

DOKUZ EYLÜL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAPAY SİNİR AĞLARI İLE ROBOT KOLU
KONTROLÜ

Kemal BERKİ

Ocak, 2013
İZMİR

YAPAY SİNİR AĞLARI İLE ROBOT KOLU KONTROLÜ

**Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi
Mekatronik Mühendisliği, Mekatronik Mühendisliği Programı**

Kemal BERKİ

**Ocak, 2013
İZMİR**

YÜKSEK LİSANS TEZİ SINAV SONUÇ FORMU

KEMAL BERKİ, tarafından **YRD.DOÇ.DR. GÜLESER KALAYCI DEMİR** yönetiminde hazırlanan “**YAPAY SİNİR AĞLARI İLE ROBOT KOLU KONTROLÜ**” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.



Yrd.Doç.Dr. Güleser KALAYCI DEMİR

Yönetici



Yrd.Doç.Dr. Hatice DOĞAN

Juri Uyesi



Yrd.Doç.Dr. Gökhan DALKILIÇ

Juri Uyesi



Prof. Dr. Mustafa SABUNCU

Müdür

Fen Bilimleri Enstitüsü

TEŞEKKÜR

Bu tez çalışmamda bana her türlü desteğini esirgemeyen, değerli yardım ve katkıları ile beni yönlendiren değerli hocam ve danışmanım Yrd.Doç.Dr. Güleser KALAYCI DEMİR'e, bana bu fırsatı tanıdığı için , aynı zamanda hayatımda çok önemli bir rolü olduğuna inandığım değerli hocam Prof.Dr. Erol UYAR'a, yüksek lisansım boyunca bana verdiği desteğinden ötürü sayın müdürüm Yıldıray BAŞKURT'a ve hayatımın her anında olduğu gibi tez çalışmalarım sırasında da beni sonsuz anlayış ve sabrı ile destekleyen eşime çok teşekkür ederim.

YAPAY SİNİR AĞLARI İLE ROBOT KOLU KONTROLÜ

ÖZ

Bu tez çalışmasında robot kollarının nasıl tasarlanabileceği anlatılmış ve bir robot tasarımı yapılmıştır. Bunun yanısıra bu tip bir robot kolunun yapay sinir ağları ile nasıl kontrol edilebileceği üzerine çalışılmıştır.

Robot kolu tasarımı ile ilgili konular; mekanik tasarım, doğru akım motorları ve kontrol sistemi şeklinde üç ana başlık altında toplandı. Mekanik tasarım başlığında; malzeme seçimi, dikkat edilmesi gereken noktalar anlatılmıştır, bir robot kolu tasarımı yapıldı ve tüm parçaların tasarımlarından ve montajlarından tek tek bahsedildi. Doğru akım motorları (DC motorlar) başlığında; DC motorlar konusunda teorik bilgi verdikten sonra doğru akım motorlarının nasıl seçilebileceği konusunda bilgiler verildi. Kontrol sistemi başlığında ise ; farklı kontrol sistemleri ile kontroller yapıldı ve bu oluşturulan sistemler ile elde edilen sonuçlar karşılaştırıldı. Matlab ve simulink yardımıyla TMS320F28335 işlemcisinin programlanması ve bir yapay sinir ağı ile robotun kontrolü üzerine çalışıldı. Diğer bir deyişle Yapay sinir ağları ile bir robotun nasıl kontrol edilebileceği gösterildi.

Sonuç olarak robot kollarında kontrol sistemi tasarımı için yapay sinir ağlarının da bir çözüm olarak kullanılabileceği gösterildi.

Anahtar sözcükler : Yapay sinir ağları, Doğru akım motoru, Robot kolu, Kontrol sistemi

NEURAL NETWORK CONTROL OF A ROBOTIC ARM

ABSTRACT

In this thesis it is explained how robotic arms can be designed. A design for a robotic arm is detailed within this thesis. A study will show how this type of robotic arm can be controlled by an artificial neural network.

The theme related with robotic arm design has been listed in three headings; mechanical design, DC motors and control system. The mechanical design section describes the material selection and significant points of the robotic arm. Also there is a detailed analysis of the materials used and the assemblies are shown in 3D pictures. The DC motor section is divided into two sub sections. The first sub section contains a theoretical presentation about DC motors. The second sub section shows how to select the correct DC motor for the application. The control system section demonstrates how a TMS320F28335 microcontroller can be programmed by using Matlab and Simulink and, using the same study method, it is demonstrated how a robotic arm can be controlled by an artificial neural network.

In conclusion it has been shown that also the artificial neural networks can be used as an alternative solution in control system of robotic arms and the benefits of this method have been discussed.

Keywords: Artificial neural network, DC motor, Robotic arm, control system

İÇİNDEKİLER

	Sayfa
YÜKSEK LİSANS TEZ SINAVI SONUÇ FORMU	ii
TEŞEKKÜR.....	iii
ÖZ	iv
ABSTRACT.....	v
BÖLÜM BİR – GİRİŞ	1
BÖLÜM İKİ – ROBOT KİNEMATİĞİ VE BİLEŞENLERİ	4
2.1 Robot Kolu Kinematığı	4
2.1.1 İleri Kinematik.....	5
2.1.2 Ters Kinematik	5
2.2 Robot Kolu Bileşenleri.....	7
2.2.1 Elektrik Motorları	7
2.2.1.1 Doğru Akım Motorları.....	8
2.2.1.2 Doğru Akım Motorlarının Çalışma Prensibi.....	8
2.2.1.3 Doğru Akım Motorlarının Yön ve Hız Kontrolü.....	9
2.2.1.4 Redüktörler	14
2.2.1.5 Tork Hesapları ve DC Motor Seçimi	14
2.2.2 Servo Motorlar ve Motor Seçimi.....	18
2.2.3 Enkoderlar.....	21
2.2.3.1 Artımsal Tip Optik Enkoderlar	21
2.2.3.2 Encoder Seçimi	23
BÖLÜM ÜÇ – MEKANİK TASARIM	24
3.1 Robot Koluna Genel Bakış.....	24
3.2 Parçaların Tasarımı ve Birleştirilmesi.....	25

BÖLÜM DÖRT – MOTORLARIN FARKLI KONTROL YÖNTEMLERİ İLE KONTROLÜ 38

4.1 Model Referanslı Uyarlamalı Kontrol.....	38
4.1.1 Plant Kavramı ve Tasarımı	38
4.1.1.1 Sistemin Plant Modelinin Bulunması	39
4.1.1.1.1 DC Motor Modelinin Parametrelerinin Bulunması.....	39
4.1.1.1.2 Encoder Modelinin Parametrelerinin Bulunması.....	43
4.1.1.1.3 Sistemin Plant Modeli	43
4.1.2 Uyarlamalı Kontrol ve Kontrol Bloğunun Oluşturulması	44
4.1.2.1 Oluşturulan Plant Modelinin Farklı Kontrol Yöntemler ile Kontrolü.....	45
4.1.2.1.1 Geri Beslemeli Kontrol	45
4.1.2.1.2 PID Kontrolü	48
4.1.2.1.3 Yapay Sinir Ağları Tabanlı Model Referans Kontrol Yöntemi.....	51
4.1.2.1.4 Kontrol Yöntemlerinin Karşılaştırılması.....	55
4.1.3 Uyarlamalı Kontrol ve Fiziki Plantin Kontrolü	57
4.1.3.1 Fiziki Plantin Farklı Kontrol Yöntemleri ile Kontrolü	57
4.1.3.1.1 Geri Beslemeli Kontrol ile Motorların kontrolü	57
4.1.3.1.2 PID kontrol ile Motorların kontrolü	60
4.1.3.1.3 DC Motorların Kontrolü için Kurulan Simulink Bloğu	62
4.2 Mini Servo Motorun Kontrolü	65

BÖLÜM BEŞ – KONTROL SİSTEMİNDE KULLANILAN ARAÇLAR 67

5.1 Matlab.....	67
5.2 Simulink ve Guide.....	67
5.3 TMS320F28335 Experimenter Kit.....	68
5.4 Code Composer Programı	69
5.5 Motor Yön ve Hız Kontrolü	70
5.6 Yapay Sinir Ağları.....	70
5.6.1 Yapay Sinir Ağları Çeşitleri	73
5.6.2 Yapay Sinir Ağları Eğitim Yöntemleri	74

5.7 Sistemin Kurulumu	75
5.8 Donanımın Genel Görünümü	78
5.9 TMS320F28335 Experimenter Kitin Simulink ile Kullanımı.....	78
BÖLÜM ALTI – SONUÇLAR	81
KAYNAKLAR	82
EKLER.....	86

BÖLÜM BİR

GİRİŞ

Günümüzde gelişen teknoloji ve bilgisayar sistemleri sayesinde kontrol sistemlerinde de ileri düzey gelişmeler sağlanmış ve birbirinden farklı birçok kontrol algoritması geliştirilmiştir. Oransal kontrol (P tipi kontrol) , oransal integral kontrol (PI tipi kontrol), oransal integral türev kontrol (PID tipi kontrol), bulanık mantık, yapay sinir ağları gibi kontrol metotları bunlardan bazılarıdır. (Hoque, Zaman ve Rahman,1995) (Buja, 1993)

Geliştirilen bu kontrol sistemlerinin birbirinden üstünlükleri ya da dezavantajları vardır. Örneğin klasik oransal integral PI ya da oransal integral türev PID kontrollü kontrol sistemleri sistem dinamiklerinin yanlışsız matematiksel modeline ihtiyaç duyar. Bu tip bir matematiksel modeli sağlamak oldukça zordur. Her ne kadar sistemin modeli elde edilmiş olsa da saturasyon, bozulma, belirsiz parametreler ya da gürültü gibi bazı bilinmeyen koşullardan kaçınılamaz ve hatasız modellenemez. Bundan başka örneğin motor kontrolünde motorun üzerine binen yük şüphelidir ve bu durum, eğer kontrol sistemi olması gerektiği gibi tasarlanmamışsa, kontrol sisteminde kararsızlıklara yol açabilecek lineer olmayan bir karakteristik gösterir. Bu klasik kontrol yöntemleri kesin kontrolün gerekli olmadığı sistemlerde kullanılabilir. Ancak bu sistemler yük altında hızlı cevap ve toparlanma gerektiren ya da parametre değişimlerine duyarsızlık gerektiren yüksek performanslı uygulamaların isteklerine cevap veremezler.(Hoque, Zaman ve Rahman,1995)

Son yıllarda kontrol sistemlerinde yapay sinir ağları dikkatleri üzerine çekmiştir. Paralel ve kollara ayrılmış yapısı ile motor kontrol sistemlerinde başarı ile kullanılabilirler. Örneğin model referanslı uyarlamalı kontrol metodu kullanarak DC motor kontrolü yapılabilir. (Weerasooriya, ve El-Sharkawi, 1993)

Yapay sinir ağları girişleri ile çıkışları arasında doğrusal olmayan bir karakteristik gösterir. Yapay sinir ağlarının bu karakteristiği kontrol sisteminin de gürültü, parametre değişimleri ya da yük değişimlerinden etkilenmemesini sağlar.

(Hoque ve diğ er. 1995) Bu tip olumsuz etkenlerin oluş umunu önceden hesaplamak zordur. Örneğ in gürültü birçok sebeple oluş uşabilir. Yüksek performanslı bir kontrol sisteminin bu tip dış etkenlere ald ırmadan, sistemi doğ ru kontrol edebilmesi gerekir. Yapay sinir ağı tabanlı kontrol yapılarının tabiatında gürültüyü kontrolün iç erisine almama durumu vardır. Bu sebeple gürültü altında da çalı ş an, yapay sinir ağı ları ile yapılmı ş kontrol sistemlerinde üstün bir kontrol performansı beklenir. (Weerasooriya, ve El-Sharkawi, 1991) Yapay sinir ağı ları ile bu tip bir kontrol sistemi kurulurken motorun dinamiklerinin bilinmesi ya da ç ıkartılması zorunlu değ ildir. (Lefranc ve Cisternas, 1994) Yapay sinir ağı nonlinear işleyen algoritması ile farklı bir yapı sergiler. Verilen eğ itim verileri ile giriş ve ç ıkış arasındaki yaklaşımı anlayıp, sonrasında bu eğ itim verilerinden farklı bir girdi değ eri verildiğ inde de doğ ru ç ıktı değ erini verebilmektedir.(Buja, 1993)

Geliş en teknoloji ile birlikte robot kollarının kinematik kontrolünde de birçok yöntem kullanılmaya baş lanmı ş tır. Bilinen en eski yöntemlerden birisi kinematik hesapları ile kontroldür. Zeb, Qaiser, Iqbal ve Ahmed(2005), uzaktan kontrol edilebilen, Radyasyon korunma Asistanı adını verdikleri dört serbestlik dereceli bir robotun kontrolünü ileri ve geri kinematik kullanarak yapmı ş lardır. Hongbo (2007), iki kollu beş serbestlik dereceli bir robotu ters pozisyon analizi ile kontrol etmi ş tir. Kosuge ve Furuta (1985), lineer bir robot kolu modelinin kinematik ve dinamik analizlerini yapmı ş tir. Robot kolunun kontrolünde doğ rusal olmayan etkilerin olduğ u durumlarda geliş miş kontrol sistemlerinin kullanılması gerektiğ ini söylemi ş tir. Bu tip durumlar için yapay sinir ağı ları gibi farklı kontrol yöntemleri çözü m olarak geli ş tirilmi ş tir. Seidl, D.R. Reineking, T.L. Lorenz, R.D. (1992), robot kontrolünde yapay sinir ağı ları kullanarak sürtünmenin olumsuz etkilerinden etkilenmeyen bir kontrol yapmı ş tir. Stuart, Vassilios ve Max (1991) , ters kinematiğ in yetersiz kald ığ ı yerlerde yapay sinir ağı ları ile robot kolunun baş arı ile kontrol edilebildiğ ini göstermi ş tir. Takehiko, Hiromichi ve Hajime (2005), 3 serbestlik derecesine sahip bir robot kolunda ters kinematikteki kötü pozisyon problemini yapay sinir ağı ları kullanarak ağı ters döndürme metodu ile çözmüş lerdir. Kinematiğ in yetersiz olduğ u durumlarda yapay sinir ağı larının yanısıra bulanık mantık ile de çözü m üretilebilir.

Örneğin Deok ve Seul (2007), robot kolunun kinematik problemlerini bulanık mantık yöntemi ile çözülebileceğini göstermişlerdir.

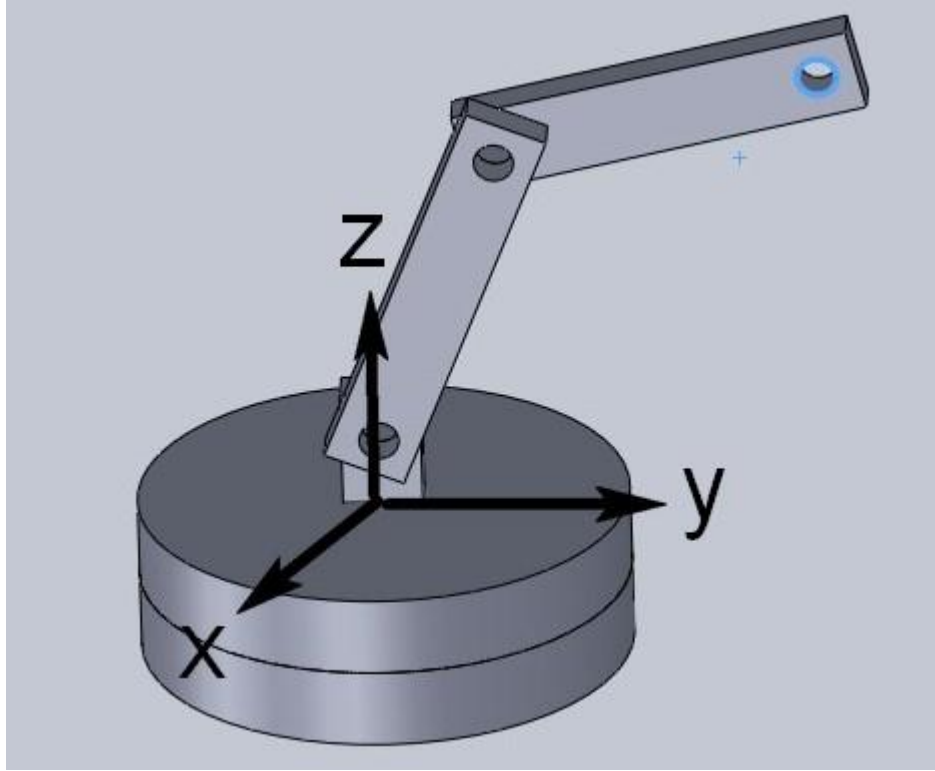
Bu çalışmada, 3 döner eksene sahip bir robot kolunun gerek mekanik olarak, gerekse kontrol sistemi tasarımı açısından nasıl tasarlanabileceği üzerinde tartışıldı. Mekanik tasarım kısmında mekanik parçaların tek tek nasıl ve hangi amaç için tasarlandığı anlatıldı. Robot kolunu oluşturan DC motor, Enkoder, Mini servo motor gibi bileşenler hakkında teknik bilgi verildikten sonra bu bileşenlerin özellikleri ve nasıl seçilebilecekleri üzerinde tartışıldı. Doğru özelliklerde malzeme seçilebilmesi için tork hesaplamalarının nasıl yapılabileceği anlatıldı. Robot kolunun kontrolünde ara bir elektronik devre olarak tasarlanan sürücü devre ve yapısı gösterildi. Kontrol sistemi olarak oransal kontrol, PID kontrol ve yapay sinir ağları ile kontrol yöntemleri ile uygulamalar yapılarak, Matlab ve Simulink ortamlarında sonuçlar birbiri ile karşılaştırıldı. Bir robot kolu kontrolünde EzDSP ve Simulink kullanarak TMS320F28335 işlemcisinin nasıl kullanılabileceği gösterildi. Tasarlanan robot kolu yapay sinir ağları tabanlı model referans kontrolü, PID kontrolü ve geri beslemeli kontrol yöntemleriyle kontrol edildi.

BÖLÜM İKİ

ROBOT KİNEMATİĞİ VE BİLEŞENLERİ

2.1 Robot Kolu Kinematığı

Robot kolu kinematığı, robot kolunun çalışma uzayında robot kolunun hareketleri ile ilgilenir. Bu hareketleri incelerken matematiksel ifadeler, konum vektörü ve yönelim matrisinden faydalanır. Matematiksel olarak nesnelerin konumu bir konum vektörü ile, yönelimi ise bir yönelim matrisi yardımı ile belirlenir. Üç boyutlu uzayda bir robot kolunun ucuna bağlı bir tutucunun yönelimini ve konumunu tanımlamak için, o robot kolunun merkezine Şekil 2.1’de görüldüğü gibi bir koordinat sistemi yerleştirilir. Bu koordinat sistemi sayesinde robot kolu ve bileşenleri arasında bir yönelim ve konum ilişkisi tanımlanır. (Bingöl ve Küçük, 2005)



Şekil 2.1 Robot kolu üzerinde koordinat sistemi

Robot kolu kinematığı ile bir robot kolunun hız, ivme ya da kuvvet analizleri yapılabilir.(Bingöl ve Küçük, 2005) Robot kolu kinematığı merkez nokta, eklemler ve uç işlevci arasında bir ilişki tanımlar. Bir robot kolunun en temel yapısı ele

alındığında, uzuvlardan ve bu uzuvları birbirine bağlayan ve öteleme ya da dönme hareketi yapan eklemlerden oluştuğu görülür. Her bir eklem, kinematik ilişki adı verilen, kendinden önceki ya da sonraki ekleme göre bir ilişki ile ifade edilir. Bu ilişki robot kolunun konum ve yönelim bilgisi içeren 4×4 dönüşüm matrisleri ile gösterilir. Her bir eklem bir dönüşüm matrisine sahiptir. Dolayısı ile dönüşüm matrislerinin sayısı robot kolunun serbestlik derecesi ile doğru orantılıdır. (Bingöl ve Küçük, 2005)

2.1.1 İleri Kinematik

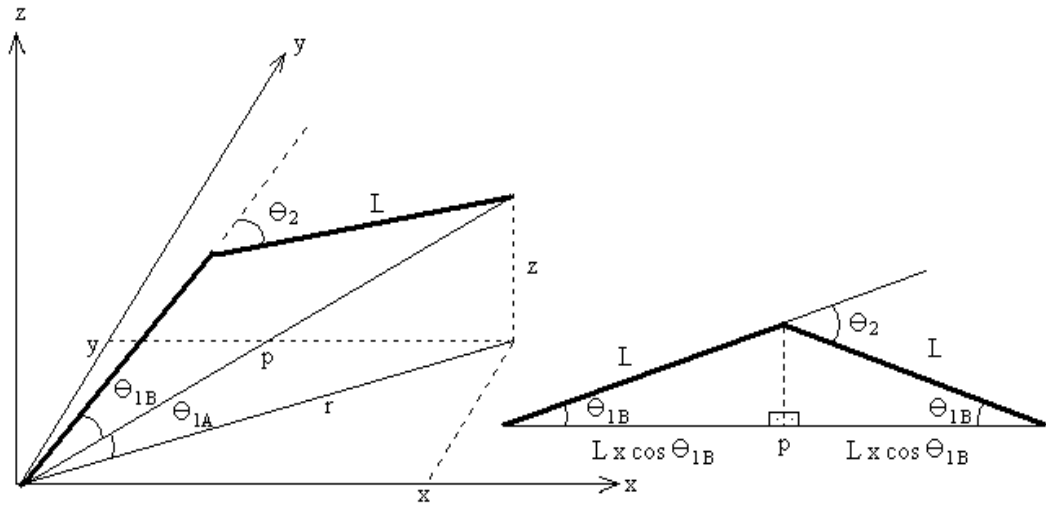
İleri kinematikte, bir robot koluna hedef açı değerleri verilir ve robot kolunun o açı değerlerine gittiğinde robot kolunun uç noktasının hangi koordinatlarda olacağı bilgisine ulaşılır. Eklemler arasındaki ilişki matematiksel olarak trigonometrik çözümler kullanılarak tanımlanabileceği gibi, homojen dönüşüm matrisleri ile de açıklanabilir. Ancak eklemlerin arasında ki bu ilişkiyi matematiksel olarak trigonometrik çözümler ile açıklamak her zaman kolay olmayabilir. Bu durumlarda homojen dönüşüm matrisleri bulunur. Homojen dönüşüm matrislerinin sırasıyla birbiriyle çarpılmaları sonucunda robot kolu merkezi ve uç nokta arasındaki ilişki elde edilir. Bu ilişkiye ileri kinematik adı verilir.

2.1.2 Ters Kinematik

Ters kinematik, bir robot koluna hedef bir son nokta verildiğinde, robot kolunun o noktaya gidebilmesi için, o hedef noktanın koordinatlarını kullanarak eklemlerin açılarının hesaplanması işlemini yapar. Diğer bir deyişle verilen hedef noktadan eklem açılarının çıkarılabilmesini sağlar. Ters kinematik çözümü yapılırken denklemlerin doğrusal olmaması sebebi ile son derece karmaşık çözümler ile karşılaşılabilir. Doğrusal olmayan bu denklemler tek bir çözüme sahip olmayıp, birden fazla sonuç üretebilirler. Robot kolu, verilen bir hedef noktaya farklı eklem açıları ile ulaşılabilir. Bu farklılar denklemlerde birden fazla çözüm olarak karşımıza çıkabilir. Dolayısı ile bir robot kolu eğer öteleme yapan eklemlerden oluşuyorsa ters kinematik çözümü oldukça kolay yapılabilir ancak döner hareket yapan eklemlerden

oluşuyorsa ters kinematik çözümü zorlaşır. Ayrıca ters kinematiği zorlaştıran etkenlerden birisi de her zaman matematiksel sonuçların fiziksel karşılıklarının olmamasıdır. Örneğin; matematiksel olarak bir tutucunun yönü diğer uzuvlar ile ters yönde çıkabilir ancak fiziksel olarak aynı yönde olmak zorundadır. (Bingül ve Küçük, 2005)

Bu çalışmada ters kinematik denklemleri çıkartılırken trigonometrik çözümlerden faydalanılmıştır. Kullanılan ters kinematik hesapları aşağıdaki gibidir.



Şekil 2.2 Robot kolunun koordinat sisteminde gösterimi

$$\theta_2 = 2 * \theta_{1B}$$

$$r = \sqrt{x^2 + y^2}$$

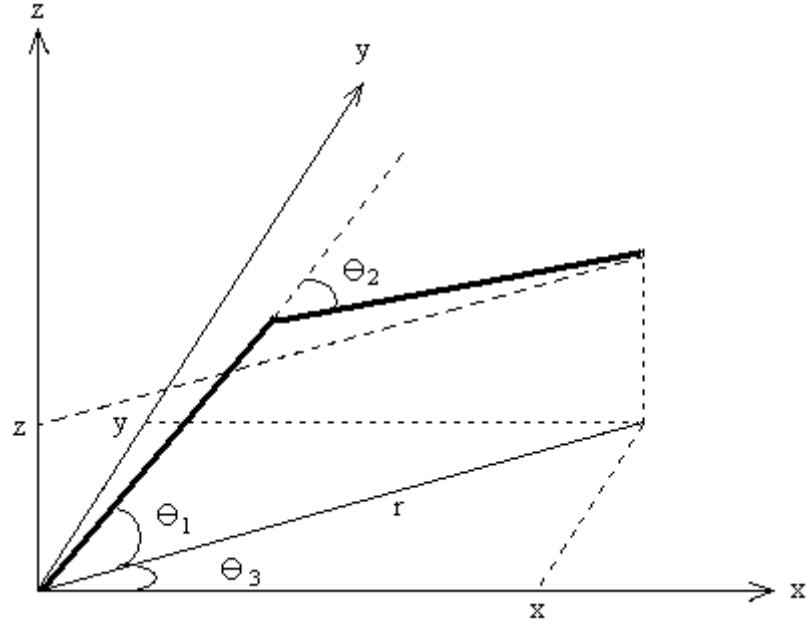
$$p = \sqrt{r^2 + z^2}$$

$$\cos\theta_{1B} = \left(\frac{p}{2}\right)/L$$

$$\cos\theta_{1B} = \frac{\sqrt{x^2 + y^2 + z^2}}{2L}$$

$$\theta_{1B} = \arccos\left(\frac{\sqrt{x^2 + y^2 + z^2}}{2L}\right)$$

$$\theta_{1A} = \text{atan2}(z, r)$$



Şekil 2. 3 Robot kolunun koordinat sisteminde gösterimi

$$\theta_1 = \theta_{1A} + \theta_{1B}$$

$$\theta_1 = \text{atan2}(z, r) + \text{acos}\left(\frac{\sqrt{x^2 + y^2 + z^2}}{2L}\right)$$

$$\theta_1 = \text{atan2}\left(z, \sqrt{x^2 + y^2}\right) + \text{acos}\left(\frac{\sqrt{x^2 + y^2 + z^2}}{2L}\right)$$

$$\theta_2 = 2 * \text{acos}\left(\frac{\sqrt{x^2 + y^2 + z^2}}{2L}\right)$$

$$\theta_3 = \text{atan2}(y, x)$$

Elde edilen denklemler sayesinde, robot koluna koordinat sisteminde bir hedef nokta verildiği zaman eklemlerin hangi açı değerlerine ulaşması gerektiği hesaplanmıştır.

2.2 Robot Kolu Bileşenleri

2.2.1 Elektrik Motorları

Elektrik motorları günümüzde hayatımızın birçok alanına girmiş yapılardır. Kabaca elektrik motorları elektrik enerjisini mekanik enerjiye çeviren makinelerdir. Elektrik motorları çalışma gerilimlerine göre iki farklı şekilde gruplandırılmaktadır:

- Doğru Akım Motorları
- Alternatif Akım Motorları

Adından da anlaşılacağı gibi doğru akım motorları DC voltaj ile çalışan, alternatif akım motorları ise AC voltaj ile çalışan motorlardır. Tezimizde sadece DC motorlar kullanıldığı için, alternatif akım motorlarına yer verilmemiştir.

2.1.1.1 Doğru Akım Motorları

Yukarıda da açıklandığı üzere doğru akım motorları DC voltaj ile çalışan motorlardır. Bu motorlar, girişlerine doğru akım verildiğinde çıkışlarından mekanik enerji üreten yapılardır.

Doğru akım motorları, hız kontrollerinin kolaylığı ve hareketlerinin düzgün, kesin olması sebebi ile endüstride birçok alanda kendine kullanım yeri bulmuştur. Özellikle pozisyonlamanın ve hızın önemli olduğu ya da durma ve çalışma frekansının yüksek olduğu uygulamalarda genellikle doğru akım motorları tercih edilir. Doğru akım motorları genel olarak kolektörü akımla besleyen bir yapıya sahiptir. İşte bu işlem esnasında kıvılcım çıkabildiğinden, yangın çıkma ihtimali yüksek olan yerlerde ve tozun, nemin yüksek olduğu yerlerde doğru akım motorları pek tercih edilmez.

2.1.1.2 Doğru Akım Motorlarının Çalışma Prensibi

Üzerinden akım geçen bir iletken eğer bir manyetik alandan geçirilirse iletkene bir kuvvet etki eder. Bu kuvvetin yönü sol el kuralı ile bulunur. Doğru akım motorunun içerisindeki iletken tel üzerine DC bir voltaj uygulanmaktadır. Bu tel aynı zamanda motorun içerisindeki manyetik alanın da içerisinde kalmaktadır. Dolayısı ile bu iletken tel üzerinden akım geçirildiği zaman, oluşan kuvvetin etkisi ile dönmeye başlar. Eğer akım ters yönde akarsa, motorun dönüşü de ters yönde olacaktır.

