

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**





KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ORCID : - - -

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde

Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : / /

Tezin Savunma Tarihi : / /

Tez Danışmanı :

ORCID : - - -

Trabzon

ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektronik Mühendisliği Yüksek Lisans Programı'nda Yapılan bir Çalışmadır.

Bu tez çalışması boyunca gerek konu seçimi ve gerekse çalışmaların yürütülmesi sırasında bilgi ve deneyimlerini bana aktaran, danışman hocam Prof. Dr. İsmail H. Altaş' a ilgi, destek ve tecrübelerini esirgemediğinden dolayı teşekkür ederim. Yüksek lisansımı yaparken burs verdiği için Türkiye burslarına da teşekkür ediyorum.

Son olarak hayatım boyunca varlıkları destekleriyle bana güç veren ve her zaman yanımda olan anneme, babama şükranlarımı sunarım.

Cemil DABBAH

Trabzon 2020

TEZ ETİK BEYANNAMESİ

Yüksek lisans tezi olarak sunduğum " İki Tekerlekli Mobil Robotun Denge Kontrolü" başlıklı bu çalışmayı baştan sona kadar danışmanım Prof. Dr. İsmail H. Altaş 'ın sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığım/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma süresinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 15/05/2020

Cemil DABBAH

İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET.....	VII
SUMMARY.....	VIII
ŞEKİLLER DİZİNİ.....	IX
TABLolar DİZİNİ.....	XII
SEMBOLLER DİZİNİ.....	XIII
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Literatür Taraması.....	1
1.2.1. İki Tekerlekli Kendini Dengeleyen Robot Türleri.....	1
1.2.1.1. Mobil Ters Sarkaç.....	2
1.2.1.2. Segway Kişisel Taşıyıcı.....	3
1.2.1.3. İki Tekerlekli Kendini Dengeleyen Otonom Robot.....	4
1.2.1.4. Otonom Kentsel Ulaşım Sistemi İçin İki Tekerlekli Araç.....	4
1.2.2. Kontrol Metodolojisi.....	5
1.3. Mekanik Tasarım Ve Elektro-Mekanik Elemanlar.....	10
1.3.1. Mekanik Tasarım.....	10
1.3.2. Elektro-Mekanik Bileşenler.....	10
1.3.3. DC Motoru.....	10
1.3.4. Dişli Dc Motor.....	11
1.3.5. Darbe Genişliği Modülasyonu (PWM).....	11
1.3.5.1. PWM Darbelerinin Görev Döngüsü.....	12
1.3.6. H-Köprü Devresi.....	12
1.3.7. L298n Sürücü.....	14
1.3.8. Artımlı Döner Kodlayıcı.....	16
1.3.9. Ataletsel Ölçüm Birimi (IMU).....	17

1.4.	Sistemin Dinamik Modeli.....	18
1.4.1.	DC Motorun Dinamik Modeli.....	19
1.4.2.	Tekerleklerin Dinamik Modeli.....	21
1.4.3.	Ters Sarkacın Dinamik Modeli.....	24
1.5.	Kontrol Sistemi Tasarımı.....	31
1.5.1.	Doğrusal Kuadratik Gauss Denetleyici Tasarımı.....	31
1.5.1.1.	Doğrusal Kuadratik Regülatör Kontrol Yöntemi.....	32
1.5.1.2.	Kalman Filtre Tasarımı.....	33
1.5.2.	Bulanık Mantık Kontrolü.....	36
1.5.2.1.	Bulanık Kümeler.....	36
1.5.2.2.	Üyelik Fonksiyonları.....	36
1.5.2.2.1.	Üçgen Üyelik Fonksiyonu.....	36
1.5.2.2.2.	Yamuk Üyelik Fonksiyonu.....	37
1.5.2.2.3.	Gauss Üyelik Fonksiyonu.....	38
1.5.2.3.	Bulanık İşlemci.....	39
1.5.2.3.1.	Bulanıklaştırma.....	39
1.5.2.3.2.	Kural Tabanı.....	39
1.5.2.3.3.	Bulanık Çıkarım.....	40
1.5.2.3.4.	Durulaştırma.....	40
2.	YAPILAN ÇALIŞMALAR , BULGULAR VE İRDELEME.....	41
2.1.	Giriş.....	41
2.2.	Benzetim Çalışması.....	41
2.2.1.	LQG Dnetleyici.....	41
2.2.1.1.	Dik Kendini Dengeleyen Durum.....	43
2.2.1.2.	Referans İzleme Durumu.....	45
2.3.	Bulanık Mantık Denetleyicisi.....	48
2.3.1.1.	Matlab'daki Bulanık Mantık Denetleyicisinin Tasarımı.....	48
2.3.1.2.	Bulanık Denetleyicinin Kuralları.....	50
2.3.1.3.	Dik Kendini Dengeleyen Durum.....	53
2.4.	LQG, LQR Ve Bulanık Mantık Denetleyicisi Arasında Karşılaştırılması.....	57
3.	SONUÇLAR.....	59
4.	ÖNERİLER.....	60

5.	KAYNAKLAR.....	61
	ÖZGEÇMİŞ	



Yüksek Lisans

Özet:

İki Tekerlekli Mobil Robotun Denge Kontrolü

Cemil DABBAH

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. İsmail H. ALTAŞ
2020, Sayfa 62

Bu tez, Matlab / Simulation kullanarak LQG ve bulanık mantık kontrol yöntemlerini kullanarak kararsız ve doğrusal olmayan iki tekerlekli kendini dengeleyen robotun kontrolünü sunmaktadır. Kontrol amacı, robot hareket ederken sistemin dengesini bir denge noktası etrafında tutmaktır.

Robotun açısını ve konumunu kontrol etmek için bir LQG kontrolörü geliştirmek amacıyla doğrusallaştırılmış bir durum-uzay modeli elde etmek için doğrusallaştırılmış bir matematiksel model çıkarılmıştır. Sistemin durum değişkenlerini tahmin etmek için optimal doğrusal kuadratik tahminci tasarlanmış ve optimal tam durum geri besleme regülatörüne (LQR) bağlanmıştır. Bundan sonra tasarlanan kontrolör robotun doğrusal olmayan dinamik modeline uygulanmıştır.

Kontrol algoritması akıllı kontrole genişletilmiş, robotun açısını kontrol etmek için bulanık mantık denetleyicisi geliştirilmiştir. Ayrıca, tasarlanmış kontrolör Matlab / Simulink'te robotun doğrusal olmayan dinamikleri üzerine uygulanmıştır.

Matlab / Simulink'te elde edilen simülasyon sonuçları, önerilen kontrolörlerin tekerlekler hareket halindeyken robotun kendi kendini dengelemesini başarabildiğini göstermektedir.

Anahtar Kelimeler: İki tekerlekli kendi kendini dengeleyen robot, Doğrusal kuadratik regülatör, Doğrusal kuadratik gauss, Bulanık mantık kontrolü

Master Thesis
Summary

Balance Control of a Two-Wheeled Mobile Robot

Cemil DABBAH

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Electrical and Electronics Engineering Graduate Program
Supervisor: Prof. Dr. İsmail H. ALTAŞ
2020, 62 Pages

This thesis presents the control of unstable and nonlinear two-wheeled self-balancing robot using LQG and fuzzy logic control methods in Matlab/Simulation. The control purpose is to keep the stability of the system around an equilibrium point while the robot is moving.

A linearized mathematical model is described to obtain a linearized state space model in order to develop LQG controller to control the angle and position of the robot. An optimal linear quadratic estimator has been designed to estimate the state variables of the system and has been connected to an optimal full state feedback regulator (LQR). The designed controller has been implemented on the nonlinear dynamics model of the robot.

The control algorithm is extended to an intelligent control since fuzzy logic controller is developed to control the angle of the robot. Also the designed controller is implemented on the nonlinear dynamics of the robot in Matlab/Simulink.

The simulation results obtained in Matlab/Simulink show that the proposed controllers are able to achieve the self-balancing of the robot while the wheels keep moving.

Key Words: Two-wheeled self-balancing robot, Linear quadratic regulator, Linear quadratic gaussian, Fuzzy logic control

ŞEKİLLER DİZİNİ

Sayfa No

Şekil 1. JOE.....	2
Şekil 2. Segway.....	3
Şekil 3. İki tekerlekli kendini dengeleyen Otonom Robot.....	4
Şekil 4. İki tekerlekli araç.....	5
Şekil 5. Tekerlek tahrikli mobil ters sarkaç.....	6
Şekil 6. Tekerlekli insan nakil aracı.....	7
Şekil 7. Arduino mikrodenetleyici kartını kullanan iki tekerlekli kendinden dengeleyici robot.....	8
Şekil 8. Kendinden dengeleyici iki tekerlekli scooter.....	9
Şekil 9. Dişli dc motor.....	11
Şekil 10. H köprüsü devresi.....	13
Şekil 11. Motor pozitif yönde (saat yönünde) döner.....	13
Şekil 12. DC motor negatif yöne döner (saat yönünün tersine).....	14
Şekil 13. Motoru hızlı bir şekilde kırma durumu.....	14
Şekil 14. L298n sürücü.....	15
Şekil 15. L298N IC blok diyagramı	16
Şekil 16. Döner kodlayıcının bir blok diyagramı.....	17
Şekil 17. Döner kodlayıcı.....	18
Şekil 18. Ataletsel ölçüm biriminin devresi.....	19
Şekil 19. Tekerleği ile ters sarkaç diyagramı.....	20
Şekil 20. Tekerlek serbest cisim diyagramı.....	23
Şekil 21. Ters sarkaç serbest cisim diyagramı.....	27
Şekil 22. LQG kontrolörün blok şeması.....	33
Şekil 23. Tahmincinin blok şeması.....	37
Şekil 24. Üçgen üyelik fonksiyonu.....	39
Şekil 25. Yamuk üyelik fonksiyonu.....	40
Şekil 26. Gauss üyelik fonksiyonu.....	40
Şekil 27. LQR kontrolörlü robotun lineer olmayan dinamiklerinin Simulink blok modeli.....	42

Şekil 28. LQG kontrolörlü robotun lineer olmayan dinamiklerinin Simulink blok modeli.....	42
Şekil 29. LQR ve LQG denetleyicileri kullanan robotun açısı.....	43
Şekil 30. Robotun LQR ve LQG denetleyicileri kullanarak konumu.....	44
Şekil 31. LQR ve LQG denetleyicileri kullanan robotun açısal hızı.....	44
Şekil 32. LQR ve LQG kontrolörleri kullanan robotun doğrusal hızı.....	45
Şekil 33. Referans izleme durumunda LQR ve LQG denetleyicileri kullanan robotun konumu.....	46
Şekil 34. Referans izleme durumunda LQR ve LQG denetleyicileri kullanan robotun açısı.....	46
Şekil 35. Referans izleme durumunda LQR ve LQG kontrolörleri kullanan robotun doğrusal hızı.....	47
Şekil 36. Referans izleme durumunda LQR ve LQG denetleyicileri kullanan robotun açısal hızı.....	47
Şekil 37. Bulanık mantık denetleyicisi ile robotun doğrusal olmayan dinamiklerinin simulink blok modeli.....	48
Şekil 38. Matlab'da Mamdani FIS arayüzü.....	49
Şekil 39. Bulanık mantık denetleyicisinin ilk girişi.....	49
Şekil 40. Bulanık mantık denetleyicisinin ikinci girişi.....	50
Şekil 41. Bulanık mantık denetleyicisinin çıkışı.....	50
Şekil 42. Bulanık mantık denetleyicisinin kuralları.....	51
Şekil 43. FIS Matlab arayüzünde kural inceleyici.....	52
Şekil 44. Bulanık mantık denetleyicisi'ni kullanarak robotun açısı.....	53
Şekil 45. Bulanık mantık denetleyicisi'ni kullanarak robotun konumu.....	53
Şekil 46. Bulanık mantık denetleyicisi'ni kullanarak robotun doğrusal hızı.....	54
Şekil 47. Bulanık mantık denetleyicisi kullanılarak robot şasisinin açısal hızı.....	54
Şekil 48. Bulanık mantık denetleyicisi'ni kullanarak robotun açısı.....	55
Şekil 49. Bulanık mantık denetleyicisi'ni kullanarak robotun konumu.....	55
Şekil 50. Bulanık mantık denetleyicisi'ni kullanarak robotun doğrusal hızı.....	56
Şekil 51. Bulanık mantık denetleyicisi kullanılarak robot şasisinin açısal hızı.....	56
Şekil 52. LQG ,LQR ve bulanık mantık denetleyicileri kullanan robotun açısı.....	57
Şekil 53. Robot şasisinin açısını kontrol etmek için bulanık mantık, LQR ve LQG kontrolörleri arasındaki karşılaştırma.....	58

