275635071E-001, -4.6262208193572008E-001, -1.0955157319620668E-  
 001,  
 2.2629948364025765E-001, 5.4577089115276145E-001, 8.4966149426531601E-  
 001,  
 1.1387311777752451E+000, 2.0654488613959558E+000,  
 1.6342075681964987E+000,  
 1.2239981610454356E+000, 8.3379490275635071E-001, 4.6262208193572008E-  
 001,  
 1.0955157319620668E-001, -2.2629948364025765E-001, -5.4577089115276145E-  
 001,  
 -8.4966149426531601E-001, -1.1387311777752451E+000, -  
 3.5291922780813373E+001,  
 5.6473286775422977E+000, 3.5291922780813373E+001, -  
 5.6473286775422977E+000,  
 3.5291922780813373E+001, -4.0939251458355670E+001 },