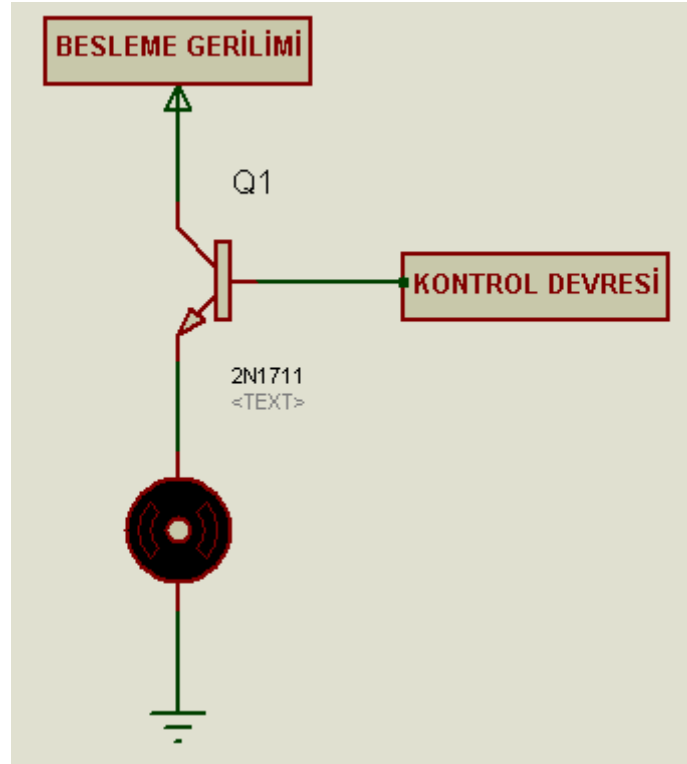
2.1.1.3 Doğru Akım Motorlarının Yön ve Hız Kontrolü

Doğru akım motorları hız ve yön kontrolünün çok kolay yapıldığı motorlardır. Bu motorlarda hız, alternatif akım motorlarından farklı olarak dış etkenlerden çok az etkilenir. Bu sebeple de belli hız değerlerinde çalışması istenen sistemlerde daha çok doğru akım motorları tercih edilir. Yine doğru akım motorlarının yön kontrolleri de çok kolay yapılabilir. Birbirine bağlı olan motor uçları ve besleme gerilimi bağlantıları sökülerek, bu iki kablo tam tersi bir şekilde bağlanırsa doğru akım motorunun dönüş yönü de değişir. Kabaca tek yönlü sürülmüş bir doğru akım motoru Şekil 2.4'te gösterilmiştir. Şekilde de görüldüğü gibi motoru sürmek için bir transistor ve kontrol devresi kullanılmıştır. Kontrol devresi motorun çalışması gerektiği sinyali aldığında transistörün base ucuna sinyal göndererek, transistörü ilettime geçirir. Böylece ilettime geçen transistör üzerinden besleme voltajı motor uçlarına ulaşır. Motora ulaşan besleme voltajı da motorun tek yönlü dönmesini sağlar. Bu işlem, kontrol devresi, transistörün base ucuna transistörü ilettime geçirecek kadar bir sinyal verdiği sürece bu şekilde devam eder. Başka bir deyişle kontrol devresi sinyal verdiği sürece motor dönecektir. Motoru durdurmak için kontrol devresinin verdiği sinyal kesilmelidir. Kontrol devresinin çalıştır ya da durdur komutlarını alabilmesi için bir anahtar ya da sensörden faydalanılabilir.

Şekil 2.5'te ise doğru akım motorunun iki yöne kontrolünü gösterilmiştir. Burada motoru sağa ya da sola döndürebilmek için kontrol devresinde 4 kontrol sinyali gelmektedir. Bu sinyallerin durumlarına göre motorun yönü değişir.

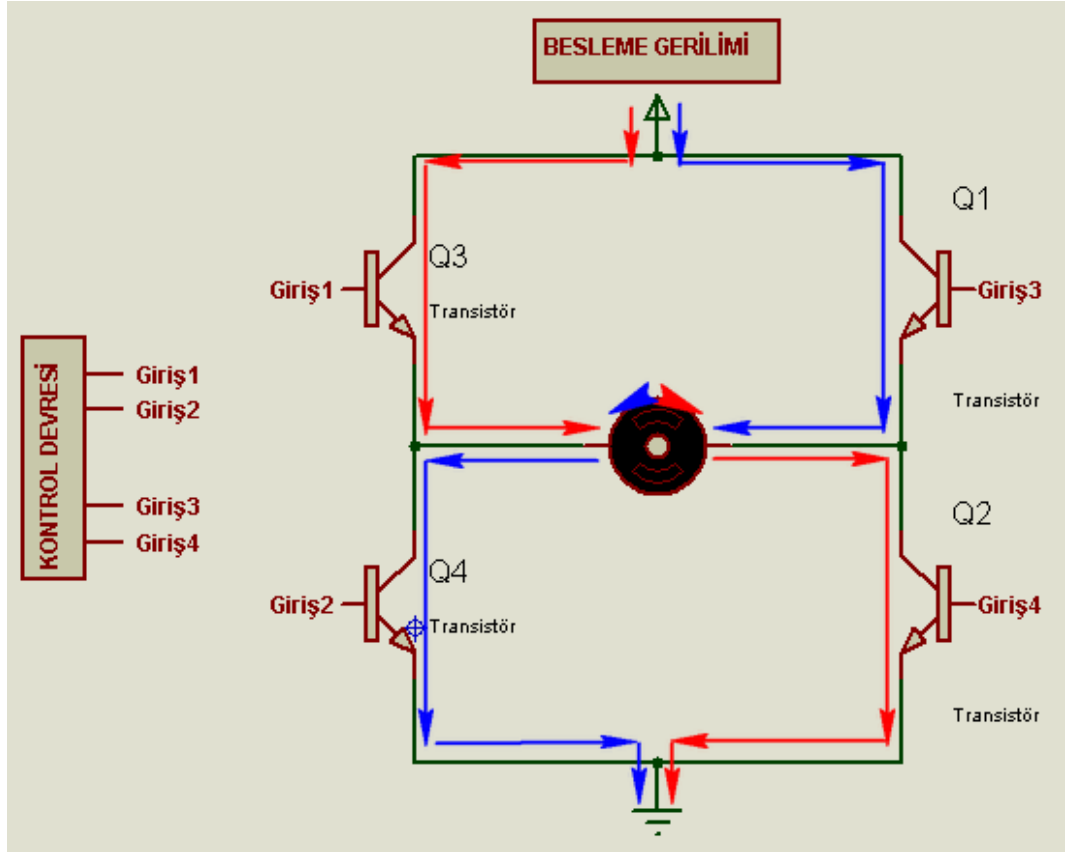
Tablo 2.1 Giriş sinyallerine göre motorun hareketi

Giriş 1	Giriş 2	Giriş 3	Giriş 4	ÇIKTI
0	0	0	0	Motor Dönmez
1	1	0	0	Motor Dönmez
0	0	1	1	Motor Dönmez
1	0	0	1	Motor X yönünde döner
0	1	1	0	Motor -X yönünde döner



Şekil 2.4 Doğru akım motorunun tek yönlü kontrolü

Tablo 2.1’de de görüldüğü gibi motoru hareket ettirebilmek için giriş sinyallerinin ya Giriş1=Giriş4=1 ve Giriş2=Giriş3=0 ya da Giriş1=Giriş4=0 ve Giriş2=Giriş3=1 olması gerekir. Eğer ki sinyaller Giriş1=Giriş4=1 ve Giriş2=Giriş3=0 şeklinde olursa motor bir tarafa doğru dönecektir. Motoru tersine çevirmek için kontrol sinyallerini Giriş1=Giriş4=0 ve Giriş2=Giriş3=1 yapmak gerekir. Giriş sinyallerinin Giriş1=Giriş4=1 ve Giriş2=Giriş3=0 şeklinde olduğu durumda devre üzerinden akan akımı Şekil 2.5’te kırmızı ile gösterilmiştir. Gelen Giriş1 sinyali ile Q3 transistörü iletime geçer ve böylece besleme geriliminin motora ulaşması sağlanır. Akım motor üzerinden geçtikten sonra Giriş4 sinyalinin 1 olması nedeni ile iletimde olan Q2 transistörü üzerinden akarak toprağa ulaşır. Bu akış esnasında da motor bir yöne döner. Motoru tam tersi yöne döndürebilmek için ise girişlerin Giriş1=Giriş4=1 ve Giriş2=Giriş3=0 şeklinde olması gerekmektedir. Burada da gelen Giriş3 sinyali ile Q1 transistörü iletime geçer ve böylece besleme geriliminin motora ulaşması sağlanır. Akım motor üzerinden geçtikten sonra Giriş2 sinyalinin 1 olması nedeni ile iletimde olan Q2 transistörü üzerinden akarak toprağa ulaşır. Bu akış esnasında da motor diğer yöne döner. Bu durumda akımın yolunu Şekil 2.5’te mavi ile gösterilmiştir. Bu devre yapısına H-Köprüsü adı verilir.



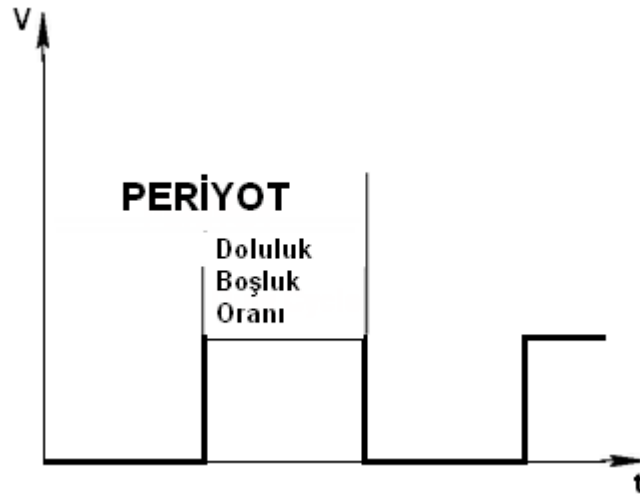
Şekil 2.5 Doğru akım motoru iki yöne kontrolü

Doğru akım motorlarını kullanırken ihtiyacımız olacak olan diğer bir kontrol de hız kontrolüdür. Doğru akım motorlarında hızın kontrol edilebileceği 3 yöntem vardır:

- Motora uygulanan gerilimin değiştirilmesi
- Manyetik alanın değiştirilmesi
- Rotor devresine ilave direnç eklenmesi

Bu üç yöntemden en çok kullanılanı gerilimin değiştirilmesi yolu ile hızın ayarlanması işlemidir. Gerilimin değiştirilmesi işlemi de birçok yöntem ile yapılabilir. Bu işlem için direnç ile gerilim bölme, PWM sinyali ve doluluk boşluk oranı değiştirme, kesici ile gerilim verme gibi bir çok yöntem kullanılabilir. Bu çalışmada DC motorların hızı PWM sinyali ile ayarlandığı için burada da sadece PWM sinyali ile hız kontrolünden bahsedilmiştir.

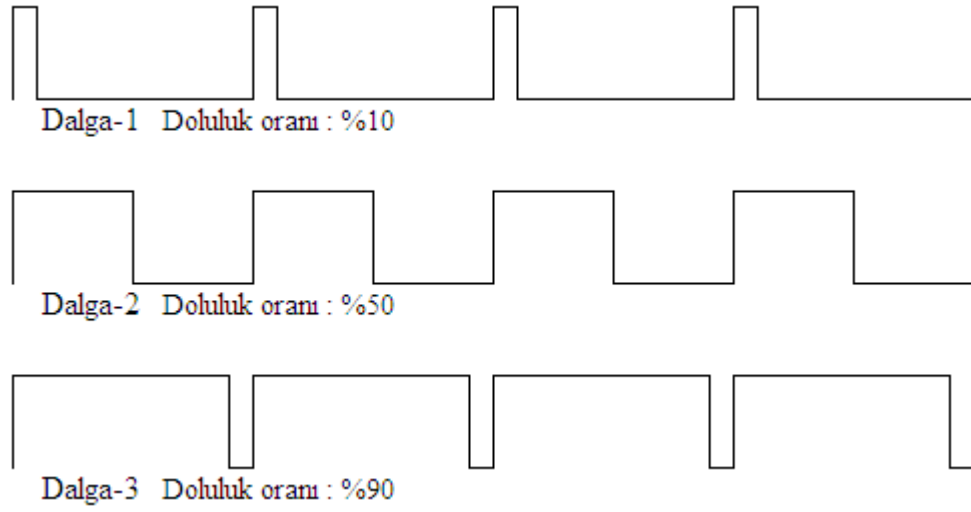
Çok verimli bir teknik olan PWM tekniği mikrokontrollör gibi elemanlar vasıtası ile kolaylıkla uygulanabildiğinden doğru akım motorlarının sürülmesi işleminde uygun bir tercih haline gelmiştir. PWM ismi Pulse Width Modulation kelimelerinin baş harflerinin birleştirilmesi ile oluşmuş olup darbe genişlik modülasyonu anlamına gelmektedir. Oluşturulan darbelerin genişlikleri ile oynayarak, voltaj seviyeleri değiştirilir. Darbe genişliklerine doluluk boşluk oranı adı verilir. Örnek bir PWM sinyali ve doluluk boşluk oranı Şekil 2.6’da gösterilmiştir.



Şekil 2.6 PWM sinyali, doluluk boşluk oranı ve Periyot

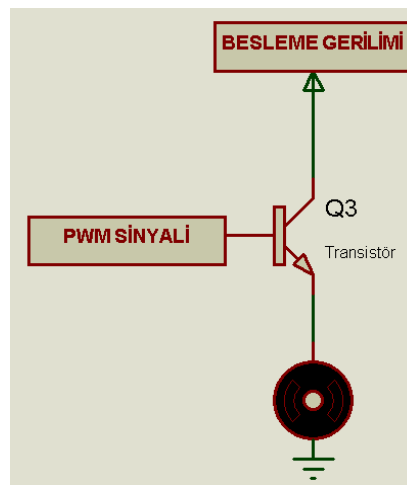
PWM sinyalinin özelliği yüksek frekanslarda oluşturulmasıdır. Sinyal düşük frekanslarda oluşturulursa, motor tarafından iki gerilim seviyesi arasında değişen bir sinyal olarak görülebilir. Ancak yüksek frekanslarda oluşturulan PWM sinyali motora uygulandığında doğru akım verilmiş gibi bir sonuç alınır.

Şekil 2.7’de farklı doluluk boşluk oranı değerlerine sahip PWM sinyalleri gösterilmektedir. Dalga1’de görünen sinyal periyodun %10 gibi bir kısmında 1 olmaktadır. İşte bu durumda doluluk boşluk oranı %10 olur. Aynı mantıkla bakıldığında dalga2’nin doluluk boşluk oranının %50 ve dalga3’ün doluluk boşluk oranının %90 olduğu görülebilir. Doluluk boşluk oranı ile PWM sinyalinin çıktısı olarak görünen DC voltaj değeri aynı orana sahiptir. Örneğin 5V’luk bir güç kaynağı ile oluşturulan %10 doluluk boşluk oranına sahip PWM sinyalinin çıktısı $5V \times \%10 = 0,5 V$ olarak ölçülür.



Şekil 2.7 Farklı doluluk boşluk oranlarında PWM sinyalleri

PWM sinyali ile motor sürülürken kullanılabilecek birçok elektronik malzeme vardır. Örneğin transistor, mosfet, optocoupler ya da özel motor sürücü entegreler gibi birçok malzeme kullanılabilir. Şekil2.8’de transistor ile sürülmüş bir örnek gösterilmiştir. Malzeme seçimi yapılırken dikkat edilmesi gereken en önemli konu, seçilen malzemenin çalışma frekansının bizim uygulayacağımız PWM sinyalinden yüksek olması gerektiğidir. Aksi halde seçilen malzeme PWM sinyalimize yeterli cevap vermeyebilir. Yine malzemenin, yüksek frekanslarda anahtarlama yaptığından, ısınacağı da göz ardı edilmemelidir. Gerek devre tasarımında gerekli önlemler almak suretiyle gerekse soğutucu takılarak ısınma problemine önlem alınmalıdır.



Şekil 2.8 Temsili olarak PWM sinyali ile motor sürme

2.2.1.4 Redüktörler

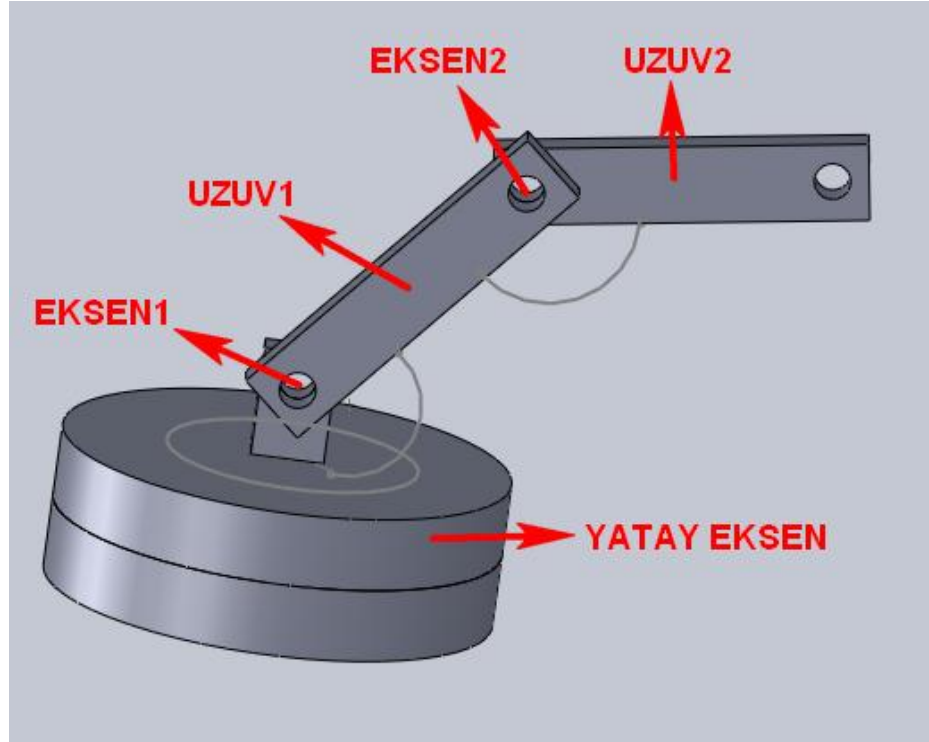
Redüktör motor çıkışlarına monte edilen dişli kutularına verilen isimdir. Genellikle motorun hızını ya da torkunu azaltmak ya da çoğaltmak amacı ile kullanılırlar. Bu değiştirme işleminde motorun hızı ile torku ters orantılıdır. Eğer redüktörün ucundan motordan alınandan daha fazla bir hız alınacaksa, redüktörün ucunda alınabilecek tork motorun çıkışındaki torktan daha düşük olacaktır. Eğer redüktörde hız düşürülüyorsa redüktörün ucundaki tork yükselecektir. Diğer yandan yüksek torklu bir motora ihtiyacınız varsa bunu redüktörsüz kendinden yüksek torklu bir motor olarak karşılamayı düşünürseniz, hem bu motorun hacmi büyük olacaktır, hem de fiyatı çok daha yüksek olacaktır. O sebeple yüksek tork ihtiyacı olan sistemlerde redüktör kullanımı oldukça yaygındır. Bunun haricinde redüktörlerin yön değiştirme gibi farklı işlemler için de kullanıldığı alanlar vardır.

Redüktörün içerisinde farklı çaplardaki dişliler birbirini döndürür. Bu çap farklılığından dolayı girişteki ve çıkıştaki hız ve tork oranları çaplar oranında değişmektedir. Bütün dişlilerin birbirleri ile olan oranları hesaplandığında redüktörün girişi ile çıkışı arasında bir oran bulunur. Bu orana redüktörün redüksiyon oranı denir.

2.2.1.5 Tork Hesapları ve Doğru Akım Motoru Seçimi

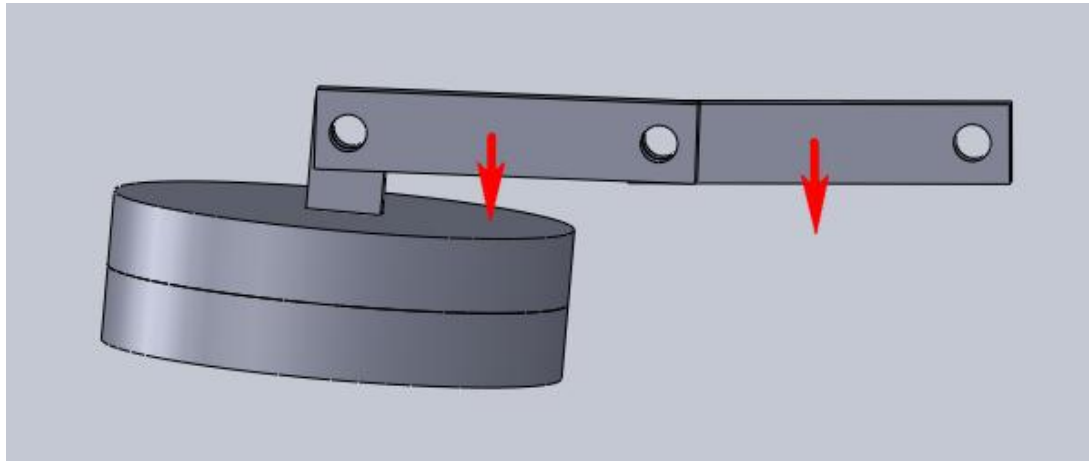
Bir robot kolu tasarlanırken hassasiyet gösterilmesi gereken en önemli konulardan birisi de doğru motor seçimidir. Çünkü kullanılacak olan motorun, robot kolunun yapacağı işlemi karşılayabilecek hızda ve güçte olması gerekmektedir. Aksi halde robot kolu kaldıracığı malzemeyi kaldıramayabilir ya da gideceği yere zamanında gidemeyebilir. Dolayısıyla; eğer robot kolu bir malzemeyi bir yerden başka bir yere taşıyacaksa, malzemeyi kaldırabilir yapıda tasarlanmalıdır. Bunun için de motor seçimi yaparken hem malzemenin ağırlığı hem de robot kollunun malzemelerinin ağırlıkları alınarak yeterli tork değerinde bir motor seçilmelidir. Burada sözü edilen tork motorun dönme momenti anlamındadır. Diğer yandan mekanik tasarımda kullanılacak olan motorların fiziksel olarak yerleştirilmesi de gerektiğinden, mekanik

tasarım ve motor seçimi işlemi paralel olarak yapılmalıdır. Diğer bir deyişle robot kolunun ağırlıklarının düzgün hesaplanabilmesi için mekanik tasarımın da şekillendirilmiş olması gerekmektedir.



Şekil 2.9 Robot kolunun ilk şekli

Şekil 2.9’da görüldüğü gibi bu çalışmada tasarlanan robot kolu 3 eksenli ya da 3 serbestlik dereceli bir robot koludur . En altta yatay eksenle dönmeyi sağlayan motor servo motor, diğer iki motor ise DC motordur.



Şekil 2.10 Ağırlıklar

Şekil 2.10’da da görüldüğü gibi bu aşamada robot kolunun mekaniğinden kaynaklanan iki adet ağırlık vardır. Ancak robot kolu tamamlandığında bağlantı noktaları, miller, motor ve enkoderlar da olacaktır. Şu aşamada bağlantı noktalarının ve millerin ağırlıkları bilinemediğinden çıkan tork değerine belli bir tolerans işlemi yaparak motorların ne olacağına karar verilmiştir.

Şekil 2.10’da görülen pozisyonda yer çekim kuvvetinin tümü uzuvlar üzerinde olduğundan tork hesabının bu pozisyonda yapılması bize en doğru sonucu verecektir. Yapılan ölçümler sonucu kullanılan parçaların ağırlıkları aşağıdaki gibidir:



Şekil 2.11 Uzuvların ağırlığı

- Prototip olarak hazırlanan uzuv ağırlıkları 90 gram ölçüldü.
- Prototip olarak hazırlanan uzuv uzunluğu 220 mm’dir.

Şekil 2.10’da da görüldüğü gibi eksenler arasında en yüksek torka ihtiyacı olan eksen, eksen1 eklemidir. Dolayısı ile bu eklemden, eksen2 ekleminde kullanılan motordan daha yüksek torklu bir motor kullanılmıştır. Aynı zamanda eksen2 ekleminde kullanılan enkoder da, eksen1 motoruna ciddi bir yük oluşturmaktadır. Bu

nedenle eksen2 eklemine kendinden enkoderlı bir motor kullanılmıştır. Bu sayede hem enkoder ağırlığı düşürülmüş hem de enkoder bağlantısının yapılacağı aparatların ve vidaların ağırlıklarından kaynaklanan yükten kurtulunmuştur. Buradan hareketle robot kolu boşta çalışırken motorlarda gerekli olan tork değerleri aşağıdaki gibi bulunmuştur.

- Eksen2 Motor Tork = (Uzuv2 uzunluk / 2) x Uzuv2 Ağırlık
- Eksen2 Motor Tork = (22 cm / 2) x 90 x 10⁻³ kg
- Eksen2 Motor Tork = 990 x 10⁻³ kg.cm
- Eksen2 Motor Tork = 0,99 kg.cm = 0,098096 Nm = 13,877 oz.in

Bu tork değeri robot kolu boşta iken uzuv2 'yi hareket ettirebilmek için gereken en küçük tork değeridir. Eğer robot kolu ile malzeme kaldırma yapılacaksa tork hesabı yapılırken kesinlikle yüklerin ağırlıklarının da hesaba katılması gerekmektedir. Burada eksen2 için Nidec-Servonun DME34BE50G isimli motor kullanılmıştır. Bu motor kendinden tek kanallı enkoderli ve üzerine redüktörü takılmış bir motordur. Bu sayede ayrı bir enkoder kullanmaya gerek kalmamıştır.

Motorun ağırlığı 300 gr, motordan alınabilecek en yüksek tork değeri 9.8 kg.cm ve 12V gerilimde çalıştırıldığında devir sayısı dakikada 92,6' devirdir. Bu devir sayısı, robot kolu için yüksek bir devir sayısıdır. Bu sebeple Mekanik tasarım aşamasında araya ek dişliler koyarak devir sayısı düşürülmüştür.

Şimdi de eksen1 'de kullanılan motora bakalım.

- Eksen1 Motor Tork = ((Uzuv1 uzunluk + Uzuv2 uzunluk / 2) x Uzuv2 Ağırlık) + ((Uzuv1 uzunluk / 2) x Uzuv1 Ağırlık) + (Eksen2 motor ağırlığı x uzuv1 uzunluk)
- Eksen1 Motor Tork = ((22 cm + 22 cm / 2) x 90 x 10⁻³ kg) + ((22 cm / 2) x 90 x 10⁻³) + 300 x 10⁻³ x 44 cm
- Eksen1 Motor Tork = 1,98 kg.cm + 0,99 kg.cm + 13,2 kg.cm
- Eksen1 Motor Tork = 16,17 kg.cm = 1,5856 Nm = 224 oz.in

Bu tork deęerleri robot kolu bořta iken uzuv1 'i hareket ettirebilmek iin gereken en kk tork deęeridir. Eęer robot kolu ile malzeme kaldırma yapılacaksa tork hesabı yapılırken kesinlikle yklerin aęırlıklarının da hesaba katılması gerekmektedir. Eksen1 iin řekil 2.12'deki motor kullanılmıřtır. Bu motorun tork deęeri 27 kg.cm ve dakikadaki devir sayısı 35 rpm'dir. Yine mekanik tasarım yapılırken bu eksen de diřliler kullanılarak hem tork deęeri arttırılmıř hem de hızı dřrlmřtr.



řekil 2.12 Eksen1' de kullanılan motor

2.2.2 Servo Motorlar ve Motor Seimi

İngilizce de hizmeti, uřak anlamına gelen servant kelimesinden ismini almıř servo motorların gnmzde aktif olarak kullanılan farklı eřitleri vardır. Buradaki robot kolunda mini DC servo motor kullanılacaęından, sadece mini DC servo motordan bahsedilmiřtir.

Mini DC servo motorlar, bir DC motor ve mili zerine baęlanmış diřli sistemi, bir potansiyometre ve kontrol devresinden oluřan yapılardır. zerine baęlanmış olan

dişli sistemi büyük redüksiyon oranlarına sahip olduğundan bu servo motorlardan, boyutu ile karşılaştırıldığında çok ciddi torklar elde etmek mümkündür.

Yukarıda da belirtildiği gibi servo motorlar içlerinde bir potansiyometre bulundurulur. Bu potansiyometre motorun pozisyonunu belirlemek için kullanılır. Yine servo motorların çalışma prensipleri daha önce bahsedilen doğru akım motorlarından farklıdır. İçerisindeki kontrol kartı sayesinde servo motor verilen pozisyona direkt gider ve o pozisyonda kalır. Pozisyon bilgisi ise PWM sinyalleri vasıtası ile verilir. Verilen PWM sinyali ve servo motorun o sinyale cevabı markalara göre hatta bazı markalarda modellere göre değişkenlik gösterir. Bu nedenle seçilen motorun teknik dokümanlarındaki değerlere göre PWM sinyali oluşturmak en doğrusudur. Bu servo motorlar, verilen PWM sinyalinin doluluk boşluk oranına göre belli bir açı değerine giderler. Bunun yanı sıra mini DC servo motorlar, normal DC motorlarda olduğu gibi sonsuz tura sahip değildir. Gidebilecekleri açı değerlerinin alt ve üst limitleri vardır ve bu limitlerin dışına çıkamazlar. Bu limitler dişli kutusunun içerisindeki mekanik parçalar ile sağlanmıştır ve bu parçalar motorun belli açılar dışına çıkmasına engel olur. Bu alt ve üst limit değerleri yine marka ve model bazlı olarak değişkenlik göstermektedir. Motor seçerken yapılacak işleme göre en uygun açı değerlerinde bir motor alınmalıdır.

Bu çalışmada en altta kalan yatay ekseninde bir adet servo motor kullanılmıştır. Mini servo motor olarak Hextronik markasının HX12K modeli seçilmiştir. (Şekil_2.13) HX12K verilen voltaj değerine göre 10 Kg.cm tork verebilen metal bir mini servo motordur. Yüksek tork özelliği ve benzer torklardaki diğer firmaların motorları ile karşılaştırıldığında fiyatının çok düşük olması, bu motorun en önemli avantajlarından birisidir.

HX12K servo motoru benzer fiyatlarda olan diğer markaların motorlarından ayırt eden en önemli özelliklerden birisi de metal dişliye sahip olmasıdır. Bu sayede dayanıklılığı da arttırılmıştır.

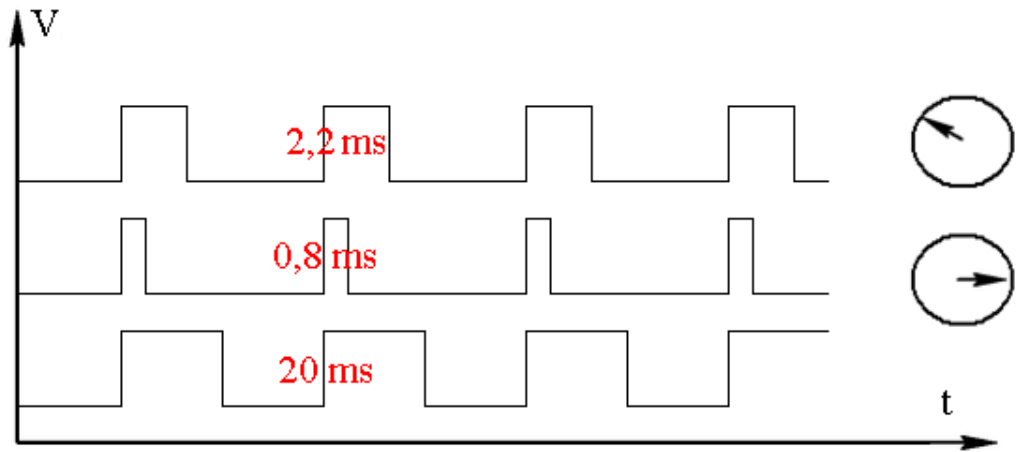


Şekil 2.13 HX12K Servo motor

Servo Motores,25.12.2013

<http://badulaquesdachina.blogspot.com/2010/03/servo-motores.html>

HX12K PWM sinyali olarak 2 ms'lik bir sinyal bekler. PWM sinyalinde 0,8 ms'lik bir doluluk boşluk oranı verilirse HX12K "0" dereceye gider, 2,2 ms'lik bir doluluk boşluk oranı verilirse HX12K 135 dereceye gider. Doluluk boşluk oranı ve açı ilişkisi Şekil 2.14'te verilmiştir.



Şekil 2.14 Doluluk boşluk oranı ve açı ilişkisi

HX12K servo motorunda sarı, kahverengi ve kırmızı renkte olmak üzere üç adet kablo vardır. Bu kabloların renklerine göre bağlantıları aşağıdaki gibidir.