TABLÖLAR DİZİNİ

Sayfa No

Tablo 1. PWM darbeleri yüzdesi ve motor hızı.....	12
Tablo 2. in1 ve in2 durumları.....	16
Tablo 3. Robotun fiziksel parametreleri	32
Tablo 4. Bulanık denetleyicinin kuralları.....	54



SEMBOLLER DİZİNİ

ε	Tahmin Hatası
w_d	Bozukluk Gürültüsü
w_n	Ölçüm Gürültüsü
K_r	LQR'nin Optimum Kazancı
K_f	Kalman Filtresinin Optimal Kazancı
PWM	Darbe Genişliği Modülasyonu
LQR	Doğrusal Kuadratik Regülatör
LQG	Doğrusal Kuadratik Gauss
FLC	Bulanık Mantık Kontrolü
IMU	Ataletsel Ölçüm Birimi
g	Yerçekimi (m/s^2)
r	Tekerlek Yarıçapı (m)
M_w	Tekerlek Ağırlığı (kg)
M_p	Vücut Kütlesi (kg)
I_w	Tekerleğin Ataleti ($kg \cdot m^2$)
I_p	Vücudun Ataleti ($kg \cdot m^2$)
l	Tekerleğin Merkezleri İle Robotun Ağırlık Merkezi Arasındaki Mesafe (m)
K_m	Motor Torku Sabiti (Nm/A)
K_e	Geri EMF Sabiti (Vs/rad)
R	Nominal Terminal Direnci (Ohm)
x	Tekerleğin Doğrusal Konumu
\dot{x}	Tekerleğin Doğrusal Hızı
φ	Şasinin Açısal Konumu
$\dot{\varphi}$	Şasinin Açısal Hızı
v_a	Motorlara Uygulanan Voltaj

1. GENEL BİLGİLER

1.1. Giriş

Son yirmi yılda, iki tekerlekli kendini dengeleyen robotların önemi, çeşitli uygulama alanlarından dolayı hızla artmıştır. Sistemin doğrusal olmayan ve kararsız dinamikleri nedeniyle kontrol ve robotik eğitimi amaçlı kullanılmıştır. İki tekerlekli kendini dengeleyen robotlar, robotik ve taşıma sistemlerindeki birçok alanda uygulamaları nedeniyle eğitim testlerinde ve analizlerinde tercih edilmektedir. Bunun yanında karmaşık dinamikleri olan bacaklı robotlara kıyasla daha basittir. İki tekerlekli kendini dengeleyen robotların kontrolü ters sarkacın kontrol kavramına dayanmaktadır. İki tekerlekli kendi kendini dengeleyen robotların kontrolü, iki dc motor tarafından tahrik edilen iki tekerlek üzerine monte edilmiş ters sarkaç kontrol konseptine dayanmaktadır.

Dört tekerlekli mobil robot ile karşılaştırıldığında, iki tekerlekli kendini dengeleyen robot, küçük şasi, küçük ağırlık olarak bir çok avantaja sahiptir. Bu tür robotlar arazilerde daha rahat hareket ve manevra kabiliyetini düşük maliyetle sağlayabilmektedir. Bu robotların karşılaştığı en önemli dönüm noktalarından biri, insanları bir Segway icadı yoluyla taşıyabilmesidir. Segway, kirliliğe neden olmayan çevre dostu bir ulaşım aracı olmasının yanı sıra, arabalara kıyasla düşük maliyetli ve güvenli bir ulaşım aracı olarak kabul edilmektedir, ancak düşük hızlarda hareket edilmektedir.

1.2. Literatür Taraması

1.2.1. İki Tekerlekli Kendini Dengeleyen Robot Türleri

Bu bölümde, kendini dengeleyen robotların kullanıldığı kontrol algoritmaları ile mekanik yapıların farklı türleri açıklanmıştır.

1.2.1.1. Mobil Ters Sarkaç

Kaynak [1] 'de arařtırmacılar JOE'yi Őekil 1'de gsterildiđi gibi iki tekerlekli bir mobil ters sarka olarak inřa etmiřlerdir. Bu tekerleklerin her biri ayrı motorla alıřtırıldıđından, ara birbirine ortak bir pivot ile bađlanmış iki tekerlekten yapılmıřtır. pivotun ortasına, ters bir sarka monte edilmiřtir. Mobil ters sarka geriye ve ileriye hareket edebilme, dikey eksenini etrafında dnebilme yeteneđine sahiptir. Sistemi kararlı hale getirmek ve istenen performansı elde etmek iin uygulanan kontrol yntemi nemlidir ve bu tezin konusudur. Kontrol tasarımı yapabilmek iin aracın durum modeli iki ayrı modele blnmřtir. İlk durum uzayı modeli , tekerleklerin konumunu ve sarkaın eđim aısını kontrol etmekten sorumludur. İkinci durum uzayı modeli, aracın dnmesini gerekleřtirmek iin kullanılmıřtır. lm durumları iin, sarkaın aısal aısını belirlemek iin bir jiroskop sensr kullanılırken, aracın dođrusal konumunu, hızını elde etmek iin iki artan enkoder kullanılmıřtır. Kullanıcı, mobilin dođrusal hızına ve dnme aısına kumanda etmek iin bir joystick kullanarak aracı ynlendirebilmektedir.



Őekil 1. JOE

1.2.1.2. Segway Kişisel Taşıyıcı

Şekil 2 de görülen Segway, Dean Kamen tarafından 1999 yılında insan transferi amacıyla icat edilen, iki tekerlekli, kendi kendini dengeleyen bir araçtır [2]. Segway'in hareket mekanizması sürücünün eğilmesine dayanmaktadır. Eğer sürücü ileri doğru eğilirse Segway ileri doğru hareket edecektir, sürücünün geriye doğru eğilmesi durumunda, Segway geri döner ve eğer sürücü dik durursa araç durur. Segway'in sağa veya sola dönmesini sağlamak için sürücünün gidonu sağa veya sola döndürmesi gerekmektedir. Segway'in durumlarını ölçmek için beş jiroskop ve iki ivmeölçer kullanmıştır ama sadece üç jiroskop dengeleme ve hareket için yeterlidir. Aracı daha güvenilir ve daha güvenli hale getirmek için ekstra iki gyros eklenmiştir.



Şekil 2. Segway

1.2.1.3. İki Tekerlekli Kendini Dengeleyen Otonom Robot

Otonom iki tekerlekli, kendi kendini dengeleyen bir robot, Rich Chi Ooi tarafından ters sarkaç kontrolü kavramına dayanarak inşa edilip, kontrolü yapılmıştır [3]. Ooi robot dengelenmesini sağlamak için, Doğrusal Kuadratik Regülatör ve kutup yerleştirme gibi

doğrusal kontrol yöntemlerini kullanmıştır. PID, aracın yörünge kontrolünü gerçekleştirmek için uygulanmıştır. Bu projedeki ölçüm durumları için jiroskop ve eğimölçer sensörleri arasında birleşme için sensör füzyon tekniğini kullanılmıştır, Robotun açısal açısını ve hızını tahmin etmek için bir Kalman filtresini kullanılmıştır.



Şekil 3. İki tekerlekli kendini dengeleyen otonom robot

1.2.2. Otonom Kentsel Ulaşım Sistemi İçin İki Tekerlekli Araç

Michael Baloh ve Michael Parent [4], B2 adı verilen akıllı iki tekerlekli kendi kendini dengeleyen bir karayolu aracı tasarlamışlardır. Aracın yapısı Şekil 4'te gösterilmektedir. Araç, iki koltuk ve sürücüyü istenen varış yerine taşıyacak ve diğer durumlarının gözlemlenmesini sağlayan verileri ekrana yönlendiren bir bilgisayarla donatılmıştır. B2'nin amacı, şehir içi bir ortamda talep üzerine, tam otomatik bir taksi hizmeti sunmaktır. İki tekerleği aynı eksene monte edildiğinden B2'nin mekanik yapısı segway'deki yapıya benzer. aracın direksiyon şekli farklı olduğundan, Segway'de direksiyon sistemi sürücünün eğim açısına dayanmaktadır. Aksine, B2'nin yönlendirilmesi tamamen bilgisayar tarafından kontrol edilmektedir. Bunun yanı sıra, B2 ana yollarda, kaldırım için değil, karayolu aracı olarak kabul edilmektedir.



Şekil 4. İki tekerlekli araç

1.2.3. Kontrol Metodolojisi

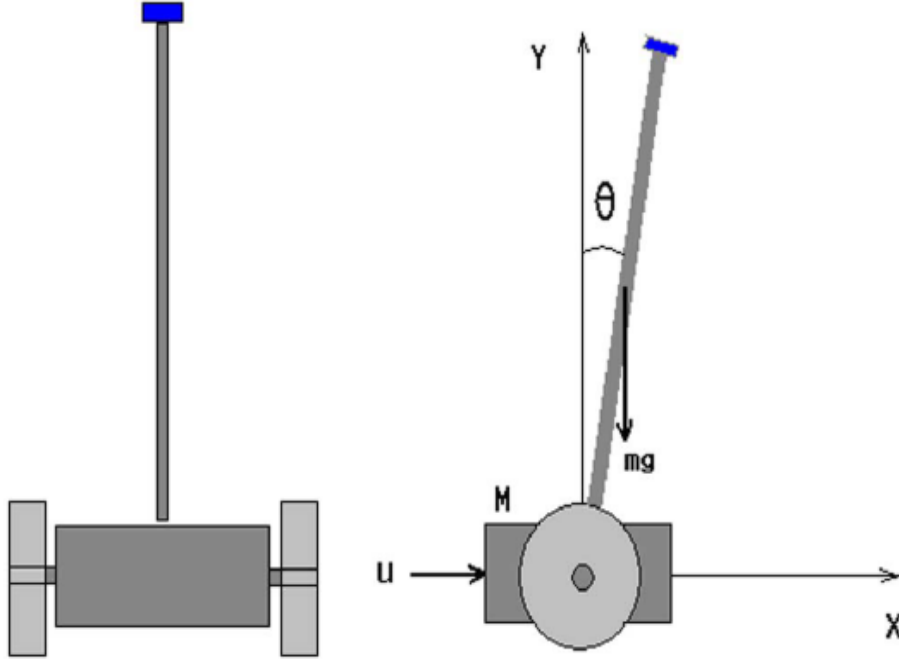
İki tekerlekli kendi kendini dengeleyen robot, doğrusal olmayan dinamiklerle içsel olarak kararsızdır. Bu tür sistemleri kontrol etmek için en çok kullanılan kontrol stratejileri doğrusal kontrol yöntemleridir. Bununla birlikte, son zamanlarda doğrusal olmayan kontrol algoritmaları, iki tekerlekli kendi kendini dengeleyen robotları kontrol etmek için de uygulanmıştır.

Bu tür robotları kontrol etmek için kullanılan kontrol algoritmalarından bazıları PID [5-10], LQR [11-13] ve yapay sinir ağı [14, 15] tabanlıdır. Bazı çalışmalarda doğrusal olmayan kayan mod kontrolü [16-20] ve adaptif kontrol [21-25] gibi tekniklerde uygulanmıştır.

Aşağıda iki tekerlekli robotlara uygulanan doğrusal ve doğrusal olmayan kontrol algoritmalarının bazıları literatürde ayrıntılı olarak gösterilmiştir.