Kırmızı kablo : Besleme voltajı

Sarı kablo : Kontrol sinyali
Kahverengi kablo : Toprak

2.2.3 Enkoderlar

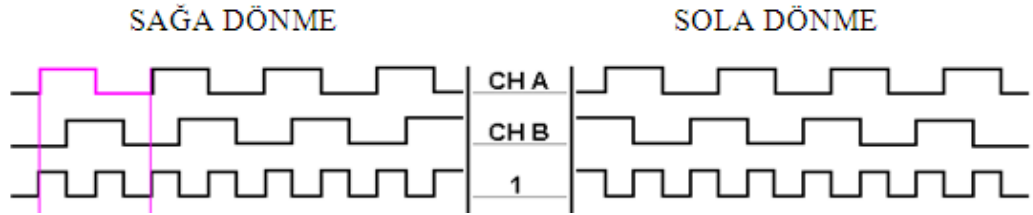
Enkoderlar, milinin dönmesi ile dijital bir elektrik sinyali üreten yapılardır. Genellikle bir motorun miline ya da hareket eden bir parçaya bağlanarak, malzemenin / milin pozisyon bilgisi; hareket hızı / ivmesini bulabileceğimiz dijital sinyaller üretirler. Bu dijital sinyaller doğru okuma tekniği ile okunarak istenilen bilgiler elde edilir. Enkoderlar çalışma Şekillerine göre, şaft enkoderlar ve lineer enkoderlar olmak üzere iki gruba ayrılır. Eğer enkoderlar bilgiyi okuma şekli olarak ayrılırsa optik enkoderlar ve manyetik enkoderlar olmak üzere iki başlıkta incelenebilir. Yine optik enkoderlar ise mutlak tip ve artımsal tip olmak üzere ikiye ayrılır. Piyasada bunlar haricinde de enkoder tipleri vardır.

2.2.3.1 Artımsal Tip Optik Enkoderlar

Optik enkoderlar içerisinde bulundurduğu hareketli disk sayesinde okuma yapabilen yapılardır. Kabaca çalışma prensibi; Bir ışık kaynağından çıkan ışıklar foto transistörler tarafından algılanır. Işık kaynağı ile foto transistörler arasına üzerinde delikler bulunan bir disk konulursa, ışık sadece diskin delikli kısımlarında karşı tarafa yani foto transistöre ulaşabilir. Delik olmayan kısımlarda foto transistör kapalı kalır. İşte optik enkoderların çalışma mantığı da budur. Ortadaki enkoderin miline bağlı olan disk dönerken, ışık eğer delikli bir bölgeye denk gelirse karşı tarafa geçer ve foto transistöre ulaşır. Eğer deliksiz bir tarafa denk geldi ise foto transistöre ulaşamaz. Bu çalışmada artımsal tip enkoder kullanılmıştır.

Artımsal tip enkoderlar içerisinde bulunan bir disk vasıtası ile ışığın foto transistöre ulaşması mantığı ile okuma yapar. Artımsal tip enkoderlarda kanallarda sadece farklı başlama noktaları olan periyodik delikler vardır. İşte artımsal tip enkoderlarda, enkoderin yönünü ve pozisyonunu bulmak için ışık bu deliklerden geçip transistöre ulaştıktan sonra bu verileri alıp işlem yapmak gerekmektedir.

Artımsal tip enkoderlar kanal sayılarına göre X ya da XY ya da YXZ şeklinde isimlendirilirler. X tipi enkoderlarda tek kanal vardır ve sadece hız ölçümü yapılabilir. XY tipi enkoderlarda ise iki kanal vardır ve aynı anda hem hız hem de yön kontrolü yapılabilir ancak bu enkoderlar tur bilgisi vermezler. XYZ tipi enkoderlarda ise hem yön hem hız bilgisi alınabileceği gibi tur bilgisi de alınabilir.



Şekil 2.15 XY tipi enkoderdan bilgi okuma.

Şekil 2.15’te bir X,Y tipi enkoderdan bilgi okuma gösterilmektedir. CHA ve CHB enkoder çıkışlarında görülen sinyallerdir. 1 numaralı sinyal ise dışarıda işlenmiş bir sinyaldir. Görüldüğü gibi 1 numaralı sinyal CHA ve CHB çıkışlarının yükselen ve alçalan kenarlarında değişim gösteren bir sinyaldir ve çözünürlüğü CHA ve CHB’nin çözünürlüğünden 4 kat daha yüksektir. İşte bu şekilde enkoderdan gelen sinyallerin çözünürlüğünü artırarak pozisyon kontrolü yapılırsa, 4 kat daha net bilgiler elde edilir. Yine bu tip bir enkoderda yön tayini yapmak için CHA ve CHB sinyallerinin arasındaki faz farkını incelemek gerekir. Şekil 2.15’teki enkoder incelenecek olursa, bu enkoderin sağa dönerken CHA sinyalinin CHB sinyalinden 90 derece önde gittiği görülür. Enkoderin yönünün tayini için birçok yöntem kullanılabilir ancak en kolay yöntemlerden birisi de CHA ve CHB sinyallerinin Xor’unu almaktır. XOR tablosu Tablo.2.2’de gösterilmiştir.

Tablo 2.2 XOR işlemi

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

Buradan da görülebileceği gibi Şekil 2.15’deki 1 numaralı sinyal aslında CHA ve CHB sinyallerinin XOR’lanması ile oluşmuş bir sinyaldir. Yine yön tayin etmek için

CHA ve CHB çıkışlarından gelen XY değerlerinin bir önceki değerleri ile karşılaştırılabilir. Karşılaştırma işleminde XOR operatörünün kullanımı yön tayin edilmesini çok kolaylaştırmaktadır. Örneğin X_0Y_0 ve X_1Y_1 çıkışları için yön tayini şu şekilde yapılır. Örneğin; eğer $(X_0 \text{ XOR } Y_1) = 0$ ise enkoder saat yönünde dönüyorsa $(X_0 \text{ XOR } Y_1) = 1$ olduğunda enkoder saat yönünün tersi yönde dönecektir. Diğer bir deyişle, bir önceki sinyalin sıfıncı biti ile son sinyalin birinci bitinin XOR çıkışının değişmesi ile enkoder yön değiştirmiş olur, bu sayede de dönüş yönü tespit edilebilir.

2.2.3.2 Enkoder Seçimi

Daha önceden de belirtildiği gibi bu çalışmada iki adet enkoder kullanılmıştır. Bu bölümde eksen1'de kullanılacak olan enkoder belirlenmiştir. Bu enkoder belirlenirken özellikle çözünürlüğünün çok düşük olmamasına dikkat edildi. Aksi halde net bir pozisyon bilgisi alınamazdı. Sonunda Opkon markalı 1024 pulse/tur sinyal çıkışlı PRI50 modelin kullanılması uygun bulunmuştur.



Şekil 2.16 Opkon PRI50

Şekil 2.16'da Opkon PRI50 gösterilmiştir. Opkon PRI50 enkoder tur başına 1024 pulse verebilir ve A,B ve X olmak üzere 3 çıkış kanalı vardır. Bu sayede hem yön kontrolü yapılır hem de hız kontrolü yapılabilir ve enkoder bu sırada tur bilgisini de verir.

BÖLÜM ÜÇ

MEKANİK TASARIM

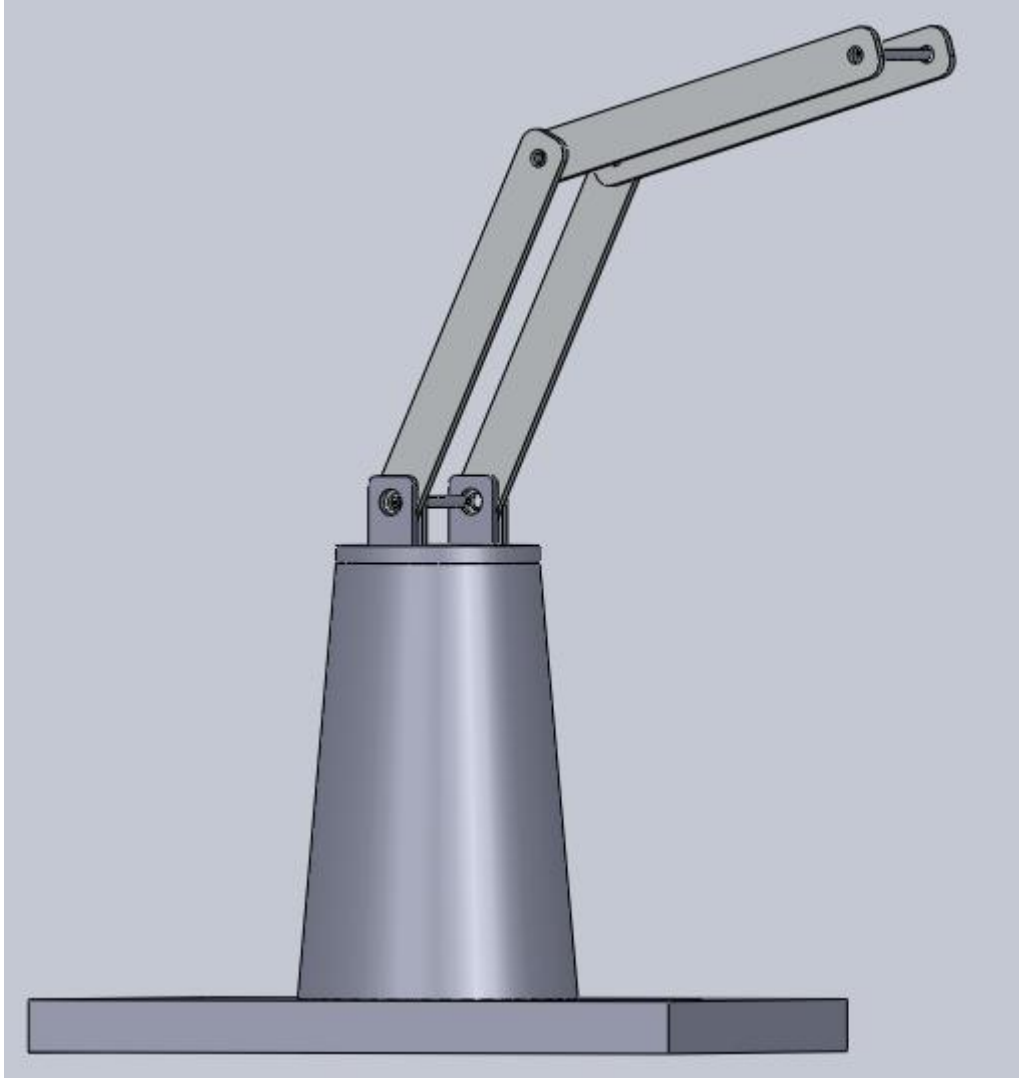
3.1 Robot Koluna Genel Bakış

Bir Robot kolunun mekanik tasarımı hiç kuşkusuz robot kolunun çalışmasındaki en önemli etkenlerden birisidir. Yanlış yapılmış bir mekanik tasarımın telafisi olmayabilir. Tüm sistem tekrardan yapılmak zorunda kalınabilir. Ancak günümüzde gelişen bilgisayar yazılımları 3D programları yapılmış olan mekanik tasarımı fiziksel olarak yapmadan önce simule edebilmeyi sağlamıştır. Böylece tasarımda yapılan yanlışlıklar önceden görülebilir ve müdahale edilebilir.

Yine mekanik tasarım yapılırken robot kolunun çalışacağı alan ve yapacağı işin gerekliliklerinden başka, malzeme seçimi, yer çekimi, hız, sürtünme, ivme gibi konular da hesaba katılmalıdır. Örneğin ağırlık kaldıracak robot kollarında eğer yüksek torklu motorlar seçilmediyse kullanılacak malzemenin de mümkün olduğunca hafif olması önemlidir. Diğer bir örnek; çok hızlı çalışacak olan motorlarda sürtünmenin en düşük seviyelere indirilmesi robot kolunun uzun ömürlü olması için önemli bir faktördür.

Bu çalışmada, robot kolunun tasarımında sadece orta millerde demir kullanılmıştır. Diğer malzemeler, hem DC motorlara çok yük binmemesi hem de bu malzemelerin işleme kolay olduğundan dolayı derlin , alüminyum gibi malzemelerden seçilmiştir.

Robot kolunun kabaca tasarımı daha önce Şekil 2.9'da verilmişti. Tasarım aşamasında bu yapı biraz daha detaylandırıldı ve son hali Şekil 3.1'de gösterilmiştir. Ancak bu tasarım robot kolunun son tasarımı değildir. Tek tek parçaları tasarlarken üzerinde değişiklikler yapılacaktır.

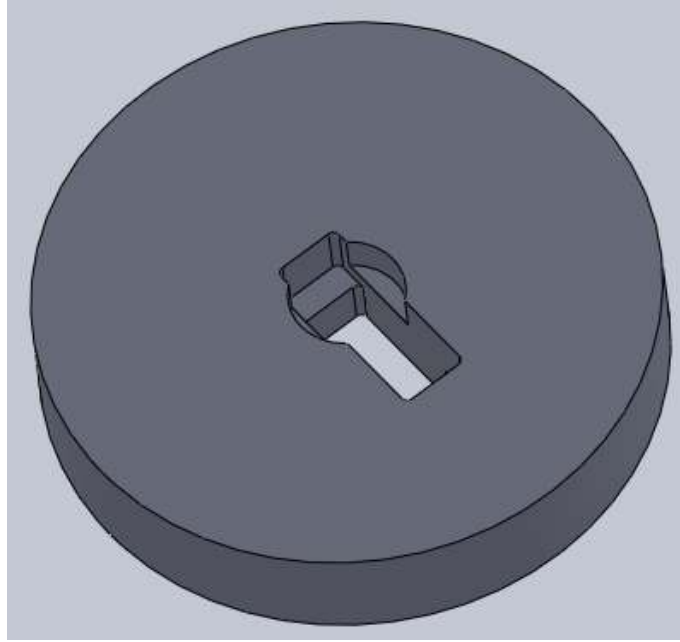


Şekil 3.1 Robot kolunun geliştirilecek olan ilk tasarımı

3.2 Parçaların Tasarımı ve Birleştirilmesi

İlk olarak üzerinde çalışılan parça alt tabladır. Alt tabla robot kolunun en altında bulunan ve gövdenin üzerine oturacağı tabladır. Alt tabla aynı zamanda ağırlığı ile robot kolunun ağırlık merkezini ortaya doğru çekme işlemini de gerçekleştirecek olan kısımdır. Dolayısı ile alt tabla ne kadar ağır olursa robot kolunun devrilmesi o kadar zorlaşacaktır. Alt tablanın içerisine aynı zamanda, yatay ekseninde dönmeyi sağlayacak olan servo motor da konulmuştur. Bu motor hareketli bir eksen olduğundan, motorun oturtulacağı yatakta kesinlikle oynama olmaması gerekmektedir. Bu sebepten alt tablayı ağır bir metal olan demirden yapmak yerine

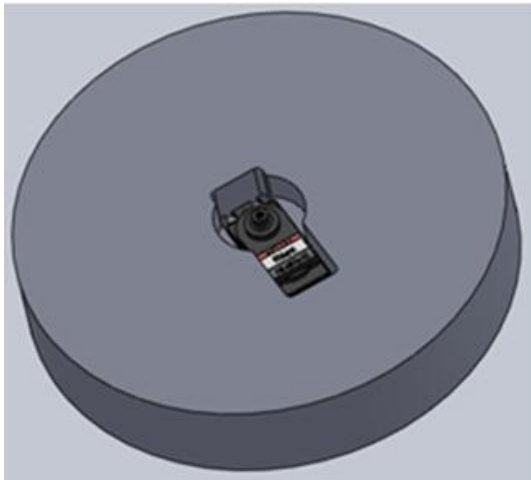
işlemesi kolay olan derlinden yapıp kalınlığı fazla tutarak ağırlık problemi de çözülmüştür. Son durumda yapılan alt tabla Şekil 3.2’de gösterilmiştir.



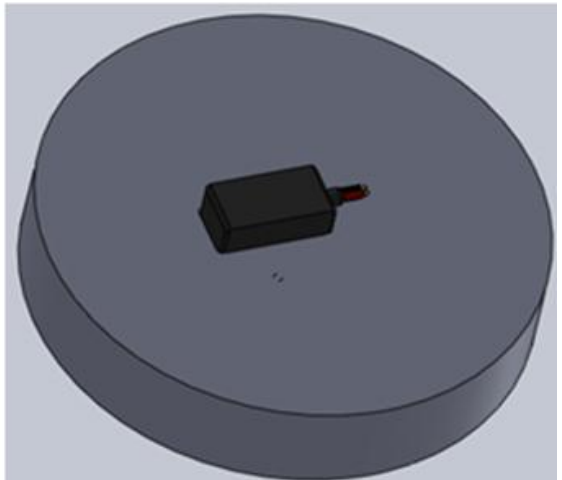
Şekil 3.2 Alt tabla

Alt tablaya seçilen HX12K servo motor bire bir oturacak şekilde bir yatak yapıldı. Motor yerleştirildikten sonra son görünüm Şekil 3.3’te gösterilmiştir.

ÜSTTEN GÖRÜNÜM

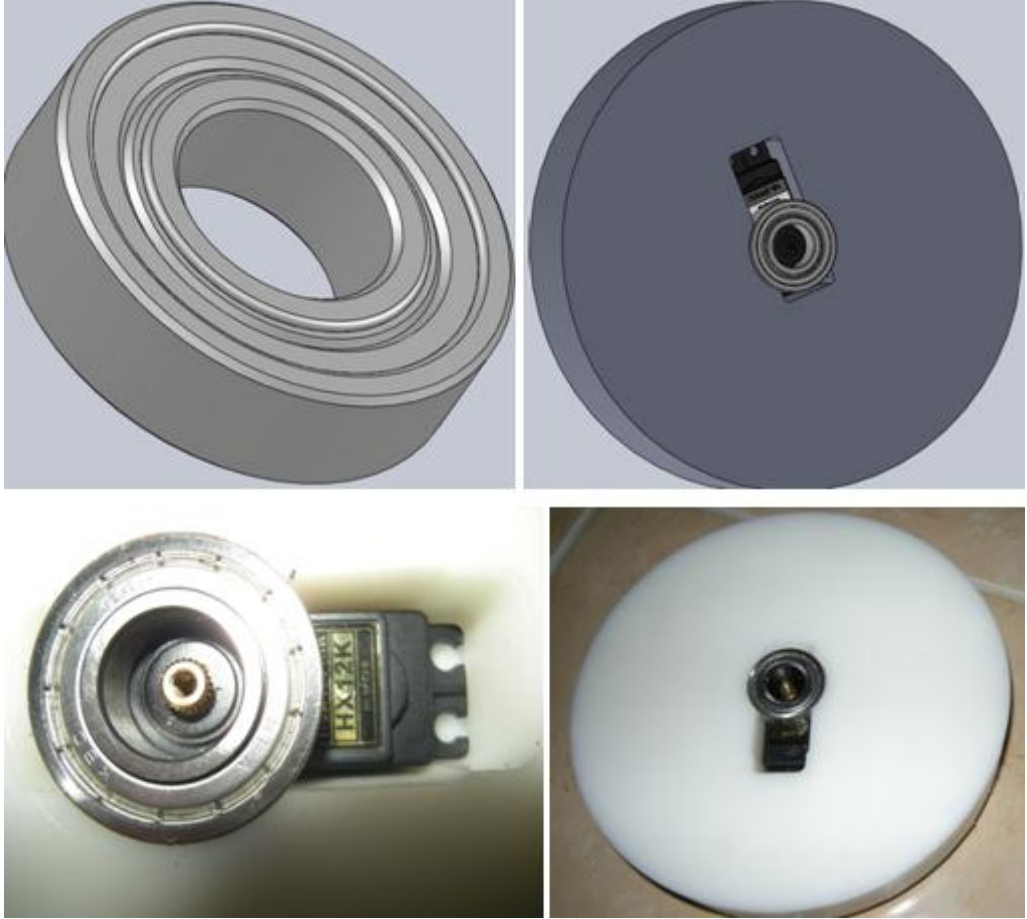


ALT TAN GÖRÜNÜM



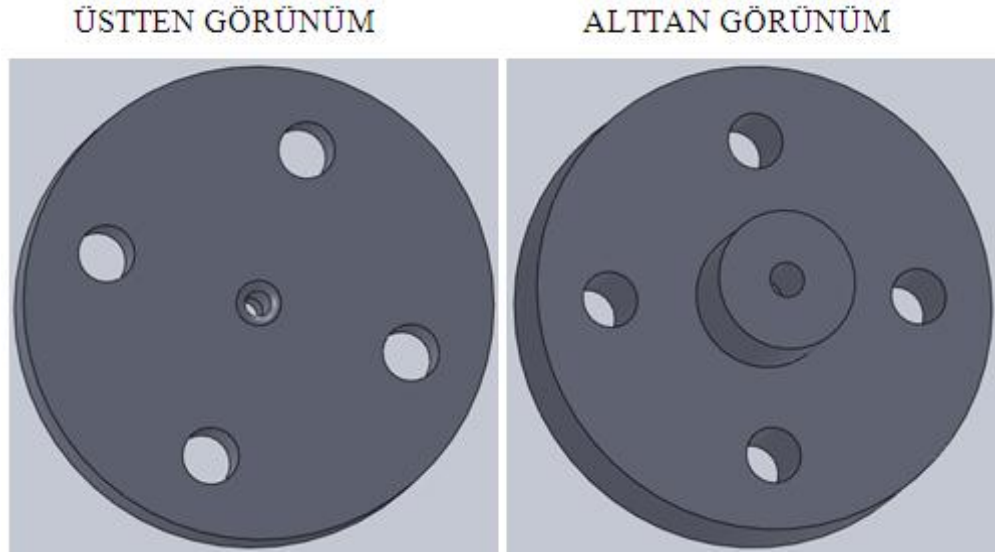
Şekil 3.3 Alt tabla ve motorun montajı

Şekil 3.4'te görüldüğü gibi alt tablanın ortasına, merkezi motorun milinin merkezine denk gelen bir daire işlendi. Bu daireye rulman oturtuldu. Servo motor ile gövde arasındaki bağlantı bu rulman sayesinde yapıldı.

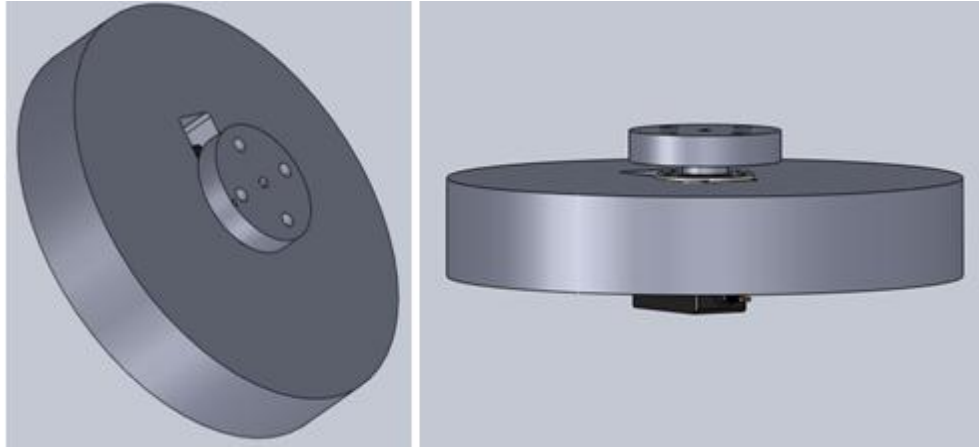


Şekil 3.4 Alt tabla rulmanı ve alt tablaya montajı

Alt tablada rulmanın iki amacı vardır. Birincisi yer çekiminin etkisi ile oluşacak yükler bu rulman üzerinden alt tablaya aktarılır ve motora gelmez, ikincisi robot kolunun alt tablaya sabitlenmesini sağlar. Dolayısı ile alt tabla ile gövde hiç birbirinden bağımsız hareket edemez. Bu ikinci amacın tam olarak yapılabilmesi için rulman ile gövde arasındaki aparatın da oluşturulması gerekir. Rulman - gövde arası aparat Şekil 3.5'te verilmiştir. Şekil 3.6'da aparatın alt tablaya bağlantısı görülebilir.

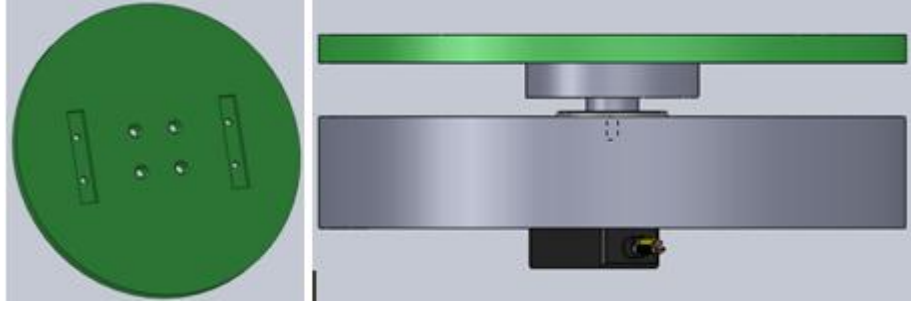


Şekil 3.5 Rulman ile gövde arasındaki aparat



Şekil 3.6 Rulman ile gövde arasındaki aparatın alt tablaya bağlantısı

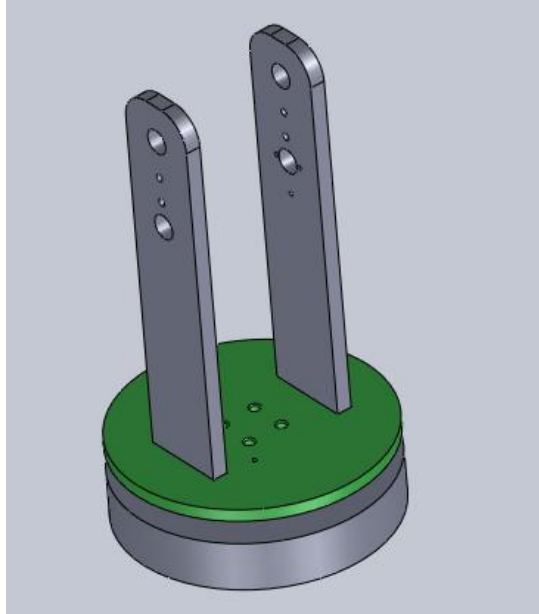
Bu aşamada aparatın üzerine bağlantı yapılabilecek ve aynı zamanda gövde kolanlarının üzerine montajlanabileceği gövde tablasına ihtiyaç vardır. Şekil 3.7’de tasarlanan gövde tablası gösterilmiştir.



Şekil 3.7 Gövde tablası ve alt tablaya montajı

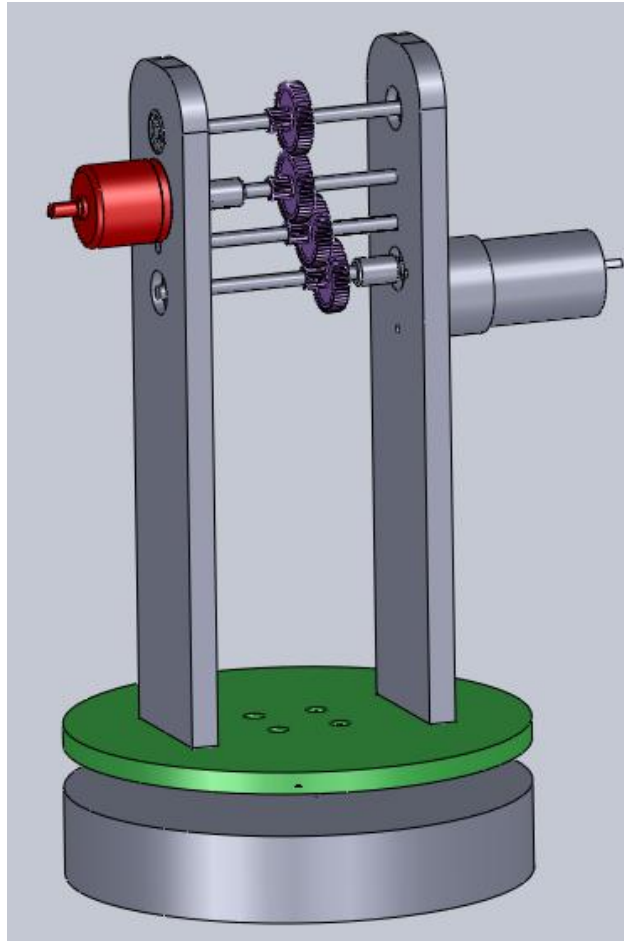
Şekil 3.7’de de görüldüğü gibi gövde tablasının tam ortasında altındaki aparata tutturulmak amacı ile 4 adet vida deliği açıldı. Yine gövde kollarının sağa sola oynamasını engellemek amaçlı iki adet yatak açıldı. Gövde kolları bu yataklara oturtuldu ve alttan iki adet vida ile sabitlendi.

Son durumda gövde iki kolon üzerinde taşınarak, robot kolunun yüksekliği biraz arttırıldı. Burada gövde kolları yine ağırlık merkezini etkileyeceğinden mümkün olduğunca kalın malzemeden yapıldı. Malzeme olarak alüminyum kullanıldı. Gövde kolları aynı zamanda robot kolunun eksen1’inin de bağlı olacağı kollardır. Bu sebeple bu kolların tasarımında bu malzemelerin de yerleştirilebileceği uygun delikler açıldı. Şekil 3.8’de gövde kollarını ve alt tablaya yerleştirilmiş hali gösterilmiştir.



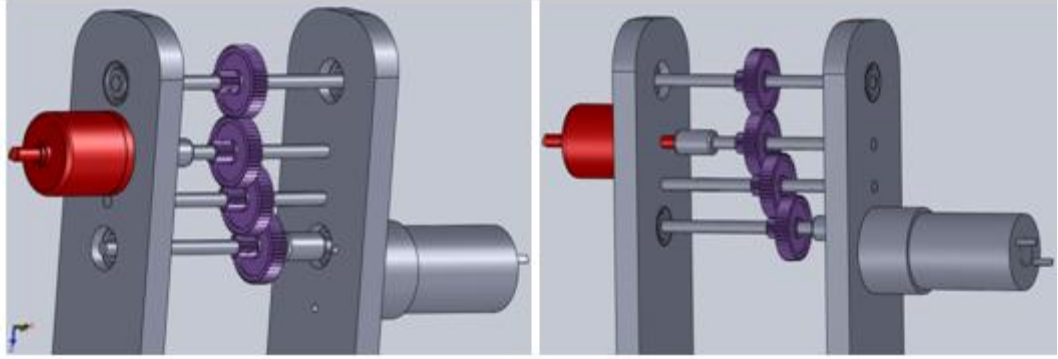
Şekil 3.8 gövde kolları ve alt tabla

Şekil 3.8’de de görüldüğü gibi gövde kollarının üst tarafında delikler vardır. Bu deliklerin en altındaki iki deliğin birine motorun mili takıldı ve 5 mm’lik bir mil vasıtası ile karşısındaki deliğe geçirildi. Karşısındaki deliğe takılan bir rulman vasıtası ile motor döndüğünde milin de dönmesi sağlandı. Aynı zamanda bu mil ve diğer millere de dişliler takılarak hem buradaki döner hareket diğer millere, oradan da uzuv1’e taşındı hem de tork arttırılıp hızı yavaşlatıldı. Şekil 3.8’de görüldüğü gibi bu deliklerden yukarı doğru 4 çift delik açıldı. Bunların alttan ikincisi sadece hareketi taşımak ve hızı yavaşlatmak amacı ile konuldu. Üçüncüsüne ise enkoder bağlandı. Dördüncü ve en üstteki delik çifti ise hareketi uzuv1’e taşımak görevini üstlenen deliktir. Bu dördüncü delik çiftinde aynı zamanda iki adet rulman yatağı deliği vardır. Son durumda motor, enkoder, miller, dişliler, kaplinler ve rulmanların takılı olduğu sistemin son hali Şekil 3.9’da verilmiştir.



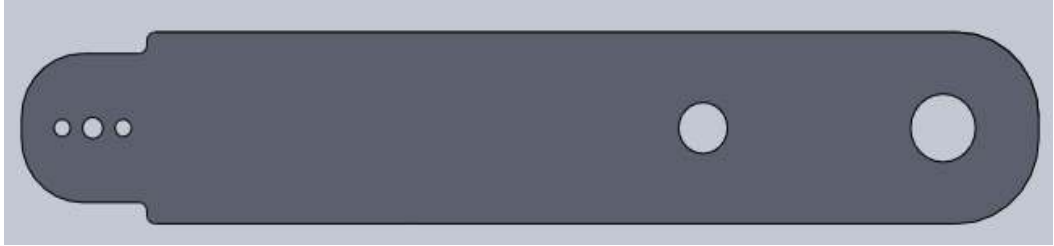
Şekil 3.9 Motor, enkoder, miller, dişliler, kaplinler ve rulmanların takılı olduğu sistemin son hali

Şekil 3.9'un yakın plan gösterimini Şekil 3.10'da verilmiştir.



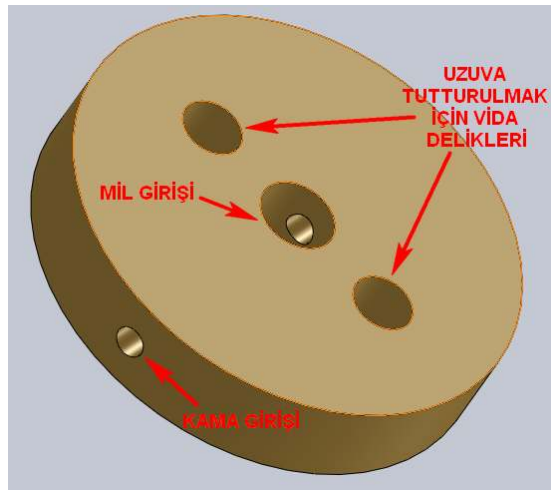
Şekil 3.10 Yakın plan gösterim

Uzuvlar, köşeleri pahtalanmış dikdörtgen yapılardan oluşturuldu. Ancak Solidwork çizimlerinde uzuvlar tamamen dikdörtgen olursa uzuvların köşelerinin alt taraftaki millere çarptığı görüldü. Bu sebeple uzuvların tasarımı değiştirilip Şekil 3.11'de gösterildiği gibi kulaklı bir yapıya çevirildi.



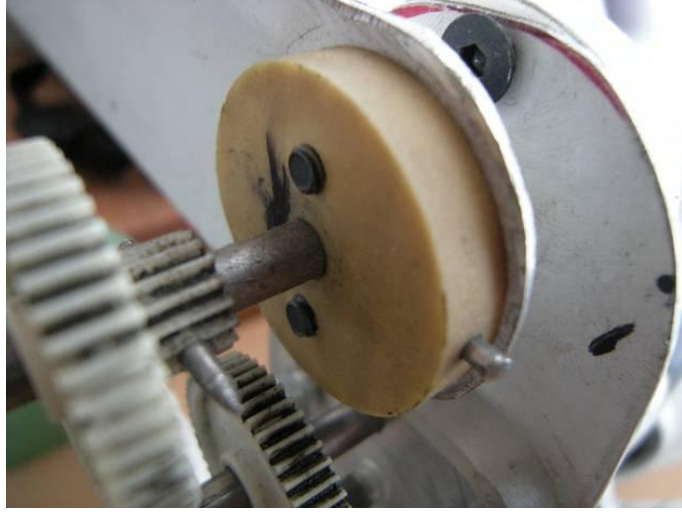
Şekil 3.11 Uzuv1 ve kulaklı yapı

Uzuvların mile bağlanabilmesi için şekil 3.12'de gösterilen aparat tasarlandı.



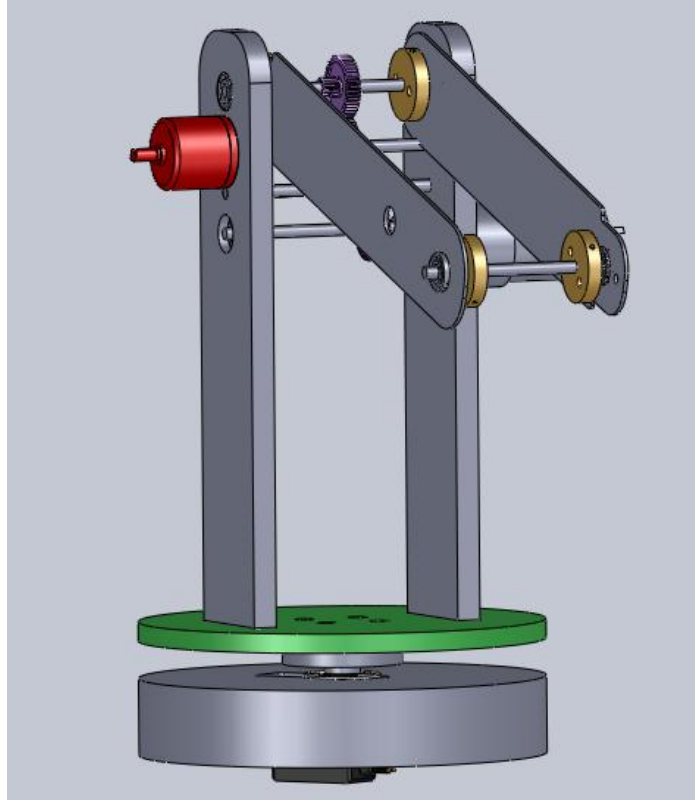
Şekil 3.12 Uzuv ve milleri bağlayan aparat

Aparat Şekil 3.13 'te gösterilmiştir.



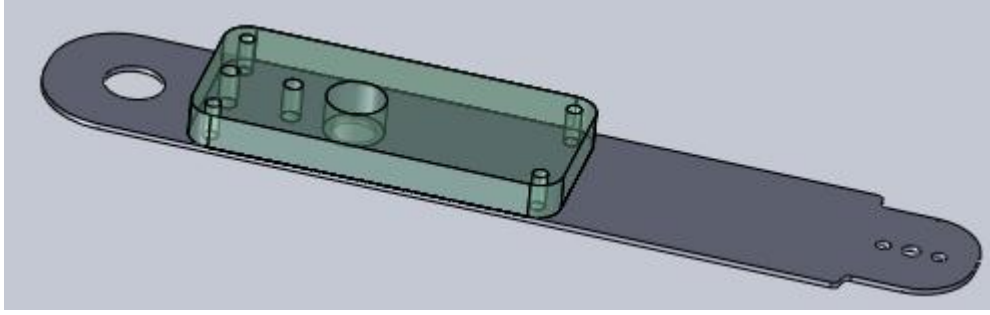
Şekil 3.13 Aparat Fotoğrafı

Uzuv ve aparatlar montajlandıktan sonra sistemin son hali Şekil 3.14'te gösterilmiştir.



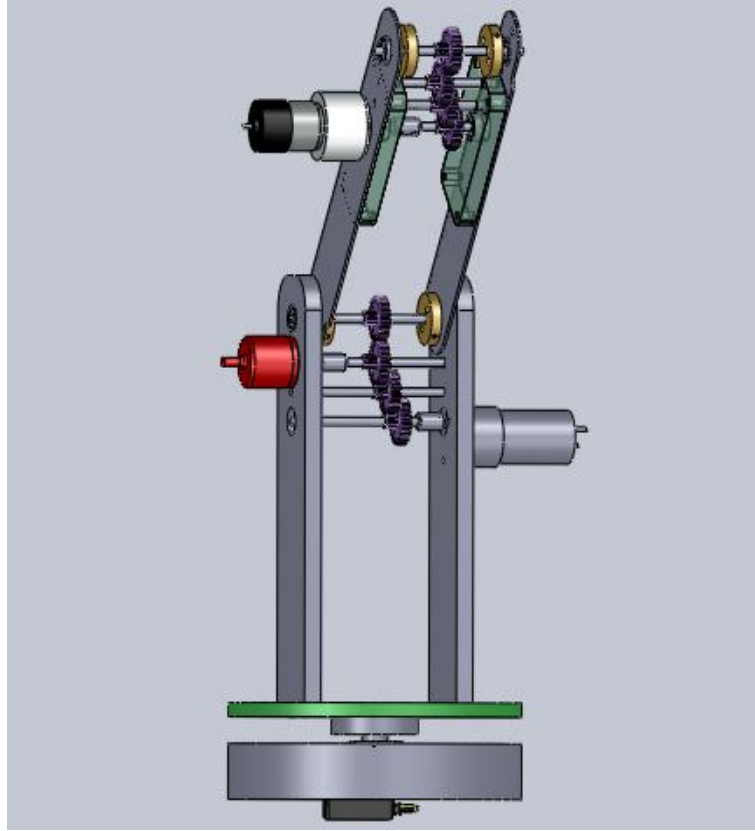
Şekil 3.14 Uzuv ve aparatlar montajlandıktan sonra sistemimizin son hali

Bir sonraki aşamada eksen2'deki motor ve diğer malzemelerin montajı yapıldı. Uzuvar çok ince malzemeler olduğundan motoru direkt uzuvlara monte etmek sorunlara yol açabilirdi. Bu sebeple uzuvlara monte edebilecek ara bir aparat tasarlandı. Bu aparat Şekil 3.15'te gösterilmiştir.



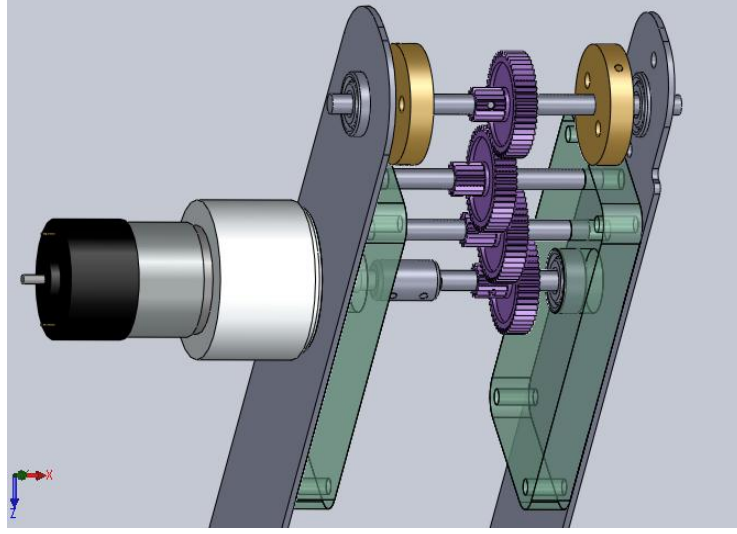
Şekil 3.15 Uzuvar ve motor bağlantısında kullanacağımız aparat

Sonrasında motor, dişliler, kaplinler ve miller uzuvlara eklenip sistemin son haline eklendi. Son durumda robot kolu Şekil 3.16'da gösterilmiştir.



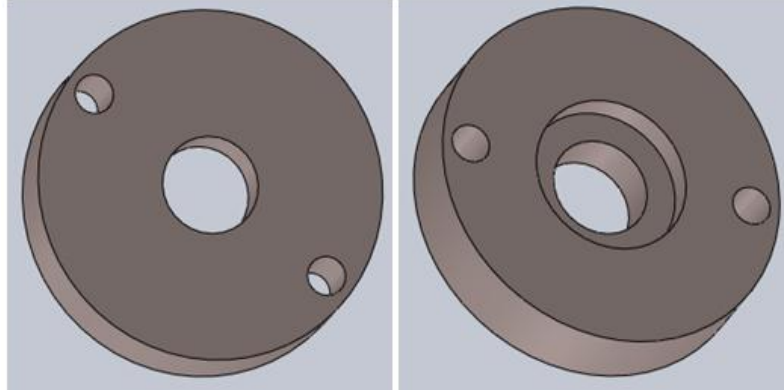
Şekil 3.16 Sistemin geldiği son nokta

Uzuv1'in bağlantıları Şekil 3.17'de detaylı bir şekilde gösterilmiştir.



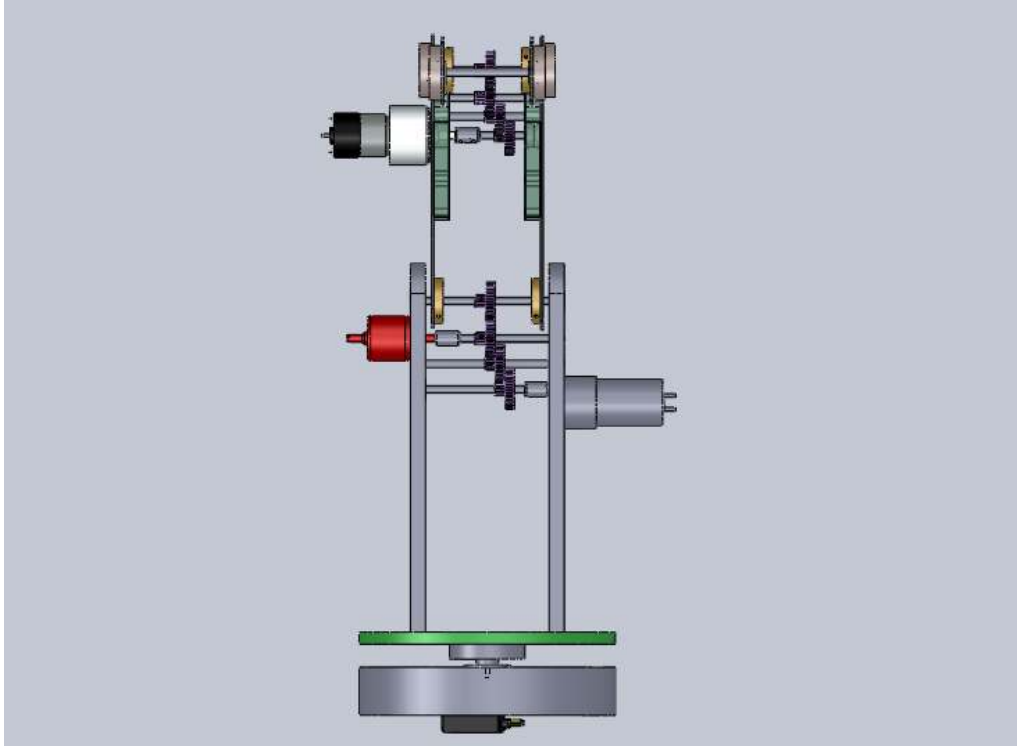
Şekil 3.17 Uzuv1'in bağlantıları

Bir sonraki aşamada uzuv2 yerleştirildi. Ancak öncesinde uzuv2 dönerken uzuv1'in bu dönüşten etkilenmemesini sağlayacak, içerisinde rulman yatağı olan ve uzuv1'e vida ile montajlanacak bir aparat tasarlandı. Bu aparat Şekil 3.18'de gösterilmiştir.

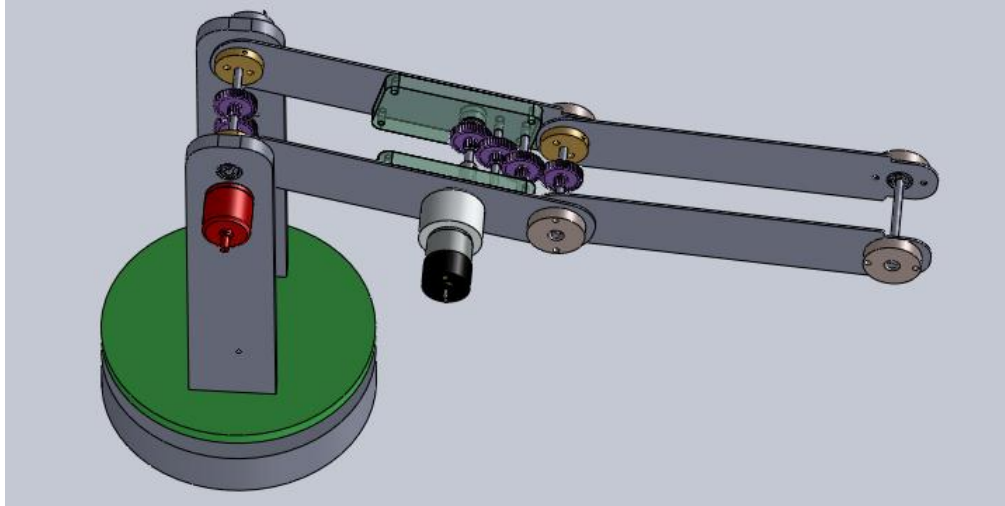


Şekil 3.18 Bağlantı aparatı

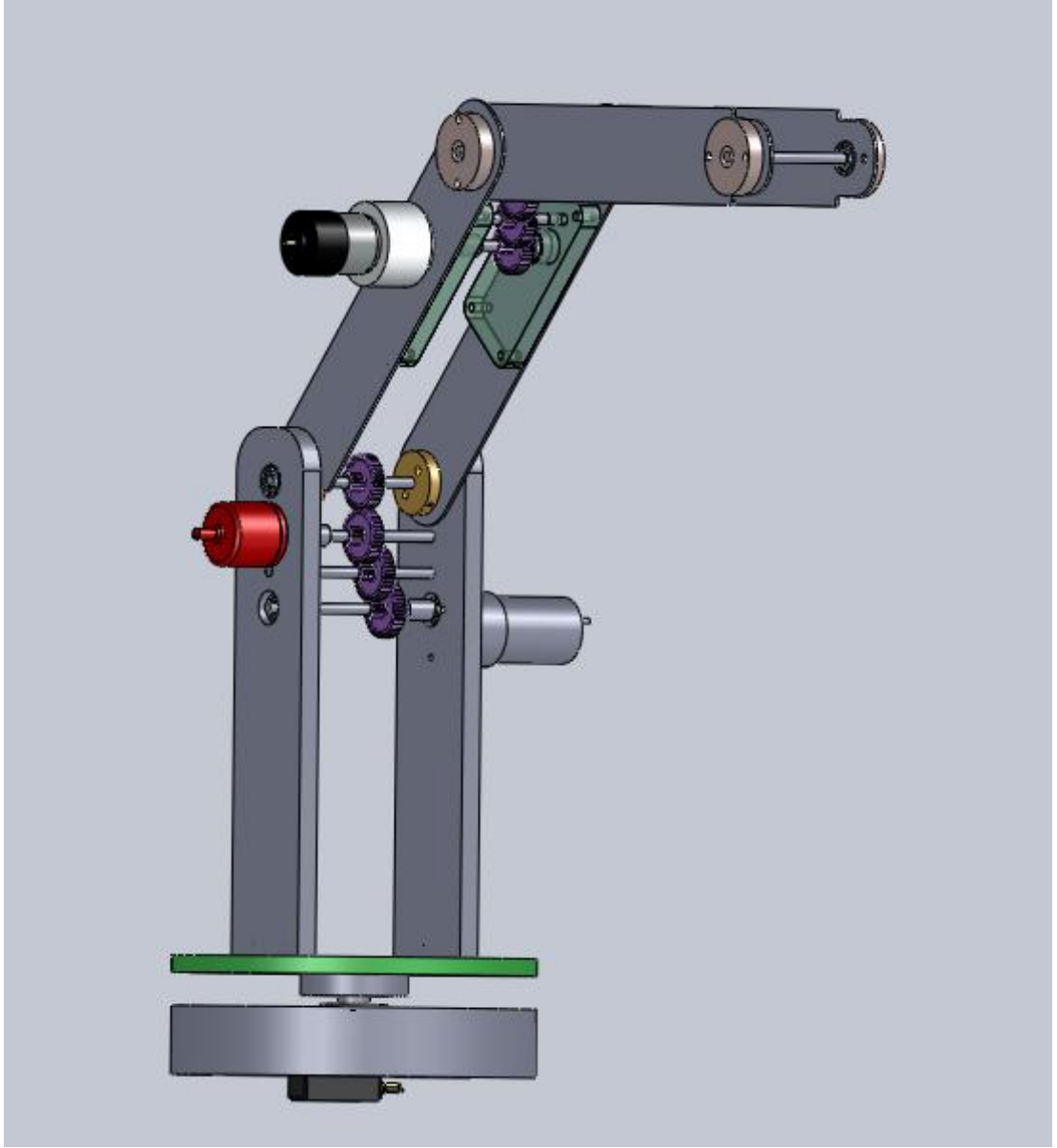
Son olarak uzuv2 ve miller sisteme eklendi. Böylece robot kolunun mekanik tasarımı tamamlanmış oldu. Oluşan robot kolunun farklı görünüşleri Şekil 3.19, Şekil 3.20 , Şekil 3. 21 ve Şekil 3.22 'de gösterilmiştir.



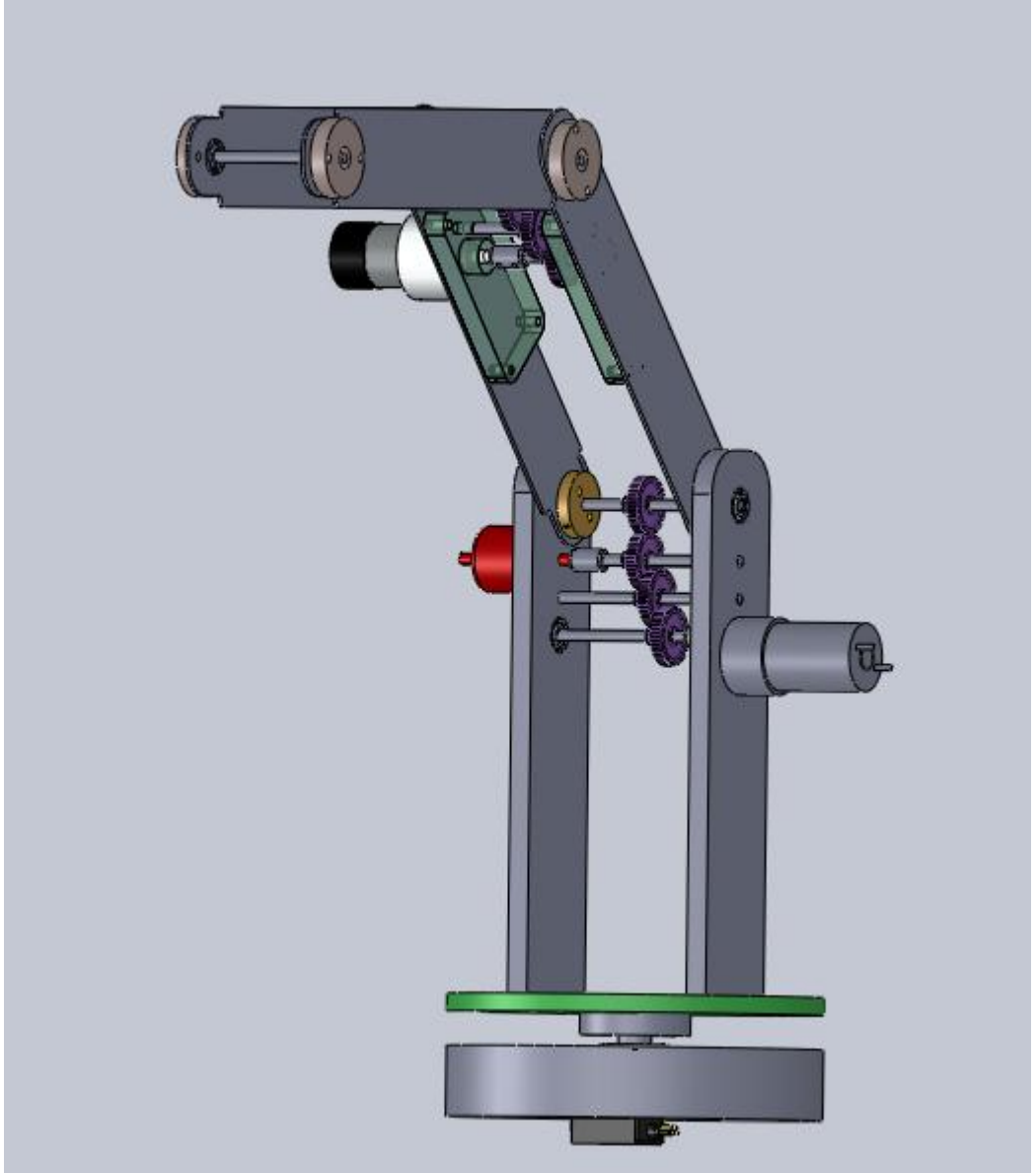
Şekil 3.19 Robot kolunun önden görünüşü



Şekil 3.20 Robot kolunun yan üstten görünüşü



Şekil 3.21 Robot kolunun yandan görünüşü



Şekil 3.20 Robot kolunun önden görünüşü

BÖLÜM DÖRT

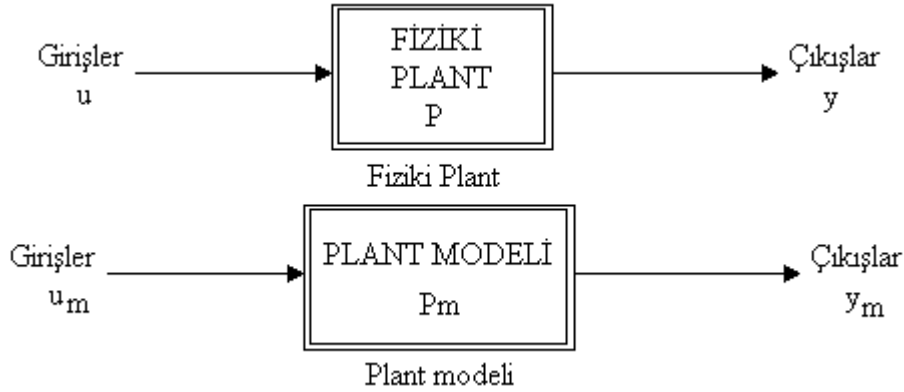
MOTORLARIN FARKLI KONTROL YÖNTEMLERİ İLE KONTROLÜ

4.1 Model Referanslı Uyarlamalı Kontrol

4.1.1 Plant Kavramı ve Tasarımı

Bir çok kontrol uygulamasında, istenilen performanslara ulaşmak için bilinmeyen bir plant'ın hareketlerini ve cevaplarını değiştirebilecek bir kontrol sistemi tasarlamak çok zor yada uğraştırıcı olabilir. Burada plant kelimesi girişler ve çıkışlar ile karakterize edilmiş bir işlem anlamındadır. Örnek bir şekil Şekil 4.1'de gösterilmiştir.

Plant'in girişleri plantin çıkışındaki verilere dönüşecek şekilde bir işleme tabi tutulur. Sistemi kontrol edecek kontrol bloğunun görevi, plant'in çıkışında istenilen değerlere ulaşabilmek için gerekli olan giriş değerlerini plant girişine uygulamaktır. Plant, mekanik, elektronik ya da hidrolik ve benzeri parçalardan oluşabileceğinden dolayı içerisindeki işlem genellikle karmaşıktır.



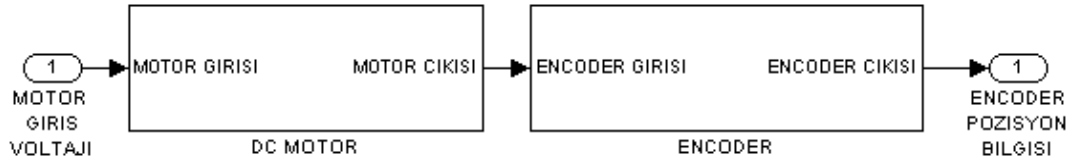
Şekil 4.1 Fiziki plant, plant modeli, girişler ve çıkışlar

Kontrol bloğu oluşturulurken plant'in işlem yapısının anlaşılması çok önemlidir. Mümkünse plant'e verilen girişleri plant içerisinde işlendikten sonra oluşan çıkışlar ile arasındaki ilişki matematiksel denklemler ile ortaya konulmalıdır. Bu denklemler aynı zamanda plant'in matematiksel modelidir. Plant'in matematiksel modeli verilen

girişlere cevabı plant ile aynı olmalıdır. Maalesef bir fiziksel plant'ın karmaşıklığı bu şekilde tamamen aynı sonuçları veren bir plant modeli oluşturabilmeyi zorlaştırır hatta imkansızlaştırabilir. Hatta kesin bir plant modeli bulunabilse dahi , plant'ın doğrusal olmaması yada zaman ile değişen çıktılar vermesi, modelin yararlılığını azaltabilir ya da yok edebilir. Bu da doğru çıktıları veren bir plant tasarımında karşılaşılabilecek zorluklardan birisidir.

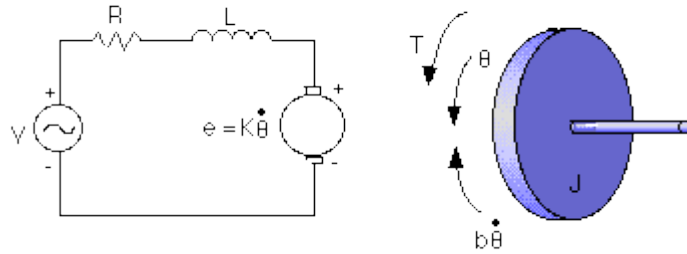
4.1.1.1 Sistemin Plant Modelinin Bulunması

Sistemi kontrol edebilmek için öncelikle sistemin plant modeli çıkartılır. Burada kontrol edilecek sistem DC motor ve enkoderdan oluştuğundan dolayı DC motor ve enkoderin modellerinin ayrı ayrı oluşturulması gerekmektedir. Şekil 4.2'de oluşturulan Simulink bloğu gösterilmiştir.



Şekil 4.2 Simulinkte DC motor ve enkoder bloğu

4.1.1.1.1 DC Motor Modelinin Parametrelerinin Bulunması. DC motorun modelini bulabilmek için öncelikle motor denklemlerini çıkartmak gerekmektedir. Bir DC motora ait elektriksel ve mekanik modeli şekil 4.3'de gösterilmiştir.



Şekil 4.3 DC motorun elektriksel ve mekanik gösterimi

<http://www.library.cmu.edu/ctms/ctms/simulink/examples/motor/motorsim.htm>

Denklemlerde kullanılan kısaltmalar aşağıdaki gibidir.

J	:	Rotor atalet momenti
B	:	Mekanik sistem damping oranı
K=K _e =K _t	:	EMK sabiti
R	:	Elektriksel direnç
L	:	Elektriksel endüktans
V	:	Giriş voltajı
θ	:	Şaftın pozisyonu
T	:	Motorun Torku
i	:	Armatür akımı
e	:	Ters EMK sabiti

Motorun akıma bağlı tork ve açısal hıza bağlı ters emk kuvveti denklemleri aşağıdaki gibidir.

$$T = K_t * i$$

$$e = K_e * \frac{d\theta}{dt}$$

Açısal ivmenin ve akımın zamana göre türevinin integralleri alınırsa aşağıdaki denklemler elde edilir.