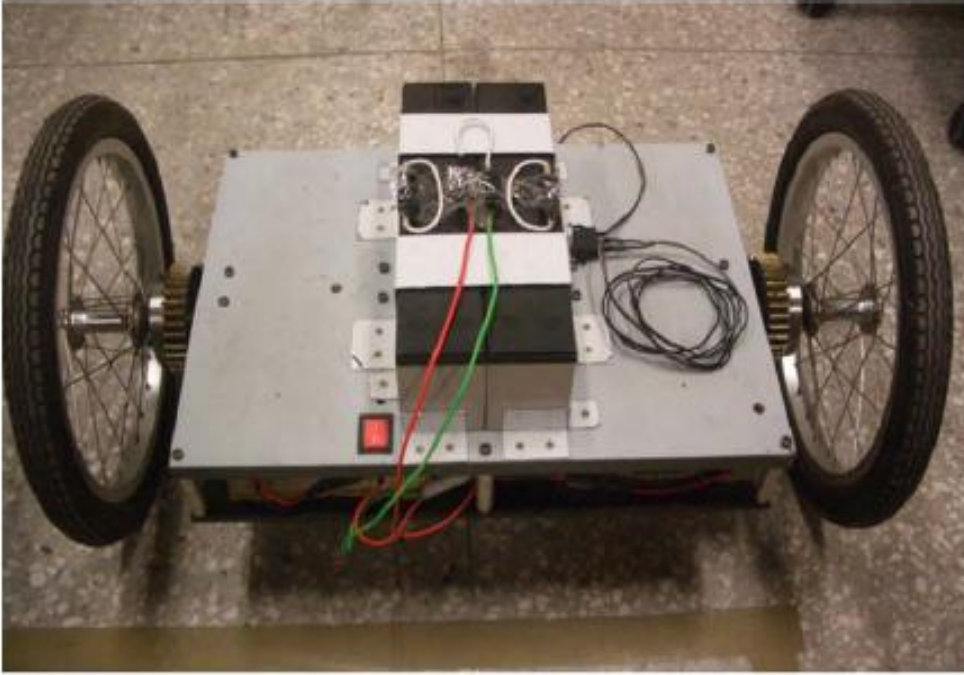
Kaynak [5] 'de S. Jung ve S. S. Kim, sinir ağı denetleyicisinin, sarkacın eğim açısını ve tekerlekli tahrik edilen ters sarkacın konumunu kontrol edebildiğini belirtmişlerdir. Ayrıca, önerilen kontrolör, aracın hareketi sırasında uygulanan bozucu dış girişlere karşı sağlamlığa sahiptir. Nöral kontrolör, yardımcı kontrolör olarak kullanılmış ve bir

yörüngeyi takip ederken robotun dengelenmesini elde etmek için birincil PID denetleyici ile birleştirilmiştir. Robotun durum ölçümlerini elde etmek için jiroskop ve enkoderler kullanılmıştır, ayrıca gecikme problemini telafi etmek için dijital filtre uygulanmıştır.



Şekil 5. Tekerlek tahrikli mobil ters sarkaç

Kaynak [6] 'da araştırmacılar, Mamdani benzeri bulanık kontrol algoritması uygulayarak iki tekerlekli bir insan taşıma aracı tasarlamışlardır. Aracın dengesi, Lyapunov kararlılık yöntemine bağlı olarak uygun koşullar sağlayarak doğrulamışlardır. Araç düşük hızlarda ileri geri gidebilmekte ve kendi etrafında dönebilmektedir. Deneysel sonuçlar, önerilen kontrol algoritmasının aracı kontrol etmek için yeterli performansa sahip olduğunu kanıtlamıştır.



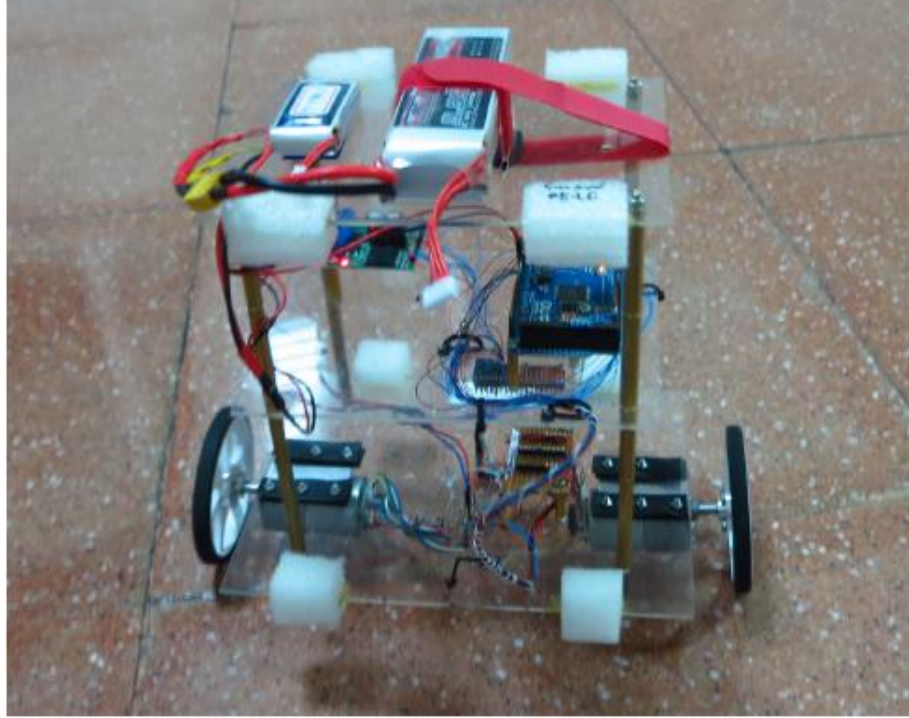
Şekil 6. Tekerlekli insan nakil aracı

Kaynak [7] 'de Chenxi Sun, Tao Lu ve Kui Yuan, Doğrusal Karesel Regülatörü (LQR) Sinir Ağı (NN) ile birleştirmeye dayanan yeni bir denge kontrol stratejisi önermişlerdir. LQR algoritması, sinir ağını başlatacak en uygun giriş değerlerini sunacaktır. Önerilen yöntemin avantajları, sistemin hata modelindeki problemi çözmesi ve diğer NN yöntemlerine ihtiyaç duyan karmaşık bir hesaplama yapmadan kendi kendine ayarlama yeteneğine sahip olmasıdır. Bununla birlikte, bu yöntemin dezavantajı, dengeleyici işlem sırasında, yakınsama hızının düşmesi küçük dengeleme süresinin diğer kontrol yöntemlerine kıyasla daha fazla artmamasına neden olmaktadır.

[8] 'de yazarlar geri besleme kontrol algoritması kullanarak kendi kendini dengeleyen insan taşıma aracı geliştirmiştir. Sistem modeli iki alt sisteme bölünmüştür, yalpa kontrolü için ilk model ve mobil ters sarkaç kontrolü için ikincisidir. PD ve faz gecikmeli kontrolörleri, aracın düşmeden dengelemesini sağlamak ve istenen yalpa açısı takibini gerçekleştirmek için araca başarıyla uygulamışlardır.

[9] 'da, Hau-Shiue Juang¹ ve Kai-Yew Lum, düşük maliyetli unsurlar kullanan iki tekerlekli, kendi kendini dengeleyen bir robot tasarlayıp üretmişlerdir. Sistem yapısı bir çift DC motor ve bir Arduino mikrodenetleyici, bir tek eksenli jiroskop ve 2 eksenli bir ivmeölçer içermektedir. Gyro taslağı sorunu, tamamlayıcı bir filtre uygulanarak

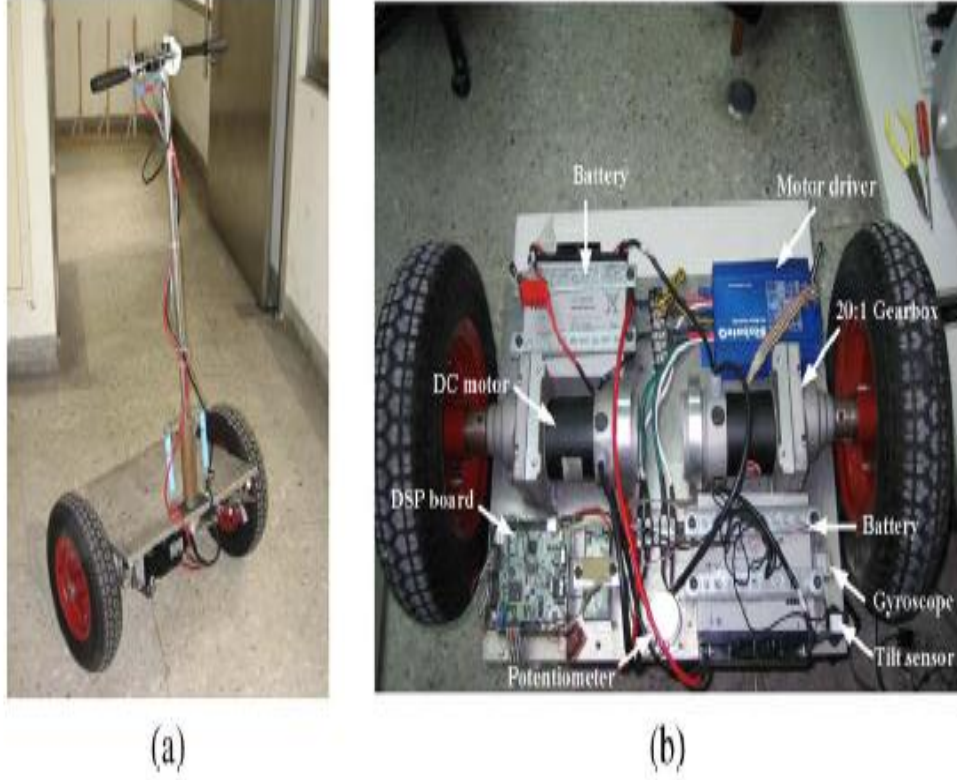
çözümüştür. Robotun dengesini kontrol etmek için PID ve LQR tabanlı PI-PD kontrol yöntemleri uygulanmıştır. Yazarlar, deneysel sonuçların, kendi kendini dengelemenin PI-PD kontrol algoritması ile gerçekleştirilebileceğini gösterdiğini belirtmişlerdir. Yazarlar, aynı kontrol girişi motor sürücü kartlarına uygulanmasına rağmen, her iki motorun aynı hızda dönmemesi sorununu çözmek için bir PI kontrolörü bir tekerlek senkronizasyon denetleyicisi gibi kullanmışlardır.



Şekil 7. Arduino mikrodeneleyici kartını kullanan iki tekerlekli kendinden dengeleyici robot

[10]'de Tsai, Ching-Chih, Huang, Hsu-Chih, Lin ve Shui-Chun, scooter'ın dengeleme ve istenen performansını elde etmek için RBFNN kullanarak adaptif bir kontrol algoritması uygulamışlardır. Sistem iki alt sisteme ayrılmıştır, bu nedenle dengeleme ve yalpalama hareketlerini gerçekleştirmek için RBFNN kullanan iki uyarlanabilir denetleyici ayarlanmıştır. Eğim açısını belirlemek için Eğim ve gyro sensörleri kullanılırken, aracın sapma açısını kontrol etmek için bir potansiyometre kullanılmıştır. İki RBFNN, tekerlekler ve yüzey arasındaki kısır ve statik sürtünmeleri bilmek ve ayrıca sürücünün ağırlık gibi belirsizliklerini ve doğrusal olmayan hareket denklemlerinin doğrusallaştırılması sırasında elde edilen modüle edilmemiş hataları bilmek

için uygulanmıştır. Uyarlanabilir kontrolörlerin asimptotik kararlılığı Lyapunov kararlılık teorisi kullanılarak sağlanmıştır.



Şekil 8. Kendinden dengeleyici iki tekerlekli scooter

1.3. Mekanik Tasarım Ve Elektro-Mekanik Elemanlar

1.3.1. Mekanik Tasarım

Bu robotun ana mekanik tasarımı, dört sütuna birbirine bağlanmış üç alüminyum katmana sahip iki tekerlek içermektedir. İlk katmanda iki DC motor sürücüsü ile iki dişli DC motorlar eklenmektedir. İkinci katman, sensörlerden gelen sinyalleri almaktan sorumlu olan mikro kontrol ünitesini içerir ve komutları içindeki kod programına dayanarak motorlara gönderir Bunun yanında robotun açı eğimini ölçmek için kullanılan atalet ölçüm birimi (IMU) içermektedir. Son katman robot ve PC arasında bir iletişim modeli olarak kullanılan bir Bluetooth modeli içerir. Ayrıca, güç kaynağını önceki tüm bileşenlere dağıtmak için bir pil içermektedir.

1.3.2. Elektro-Mekanik Bileşenler

Bu tür robotlardaki ana elektro-mekanik elemanlar:

- İki adet DC motor.
- İki Enkoder.
- H-köprüsü DC motor sürücü.
- Mikrodenetleyici ünitesi.
- Atalet ölçü birimi (IMU).
- Bluetooth modeli.
- Batarya.