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

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

DC motorun mekanik ve elektriksel modellerinden aşağıdaki denklemler yazılabilir.

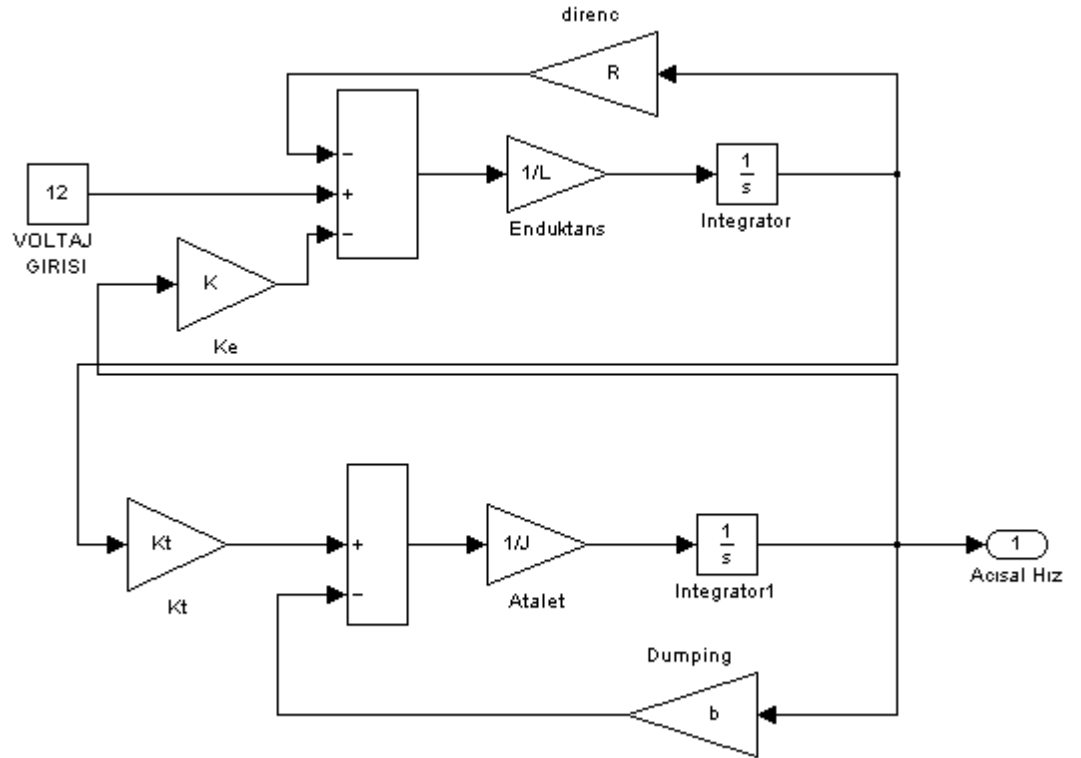
$$j * \frac{d\theta^2}{dt^2} = T - b * \frac{d\theta}{dt}$$

$$\frac{d\theta^2}{dt^2} = \frac{1}{j} * (K_t * i - b * \frac{d\theta}{dt})$$

$$L * \frac{di}{dt} = -R * i + V - e$$

$$\frac{di}{dt} = \frac{1}{L} * (-R * i + V - K_e * \frac{d\theta}{dt})$$

Yukarıda elde edilen DC motor denklemleri kullanılarak oluşturulan Simulink modeli şekil 4.4’de gösterilmiştir.



Şekil 4.4 DC motor modeli

Tablo.4.1’de robot kolunda kullanılan DC motorun Voltaj – Açısai Hız karakteristiği verilmiştir.

Tablo 4.1 1 numaralı DC motorun Voltaj – Açısal Hız tablosu

VOLTAJ (V)	DEVİR/DAKİKA (RPM)	AÇISAL HIZ (rad/s)
12	35	3,66
11	32	3,35
10	29	3,03
9	26	2,72
8	23	2,40
7	20	2,09
6	17	1,78
5	13,5	1,41
4	10,5	1,10
3	7	0,73
2	4	0,41

DC motor modeline, girdi ve çıktı olarak, DC motor voltaj – açısal hız değerleri verildiğinde DC motor modelindeki katsayılar aşağıdaki gibi bulundu.

$$J = 0,00096699;$$

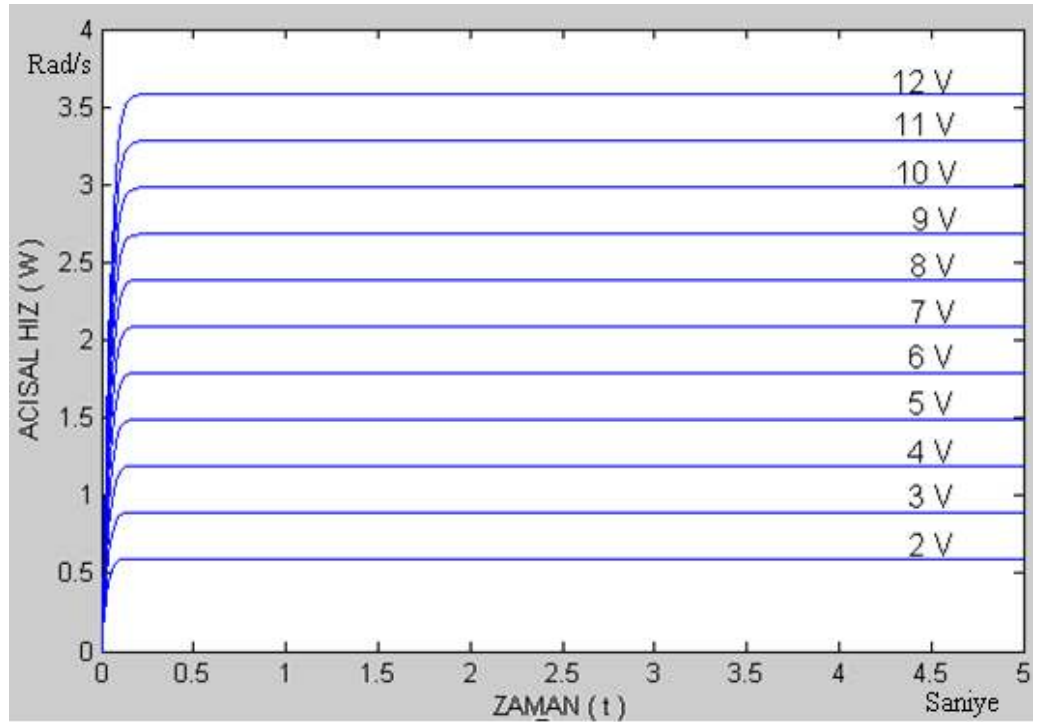
$$K = 0,39388;$$

$$K_t = 0,039692;$$

$$L = 0,059563;$$

$$R = 1,7275;$$

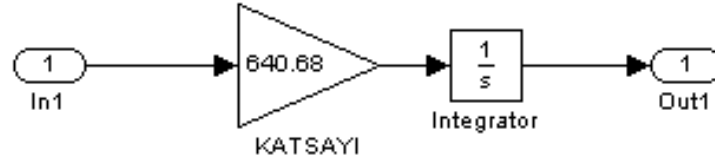
$$B = 0,067957;$$



Şekil 4.5 DC motor modeli Voltaj – Açısal hız

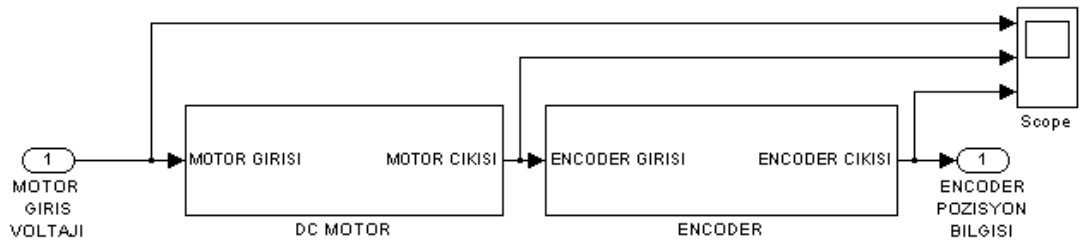
Bulunan bu katsayılar DC motor modeline konulup 2-12V arasında sistem simule edildiğinde, DC motor modeli çıkışında elde edilen açısal hız değerlerinin Şekil 4.5’de ki gibi olduğu gözlemlendi. Bu değerlerin Tablo.4.1’deki değerler ile aynı olması, oluşturmuş olan DC motor modelinin doğruluğunu gösterir.

4.1.1.1.2 Encoder Modelinin Parametrelerinin Bulunması. Enkoder bloğu için oluşturulan Simulink modeli aşağıdaki Şekil 4.6’da verilmiştir. Bu model, tıpkı robot kolunda kullanılan enkoder gibi motor bir tur attığında çıktı olarak 4096 değeri vermektedir.

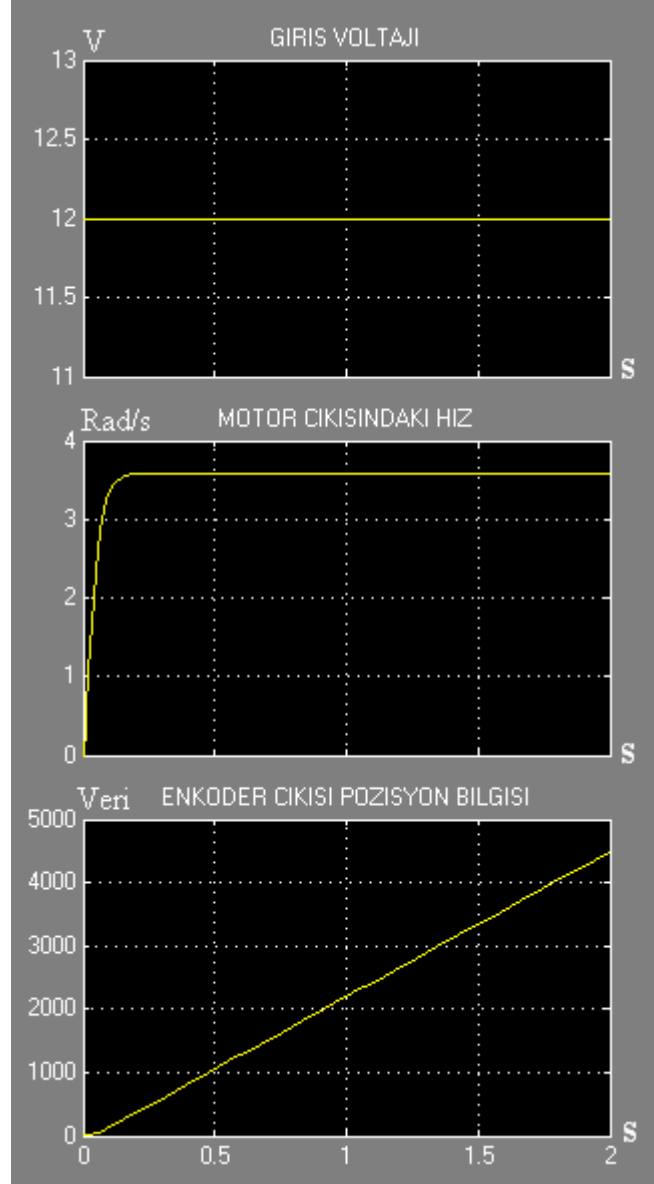


Şekil 4.6 Enkoder Simulink modeli

4.1.1.1.3 Sistemin Plant Modeli. Oluşturulan DC motor ve enkoder bloğu Şekil 4.7’deki gibi bağlanıp scope ile çıktılar alındığında Şekil 4.8’deki sonuçlara ulaşıldı. Bu sonuçlardan da görüldüğü gibi elde edilen DC motor + enkoder modeli, robot kolunda kullanılan motor_+_enkoder ile aynı çıktıları vermektedir.



Şekil 4.7 Scope bağlantısı



Şekil 4.8 Scope çıktıları

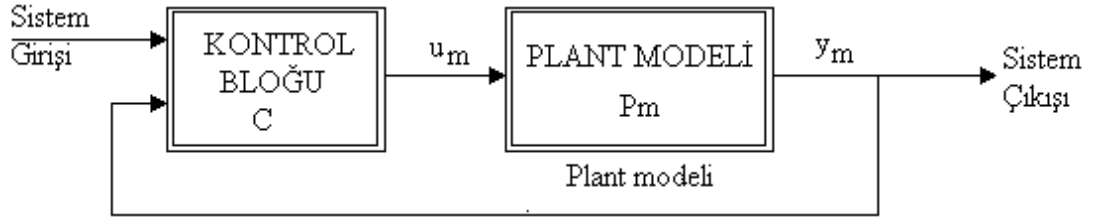
4.1.2 Uyarlamalı Kontrol ve Kontrol Bloğunun Oluşturulması

Uyarlamalı kontrolde öncelikle oluşturulan plant modelini istenilen şekilde kontrol edebilen bir kontrol bloğu tasarlanır. Plant modeli bu kontrol bloğu ile çalıştırılır ve sonuçlar gözlemlenir. Eğer kontrol bloğu plant modelinin çıkışında istenilen değerleri sağlayabiliyorsa, bu kontrol bloğu fiziki plant'a uygulanır.

Kontrol bloğu tasarlanırken plant'ın giriş ve çıkışları arasındaki ilişkiyi doğru olarak tanımlayan mümkün olduğunca basit bir denklem kullanılmalıdır. Basit bir

modelin kontrolü genellikle anlaşılması ve uygulanması basit bir kontrol bloğu ile sağlanabilir. Bu yapı aynı zamanda uygulamadaki hedeflere ulaşması açısından daha güvenlidir.

Doğru plant modeline ulaşıldıktan sonra bu plant modeli Şekil 4.9’da görüldüğü gibi kontrol bloğuna bağlanır. Kontrol bloğu plant modelinin (P_m) istenilen çıkışları (y_m) verebilmesi için gerekli olan giriş (u_m) değerlerini sağlamak ile görevlidir. Başka bir deyişle kontrol bloğu, sistem çıkışından geri besleme ile alınan çıkış bilgisini sistem girişi ile karşılaştırarak plant modelinin çıkışında istenilen değerlerin elde edilmesini sağlayacak şekilde tasarlanır.

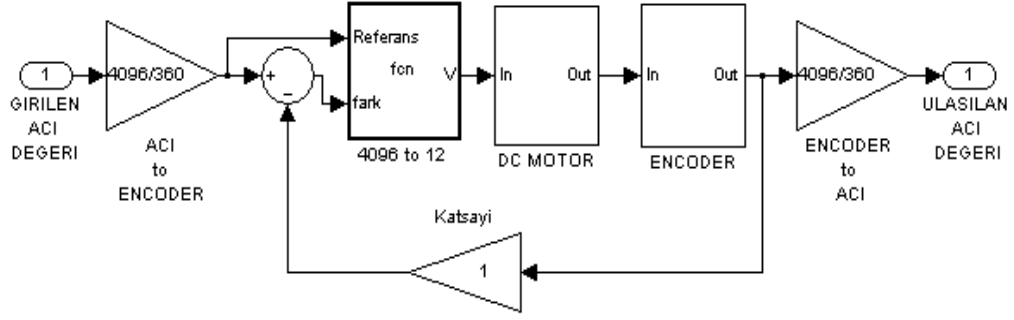


Şekil 4.9 Uyarlamalı kontrol ile plant modelinin kontrol edilmesi

4.1.2.1 Oluşturulan Plant Modelinin Farklı Kontrol Yöntemleri ile Kontrolü

Bu bölümde oluşturulan plant modeli Geri beslemeli kontrol, PID kontrol ve yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrol edilerek, bu kontrol yöntemlerinin birbirlerine karşı üstünlükleri ve dezavantajları üzerinde durulmuştur.

4.1.2.1.1 Geri Beslemeli Kontrol. Geri beslemeli kontrolde, sistemin çıkışı bir çarpan ile çarpılarak oluşan hata değeri sistemin girişine verilir ve sistemin çıkışının, geri besleme katsayısı vasıtası ile sistemin girişine verilen referans değerine eşitlenmesi sağlanır. Şekil 4.10’da DC_motor+_enkoder sisteminde kullanılabilecek örnek bir geri besleme yöntemi görülebilir. Şekilde de görülebileceği gibi geri besleme katsayı değeri bizim sistemimiz için 1 olarak bulunmuştur.



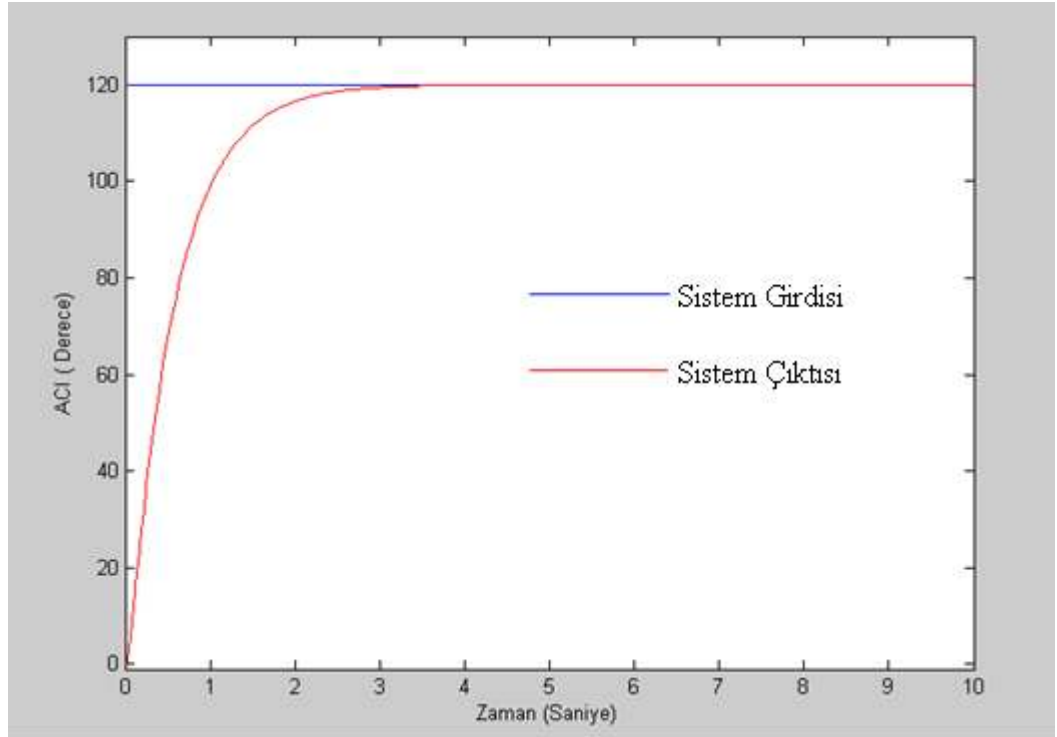
Şekil 4.10 Geri besleme kontrollü DC motor + enkoder bloğu

FCN bloğunu içeriği aşağıdaki gibidir.

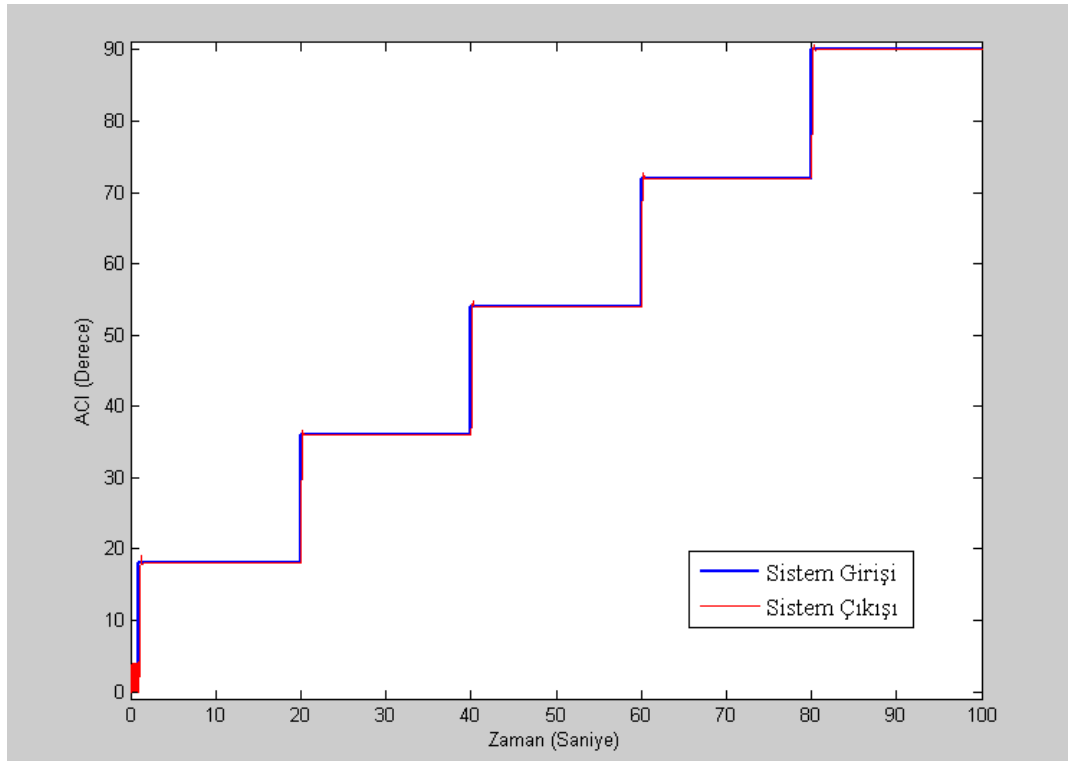
```
function V = fcn(fark, Referans)
persistent ilkp;      % Sabit değişken tanımla
if isempty(ilkp)      % Eğer ilkp değeri boş ise -10'a eşitle
    ilkp = -10;
end
persistent blm;      % Sabit değişken tanımla
if isempty(blm)      % Eğer blm değeri boş ise 1'e eşitle
    blm = 1;
end
if ilkp ~= Referans   % ilkp değeri Referans değerinden farklı ise eşitle
    ilkp=Referans;    % ilkp değerini Referans değerine eşitle
    blm=abs(fark);    % blm'yi fark değerine eşitle
end
V=fark*12/blm;        %Voltaj çıktısı ver
```

Şekil 4.11'da sisteme 120 derecelik açı girdisi verildiğinde sistem çıkışı gösterilmiştir.

Şekil 4.12'de ise sisteme 1°- 90° arasında girdiler verildiğinde sistemin çıktıları görülmektedir. Bu şekilde de görüldüğü gibi yaptığımız kontrol vasıtası ile sistemimiz girdi olarak verilen açı değerine ulaşabildiği ancak aynı zamanda küçük açı değişimlerinde sistemin referans nokta etrafında dolaştığı ancak istenen açı değerinde ulaşamadığı görülmüştür.



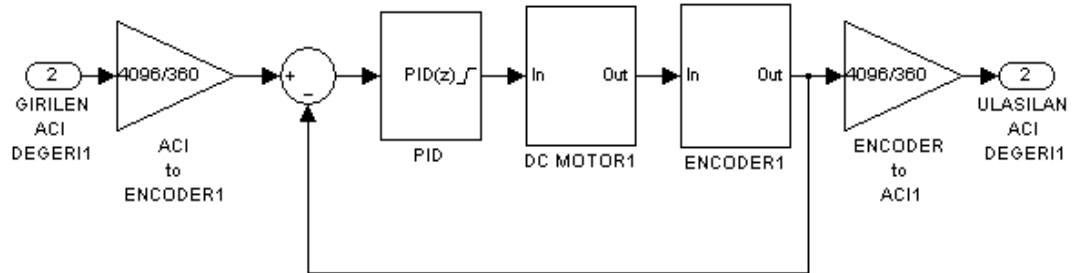
Şekil 4.11 Geri besleme ile kontrol edilen sistemde 120 derecelik açılı girdisi verildiğinde sistemin çıkışı



Şekil 4.12 Geri besleme ile kontrol edilen sistemde 0-90 derece arası girdiler verildiğinde sistemin çıktıları

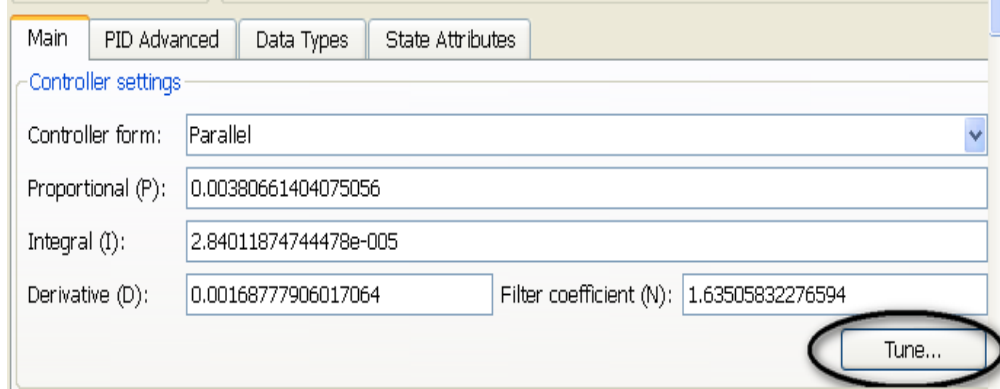
Sonuç olarak geri besleme ile yapılan kontrolde referans olarak verilen girdi değeri, diğer bir deyişle açılı farkı yükseldikçe daha iyi sonuçlar elde edildiği ancak küçük açılı değişimlerinde sistemin kararsız rejimde kalma süresinin uzadığı ve 3 derece ile 2 derecelik arasındaki farklarda sistemin kararsız çalıştığı gözlemlenmiştir.

4.1.2.1.2 PID Kontrolü. Açılımı oransal (proportional), integral ve türev (derivative) kontrol olan PID kontrol, sistemin referans değeri ile sistemin çıkışı arasındaki hata değerini kullanarak sistem çıkışının referans değerine ulaşmasını sağlar. Hata değerine PID girişine verilerek, paralel olarak, bir çarpan ile çarpılma, integral işlemi ve türev işlemi uygulanır. Bu işlemlerin ağırlıklı toplamaları PID kontrolün çıkışını oluşturur. Bu çalışmada PID bloğu Şekil 4.13’de görüldüğü gibi DC motorun önüne eklendi ve enkoderin çıkışındaki veri geri besleme ile referans değerinden çıkartılarak, hata değeri PID bloğuna girildi. Kurulan sistem Şekil 4.21’de gösterilmiştir.



Şekil 4.13 PID kontrollü DC motor + enkoder bloğu

PID ile kontrolde PID bloğunun çıkışının doğru değerleri vermesi başka bir deyişle sistemin istenildiği gibi kontrol edilebilmesi için doğru PID katsayılarının bulunması çok önemlidir. Bu çalışmada PID değerlerini bulmak için PID bloğunun tune özelliğinden faydalanılmıştır. Tune özelliği, oluşturmuş olduğumuz blokta istenilen sonuçları alabilmemiz için PID bloğundan doğru çıktıların çıkmasını sağlayan PID katsayılarının bulunmasında faydalı bir araçtır. Şekil 4.13’de PID bloğunda tune diyalog kutusunu açmak için basılacak buton gösterilmiştir.

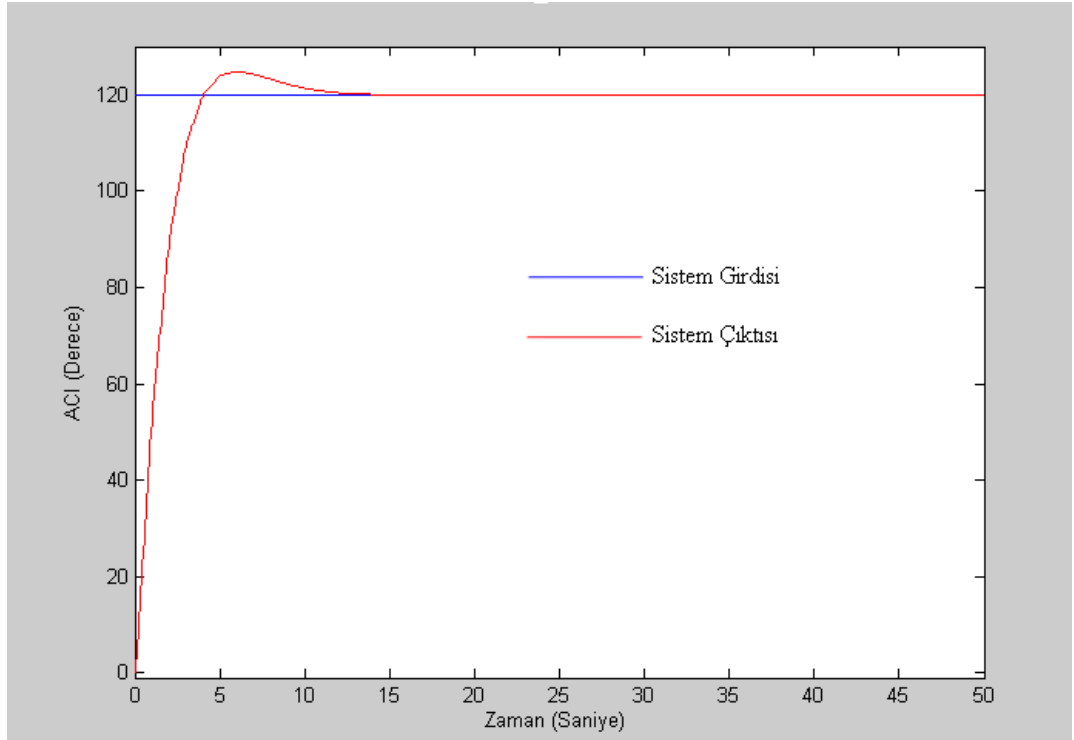


Şekil 4.13 PID bloğunda tune diyalog kutusunu açmak için basılacak buton

Tune özelliğinden yararlanarak sistemin kararlı hale geçişi ve referans noktaya ulaşma hızı ile ilgili ayarlar yapıldı ve PID katsayıları bulundu. Bulunan PID katsayıları ile simülasyon çalıştırıldığında sistemin verilen 20 s'lik zaman aralığında %0.5'lik bir hata ile çalıştığı gözlemlendi. Bu hatayı sıfıra indirebilmek için PID katsayıları değiştirildi ve değişikliğin sonuçlara etkisi gözlemlendi. Bu metot ile doğru parametrelerde doğru değişiklikler yapılarak sistem %0 hataya ulaşınca kadar PID katsayıları değiştirildi. Sonuç olarak %0 hata için PID katsayıları aşağıdaki gibi bulundu.