1.3.3. DC Motor

Fırçalı DC motor, fırçasız motor, step motor ve dişli DC motor gibi İki tekerlekli kendi kendini dengeleyen Robotlara uygulanabilecek birçok DC motor tipi vardır.

1.3.4. Dişli Dc Motor

Şekil 9’da bir fotoğraflı verilen fırçalı bir motor tipi olan dişli DC motor, yüksek tork vererek DC motorun hızını azaltabilen dişliler içermektedir, bu nedenle düşük hızlarda yüksek tork gerektiren birçok uygulama için çok faydalıdır. Bunun yanı sıra, bu tür motorlardan bazıları, motorun dönme pozisyonunu ölçen bir kodlayıcı içermektedir.



Şekil 9. Dişli dc motor

1.3.5. Darbe Geniřlięi Modülasyonu (PWM)

PWM, DC motorun hızını ayarlamak için yaygın bir yöntemdir. Motorun hızını deęiřtirmek için darbe geniřliklerinin deęiřmesine baęlıdır. Darbe geniřlięi ne kadar büyük olursa motor o kadar hızlı döner. Motorun durması için sıfır darbeler kullanılmalıdır. Tam hızda çalışmasını sağlamak için sürekli tam voltaj kullanılmalıdır.

1.3.5.1. PWM Darbelerinin Görev Döngüsü

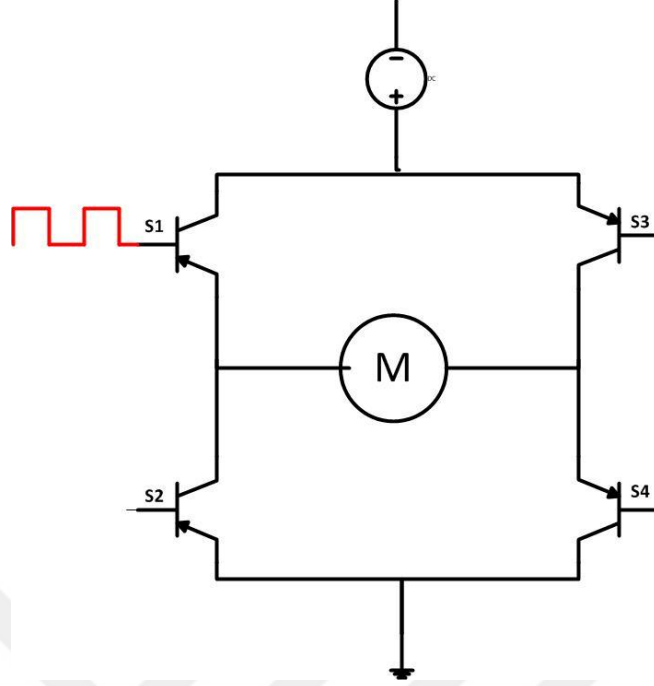
Bu tabloda PWM darbelerinin yüzdesi ile motor hız oranı arasındaki ilişki açıklanmıştır.

Tablo 1. PWM darbeleri yüzdesi ve motor hızı

PWM yüzdesi	Motor hızı ve durumu
0%	Motorı durdurma
25%	Motorı hızının çeyreğinde döner
50%	Motorı hızının yarısında çalışır
75%	Hızının dörtte üçünde motorı dönüşür
100%	Motorı tam hızda döner

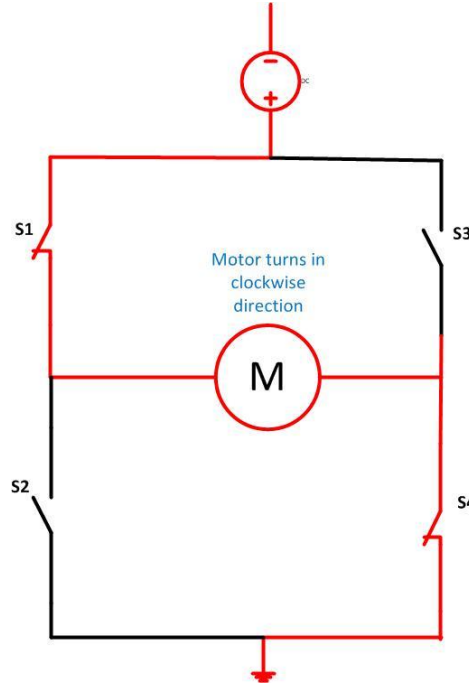
1.3.6. H-Köprü Devresi

Transistörler veya MOSFETler gibi anahtarlar kullanarak DC motorun terminallerine uygulanan gerilim deęiřtirilir. DC motorun dönüş yönünü kontrol etmek için bir H köprü devresi kullanılır. H-köprü devresi dört anahtardan (S1, S2, S3 ve S4) oluşur ve devrenin merkezinde DC motor bulunur [41].



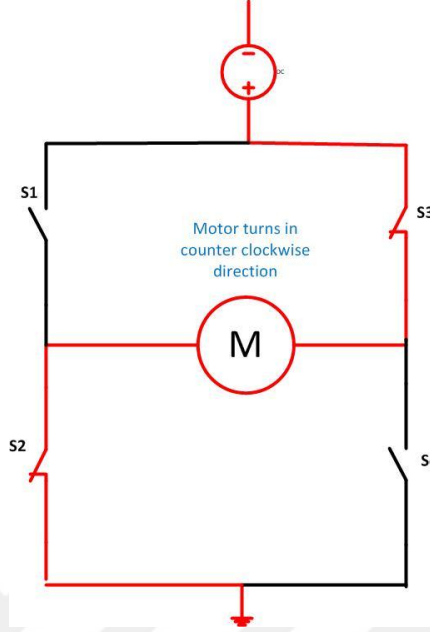
Şekil 10. H köprüsü devresi

Şekil 11'de H-köprüsünün genel konsepti gösterilmektedir. Tüm anahtarlar kapalıysa motor kapatılır, S1 ve S4 anahtarı açıksa ve S2 ve S3 anahtarı kapalıysa, motor şekil 11'da gösterildiği gibi pozitif yönde (saat yönünde) dönecektir.



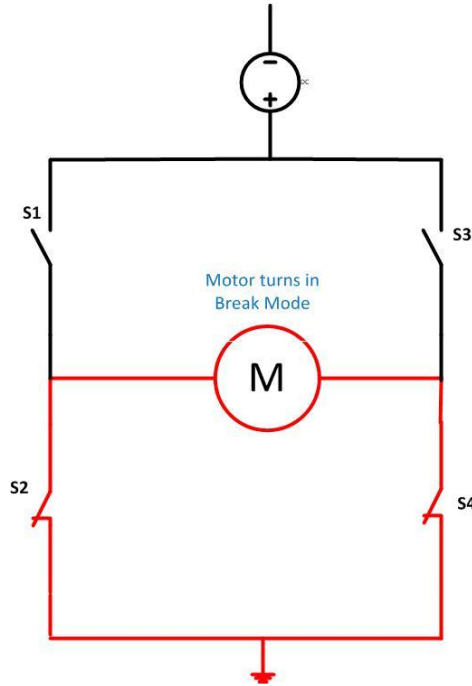
Şekil 11. Motor pozitif yönde (saat yönünde) döner

S1 ve S4 anahtarları kapalıysa ve S2 ve S3 anahtarları açıksa, DC motor şekil 12’da gösterildiği gibi negatif yönde (saat yönünün tersine) dönecektir.



Şekil 12. DC motor negatif yöne döner

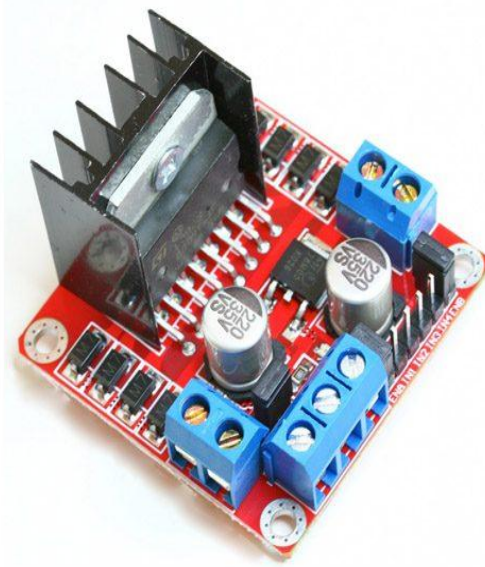
S2 ve S4 anahtarları açıksa, motorun her iki terminali de zemine bağlanacak ve motoru hızlı bir şekilde durduracaktır.



Şekil 13. DC motoru hızlı bir şekilde kırma durumu

1.3.7. L298n Sürücü

L298N sürücüsü DC motorun hızını ve dönüşünü kontrol etmek için yaygın sürücü devrelerinden biridir. Model, 2 DC motoru kontrol etmek için, 2A'ya kadar maksimum akımla 5 ila 35 V arasında voltaj beslemesi olan iki H köprüsü içermektedir.



Şekil 14. L298n sürücü

L298N modelinin en çok kullanılan pinlerinden bazıları:

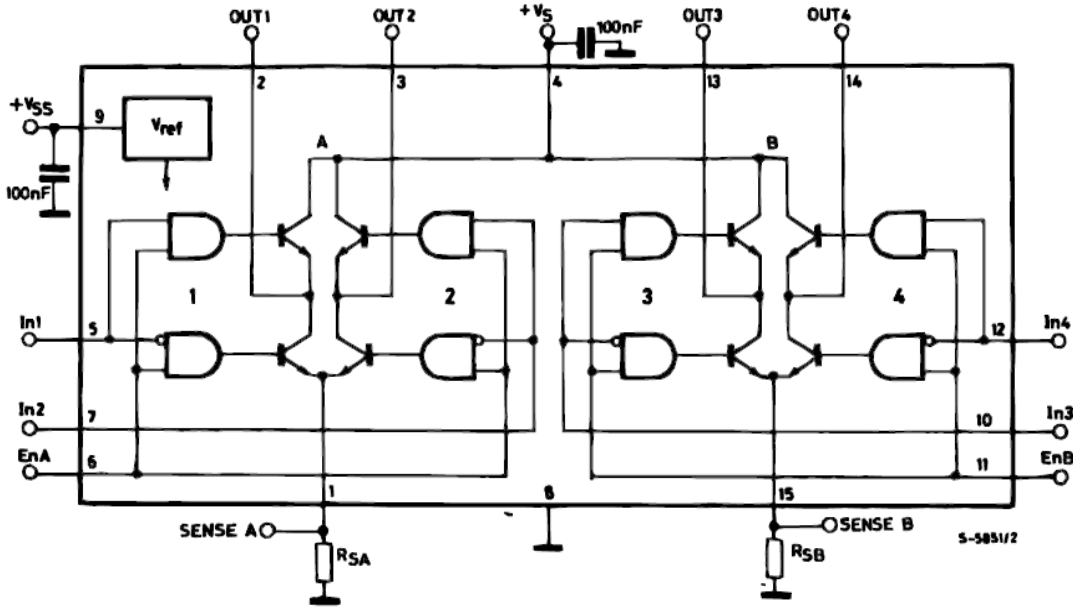
- Her DC motor A ve B motoru için iki çıkış
- Zemin pimi
- Motor beslemesi için VCC
- 5 V pimi giriş veya çıkış olarak kullanılabilir regülatörden gelen
- Motor A ve B motorunun hızını etkinleştirmek ve kontrol etmek için kullanılan En A, En B. atlamacı bağlıysa motor maksimum hızda çalışır. Jumper bağlantısı kesilirse PWM sinyal girişi bu pime bağlanabilir ve DC motorun hızını kontrol edebilir.
- A motorunun dönüş yönünü kontrol etmek için In1 ve In2 pimleri
- B motorunun dönüş yönünü kontrol etmek için In 3 ve In4 pimleri

Aşağıdaki tabloda in1 ve in2 durumları gösterilmektedir:

Tablo 2. in1 ve in2 durumları

In 1	In 2	DC motorun yönü
0 v	0 v	Motor kapalı
5 v	0 v	Motoru ileri doğru döndür
0 v	5 v	Motoru geriye doğru dönür
5 v	5 v	Kullanılmamış

In 3 ve In 4 için aynı durum. L298n IC'nin blok şeması şekil 15'da gösterilmektedir.

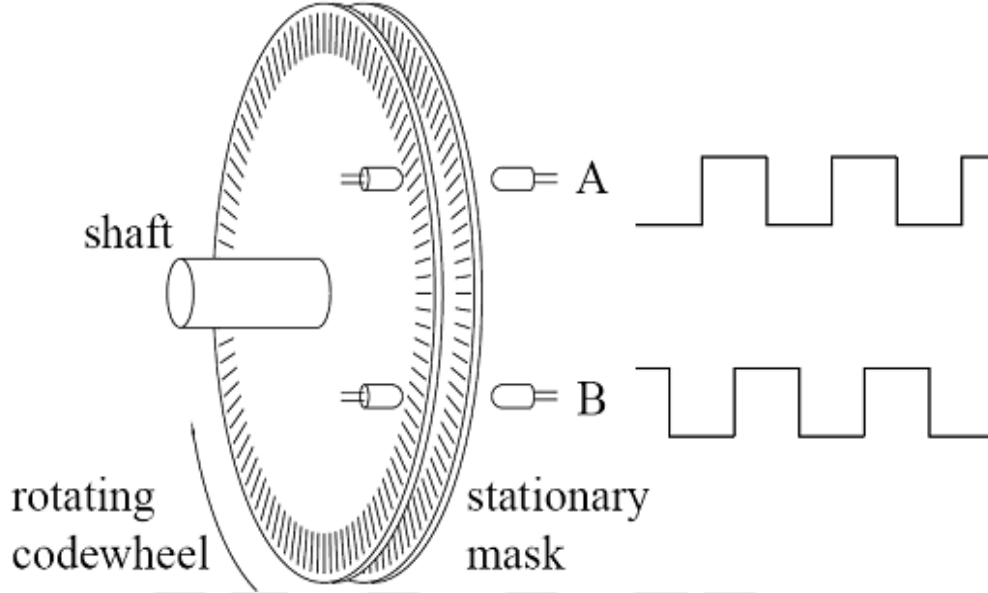


Şekil 15. L298N IC blok diyagramı [38]

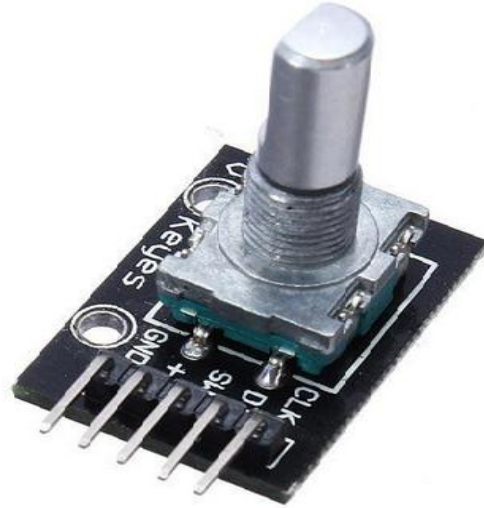
1.3.8. Artımlı Döner Kodlayıcı

İki tekerlekli kendi kendini dengeleyen robotun konumu, DC motorlu bir döner kodlayıcı uygulanarak ölçülebilir. Döner kodlayıcı, motor şaftının açılal pozisyonunu ölçen ve analog veya dijital çıkış sinyaline dönüştüren bir elektro-mekanik sensördür. Artımlı döner kodlayıcı, motor şaftı döndürüldüğünde A fazı ve B fazı darbeleri üreten iki ışık ledinden oluşur. Sayaç sadece A fazını sayarken, B fazı DC motorun dönüş yönünü

belirlemek için kullanılır. B fazı düşükse, motor saat yönünde döndüğü anlamına gelir, B fazı yüksekse DC motor saat yönünün tersine döner.



Şekil 16. Döner kodlayıcının bir blok diyagramı



Şekil 17. Döner kodlayıcı

Döner kodlayıcı aşağıdaki pimleri içerir:

GND: zemin.

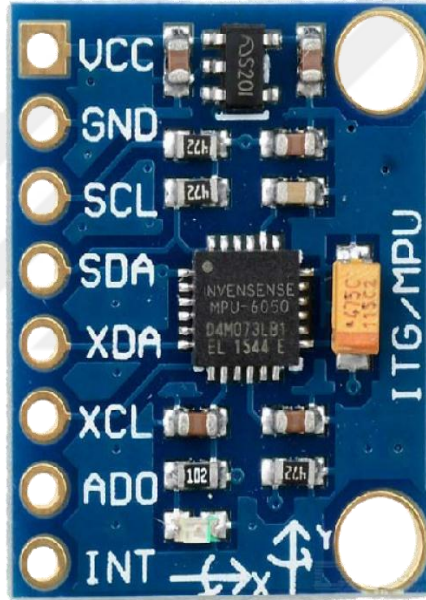
VCC: güç kaynağı voltajı.

DT: Bir faz çıkış sinyali.

CLK: B faz çıkış sinyali.

1.3.9. Ataletsel Ölçüm Birimi (IMU)

Ataletsel ölçüm birimi, insansız Aerial araçlar, dronlar, iki tekerlekli kendi kendini dengeleyen robot telefonlar ve benzeri birçok uygulamada kullanılan iki ana sensör ivmeölçer ve Jiroskoptan oluşur. ivmeölçer bir cisimin ivmesinin ölçülmesinden sorumludur, Jiroskop x, y ve z eksenini etrafındaki açısal momentu veya dönüşü ölçebilir. Ayrıca IMU ünitesi, eksen üzerindeki manyetik alanları ölçmekten sorumlu bir manyetometre gibi I2C veri yolu kullanılarak harici sensörlere bağlanabilir. Yaygın IMU cihazlarından biri, Şekil 18'te gösterildiği gibi MPU-6050'dir.



Şekil 18. Ataletsel ölçüm biriminin devresi

MPU-6050 aşağıdaki pinleri içerir:

VCC: 5 volt güç kaynağı.

GND: Zemin.

SCL: I2C saat çizgisi.

XDA: Harici sensörü bağlamak için kullanılan harici I2C veri hattı.

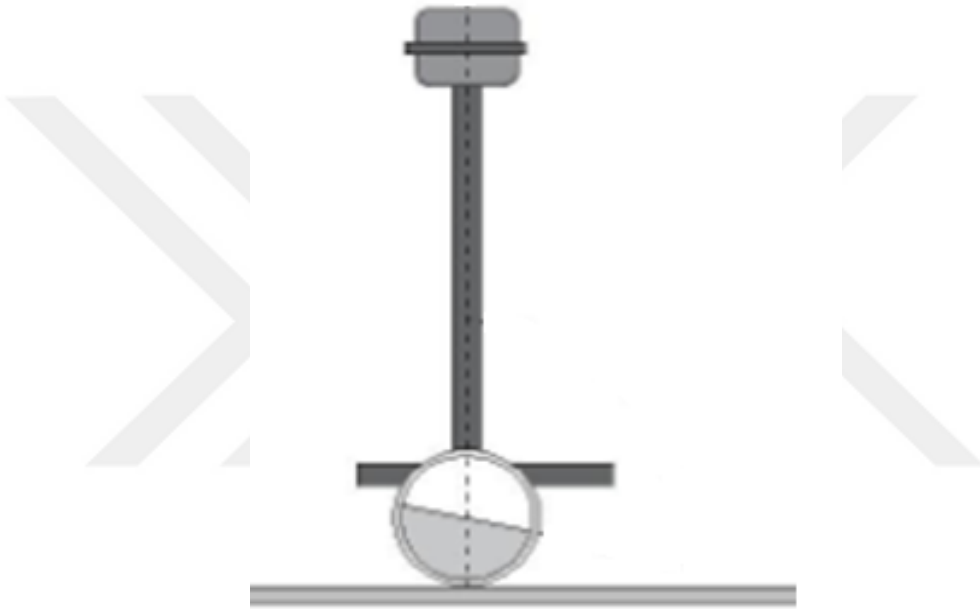
XCL: Harici I2C saat çizgisi.

AD0: Bu pim dahili I2C'nin adresini değiştirmek için kullanılır.

INT: Bu pim kesinti içindir.

1.4. Sistemin Dinamik Modeli

İki tekerlekli kendini dengeleyen sistemin dinamik modeli, esas olarak DC motor, tekerlekler ve ters sarkaç içermektedir. Her bir parça için hareket denklemleri ayrı ayrı türetilmiştir. Bundan sonra tüm parçaların durum uzay formu elde edilmiştir. Robotun performansı, motorun hareketi sırasında robota uygulanan bozucu girişlerden gelen torklardan etkilenebilir [3].



Şekil 19. Tekerlekli ters sarkaç diyagramı

1.4.1. DC Motorun Dinamik Modeli

Robotun dengesini kontrol etmek için DC motorda üretilen tork kontrol edilecek şekilde motorun dinamik modeli elde edilmelidir. Motor girişine uygulanan voltaj, motor armatüründe bir akım oluşturmaktadır. Bunun bir sonucu olarak motor akımının miktarı ile orantılı bir tork üretmektedir [3].

Tork ve akım arasındaki ilişki şöyle ifade edilebilir:

$$\tau_m = k_m i \quad (1)$$

Motorun açılma hızıyla doğru orantılı olan geri elektromotor kuvvet gerilimi üretilmiştir:

$$u_e = k_e \omega \quad (2)$$

Kirchhoff'un Gerilim Yasasını DC motoruna uygulayarak:

$$\sum U = 0 \quad (3)$$

Olması gerektiğinden,

$$U_a - Ri - L \frac{di}{dt} = 0 \quad (4)$$

Newton'un ikinci yasası döner hareketli sistemler için yazılırsa:

$$\sum T = 0 \quad (5)$$

Döner kütleyle etkiyen momentlerin toplamı:

$$\tau_m - k_f \omega - \tau_a = I_R \frac{d\omega}{dt} \quad (6)$$

(1) ve (2) denklemini (4)teki denklemin yerine koyarak

$$\frac{di}{dt} = \frac{R}{L} i + \frac{k_e}{L} u_a \quad (7)$$

Ayrıca, denklem (1) ve (2)yi ,(6)daki denklemin yerine koyarak

$$\frac{d\omega}{dt} = \frac{k_m}{I_R} i - \frac{k_e}{I_R} \omega - \frac{\tau_a}{I_R} \quad (8)$$

Son iki denklem, motorun temel hareket denklemlerini temsil eder.

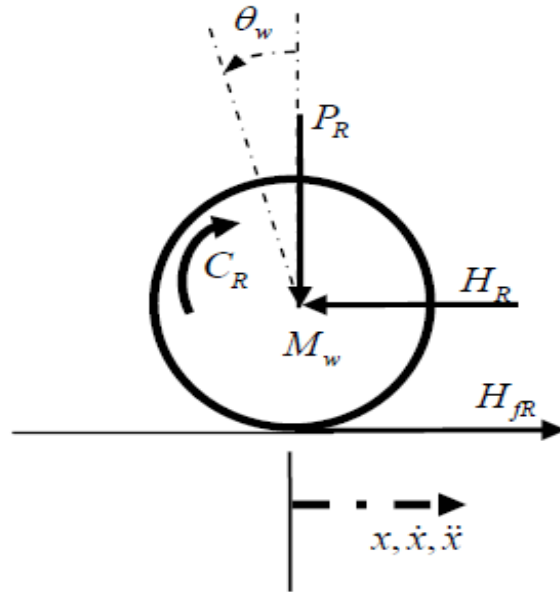
DC motorun durum uzayı modeli şu şekilde elde edilir:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-k_m k_e}{I_R R} \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_m}{I_R R} & \frac{-1}{I_R} \end{bmatrix} \begin{bmatrix} u_a \\ \tau_a \end{bmatrix} \quad (9)$$

$$y = [1 \quad 0] \begin{bmatrix} \theta \\ \omega \end{bmatrix} + [0 \quad 0] \begin{bmatrix} u_a \\ \tau_a \end{bmatrix} \quad (10)$$

1.4.2. Tekerleklerin Dinamik Modeli

Tekerlek serbestlik durumu şekil 20' de verilmiştir. Tekerleklerin hareket denklemlerinden sonra ters sarkaç denklemleri elde edilmiştir. Her iki tekerleğin hareket denklemleri benzer olduğu için bir tekerleğin dinamik modeli gösterilmektedir. Robotun matematiksel modelini elde edebilmek için Newton'un ikinci yasasını uygulamak gerekir [3]:



Şekil 20. Tekerlek Serbestlik diyagramı

$$\sum F_x = Ma \quad (11)$$

$$M_w \ddot{x} = H_{fR} - H_R \quad (12)$$

H_{fR} zemin ve tekerlekler arasındaki sürtünme kuvvetleridir, H_R tekerlekler ve şasi arasındaki reaksiyon kuvvetleridir. Tekerleğin merkezi etrafına uygulanan toplam momentler:

$$\sum M_o = I\alpha \quad (13)$$

$$I_w \ddot{\theta}_w = C_R - H_{fR}r \quad (14)$$

DC Motorun hareket denklemlerinden motordan üretilen momen:

$$T_m = I \frac{dw}{dt} - T_a \quad (15)$$

Bu nedenle DC motordan tekerleklere uygulanan çıkış momenti:

$$C = I_R \frac{dw}{dt} = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m}{R} V_a \quad (16)$$

(16) denklemini alıp (14) denklemindeki yerine yerleştirerek

$$I_w \ddot{\theta}_w = \frac{-k_m k_e}{R} \dot{\theta}_w + \frac{k_m}{R} V_a - H_{fR}r \quad (17)$$

Sonuç olarak, zemin ile tekerlekler arasındaki sürtünme kuvvetleri:

$$H_{fR} = \frac{-k_m k_e}{Rr} \theta_w \dot{\cdot} + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w \quad (18)$$

Sol ve sağ tekerlekler için hareket denklemini elde etmek için son denklemin (12)deki yerine konması gerekmektedir.