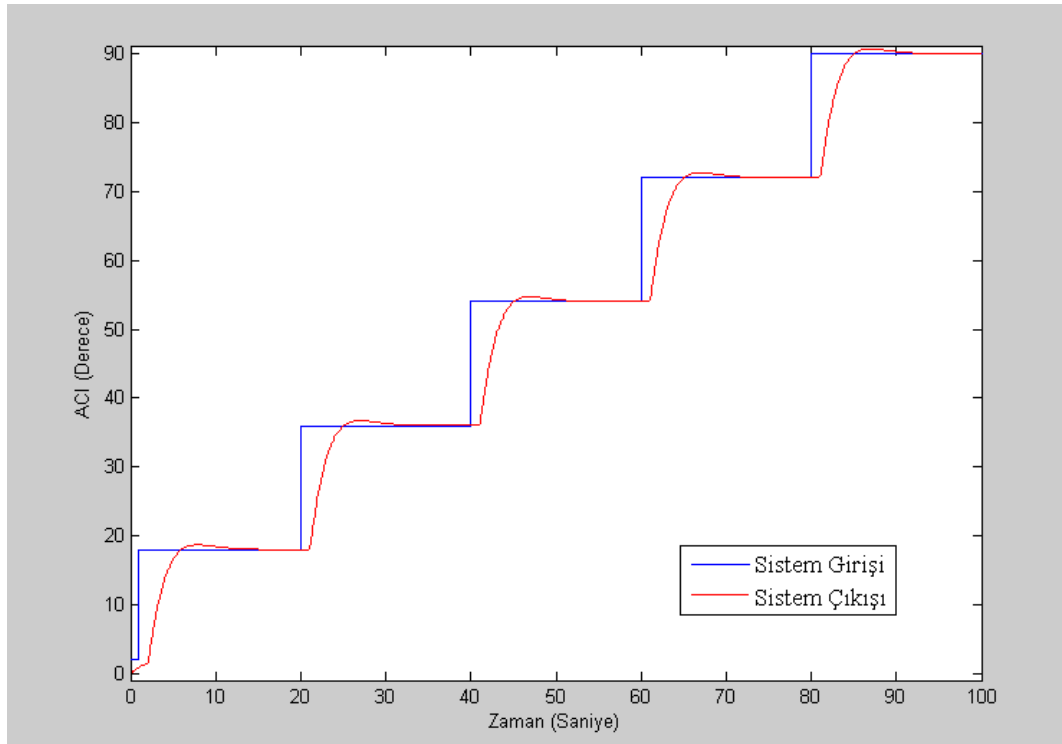
Proportional	: 0,00299651499951687
Integral	: 0,8702992459878e-006
Derivative	: - 0,0018601013624742
Filter coefficient	: 0,32726865106461100

Bu PID katsayıları ile oluşturduğumuz sisteme 120° 'lik bir aç girdisi verildiğinde sistem çıktısının Şekil 4.14'deki gibi olduğu görüldü.



Şekil 4.14 PID kontrollü sistemde 120 derecelik açı girdisi verildiğinde sistemin çıkışı

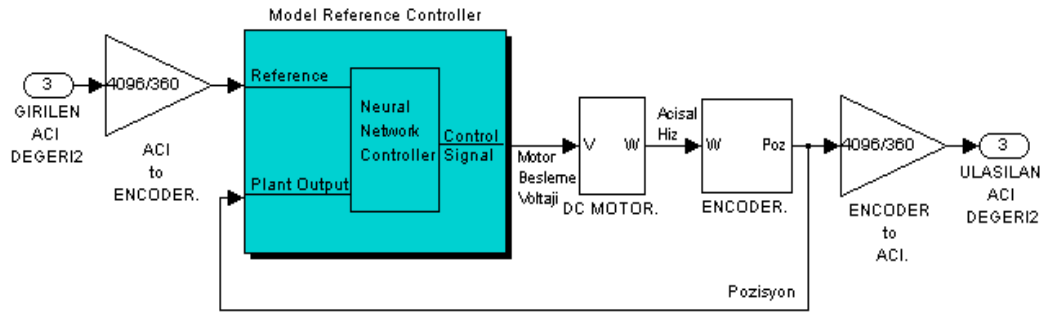
Şekil 4.15 de ise sisteme 1° - 90° arasında girdiler verildiğinde sistemin çıktıları gösterilmiştir.



Şekil 4.15 PID kontrollü sistemde 0-90 derece arası girdiler verildiğinde sistemin çıktıları

Sonuç olarak PID ile yapılan DC motor kontrolünde, sistemin geri beslemeli kontrolün aksine küçük açı değerlerinde de istenilen noktaya başarıyla gidebildiği ve bu yönü ile geri beslemeli kontrolden daha iyi sonuçlar verdiği, ancak düzensiz rejim süresinin geri beslemeli kontrolden daha uzun olduğu ve kararlı hale geçme süresi bakımından geri beslemeli kontrolden daha kötü sonuçlar verdiği gözlemlendi.

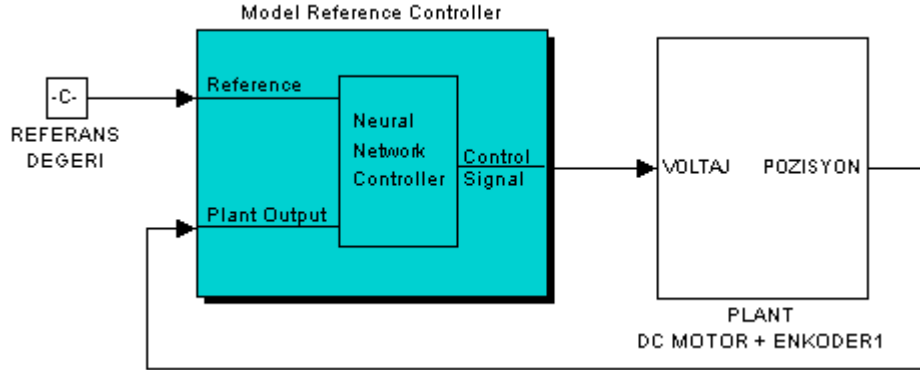
4.1.2.1.3 Yapay Sinir Ağları Tabanlı Model Referans Kontrol Yöntemi. Yapay sinir ağları tabanlı model referans yöntemi ile DC motor kontrolü tıpkı PID kontrolündeki gibi sistemin önüne bir yapay sinir ağı bloğu konularak yapılabilir. Bizim sistemimizde de yapay sinir ağı bloğu DC motorun önüne eklendi ve enkoderin çıkışındaki veri geri besleme ile yapay sinir ağına geri verilerek, yapay sinir ağı çıktısı motor girişine verildi. Yapay sinir ağı bloğu olarak Simulink'deki model referans denetleyicisi bloğu kullanıldı. Kurulan sistem Şekil 4.16'da gösterilmiştir.



Şekil 4.16 Yapay sinir ağı tabanlı model referans kontrolü metodu ile kontrol edilen DC motor_+_enkoder bloğu

Sistem içerisine yerleştirilen yapay sinir ağı bloğunun çıkışının doğru değerleri vermesi başka bir deyişle sistemin istenildiği gibi kontrol edilebilmesi için öncelikle yapay sinir ağının eğitilmesi gerekmektedir. Model Reference bloğu aynı zamanda yapay sinir ağının eğitilebileceği bir bloktur. Eğitimin başarısı, Model Reference bloğuna girilen parametrelere, örnek verilere, sisteme tanıtılan plant'a ya da bloğa referans olacak modele göre değişeceğinden dolayı birden fazla deneme yapmak gerekebilir. Tezimizde de yapay sinir ağını eğitmek için Model Reference bloğunun

bu özelliklerinden faydalandı. Öncelikle DC_motor+_enkoder bloğu Model Reference bloğuna plant olarak tanıtıldı. Şekil 4.17’de kurulan sistem gösterilmiştir.



Şekil 4.17 Model referans bloğu ve plant

Yapay sinir ağının eğitilmesi için ilk olarak plant identification penceresinden DC motor+_enkoderdan oluşan plantimiz Model Reference Control bloğuna tanıtıldı; örnek veriler alındı ve yapay sinir ağının plant bölümü eğitildi. Eğitim sonucunda en doğru sonuçlara Şekil 4.18’de ki girdiler ile ulaşıldı.

Şekil 4.18 Plant eğitiminde kullanılan parametreler

Plant eğitildikten sonra model bloğumuz eğitildi. Bunun için model Reference Control bloğunun referans alabileceği bir blok ya da yapay sinir ağlarının eğitiminde kullanılacak veri setinin verilmesi gerekir. Bizim sistemimizde giriş değerleri çıkış değerlerine eşit olması istendiğinden dolayı $çıkış(t) = giriş(t)$ olacak Şekilde “model9” adında bir Simulink bloğu oluşturuldu (şekil 4.19) ve sistem eğitildi. Eğitim sonucunda en doğru sonuçlara Şekil 4.20’deki girdiler ile ulaşıldı.



Şekil 4.19 Model9 bloğu

Model Reference Control

File Window Help

Network Architecture

- Size of Hidden Layer: 13
- Sampling Interval (sec): 0.05
- No. Delayed Reference Inputs: 2
- No. Delayed Controller Outputs: 1
- No. Delayed Plant Outputs: 2
- ☐ Normalize Training Data

Training Data

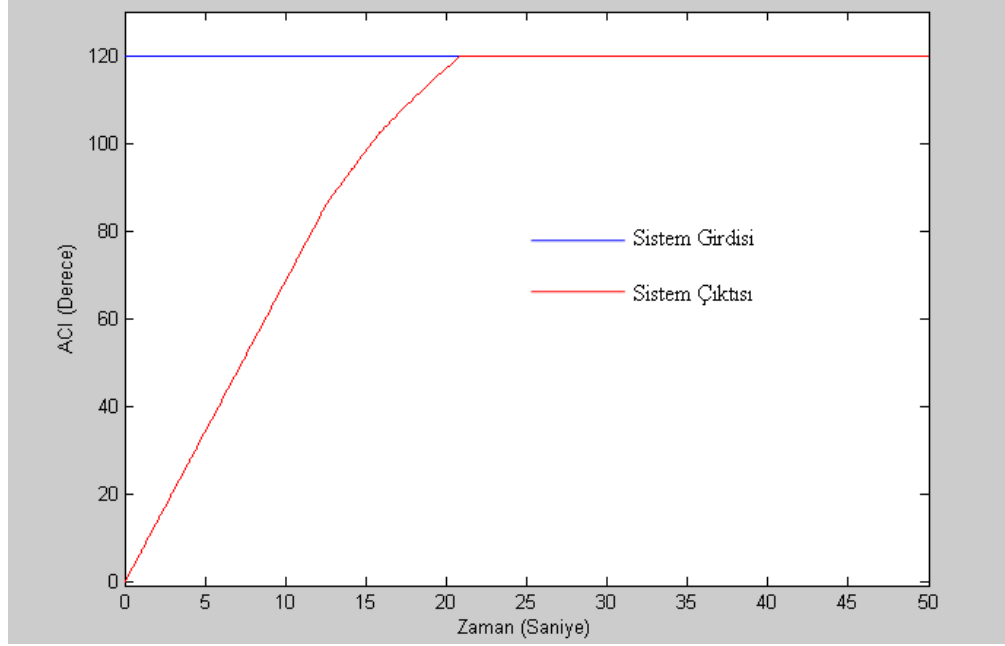
- Maximum Reference Value: 1
- Minimum Reference Value: 0
- Maximum Interval Value (sec): 5
- Minimum Interval Value (sec): 0.1
- Controller Training Samples: 6000
- Reference Model: Browse
- model9
- Generate Training Data
- Import Data
- Export Data

Training Parameters

- Controller Training Epochs: 30
- Controller Training Segments: 1
- ☒ Use Current Weights
- ☐ Use Cumulative Training
- Plant Identification
- Train Controller
- OK
- Cancel
- Apply

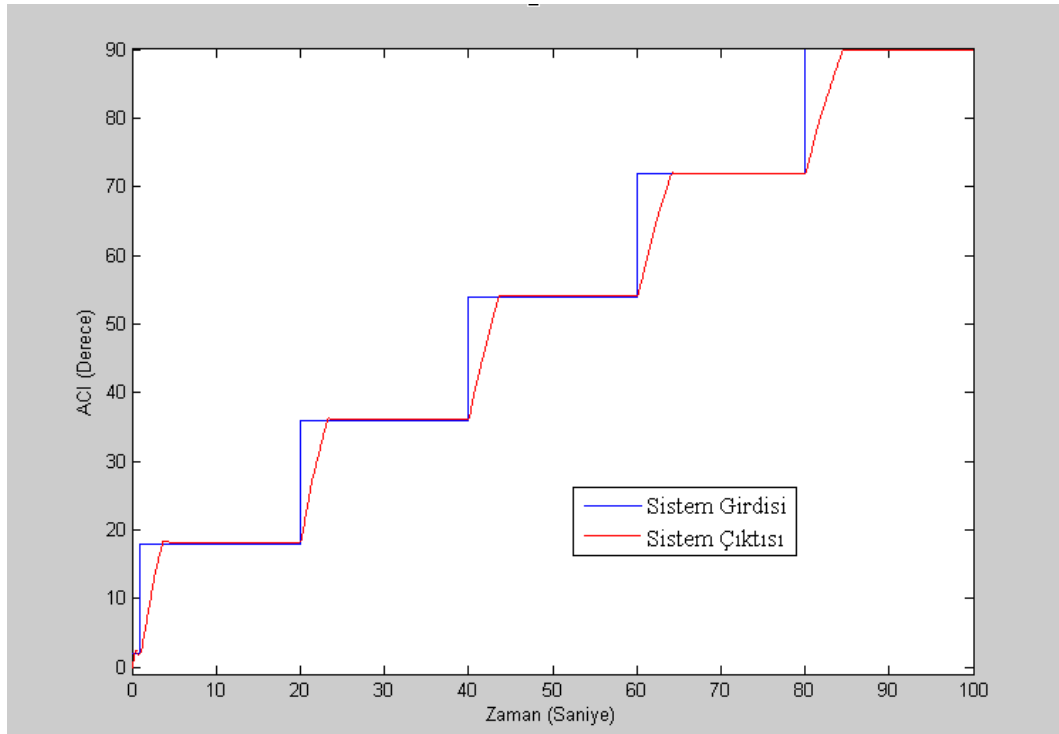
Şekil 4.20 Model Reference Control bloğunun eğitiminde kullanılan parametreler

Eğitim işlemi tamamlandıktan sonra oluşturduğumuz sisteme 120° ‘lık bir açı girdisi verildiğinde sistem çıktısının Şekil 4.21 ‘da ki gibi olduğu görüldü.



Şekil 4.21 Yapay sinir ağları tabanlı model referans yöntemi ile kontrol ile kontrol edilen sisteme 120 derecelik açı girdisi verildiğinde sistemin çıkışı

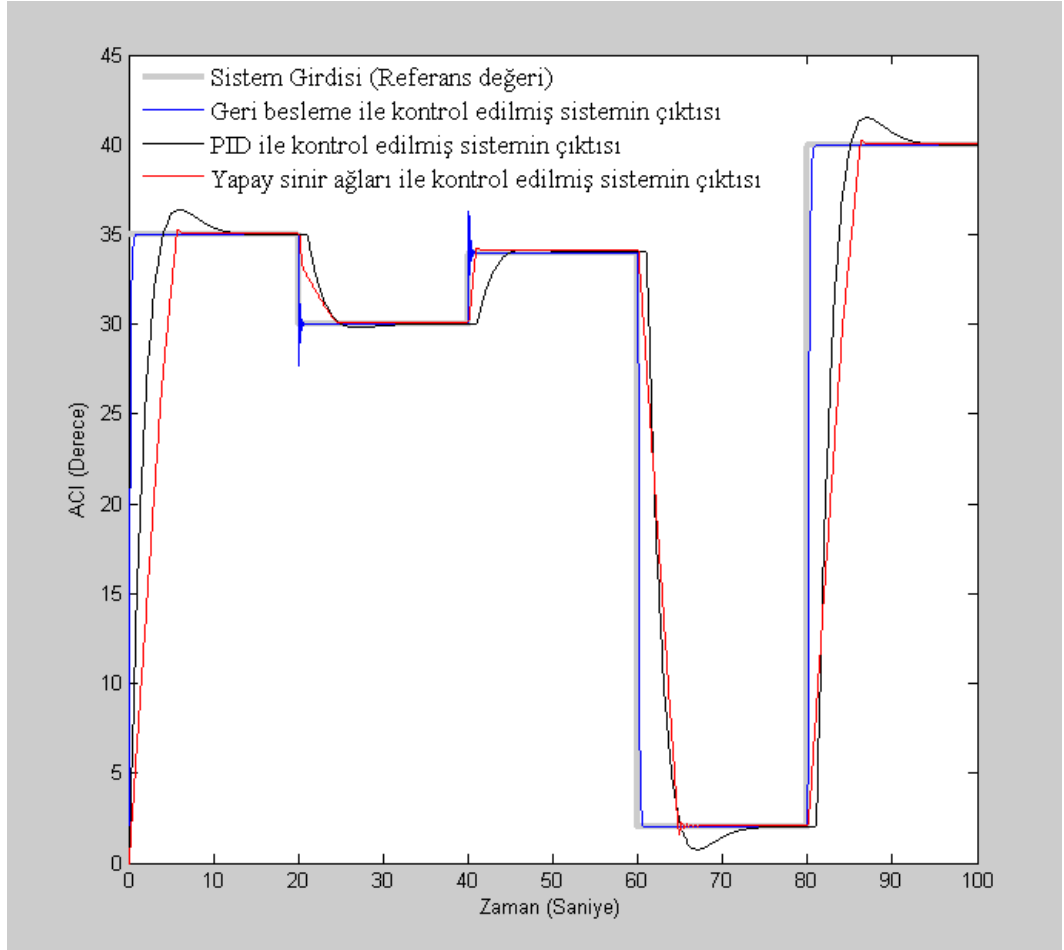
Şekil 4.22 de ise sisteme 1° - 90° arasında girdiler verildiğinde sistemin çıktıları gösterilmiştir.



Şekil 4.22 Yapay sinir ağları tabanlı model referans yöntemi ile kontrol ile kontrol edilen sisteme 0-90 derece arası girdiler verildiğinde sistemin çıktıları

Sonuç olarak yapay sinir ağları tabanlı model referans kontrol yöntemi ile yapılan kontrolde, sistemin düşük açı değerlerinde geri beslemeli kontrolden daha iyi sonuçlar verdiği ve 0-90 derece arasında referanslar verildiğinde kararlı hale geçme süresinin ya da referans noktaya varma süresinin PID kontrolden daha iyi olduğu gözlemlendi.

4.1.2.1.4 Kontrol Yöntemlerinin Karşılaştırılması. Şuana kadar incelenen 3 farklı kontrol yönteminin bir birlerine karşı üstünlüklerini görebilmek amacı ile bu üç sisteme aynı girdi değerleri verildi. Sonuçlar Şekil 4.23'de ve Tablo 4.2'de gösterilmiştir.



Şekil 4.23 Kontrol yöntemlerinin sonuçlarının karşılaştırılması

Tablo 4.2 Kontrol yöntemlerinin sonuçlarının karşılaştırılması

Süre (s)	Referans Değeri Derece	PID ile Kontrol			Yapay Sinir Ağları Tabanlı Model Referans Kontrol			Geri Beslemeli Kontrol		
		Kalıcı Duruma Ulaşma (s)	Kalıcı Durum Hatası Derece	Taşma Yüzdesi	Kalıcı Duruma Ulaşma (s)	Kalıcı Durum Hatası Derece	Taşma Yüzdesi	Kalıcı Duruma Ulaşma (s)	Kalıcı Durum Hatası Derece	Taşma Yüzdesi
0	35	4	36,412	4,02%	5,63	35,286	0,90%	0,7	35,000	0,00%
20	30	24,73	29,816	0,61%	24,5	30,121	0,00%	20,06	27,703	7,66%
40	34	44,55	34,176	0,51%	40,88	34,294	0,92%	40,05	36,249	6,61%
60	2	65	0,726	63,73%	64,7	1,620	19,17%	60,06	2,000	0,00%
80	40	85	41,534	3,82%	86,16	40,247	0,66%	80,75	40,000	0,00%

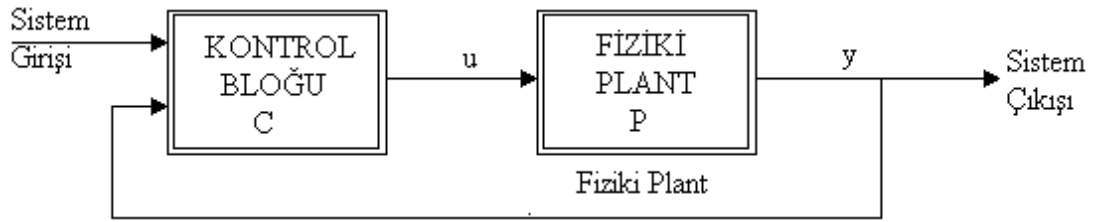
Şekilde de görülebileceği üzere sistem referans değerine en hızlı ulaşan sistemimiz geri beslemeli kontrol ile kontrol edilen sistem oldu ancak geri beslemeli kontrolün küçük aç değişimlerinde doğru değerlere ulaşamaması ya da referans değerinden uzaklaşması ve diğer kontrol metotlarında böyle bir sorunun olmaması PID ile kontrol ya da yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrolün, geri beslemeli kontrolden daha avantajlı olduğunu gösterdi.

Şekilde PID kontrol ile kontrol edilmiş sistemin çıktıları incelendiğinde, PID kontrolün, test edilen aç değerleri için referans noktasına en uzun sürede ulaşan kontrol olduğu gözlemlendi. Aynı zamanda referans noktasını geçtikten sonra belli bir süreliğine düzensiz rejime girdiği, sonrasında referans değerine geri döndüğü gözlemlendi. Bu dezavantajı nedeni ile PID ile kontrolün, yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrolden daha kötü sonuçlar verdiği görüldü.

Yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrol edilmiş sistemin çıktıları incelendiğinde ise küçük aç değerlerinde yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrolün, geri beslemeli kontrole göre çok daha iyi sonuçlar verdiği, PID kontrol sistemi ile karşılaştırıldığında ise referans değerine PID kontrol sisteminden daha hızlı ulaştığı ve referans değeri çevresinde PID kontrole göre daha kararlı davrandığı gözlemlendi. Bu sonuçlardan hareketle, bu üç kontrol sistemi arasında, robot kolunda kullanılabilecek en uygun kontrol sisteminin yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrol olduğu kararı verildi.

4.1.3 Uyarlamalı Kontrol ve Fiziki Plantin Kontrolü

Sistem çıkışında istenilen değerlerin elde edilmesini sağlayan kontrol bloğu tasarlandıktan sonra sistemden plant modeli çıkartılır ve fiziki plant sisteme eklenir. Eğer plant modelimiz doğru olarak tasarlandıysa ve kontrol bloğu plant modelimizin istenilen değerlere ulaşmasını sağlıyorsa, plant modeli yerine fiziki plant yerleştirildiğinde de kontrol bloğu sistem çıkışının istenilen değerlere ulaşmasını sağlar. Sistemin son hali şekil 4.24’de gösterilmiştir.



Şekil 4.24 Uyarlamalı kontrol ile fiziki plant’in kontrol edilmesi

4.1.3.1 Fiziki Plantin Farklı Kontrol Yöntemleri ile Kontrolü

Bu bölümde oluşturulan plant modeli Geri beslemeli kontrol , PID kontrol ve yapay sinir ağları tabanlı model referans kontrol yöntemi ile kontrol edilmiştir ve sistem çıktıları gösterilmiştir.

4.1.3.1.1 Geri Beslemeli Kontrol ile Motorların kontrolü. Geri beslemeli kontrol ile motorları kontrol edebilmek için gerekli olan Simulink bloğu oluşturuldu. Oluşturulan blog Şekil 4.25’te gösterilmiştir.

Şekil 4.25’deki Fcn bloğunun içeriği aşağıdaki gibidir.

```
function y = fcn(ilk,giris)
persistent ilkp;           % Geçici değişken tanımla
if isempty(ilkp)           % Eğer ilkp boş ise
    ilkp = -10;
end
```

```

persistent blm;           % Geçici değişken tanımla
if isempty(blm)           % Eğer blm boş ise
    blm = 1;
end
if ilkp ~= ilk             % Eğer ilkp değeri ilk değerinden farklı ise
    ilkp=ilk;
    blm=abs(giris);        % blm'yi girişin mutlak değerine eşitle
end
if blm==0                 % Eğer blm=0 ise
    ara=0;
else                      % Eğer blm sıfırdan farklı ise
    ara=giris*100/blm;
end
y=ara;                   % Çıkışı ara değerine eşitle

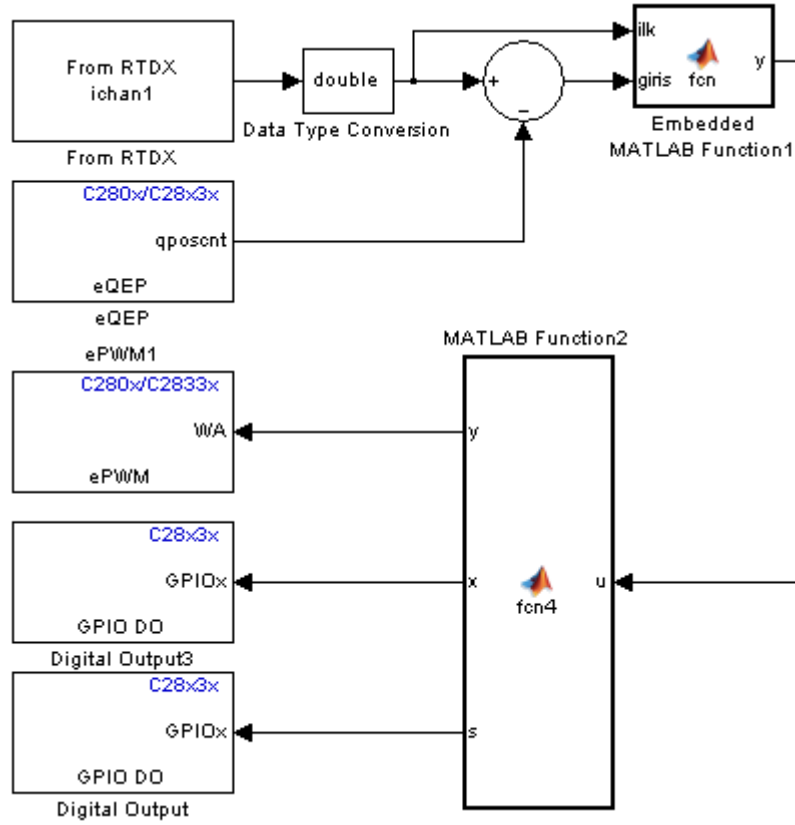
```

Fcn4 bloğunun içeriği aşağıdaki gibidir.

```

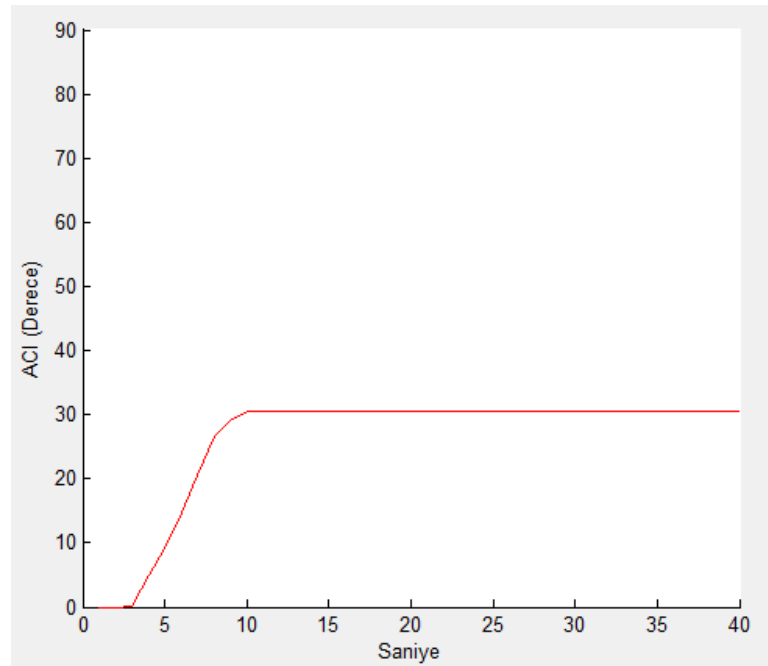
function [y,x,s]= fcn4(u)
m=0;                     % PWM değerini sıfırla
x1=0;                   % Motor kontrol uçlarını low seviyeye indir
s1=0;                   % Motor kontrol uçlarını low seviyeye indir
    if u<0               % Eğer giriş değeri sıfırdan küçük ise
        x1=1;           % x1 değerini High seviyeye yükselt
        s1=0;           % s1 değerini low seviyeye düşür
        m=abs(u)+75;     % PWM hesapla
    elseif u>0           % Eğer giriş değeri sıfırdan büyük ise
        x1=0;           % x1 değerini low seviyeye düşür
        s1=1;           % s1 değerini high seviyeye yükselt
        m=abs(u)+75;     % PWM hesapla
    end
y=m;                     % y değerini hesaplanan PWM değerine eşitle
x=x1;                   % x çıkışını x1 değerine eşitle
s=s1;                   % s çıkışını s1 değerine eşitle

```



Şekil 4.25 DC motor kontrolü için Simulinkte uygulanan yazılım bloğu

Sistem girişine 30 derecelik açı verildiğinde enkoderdan okunan pozisyon bilgisi Şekil 4.26’da gösterilmiştir.



Şekil 4.26 30 derecelik açı girdisine sistemin çıktısı

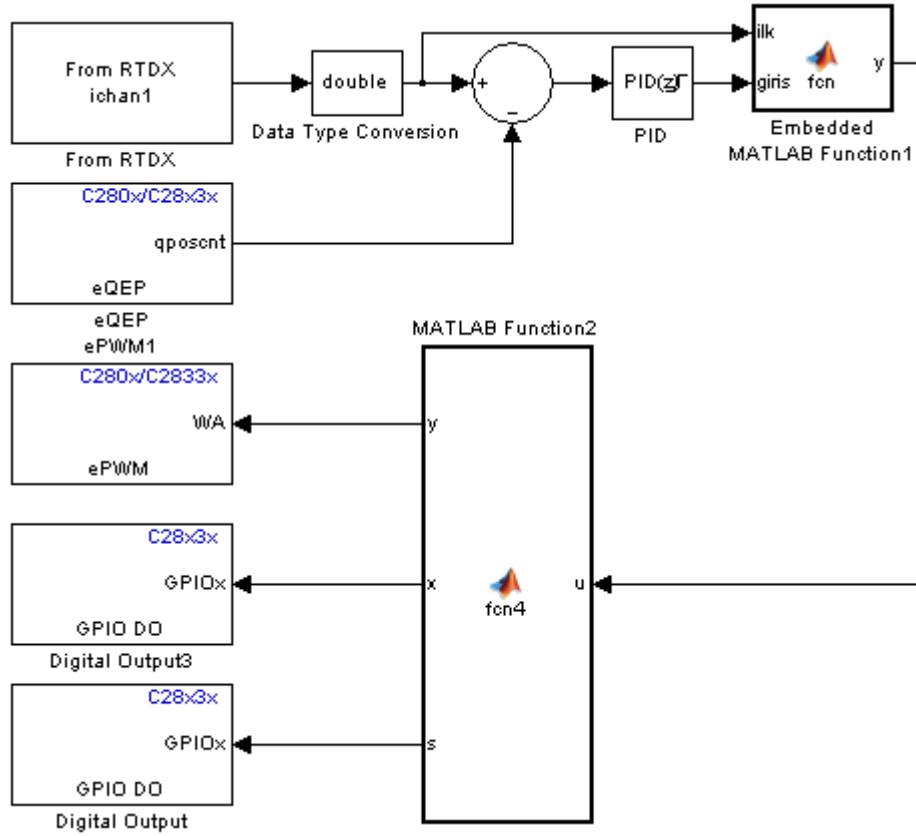
4.1.3.1.2 *PID kontrol ile Motorların kontrolü.* PID kontrol ile motorları kontrol edebilmek için gerekli olan Simulink bloğu oluşturuldu. Oluşturulan blog Şekil 4.27 de gösterilmiştir.