Sol tekerleğin denklemi:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \theta_w \dot{\cdot} + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w - H_L \quad (19)$$

Sağ tekerleğin denklemi ise:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr} \theta_w \dot{\cdot} + \frac{k_m}{Rr} V_a - \frac{I_w}{r} \ddot{\theta}_w - H_R \quad (20)$$

Açısal dönme, aşağıdaki denklemler kullanılarak doğrusal dönmeye dönüştürülebilmektedir:

$$\ddot{\theta}_w = \frac{\ddot{x}}{r} \quad \rightarrow \quad \ddot{\theta}_w r = \ddot{x} \quad (21)$$

Doğrusal hız için ise:

$$\dot{\theta}_w = \frac{\dot{x}}{r} \quad \rightarrow \quad \dot{\theta}_w r = \dot{x} \quad (22)$$

Önceki doğrusal dönüşümleri kullanarak sağ ve sol tekerlek denklemleri elde edilir:

Sol tekerleğin denklemi:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m}{Rr} V_a - \frac{I_w}{r^2} \ddot{x}_w - H_L \quad (23)$$

Sağ tekerleğin denklemi ise:

$$M_w \ddot{x} = \frac{-k_m k_e}{Rr^2} \dot{x} + \frac{k_m}{Rr} V_a - \frac{I_w}{r^2} \ddot{x} - H_R \quad (24)$$

Tekerleklerin dinamik denklemini tanımlayan toplam denklem:

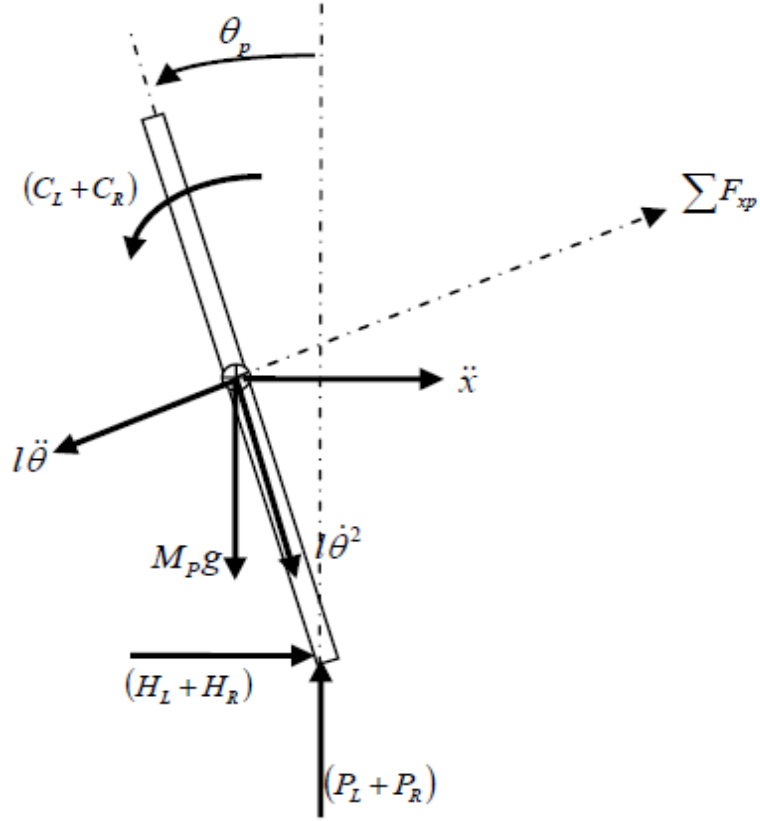
$$2(M_w - \frac{I_w}{r^2}) \ddot{x} = \frac{-2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{Rr} V_a - (H_R + H_L) \quad (25)$$

1.4.3. Ters Sarkacın Dinamik Modeli

Şekil 21' de gösterildiği gibi robotun şasisinin dinamik modeli ters sarkacın dinamik modeli ile temsil edilebilmektedir [3]. Tekerleğin dinamik modelinde yapılanlar gibi newton yasası uygulanmıştır böylece yatay yönde uygulanan kuvvetlerin toplamı elde edilmiştir.

$$\sum F_x = M_p \ddot{x} \quad (26)$$

$$M_p \ddot{x} = (H_L + H_R) - M_p l \ddot{\theta}_p \cos \theta_p + M_p l \dot{\theta}_p^2 \sin \theta_p \quad (27)$$



Şekil 21. Ters sarkacın serbestlik diyagramı

Tekerlekler ile şasi arasındaki reaksiyon kuvvetleri ayrıldığından son denklemi yeniden düzenleyerek:

$$(H_L + H_R) = M_p \ddot{x} + M_p l \ddot{\theta}_p \cos \theta_p - M_p l \dot{\theta}_p^2 \sin \theta_p \quad (28)$$

Şasiye uygulanan toplam dikey kuvvetler:

$$\sum F_{xp} = M_p \ddot{x}_p \cos \theta_p \quad (29)$$

Ve böylece denklem (30) elde edilir:

$$(H_L + H_R) \cos \theta_p + (P_L + P_R) \sin \theta_p - M_p g \sin \theta_p - M_p l \ddot{\theta}_p = M_p \dot{x}_p \cos \theta_p \quad (30)$$

Şasi merkezinde toplam moment:

$$\sum M_o = I\alpha \quad (31)$$

$$-(H_L + H_R)l \cos \theta_p - (P_L + P_R)l \sin \theta_p - (C_L + C_R) = I_p \ddot{\theta}_p \quad (32)$$

Şasiye DC motordan uygulanan çıkış momenti:

$$C = \frac{-k_m k_e}{R} \dot{\theta}_p + \frac{k_m}{R} V_a \quad (33)$$

Doğrusal dönüşümü kullandıktan sonra denklem (34) deki gibi olur:

$$(C_L + C_R) = \frac{-2k_m k_e}{R} \frac{\dot{x}}{r} + \frac{2k_m}{R} V_a \quad (34)$$

Önceki denklem, (28) denklemindeki yerine yerleştirerek:

$$-(H_L + H_R)l \cos \theta_p - (P_L + P_R)l \sin \theta_p - \left(\frac{-2k_m k_e}{R} \frac{\dot{x}}{r} + \frac{2k_m}{R} V_a \right) = I_p \ddot{\theta}_p \quad (35)$$

Denklem (35)i (30)daki denklemin yerine yerleştirerek:

$$I_p \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{R} + \frac{2k_m}{R} V_a + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p = -M_p \dot{x}_p l \cos \theta_p \quad (36)$$

$$-(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p = I_p \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{R} + \frac{2k_m}{R} V_a \quad (37)$$

$-l$ ile denklem (30)'un çarpımı:

$$\begin{aligned} & -(H_L + H_R) l \cos \theta_p - (P_L + P_R) l \sin \theta_p + M_p g l \sin \theta_p + M_p l^2 \ddot{\theta}_p \\ & = -M_p \dot{x}_p l \cos \theta_p \end{aligned} \quad (38)$$

Ve denklem (28)i (25)deki denklemin yerine yerleştirerek:

$$\begin{aligned} 2(M_w - \frac{I_w}{r^2}) \ddot{x} &= \frac{-2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{Rr} V_a - M_p \ddot{x} - M_p l \ddot{\theta}_p \cos \theta_p \\ &+ M_p l \dot{\theta}_p^2 \sin \theta_p \end{aligned} \quad (39)$$

Doğrusal olmayan hareket denklemleri, son iki denklemin yeniden düzenlenmesi ile elde edilmiştir.

$$(I_p + M_p l^2) \ddot{\theta}_p - \frac{2k_m k_e \dot{x}}{Rr} + \frac{2k_m}{R} + M_p g l \sin \theta_p = -M_p l \ddot{x} \cos \theta_p \quad (40)$$

$$\frac{2k_m}{Rr} v_a = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e}{Rr^2} \dot{x} + M_p l \ddot{\theta} \cos \theta_p - M_p l \dot{\theta}_p^2 \sin \theta_p \quad (41)$$

Önceki doğrusal olmayan denklemleri doğrusal olana dönüştürmek için aşağıda varsayımlar kullanılmalıdır.

$$\theta_p = \pi + \varphi$$

Ve,

$$\cos \theta_p = -1, \quad \sin \theta_p = -\varphi, \quad \dot{\theta}_p^2 = 0$$

φ dikey konumdan küçük açıdır. Sonuç olarak doğrusallaştırılmış denklemler:

$$(I_p + M_p l^2) \ddot{\varphi} - \frac{2k_m k_e}{Rr} \dot{x} + \frac{2k_m}{R} v_a - M_p g l \varphi = M_p l \dot{x} \quad (42)$$

$$\frac{2k_m}{Rr} v_a = \left(2M_w + \frac{2I_w}{r^2} + M_p \right) \ddot{x} + \frac{2k_m k_e}{Rr^2} \dot{x} - M_p l \ddot{\varphi} \quad (43)$$

Doğrusallaştırılmış durum-uzay modeli, denetleyiciyi geliştirmek için kullanılmış olan yukarıdaki doğrusallaştırılmış denklemlerden elde edilebilir.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\varphi} \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & A_{22} & A_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & A_{42} & A_{43} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{21} \\ 0 \\ B_{41} \end{bmatrix} v_a \quad (45)$$

x, \dot{x} : Tekerleğin doğrusal konumu ve hızı.

$\varphi, \dot{\varphi}$: Şasinin açısal konumu ve hızı.

v_a : Motorlara uygulanan voltaj.

$$A_{22} = \frac{2k_m k_e (M_p l r - I_p - M_p l^2)}{Rr^2 \alpha} \quad (46)$$

$$A_{23} = \frac{M_p^2 g l^2}{\alpha} \quad (47)$$

$$A_{42} = \frac{2k_m k_e (R\beta - M_p l)}{Rr^2 \alpha} \quad (48)$$

$$A_{43} = \frac{M_p g l \beta}{\alpha} \quad (49)$$

$$B_{21} = \frac{2k_m(I_p + M_p l^2 - M_p l r)}{R r \alpha} \quad (50)$$

$$B_{41} = \frac{2k_m(M_p l - r \beta)}{R r \alpha} \quad (51)$$

İse

$$\alpha = I_p \beta + 2M_p l^2 \left(M_w + \frac{I_w}{r^2} \right) \quad (52)$$

$$\beta = 2M_w + \frac{2I_w}{r^2} + M_p \quad (53)$$

Dinamik modelde kullanılan parametrelerin değerleri aşağıdaki tabloda gösterilmiştir:

Tablo 3. Robotun fiziksel parametreleri [3]

Parametreler	Değerler	Tanım
g	9.81	Yerçekimi (m/s ²)
r	0.051	Tekerlek yarıçapı (m)
M _w	0.03	tekerlek ağırlığı (kg)
M _p	1.13	Vücut kütlesi (kg)
I _w	0.000039	Tekerleğin ataleti (kg*m ²)
I _p	0.0041	Vücutun ataleti (kg*m ²)
l	0.07	Tekerleğin merkezleri ile robotun ağırlık merkezi arasındaki mesafe (m)
K _m	0.006123	Motorun tork sabiti (Nm/A)
K _e	0.006087	Geri EMF sabiti (Vs/rad)
R	3	Nominal Terminal Direnci (Ohm)

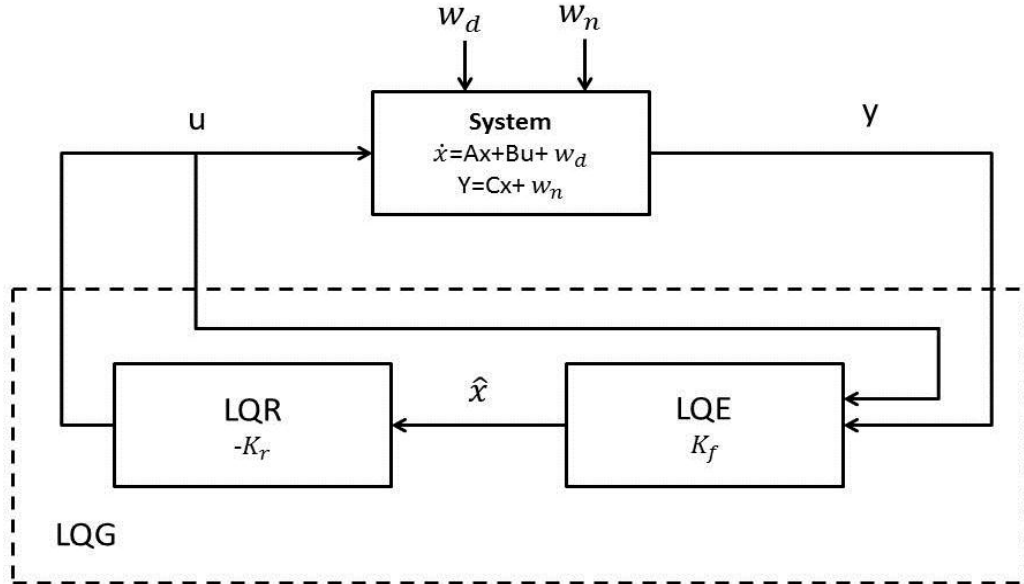
1.5. Kontrol Sistemi Tasarımı

1.5.1. Doğrusal Kuadratik Gauss Denetleyici Tasarımı

LQG kontrol ünitesi birbirine bağlı iki parçadan oluşur. İlk kısım, tahmin edici ve filtre olarak çalışan Kalman filtresidir, İkinci kısım ise LQR denetleyicisidir. Ancak LQR kontrolörü hedefleri gerçekleştirebilir sistemin tüm durumlarını ölçebilme özelliğine sahiptir, LQG'de sadece robotun pozisyonu ölçülmelidir. LQG kontrolörün kapalı döngüsü aşağıdaki şekilde verilebilir [32,33]:

$$\begin{bmatrix} \dot{x} \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} A - BK_r & BK_r \\ 0 & A - K_f C \end{bmatrix} \begin{bmatrix} x \\ \varepsilon \end{bmatrix} + \begin{bmatrix} I & 0 \\ I & -K_f \end{bmatrix} \begin{bmatrix} w_d \\ w_n \end{bmatrix} \quad (54)$$

$\varepsilon = x - \hat{x}$ tahmin hatası, K_r LQR'nin optimum kazancı ve K_f Kalman filtresinin optimal kazancıdır. , w_d bozukluk gürültüsü ve w_n ölçüm gürültüsü.



Şekil 22. LQG kontrolörün blok şeması

1.5.1.1. Doğrusal Kuadratik Regülatör Kontrol Yöntemi

LQR, durum uzayındaki kontrol sistemleri tasarım yöntemlerinden biridir. Kutup yerleştirme yaklaşımı ile karşılaştırıldığında, bu yöntem bize sistemin en iyi kutuplarını bulmak için bir yaklaşım sunmaktadır. Ayrıca, bu yöntem yüksek tepkiyi ve aktüatör tüketilen enerjiyi vererek kutuplar performansı arasındaki dengeyi tutmak mümkündür. Dengelemesiz sorumlu kuadratik maliyet fonksiyonudur [34,39]:

$$J = \int_0^{\infty} x^T Q x + u^T R u \quad (55)$$

Optimal geri besleme kontrol yasası:

$$u = -K_r x \quad (56)$$

K_r 'nin optimal geri besleme kazanç matrisi şu şekilde elde edilmektedir:

$$K_r = R^{-1} B^T P \quad (57)$$

P gerçek simetrik bir matristir. Riccati denkleminin çözümü:

$$P A + A^T P - P B R^{-1} B^T P + Q = 0 \quad (58)$$

LQR kontrolörü Matlab'da tasarlamak için, öncelikle kontrol edilebilir olup olmadığını kontrol etmek için sistemi incelemeliyiz. Sistem kontrol edilebilir ise kutuplar sistemin kararlı olduğu yere yerleştirilmiştir. Ardından, LQR kontrol cihazı, Matlab'da aşağıdaki fonksiyon kullanarak en uygun durum geri besleme kazanç matrisini belirleyerek tasarlanmıştır.

$$K_r = lqr(A, B, Q, R)$$

Simülasyonda Q ve R değerleri denemeyle seçilmiştir.

1.5.1.2. Kalman Filtre Tasarımı

Tahmin ediciyi kullanmanın amacı, x durum değişkeninin tam değerinin her zaman ölçülememesidir, çünkü bazı durumlarda ölçüm kabiliyeti sınırlıdır. Tahmin edici, sistemin tam durum değişkenler (\hat{x}) tahmin etmeye çalışmaktadır. Ayrıca, ölçümler ve sistemin modelini etkileyen gürültüleri ve bozuklukları filtreler. Tahmin edicinin girişleri kontrol sinyali u ve gürültülü ölçüm y iken, tahmin edicinin çıktısı, sistemi stabilize etmek için kontrol ünitesine döndürülecek olan tahmin edilen durum vektörü \hat{x} dir. Kontrol ünitesinin çıktısı şöyle olur [38,40]:

$$u = -K_r \hat{x} \quad (59)$$

Bozucu etkiyi ve gürültü ölçümlerini içeren sistemin genel durum uzay formu:

$$\begin{aligned} \dot{x} &= Ax + Bu + W_d \\ y &= Cx + Du + W_n \end{aligned} \quad (60)$$

Tam durumlu tahmin edicinin durum uzay formu şöyledir:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + k_f(y - \hat{y}) \\ \hat{y} &= C\hat{x} \end{aligned} \quad (61)$$

k_f filtrenin kazancıdır.

$$\hat{\dot{x}} = A\hat{x} + Bu + k_f y - k_f c\hat{x} \quad (62)$$

Yukarıdaki denklemi yeniden düzenleyerek durum tahmini:

$$\hat{\dot{x}} = (A - k_f c)\hat{x} + [B \quad k_f] \begin{bmatrix} u \\ y \end{bmatrix} \quad (63)$$

Tahmin hatası:

$$\varepsilon = x - \hat{x} \quad (64)$$

Yukarıdaki denklemin türevini alarak:

$$\dot{\varepsilon} = \dot{x} - \dot{\hat{x}} \quad (65)$$

(60) ve (61) denklemleri yukarıdaki denklemde yerine koyarak:

$$\dot{\varepsilon} = Ax + Bu - A\hat{x} + k_f c\hat{x} - k_f y - Bu \quad (66)$$

Sonra yukarıdaki denklem yeniden düzenlenmiştir:

$$\dot{\varepsilon} = A(x - \hat{x}) + k_f c(\hat{x} - x) \quad (67)$$

Ardından, işaretin denkleminin ikinci kısmı çevrilmiştir:

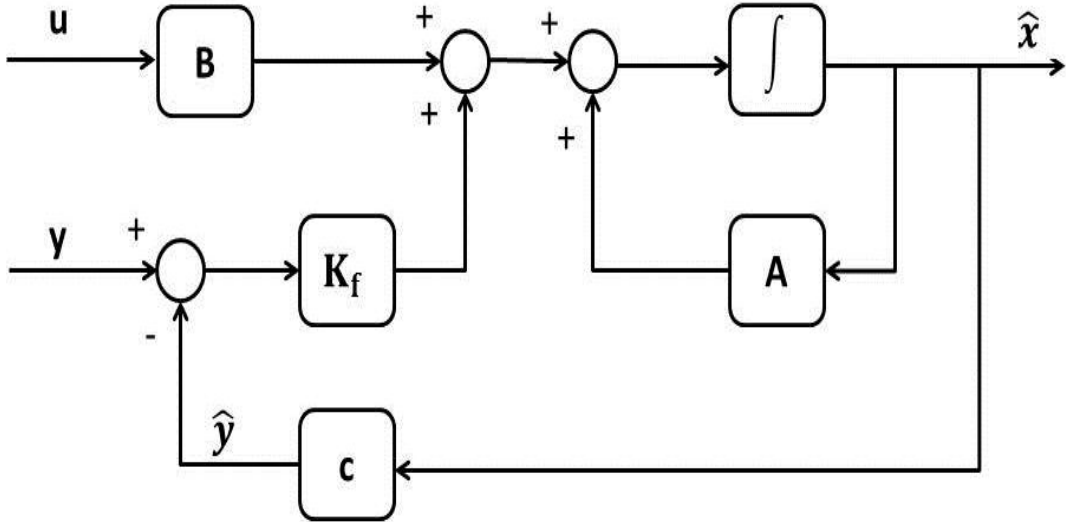
$$\dot{\varepsilon} = A(x - \hat{x}) - k_f c(x - \hat{x}) \quad (68)$$

Yukarıdaki denklemin yeniden düzenlenmesi şunları verir:

$$\dot{\varepsilon} = (A - k_f c) (\varepsilon) \quad (69)$$

Hata tahmininin türevi şu şekilde olur:

$$\dot{\varepsilon} = (A - k_f c) \varepsilon \quad (70)$$



Şekil 23. Tahmincinin blok şeması

Şekil 23'de gösterildiği gibi, \hat{x} vektörünün değerleri, $e = y - \hat{y}$ error hatasını tahmin edicinin k_f kazançları ile çarparak hata sıfırlanmaya kadar değiştirilir, bu da $x = \hat{x}$ anlamına gelir. Matlab'da Kalman filtresini tasarlamadan önce, sistemin gözlenebilirliği de incelenmelidir.

1.5.2. Bulanık Mantık Kontrolü

Bulanık mantık denetleyicisi, insan operatörünün bilgisini temsil etmek için bulanık küme kavramları kullanan doğrusal olmayan bir denetleyicidir. Bulanık mantık denetleyicisi, sistemin matematiksel modeline ihtiyaç duymadan karmaşık doğrusal olmayan sistemleri kontrol etmek için istenen kontrol eylemlerini üretebilmektedir. Bulanık mantık denetleyicisi, bulanıklaştırma, kural tabanı, bulanık çıkarım ve durulaştırma gibi istenen denetim eylemini oluşturmak için çeşitli işlemler içermektedir. [35,36].

1.5.2.1. Bulanık Kümeler

Bulanık küme, üyelik işlevinin 0 ile 1 arasında değerlere sahip olduğu bir kümedir. A'nın x ekseninde bulanık bir küme olduğu varsayılırsa A şöyle tanımlanabilir:

$$A = \{ (x, \mu_A(x)), x \in X \} \quad (71)$$

$\mu_A(x)$, A bulanık kümesindeki x ögesinin üyelik değeri olarak adlandırılır. Bulanık kümeler sürekli formda veya ayrık formda temsil edilebilir:

$$A = \sum_{x_i \in X} \frac{\mu_A(x_i)}{x_i} \quad (\text{Discrete form}) \quad (72)$$

$$A = \int \frac{\mu_A(x)}{x} \quad (\text{continuos form}) \quad (73)$$

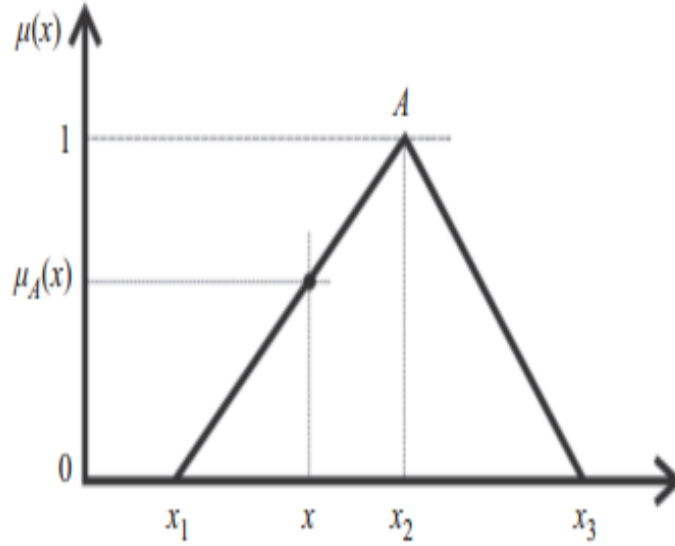
1.5.2.2. Üyelik Fonksiyonları

Üyelik fonksiyonları üçgen, trapez, Gauss ve benzeri şekillerde gösterilebilir[35].