Şekil 4.27'deki Fcn bloğunun içeriği aşağıdaki gibidir.

```
function y = fcn(ilk,giris)
y=giris*100/12;
```

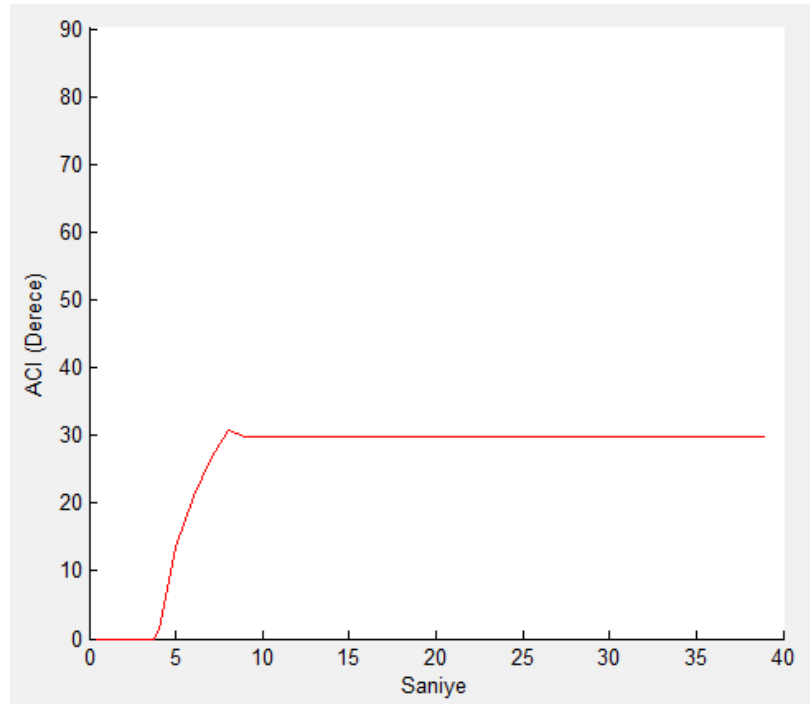
Şekil 4.27'deki Fcn4 bloğunun içeriği aşağıdaki gibidir.

```
function [y,x,s]= fcn4(u)
m=0;                % PWM değerini sıfırla
x1=0;               % Motor kontrol uçlarını low seviyeye indir
s1=0;               % Motor kontrol uçlarını low seviyeye indir
    if u<0           % Eğer giriş değeri sıfırdan küçük ise
        x1=1;        % x1 değerini High seviyeye yükselt
        s1=0;        % s1 değerini low seviyeye düşür
        m=abs(u)+75;  % PWM hesapla
    elseif u>0       % Eğer giriş değeri sıfırdan büyük ise
        x1=0;        % x1 değerini low seviyeye düşür
        s1=1;        % s1 değerini high seviyeye yükselt
        m=abs(u)+75;  % PWM hesapla
    end
y=m;                % y değerini hesaplanan PWM değerine eşitle
x=x1;               % x çıkışını x1 değerine eşitle
s=s1;               % s çıkışını s1 değerine eşitle
```



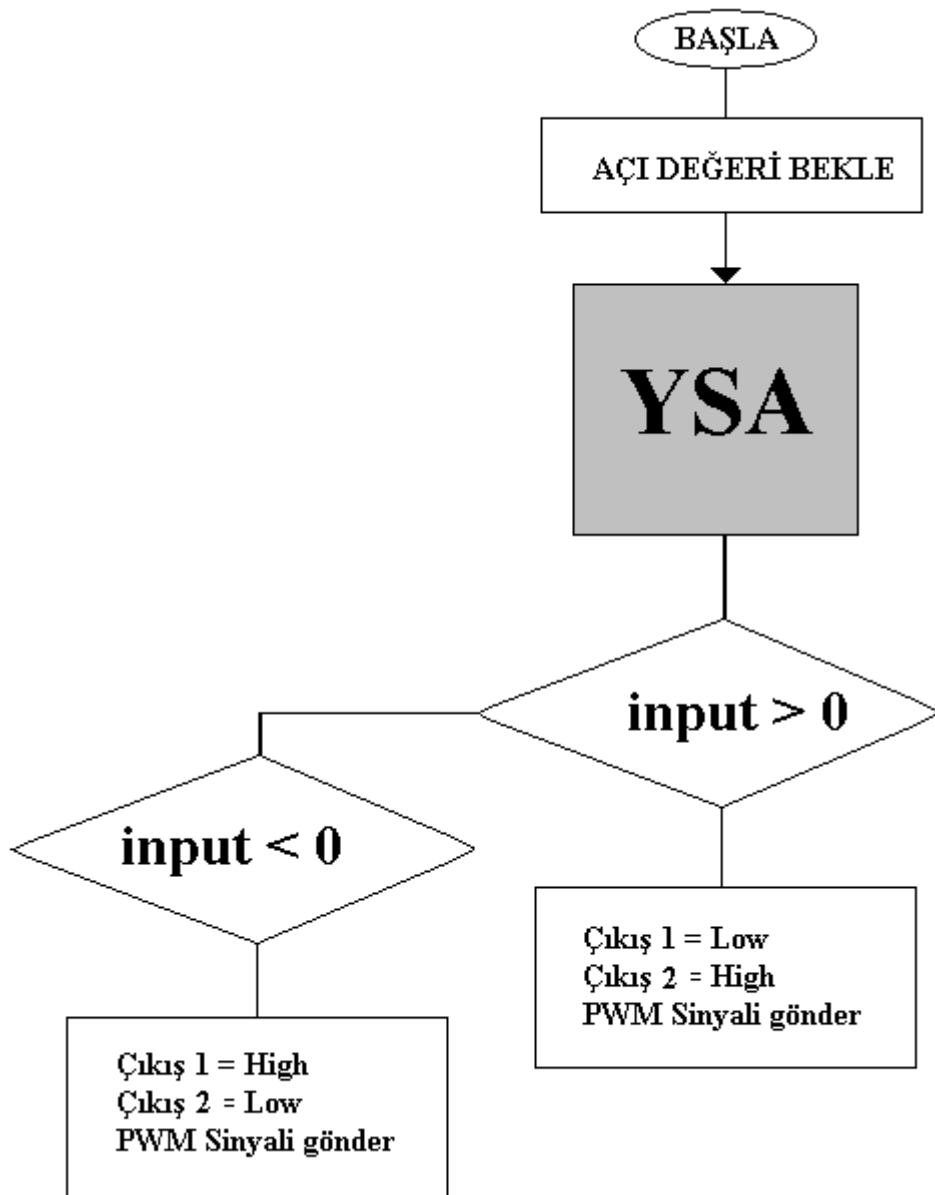
Şekil 4.27 DC motor kontrolü için Simulinkte uygulanan yazılım bloğu

Sistem girişine 30° açı verildiğinde enkoderdan okunan pozisyon bilgisi Şekil 4.28’da gösterilmiştir.



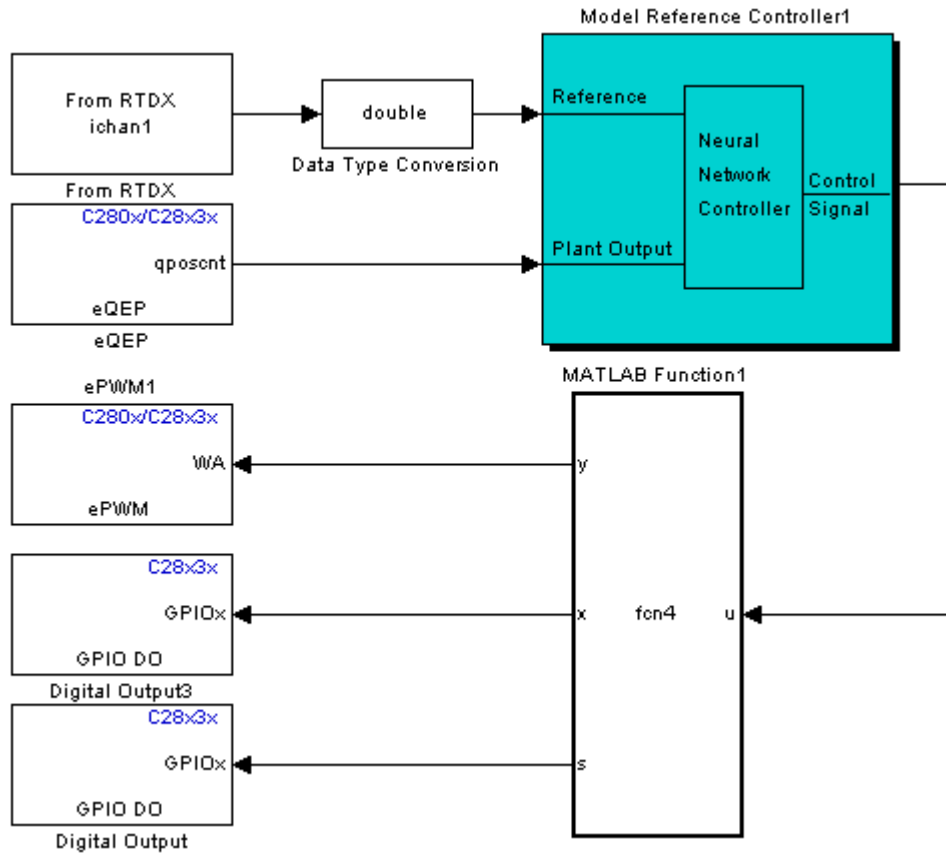
Şekil 4.28 30° derecelik açı girdisine sistemin çıktısı

4.1.3.1.3 DC Motorların Kontrolü için Kurulan Simulink Bloğu. Açı bilgisinin gelmesi ile birlikte Şekil 4.29'daki algoritma çalışmaya başlayacaktır. Şekilden de görülebileceği gibi programa açı bilgisi geldiğinde Yapay sinir ağları bloğu, girişlerindeki gidilmek istenilen açı ve motorun o an bulunduğu açı değerlerini kullanarak bir çıktığı değeri oluşturur. Eğer bu çıktı değeri sıfırdan büyük ise işlemcinin 1 nolu çıkışı low , 2 nolu çıkışı high olur. Aynı anda yapay sinir ağı bloğunun çıkışı motor besleme voltajıdır ve bu voltaj PWM sinyali olarak motorun girişlerine verilir.



Şekil 4.29 Yapay sinir ağları bloğu ve bu bloğun çıkışına göre sinyallerin oluşumu

Bu durumda motor saat yönünde döner. Sistem çıkışının başka bir deyişle pozisyon bilgisinin istenilen referans değere ulaşması ile birlikte yapay sinir ağı bloğu çıkışını sıfıra yaklaştırır. Sinir ağının çıkışının sıfır olması aynı zamanda motor besleme voltajının da sıfır olması demektir. Böylece motor hareket etmez. Eğer yapay sinir ağı bloğunun çıktı değeri sıfırdan küçük ise işlemcinin 1 nolu çıkışı high , 2 nolu çıkışı ise low olur. Bu çıkışlar ve gelen PWM değeri ile birlikte motor saat yönünün tersi yönde dönmeye başlar. Yine pozisyon bilgisinin referans değerine ulaşması ile birlikte yapay sinir ağı bloğu çıkışı sıfır olur. Motorun besleme voltajı kesilir ve motor durur. Bu sistemin çalıştığı Simulink bloğu Şekil 4.30’de gösterilmiştir.



Şekil 4.30 DC motor kontrolü için Simulinkte uygulanan yazılım bloğu

FCN4 bloğunun içeriği aşağıdaki gibidir.

```
function [y,x,s]= fcn4(u)
```

```
V=u*100/12
```

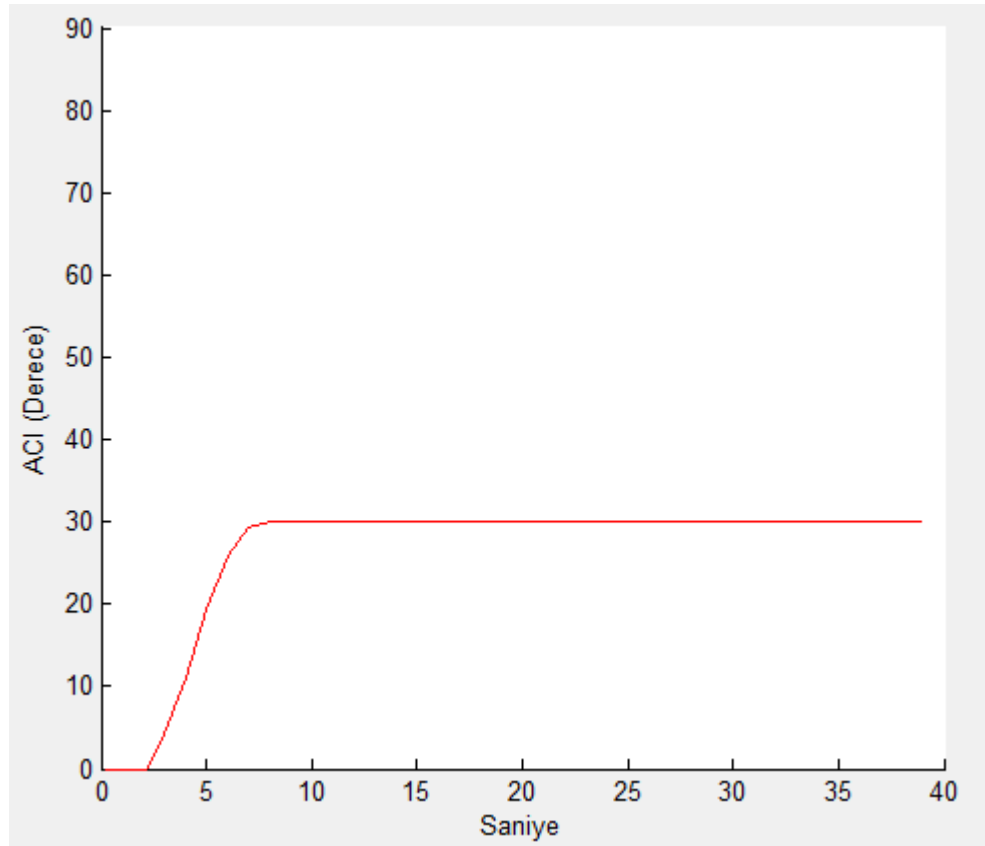
```
% Gelen Voltaj Değerini PWM değerine dönüştür
```

```

m=0; % Başlangıç değerlerini sıfırla
x1=0; s1=0;
if V<0 % Voltaj değeri sıfırdan küçükse motor bağlantıları
    x1=1; s1=0; % x1=1; s1=0; şeklinde olsun.
    m=abs(V)+22; % PWM değerini hesapla
else V>0 % Voltaj değeri sıfırdan büyükse motor bağlantıları
    x1=0; % x1=0; s1=1; şeklinde olsun.
    s1=1;
    m=abs(V)+22; % PWM değerini hesapla
end
y=m; % Çıkıştan PWM değerini gönder
x=x1; % Motorun uçlarını aktif et
s=s1;

```

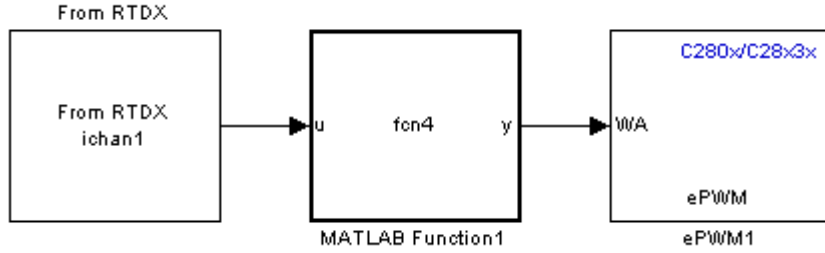
Sistem girişine 30° açı verildiğinde enkoderdan okunan pozisyon bilgisi Şekil 4.31’da gösterilmiştir.



Şekil 4.31 30 derecelik açı girdisine sistemin çıktısı

4.2 Mini Servo Motorun Kontrolü

Robotumuzun en altında bulunan mini servo motor için ise Şekil 4.32’de ki gibi bir Simulink bloğu oluşturuldu ve bu şekilde kontrol edildi.



Şekil 4.32 Mini servo motor kontrolü için Simulinkte uygulanan yazılım bloğu

FCN4 bloğunun içeriği aşağıdaki gibidir.

```

function [y]= fcn(u)
persistent t;           % Sabit değişken tanımla
persistent k;           % Sabit değişken tanımla
if isempty(t)           % Eğer t boş ise
    t = 6.692308;        % 6.692308'e eşitle
end
if isempty(k)           % Eğer k boş ise
    k = 0;               % sıfıra eşitle
end
persistent ara;         %Sabit değişken tanımla
if isempty(ara)         % Eğer ara boş ise
    ara = 0;             % sıfıra eşitle
end
if k ~= u;              % k ile u eşit değilse
    if t>(4+(u-40)*(7/130)); % t u'nun açılışından büyük ise
        ara=1;           % ara değeri 1 yap
        k=u;             % k ile u'yu eşitle
    else t<(4+(u-40)*(7/130)); % t u'nun açılışından küçük ise

```

```

    ara=2;                % ara değeri 2 yap
    k=u;                  % k'yı U'a eşitle
    end
end
if ara==1;                % Eğer ara 1 ise
    if t>(4+(u-40)*(7/130)); % t u'nun açılışından büyük ise
        t=t-0.05;          % t değerini 0.05'lik adımlar ile küçült
    else t<(4+(u-40)*(7/130)); % t u'nun açılışından küçük ise
        ara=0;              % ara'yı sıfır yap
    end
else
    if ara==2;            % Eğer ara 1 ise
        if t<(4+(u-40)*(7/130)); % t u'nun açılışından küçük ise
            t=t+0.05;        % t değerini 0.05'lik adımlar ile arttır.
        else t>(4+(u-40)*(7/130)); % t u'nun açılışından büyük ise
            ara=0;           % ara'yı sıfır yap
        end
    end
end
end
y = t;                    % çıkışı y'ye eşitle

```

BÖLÜM BEŞ

KONTROL SİSTEMİNDE KULLANILAN ARAÇLAR

Robot uygulamalarında kontrol sistemi önemli bir yer teşkil etmektedir. Kullanıcı bağımlı çalışan robotlarda robotun yapacağı işlemlerden, yazılım yüklenmesine kadar tüm işlemlerin kontrol edilebiliyor olması gerekmektedir. Kullanıcı bağımlı çalışmayan robotlarda ise özellikle yazılım yükleme işlemleri ve robotun kendi kendini kontrol etmesi işlemi yine kontrol sisteminin görevidir.

Bu bölümde sistemin farklı kontrol metotları ile nasıl kontrol edilebileceği ve kontrol sisteminde kullanılacak araçlar incelenmiştir.

5.1 Matlab

Matlab ilk olarak 1985 yılında Cleve Barry Moler tarafından geliştirilmiş bir paket programdır. Günümüzde mühendislik, matematik, fizik ve daha birçok alanda kullanıcılara kolaylıklar sunmaktadır. Sayısal hesaplamalar, denklem çözümleri, grafik işlemleri, istatistiksel incelemeler ve daha birçok matematiksel işlemler matlabda çok kolay bir şekilde sonuçlandırılabilir. Bunların yanı sıra Matlab içerisinde bulundurduğu araçlar sayesinde, farklı dallarda da işlem yapabilmemize olanak tanır. Örneğin Simulink ile simülasyonlar yapabileceğiniz gibi yapay sinir ağları aracını kullanarak , yapay sinir ağları ile uygulamalar yapılabilir.

Bu çalışmada Matlab, bir yandan CCS'in (Code Composer Studio) yönetimini sağlarken diğer yandan CCS ile Simulink arasındaki iletişimi sağlamaktadır.

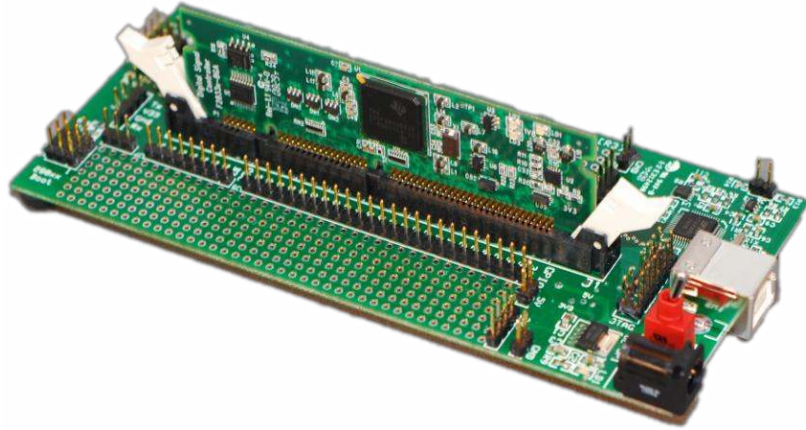
5.2 Simulink ve Guide

Simulink ve Guide Matlabın altında çalışan araçlardır. Her ne kadar Simulink aracı çok farklı şekillerde bizlere yardımcı olsa da en çok kullanıldığı alan, mevcut sistemlerin simülasyonunun yapılmasıdır. Bu çalışmada ise Simulinkte kurulmuş olan sistemin robot kolu üzerinde çalıştırılması özelliğinden yararlanılmıştır. Guide

penceresi ise kullanıcı ile Matlab programı arasında bir ara yüzey sağlamaktadır. Guide penceresi sayesinde kullanıcı çalışan bir programa istediği bilgileri gönderebilir ya da istediği bilgileri alabilir.

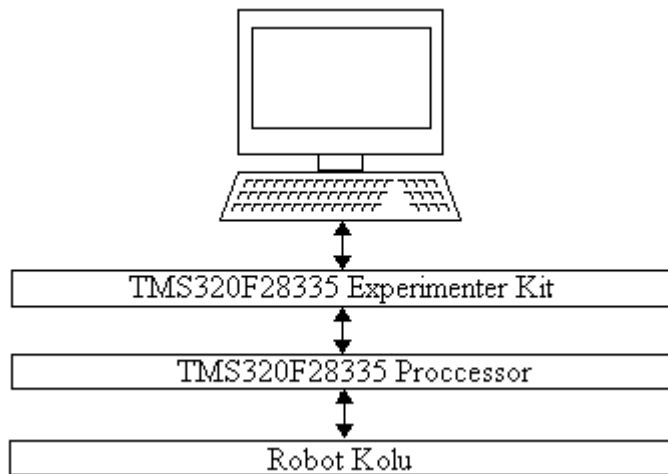
5.3 TMS320F28335 Experimenter Kit

TMS320F28335 experimenter kit, Texas Instruments'in çıkardığı, içerisinde TMS320F28335 docking station ve TMS320F28335 işlemci kartını bulunduran uygulama geliştirme amaçlı üretilmiş uygulama kartıdır. Şekil 5.1'de TMS320F28335 experimenter kit gösterilmiştir. Bu kit CCS ile haberleşebilir ya da CCS üzerinden gelen yazılımı işlemciye yükleyebilir.



Şekil 5.1 TMS320F28335 Experimenter Kit

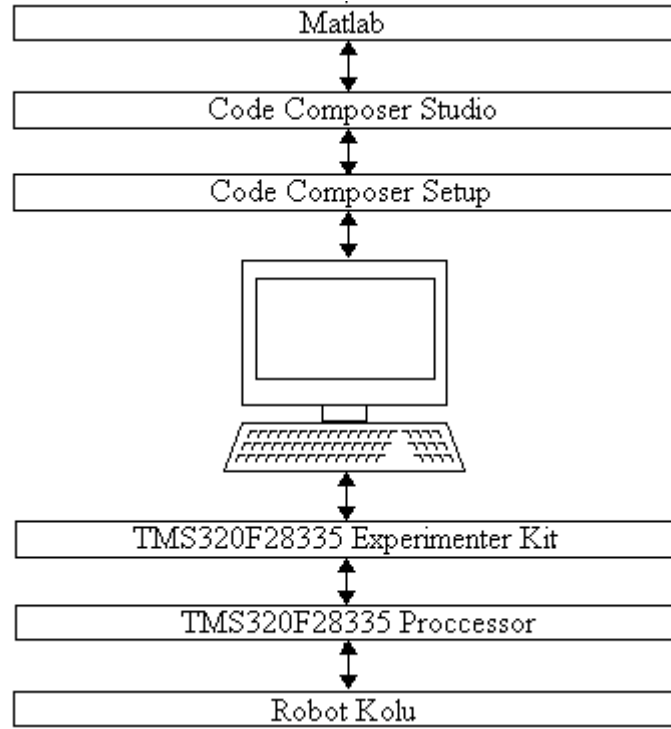
Şekilde 5.2'de TMS320F28335 experimenter kitin bu çalışmadaki yeri gösterilmiştir.



Şekil 5.2 TMS320F28335 Experimenter Kitin yeri

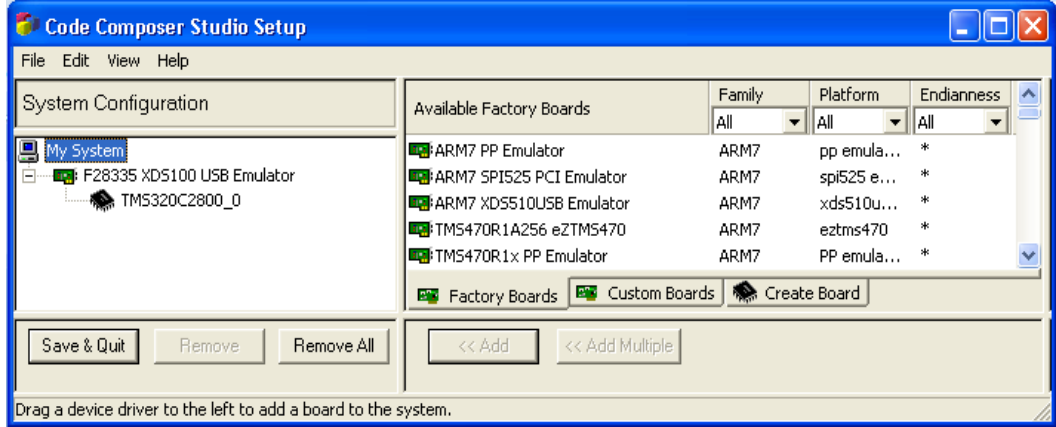
5.4 Code Composer Programı

Code Composer programı Code Composer Studio kısaca CCS ve Code Composer Studio Setup programlarından oluşur. Bu ikili, TMS320F28335 experimenter kiten veri alıp gönderebilir ya da işlemciye yazılım yükleyebilir. Burada CCS daha çok derleyici görevini üstlenmektedir. Code Composer Studio Setup ise TMS320F28335 experimenter kit ile bağlantıyı kurar. Bağlantıyı kurabilmek için kullanılacak olan işlemcinin ve uygulama kartının Şekil 5.3’de gösterildiği gibi Code Composer Studio Setup’a tanıtılmış olması gerekmektedir.



Şekil 5.4 Code Composer programının yeri

Bu çalışmada CCS programı Şekil 5.4’de de görüldüğü gibi, Matlabdan gelen kodları uygulama kartına gönderir. Bu kodlar gönderilmeden önce C koduna çevrilir ve bu şekilde derlenir. Matlab eğer yeni bir yazılım gönderiyorsa CCS bu yazılımı işlemciye yükler. Sonrasında programın çalışması esnasında haberleşme görevini yürütür.



Şekil 5.3 Code Composer Studio Setup programı

5.5 Motor Yön ve Hız Kontrolü

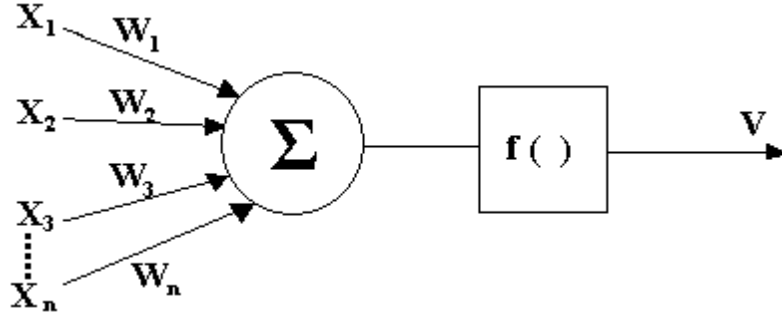
Daha önceki bölümlerde de anlatıldığı gibi doğru akım motorları birkaç yol ile sürülebilir. Örneğin transistör kullanarak, mosfet kullanarak, röle kullanarak, optocoupler kullanarak ya da özel motor sürücü entegreler kullanılarak DC motorlar sürülebilir. Bu tezde 4 kanallı LD293B motor sürücü entegresi kullanılmıştır. Bu entegre 4 kanalı sayesinde aynı anda 4 farklı motoru sürmemize olanak tanıyan 16 bacaklı bir entegredir. Çıkış akımı 1A olmasının yanı sıra, değerlerinde 2A'ye kadar çıkabilir. Aynı zamanda yüksek sıcaklıklarda kendi kendini korumaya aldığından güvenilir bir entegredir.

5.6 Yapay Sinir Ağları

Yapay sinir ağları insan sinir sisteminin yapay olarak benzerini oluşturma çabaları sonucunda ortaya çıkmış yapılardır. Kısaca YSA olarak da adlandırılabilir. Yapay sinir ağları günümüzde mühendislik, elektronik, matematik ve fizik gibi birçok dalda kullanılmaktadır. Yapay sinir ağları klasik teknikler kullanarak çözülmesi zor olan problemlerde karşımıza alternatif bir çözüm olarak çıkmaktadır. Yapay sinir ağları çalışma şekli olarak biyolojik sinir ağları ile benzer bir yapıya sahiptir. Her bir yapay sinir ağı yapay sinir hücrelerinden oluşmuştur. Genel olarak bir yapay sinir hücresi 5 parçadan oluşur. Bunlar:

- Girişler ($X_1, X_2, X_3, \dots, X_n$)
- Ağırlıklar ($W_1, W_2, W_3, \dots, W_n$)

- Toplama fonksiyonu (Σ)
- Aktivasyon fonksiyonu (f)
- Çıkış (V)

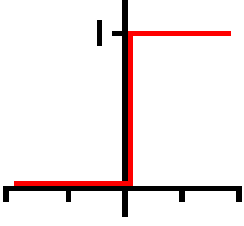
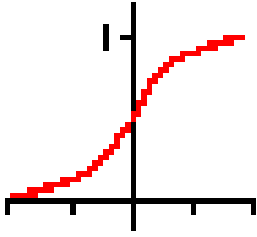
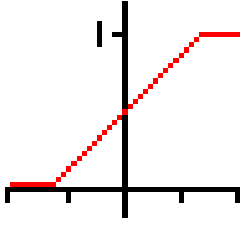
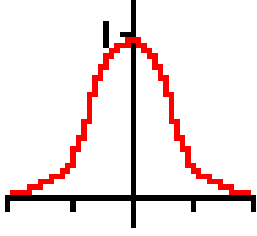
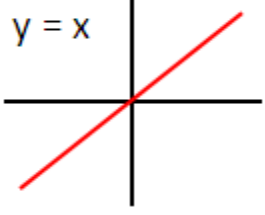
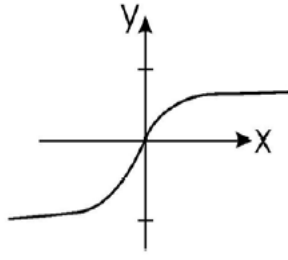


Şekil 5.5 Yapay sinir hücresi

Şekil 5.5’de de görülebileceği üzere tıpkı biyolojik sinir hücresinde olduğu gibi yapay sinir hücresinde de birçok giriş olmasına karşın tek bir çıkış vardır. Girişler görev olarak dentritlere benzerler. Gelen sinyalleri yapay sinir hücresine aktarmak ile görevlidirler. Şekil 5.5’de de görüldüğü gibi her bir giriş, ağırlığı olan bir bağlantı ile hücreye bağlıdır. Burada bağlantının ağırlık bilgisi, diğer işlem elemanlarının üzerindeki o işlem elemanın etkisi olarak düşünülebilir. Ağırlık büyüdükçe etki de büyüyecektir. Eğer ağırlık sıfır ise hiçbir etkisi yok demektir. Eğer ağırlık negatif ise ters etkisi olduğu anlamına gelir. Ortadaki toplama işlemi ise işlem elemanına gelen ağırlıklı toplam girişi hesaplar. Ancak burada toplam girişten kasıt girişlerin toplanması değil girişlerin ağırlıklar ile çarpıldıktan sonra toplanmasıdır. Aktivasyon fonksiyonunun görevi ise toplama işleminden çıkan sonuca göre bir çıkış vermektir. Burada farklı aktivasyon fonksiyonları kullanılabilir. Dolayısı ile kullanılan aktivasyon fonksiyonuna göre çıkış değerleri de değişiklik gösterir. Tablo 5.1’de değişik aktivasyon fonksiyonlarından örnekler gösterilmiştir.

Sonuç olarak; yapay sinir ağları karmaşık problemlere alternatif bir çözüm yolu öneren bir metot olarak tanınır ve birçok sistemde başarı ile uygulanır. Günümüzde yapay sinir ağları robotik sistemler, sınıflandırma, tahmin yapma, pattern ya da el yazısı , yüz gibi görüntü tanıma işlemleri, veri ilişkilendirme, filtreleme ya da yorumlama işlemleri, görüntü işleme gibi birçok işlemlerde kullanılır hale gelmiştir.

Tablo 5.1 Bazı Aktivasyon fonksiyonları

<i>Unit Step</i>		$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
<i>Sigmoid</i>		$f(x) = \frac{1}{1+e^{-\beta x}}$
<i>Piecewise Linear</i>		$f(x) = \begin{cases} 0 & \text{if } x \leq x_{min} \\ mx+b & \text{if } x_{min} < x < x_{max} \\ 1 & \text{if } x \geq x_{max} \end{cases}$
<i>Gaussian</i>		$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
<i>Identity</i>		$f(x) = x$
<i>Hiperbolik Tanjant</i>		$f(x) = (1 - \exp(-2s))/(1 + \exp(-2s))$

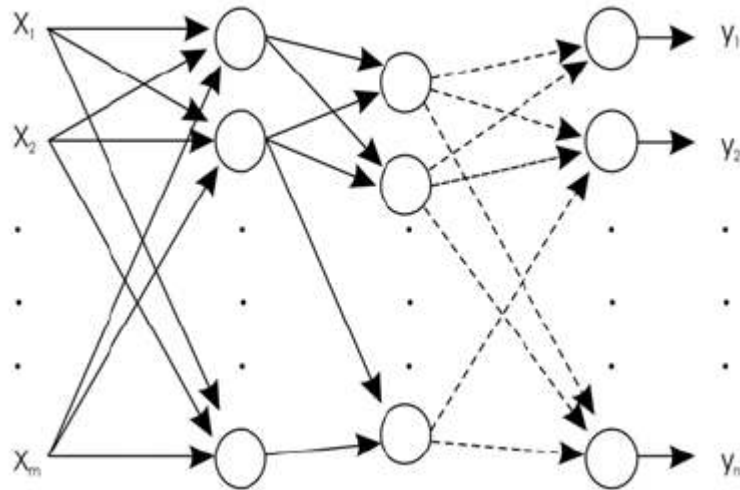
5.6.1 Yapay Sinir Ağları Çeşitleri

Günümüzde yapay sinir ağları çeşitlerine genel olarak bakarsak çok ciddi sayılara ulaştığı görülür. O sebeple burada yapay sinir ağları çeşitleri sadece, yapılarına göre yapay sinir ağları ve öğrenme algoritmalarına göre yapay sinir ağları olarak incelenmiştir.

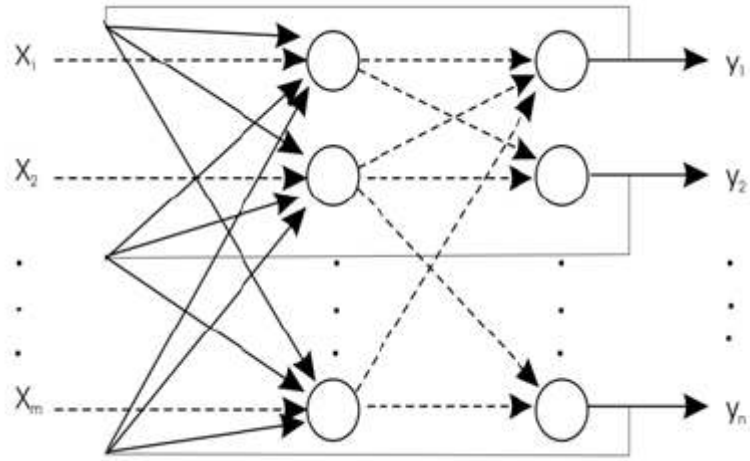
Yapılarına göre yapay sinir ağlarını üç grupta inceleyebiliriz:

- İleri Beslemeli Yapay Sinir Ağları
- Geri Beslemeli Yapay Sinir Ağları
- Kaskat Bağlantılı Yapay Sinir Ağları

İleri Beslemeli ağlarda Şekil 5.6'da da görüldüğü gibi sinir hücreleri girişten çıkışa doğru bir yapı sergilerler. Bir katman sadece kendinden sonraki katmana bağlıdır. Geri beslemeli yapay sinir ağlarında ise Şekil 5.7'de de görüldüğü gibi bir sinir hücresinin çıkışı ya kendinden önceki katmandaki sinir hücresine ya da kendi katmanındaki bir sinir hücresine verilir. Kaskat bağlantılı yapay sinir ağlarında ise sinir hücresi yalnız daha önceki katmanlardaki sinir hücrelerinden veri alır.



Şekil 5.6 İleri beslemeli YSA (Yurtoğlu, H.)



Şekil 5.7 Geri beslemeli YSA (Yurtoğlu, H.)

5.6.2 Yapay Sinir Ağları Eğitim Yöntemleri

Öğrenme algoritmalarına göre ise yapay sinir ağları yine üç grupta incelenebilir.

- Eğiticili Öğrenme : Öğrenme sırasında yapay sinir ağına girdi verilerinin yanı sıra olması gereken çıktı verileri verilir ve ağ kendi çıktısı ile olması gereken çıktı arasındaki farkı bağlantı ağırlıklarına ekler.
- Eğiticişiz Öğrenme : Öğrenme sırasında yapay sinir ağına herhangi bir çıktı bilgisi verilmez. Yapay sinir ağı girdilere göre bilgileri sınıflandırır.
- Destekleyicili Öğrenme : Öğrenme sırasında her iterasyon sonrası ağa çıktının iyi ya da kötü olduğu bilgisi verilir. Ağ bu bilgiler ışığında kendisini tekrardan yapılandırır.

Bu çalışmada MLP (Multilayer Perceptron) tipi yapay sinir ağı kullanılmıştır. MLP tipi yapay sinir ağları nöronlardan oluşan katmanlarından oluşan yapılardır. Bir önceki katmandaki tüm nöronlar bir sonraki katmandaki nöronlar ile bağlantılıdır. İlk katman girdilerin verildiği katmandır ve girdi katmanı olarak isimlendirilir. Girdi katmanı yapay sinir ağına bilgi alma işlemini üstlenir. Son katman ise çıktı katmanıdır. Çıktı katmanı yapay sinir ağının çıktılarının alınacağı katmandır. Girdi katmanı ile çıktı katmanı arasında kalan katmanlara ise gizli katmanlar adı verilir. MLP tipi ağlarda bilginin akışı ileri yönlüdür. Eğitim algoritması olarak genelde geriye yayılım algoritması kullanılır. Sistem eğiticili öğrenme yoluyla eğitilir.

Çok katmanlı perceptron yapısı (MLP tipi yapı) geri yayılım eğitim algoritmasına uygunluğu ve eğitim sonuçlarındaki başarısı nedeni ile günümüzde en yaygın kullanılan yapay sinir ağı yapılarından birisidir. Bu sebeple bu çalışmada da MLP tipi yapay sinir ağı kullanılmıştır.

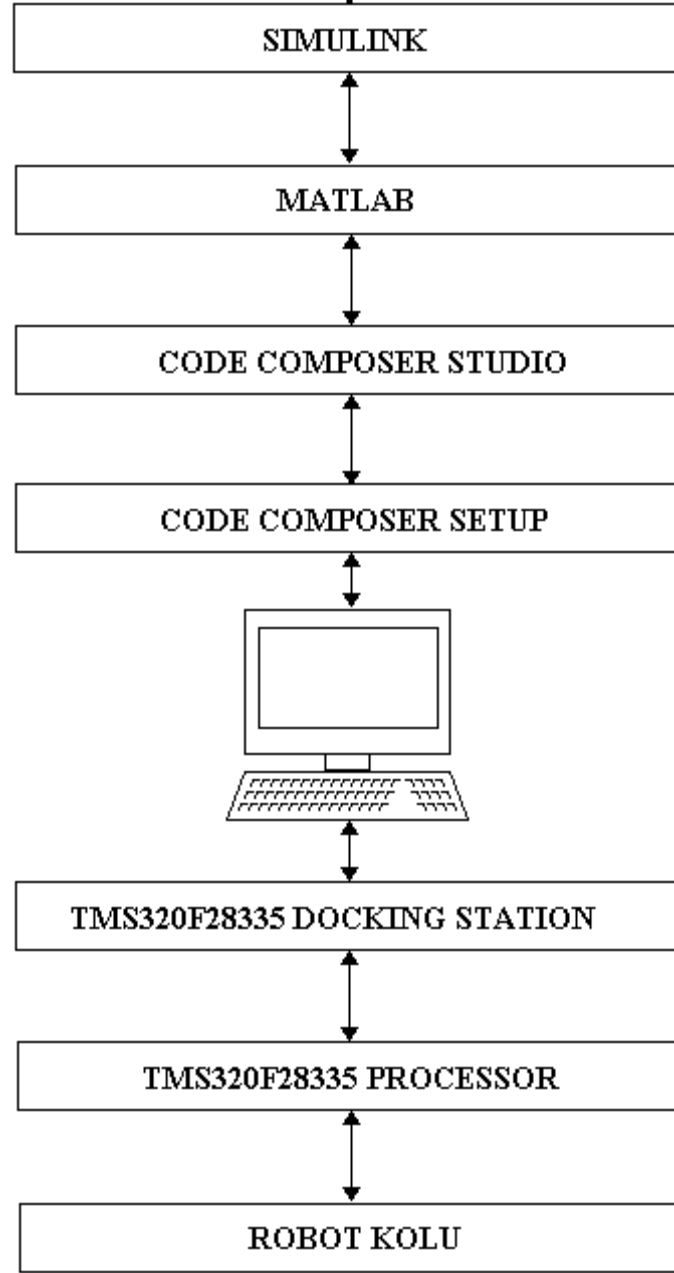
5.7 Sistemin Kurulumu

Kontrol sistemi robot kolu kontrolü işlemini gerçekleştiren sistemdir. Kontrol sisteminde kullanılan araçlardan daha önce bahsedilmişti. Burada ise genel olarak sistemin nasıl çalıştığından bahsedilecektir. Kontrol sistemi bu noktada iki farklı işlem için incelenecektir.

- Yazılım yükleme işlemi
- İşlemci çalışırken haberleşme

Burada yazılım yükleme işleminden kasıt, yapılmış olan programın işlemciye yazılmasıdır. Şekil 5.8’de de görüldüğü gibi işlem Simulink ile başlar. Kullanıcının işlemi başlatması ile birlikte Simulink, bloklar ile hazırlanmış olan programı Matlab’a gönderir.

Matlab bu esnada, eğer CCS (Code Composer Studio) programı açık değilse CCS programını çalıştırır ve veriler CCS programına gönderilir. CCS programı Matlabdan gelen bilgilere göre C kodlarını hazırlar. Bu esnada yine Matlab tarafından açılmış olan Code Composer Setup programının kurmuş olduğu bağlantı vasıtası ile assembly kodları docking station’a gönderilir ve işlemciye yazılır. Bu işlem programın büyüklüğüne göre birkaç dakika sürebilir ve işlemin hangi aşamasında olduğu CCS ekranından izlenebilir.

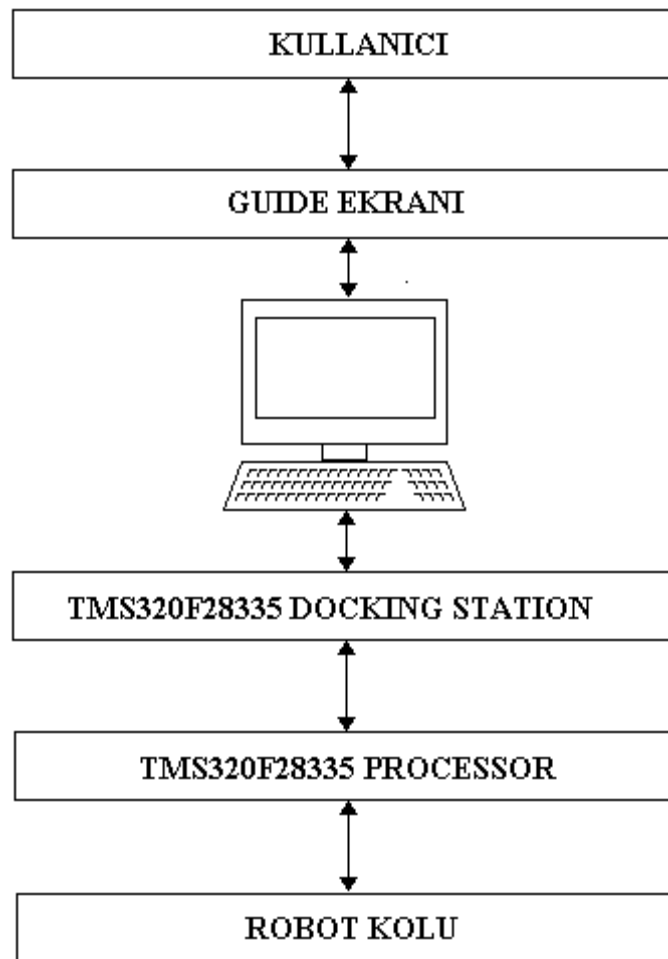


Şekil 5.8 Mikrokontroller’a yazılım atılırken araçların kullanımı

Bu işlemler bittikten sonra işlemci çalışmaya başlar. Artık robot kolunu kontrol edecek seviyededir ancak kullanıcıdan, robot kolunun gideceği noktanın koordinatlarının gelmesi gerekmektedir. İşte burada kullanıcının veri girmesi ya da veri alması işlemi için arayüz penceresi kullanılabilir. Şekil 5.9’da örnek bir guide penceresi gösterilmiştir.

The screenshot shows a window titled 'interface' with a blue title bar. Inside, there is a section titled 'GIDILECEK NOKTANIN KOORDINATLARINI GIRINIZ'. Below this title are three input fields labeled 'X KOORDINATI', 'Y KOORDINATI', and 'Z KOORDINATI'. At the bottom right of the window are two buttons labeled 'GİT' and 'ÇIKIŞ'.

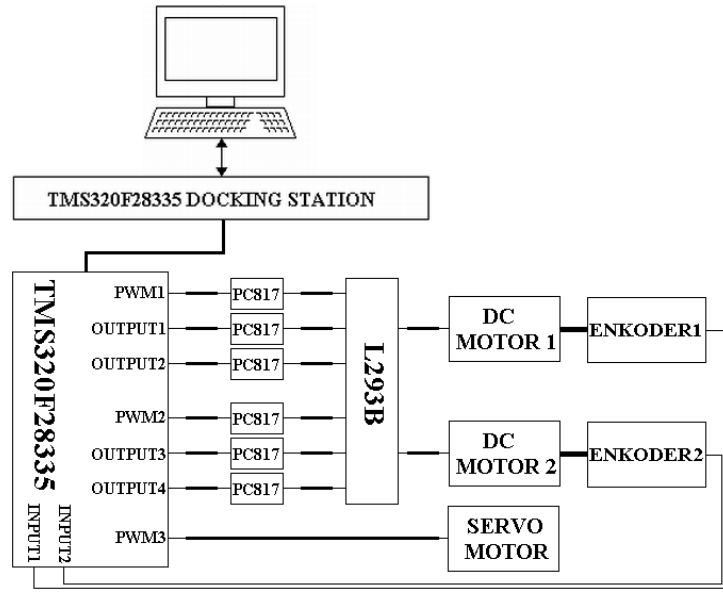
Şekil 5.9 Guide penceresi



Şekil 5.10 Çalışma zamanında kontrol

Bu çalışmada Guide penceresinin açılması için Simulinkte program çalıştırıldığında Guide penceresinin de açılmasını sağlayan bir komut eklenmiştir. Burada Guide penceresi ile Robot kolumuz arasındaki haberleşme real time data exchange (RTDX) vasıtası ile sağlanmaktadır. Başka bir deyişle, Guide penceresinde kullanıcının girmiş olduğu koordinat bilgileri, RTDX üzerinden işlemciye iletilir ve robot kolunun o koordinatlara gitmesi sağlanır. Guide penceresinin bu çalışmadaki yeri Şekil5.10’da gösterilmiştir.

5.8 Donanımın Genel Görünümü



Şekil 5.11 Donanımın Genel Görünümü

Sürücü kartı olarak optocoupler ve L293B entegresinden oluşan bir sürücü kartı tasarlandı. İşlemci ile L293B arasına koruma amaçlı PC817 optocouplerler konuldu. DC motorlar L293B entegresi ile sürüldü. Donanımın genel görünümü Şekil 5.11’de gösterilmiştir.

5.9 TMS320F28335 Experimenter Kitin Simulink ile Kullanımı

Bu çalışmada motorların kontrolü için Simulink ve eZdsp kartı kullanıldı. Simulinkte bu şekilde program yazarken , klasik birçok derleyicinin aksine programın yazılması blokların yerleştirilmesi yoluyla yapılır. eZdsp kartı ile Simulinkte program yazabilmek için öncelikle doğru işlemci bloğunun Simulink

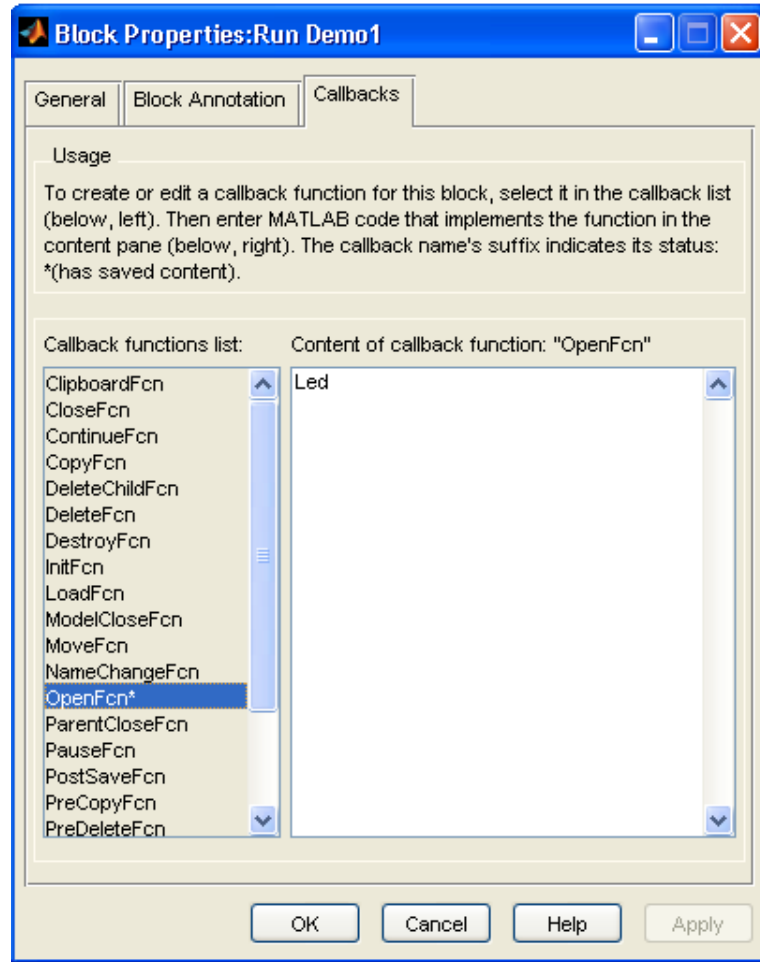
penceresine eklemek gerekmektedir. Bu çalışmada kullanılan işlemci 28335 işlemcisinin bloğunu Şekil 5.33 de gösterilmiştir.



Şekil 5.33 28335 işlemcisinin için Simulink bloğunu ve başlama butonu

Şekilde görülen işlemci bloğu TMS320F28335 işlemcine ait bir bloktur. Tezimizde de bu blok kullanıldı. CCS üzerinde de bu işlemci kaydedilerek TMS320F28335 docking station ile bağlantı sağlandı.

Simulink Programı üzerine yukarıdaki Şekil 5.33’de görülen başlama butonu eklendi ve tüm sürecin “Build / Rebuild & Run” butonuna basılması ile başlaması için gerekli ayarlamalar yapıldı. Başlama butonuna basılması ile birlikte Simulink blok diyagramı Matlaba gönderilir ve Matlab, CCS ve işlemci arasında daha önceki bölümlerde anlatılan haberleşme başlatılır. İşlemci artık açılışının gelmesini beklemektedir. Burada programın çalıştırılması ile birlikte son kullanıcı ile sistemin haberleşmesini sağlayan guide penceresinin de açılması için “Build / Rebuild & Run” butonuna aşağıdaki Şekil 5.34’de de görüldüğü gibi open fcn eklendi.



Şekil 5.34 Guide penceresinin açılmasını sağlayan işlem

BÖLÜM ALTI

SONUÇLAR

Bu çalışmada, 3 döner eksene sahip bir robot kolu ve bu robot kolunun kontrol sistemi tasarlandı. Robot kolu mekanik tasarımında, kullanılan mekanik parçalar tek tek nasıl ve hangi amaç için tasarlandığı anlatıldı. Robot kolunda kullanılan DC motor, enkoder, mini servo motor gibi bileşenlerin nasıl seçildikleri ile ilgili bilgiler verildi. Robot kolunun mekanik tasarımında kullanılacak olan uzuvların seçimi ve bu uzuvların ağırlıklarına bağlı olarak hesaplanan tork değerleri hakkında bilgiler verildi. Robot kolunun kontrolünde ara bir elektronik devre olarak tasarlanan sürücü devre ve yapısı gösterildi. Kontrol sistemi olarak geri beslemeli kontrol, PID kontrol ve yapay sinir ağları ile kontrol yöntemleri ile uygulamalar yapılarak, Matlab ve Simulink ortamlarında sonuçlar birbiri ile karşılaştırıldı. Yapılan incelemelerde geri beslemeli kontrolün referans noktaya hızlı ulaştığı ancak küçük açı değişimlerinde kararlı hale geçemediği, bu yönüyle özellikle küçük açı değerleri için PID ve yapay sinir ağları ile kontrol yöntemlerinden daha başarısız olduğu, PID kontrolün ise referans noktaya ulaşma süresinin yapay sinir ağları ile kontrolden daha uzun olduğu, yapay sinir ağları ile kontrolün diğer iki kontrolden daha iyi sonuçlar verdiği gözlemlendi.

KAYNAKLAR

- Ata, R. (1995). Kablolulu alçak gerilim fiderlerinin optimum kesit değerlerinin yapay sinir ağları ile tayini. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 1 (1), 1 – 4.
- Bal, C. (2002). *Doğru akım motorlarının hız denetiminde yapay sinir ağlarının performans analizi*. Fırat Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar anabilim dalı, Yüksek Lisans Tezi
- Bingül, Z. ve Küçük, S.(2005). *Robot tekniği 1* (1). İstanbul: Birsen yayınevi
- Buja, G.S. (1993), Neural network implementation of a fuzzy logic controller, *Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93., International Conference on*, (1), 414 – 417.
- Çoturoğlu, F. (2004). *Bilgisayar ve Programlama Matlab (Bölüm1-Giriş)*. Zonguldak Karaelmas Üniversitesi Makine Mühendisliği Bölümü ders notu
- Deok, H. S.ve Seul, J.(2007). Geometrical analysis of inverse kinematics solutions and fuzzy control of humanoid robot arm under kinematics constraints, *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, 1178 – 1183.
- El-Khouly, F.M., Sharaf, A.M., Abdel-Ghaffar, A.S. ve Mohammed, A.A. (1994). An adaptive neural network speed controller for permanent magnet dc motor drives, system theory, 1994. *Proceedings of the 26th Southeastern Symposium*, 416 – 420
- Hongbo, W. (2007). Kinematics and control for a personal robot with five degrees-of-freedom arms networking, *Sensing and Control, 2007 IEEE International Conference on*, 507 – 512.
- Hoque, M.A., Zaman, M.R.ve Rahman, M.A. (1995). Artificial neural network based permanent magnet dc motor drives, *Industry Applications Conference, 1995*.

Thirtieth IAS Annual Meeting, IAS '95. Conference Record of the 1995 IEEE, (1), 98 – 103.

Jae-Sub, K., Byung-Jin, J., Ki-Tae, P., Chung-Hoon, C. ve Dong-Hwa, C. (2008). Maximum power point tracking control of pv system for dc motors drive with neural network, *Smart Manufacturing Application, 2008. ICSMA 2008. International Conference*, 514 – 519.

Kıvrak, A.Ö. (2006) *Design of control systems for a quadrotor flight vehicle equipped with inertial sensors*, Atılım Üniversitesi Mekatronik Mühendisliği, Yüksek Lisans Tezi

Kieffer, S., Morellas, V. ve Donath, M. (1991). Neural network learning of the inverse kinematic relationships for a robot arm, *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, (3), 2418 – 2425.

Kosuge, K. ve Furuta, K. (1985). Kinematic and dynamic analysis of robot arm, robotics and automation. proceedings. *1985 IEEE International Conference on*, 1039 – 1044.

Lefranc, G. ve Cisternas, B. (1994). Comparison of two neural network methods applied to a tracking control system, *Industrial Electronics, 1994. Symposium Proceedings, ISIE '94., 1994 IEEE International Symposium on*, 66 – 71.

Merdan, O. (b.t.). *Elektrik Motoru Nedir? Nasıl Çalışır?*. 15.Eylül.2012,
<http://www.bilgiustam.com/elektrik-motoru-nedir-nasil-calisir/>

Messner, B., Tilbury, D., Oms, L., Pagel, J., Sun, Y., Suri, M., Caruana, C., Kawano, D., Nakai, B., Prempraneerach, P. ve Luntz, J. (b.t.). *Dc motor speed modelling in simulink*. 01.Şubat.2013,
<http://www.library.cmu.edu/ctms/ctms/simulink/examples/motor/motorsim.htm>

Ogawa, T., Matsuura, H. ve Kanada, H. (2005). A solution of inverse kinematics of robot arm using network inversion, *Computational Intelligence for Modeling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, (1), 858 – 862.

Özel, E. (2009). *PWM yöntemi ile dc motorun pc üzerinden kontrolü*. Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Elektrik – Elektronik mühendisliği anabilim dalı, Yüksek Lisans Tezi

Petros, I. (2003). *Robust Adaptive Control*. 10.Şubat.2013, http://www-bcf.usc.edu/~ioannou/RobustAdaptiveBook95pdf/Robust_Adaptive_Control.pdf

Seidl, D.R., Reineking, T.L. ve Lorenz, R.D. (1992). Use of neural networks to identify and compensate for friction in precision, *position controlled mechanisms, Industry Applications Society Annual Meeting, 1992, Conference Record of the 1992 IEEE*, (2), 1937 – 1944.

Servo *Motors*, (2010). 25.12.2012 <http://badulaquesdachina.blogspot.com/2010/03/servo-motors.html>

Stergiou, C. ve Siganos, D. (b.t.). *Neural networks*. 14.Aralık.2012, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Contents

T.C. Milli Eğitim Bakanlığı. (2011). *Elektrik - Elektronik Teknolojisi, AC ve DC Makineler*. 523EO0070

Takefuji, Y. (1991). Artificial neural networks for four-coloring map problems and k-colorability problems. *Circuits and systems*, (38) içinde (326 – 333). IEEE Transactions on.

Tezioğlu, H. (2008). *DC motorun hız performans eğrisi kullanılarak kazanç parametrelerinin (p,i,d), bulunması*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi

- Weerasooriya, S. ve El-Sharkawi, M.A. (1991). Identification and control of a dc motor using back-propagation neural networks, *Energy Conversion, IEEE Transactions on*, 663 – 669.
- Weerasooriya, S. ve El-Sharkawi, M.A. (1993). Laboratory implementation of a neural network trajectory controller for a dc motor, *Energy Conversion, IEEE Transactions on*, 107 – 113.
- Yanyan, J. ve Michel, H. (2012). How to control a dc motor using ZX, *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, 2376 – 2380.
- Yurtoğlu, H. (2005). *Yapay sinir ağları metodolojisi ile öngörü modellemesi: bazı makroekonomik değişkenler için Türkiye örneği*, Devlet Planlama Teşkilatı, Uzmanlık tezi
- Zeb, J., Qaiser, N., Iqbal, N. ve Ahmed, N. (2005). Kinematics modeling of the arm of a radiation protection assistant robot, *9th International Multitopic Conference, IEEE INMIC 2005*, 1 – 6.

EKLER

Hextronik HX12k 208oz@6V(TowerPro MG995) High Speed, High torque Servo



Detailed Product Description

Hextronik HX12k 208oz@6V(TowerPro MG995) High Speed, High torque Servo Stall torque:15.0kg/cm (6.0V)

TowerPro MG995 Metal Gear High Speed Servo

- Size: 40.6mm x 19.8mm x 37.8mm
- Weight: 55g
- Operating Speed: 0.16sec/60degree (6.0V)
- Stall torque:10.0 kg/cm (6.0V)
- Temperature Range: 0 degree to 135 degree
- Dead band width: 4ms
- Operating Voltage: 4.8V~6V

Product Config Table			
Weight (g)	55		
Torque (kg)	10		
Speed (Sec/60deg)	0.16		
A(mm)	43		
B(mm)	42		
C(mm)	39		
D(mm)	20		
E(mm)	54		
F(mm)	27		
	0		
	0		