1.5.2.2.1. Üçgen Üyelik Fonksiyonu

Üçgen üyelik işlevi aşağıdaki matematiksel formül olarak tanımlanabilir:

$$\mu_A(x) = \begin{cases} \frac{|x| - |x_1|}{|x_2| - |x_1|} & , \text{For } x_1 \leq x \leq x_2 \\ \frac{|x_3| - |x|}{|x_3| - |x_2|} & , \text{For } x_2 \leq x \leq x_3 \\ 0 & , \text{otherwise} \end{cases} \quad (74)$$



Şekil 24. Üçgen üyelik fonksiyonu

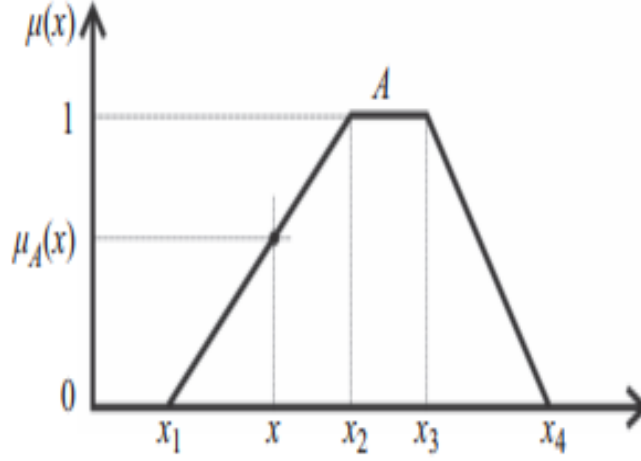
x_1, x_2, x_3 , Şekil 24'te gösterildiği gibi x eksenindeki A bulanık üyelik işlevinin konumunu temsil etmektedir.

1.5.2.2.2. Yamuk Üyelik Fonksiyonu

Yamuk üyelik işlevi aşağıdaki matematiksel formül olarak tanımlanabilir:

$$\mu_A(x) = \begin{cases} \frac{|x| - |x_1|}{|x_2| - |x_1|} & , \text{IF } x_1 \leq x \leq x_2 \\ 1 & \text{IF } x_1 \leq x \leq x_2 \\ \frac{|x_4| - |x|}{|x_4| - |x_3|} & , \text{IF } x_3 \leq x \leq x_4 \\ 0 & , \text{otherwise} \end{cases} \quad (75)$$

x_1 , x_2 , x_3 ve x_4 , Şekil 25'te gösterildiği gibi x eksenindeki A bulanık üyelik işlevinin konumunu temsil etmektedir.

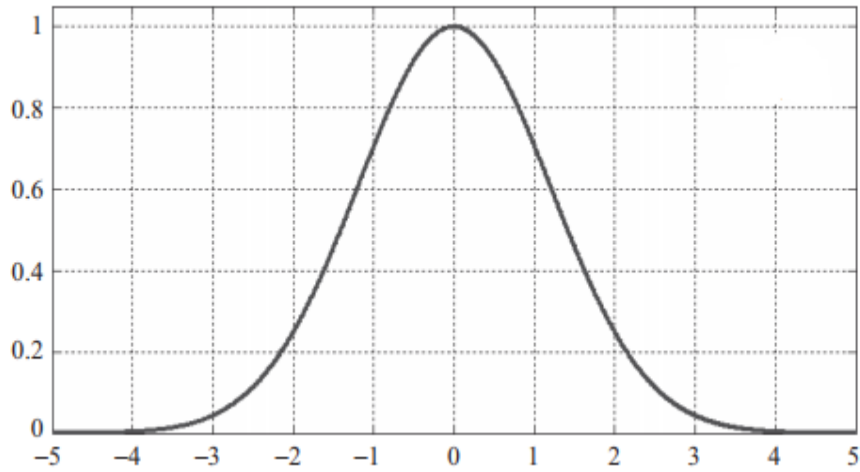


Şekil 25. Yamuk üyelik fonksiyonu

1.5.2.2.3. Gauss Üyelik Fonksiyonu

Gauss üyelik fonksiyonu aşağıdaki matematiksel formül olarak tanımlanabilir:

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-x_p}{w}\right)^2} \quad (76)$$



Şekil 26. Gauss üyelik fonksiyonu

x , X uzayında tanımlı kesin değişkendir. x_p , gaussian üyelik fonksiyonunda maximum üyeliğe sahip kesin sayıdır. W ise bant genişliğidir.

1.5.2.3. Bulanık İşlemci

Mamdani bulanık mantık sonuçlandırma algoritması dört aşamadan oluşur: Bulanıklaştırma, kural tabanı, bulanık çıkarım ve durulaştırma, bu aşamalar aşağıdaki açıklanmıştır:

1.5.2.3.1. Bulanıklaştırma

Bulanık işlemedeki ilk adım, kesin girdilerin sayısal değerlerini üyelik değerlerine dönüştürmektir. Sıfır olmayan her üyelik değeri kurallarını kural tabanı aşamasında etkinleştirmektedir.

1.5.2.3.2. Kural Tabanı

Bulanık işlemede en önemli aşama sistemin doğru kararı vermesine yardımcı olan doğru kuralları seçmektir. Bu kurallar bulanık denetleyiciye uygulanır ve bir uzmanın düşüncelerine ve deneyimlerine bağlıdır. Bu kuralların inşası, şu şekilde ifade edilen IF-Then yapısına dayanmaktadır:

IF x is A and y is B Then z is C

A , B , C dilsel değerlerdir.

1.5.2.3.3. Bulanık Çıkarım

Bu aşamada, kural tabanı aşamasından elde edilen bulanık üyelikler bir bulanık kümede toplanır. Başka bir deyişle, bu aşama her kuralın etkinliğini değerlendirir. Her kuralın etkinliği aşağıdaki ilişki ile belirlenmektedir:

$$\mu_{Rj}(P) = \min[w_{Rj}, \mu_{CP}(P)] \quad (77)$$

$\mu_{Rj}(P)$, jth kuralının üyelik fonksiyonudur. Bundan sonra, tüm aktif kurallardan sonuç üyelik işlevlerinin birliği uygulanır:

$$\mu_{RT}(P) = \bigcup_{j=1, M} \mu_{Rj}(P) \quad (78)$$

M kullanılan tüm kuralların sayısı.

1.5.2.3.4. Durulaştırma

Bulanık denetleyicideki son aşama Durulaştırma'dır. Bu aşamada, bulanık sonuç aşamasından bulanık çıktı kesin değere dönüştürülmektedir. Durulaştırmada kullanılan ünlü yöntemlerden biri yerçekiminin merkezidir. Bu yöntemi temsil eden formül:

$$Z_{COG} = \frac{\sum_{j=1}^M w_{Rj} Z_j}{\sum_{j=1}^M w_{Rj}} \quad (79)$$

M, aktif kuralların sayısıdır. w_{Rj} , i'inci kurala ait ağırlık faktörüdür, Z_j kesin değerdir.

2. YAPILAN ÇALIŞMALAR , BULGULAR VE İRDELEME

2.1. Giriş

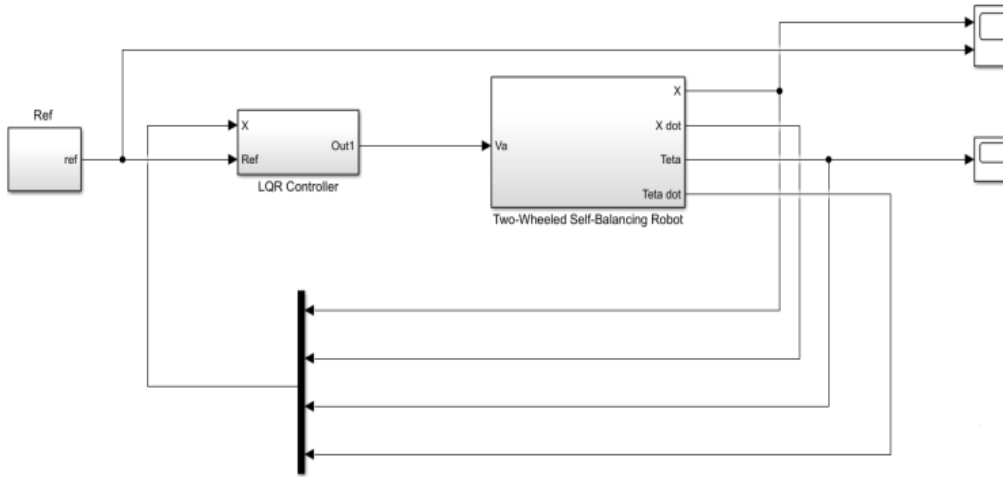
Bu bölümde Matlab / Simulink tarafından elde edilen sonuçlar gösterilmektedir. İlk olarak, robotun açısını ve konumunu kontrol etmek için kullanılan LQG ve LQR sonuçları açıklanmaktadır. Bundan sonra, robotun dengesini kontrol etmek için bulanık mantık denetleyicisi uygulanmaktadır.

2.2. Benzetim Çalışması

2.2.1. LQG Denetleyici

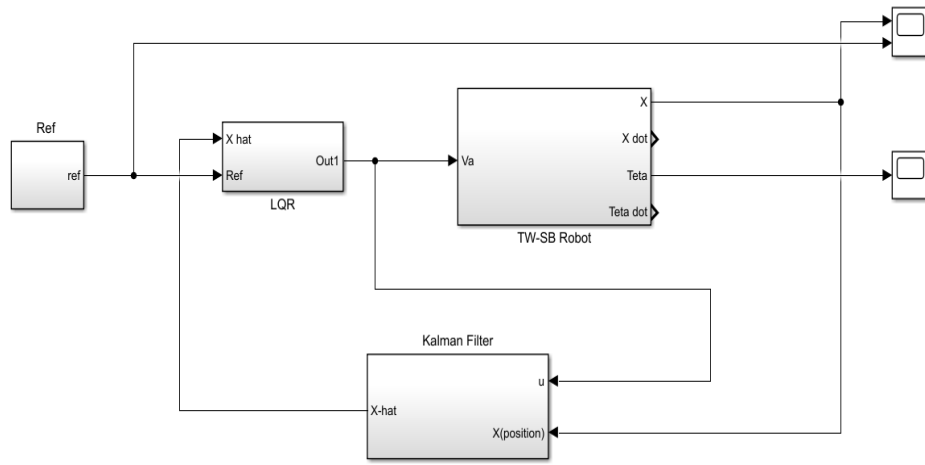
LQR ve LQG kontrolörleri olan lineer olmayan İki tekerlekli Kendi Kendini dengeleyen robotun Simulink modeli, sırasıyla Şekil 27 ve Şekil 28'te gösterilmiştir. Önerilen kontrolörlerin performansı iki durumda incelenmiştir:

- Dik kendini dengeleyen durum.
- Referans izleme durumu.



Şekil 27. LQR kontrolörlü robotun lineer olmayan dinamiklerinin Simulink blok modeli

LQR’de olduğundan, sistemin çıkışı, sistemi stabilize etmek için denetleyiciye dönen robotun durumlarıdır. LQG kontrolöründe, sistemin çıkışı sadece robotun pozisyonudur. LQG denetleyicisindeki Kalman filtresi diğer durumları tahmin edecektir. Kalman filtrelerinin girişi robotun x pozisyonu ve kontrol sinyali u , Kalman filtresinin çıkışı, Şekil 30’te gösterildiği gibi tüm tahmin edilen sistemin durumları \hat{x} . Tahmin edilen durumları robot stabilize etmek ve arzu edilen performans elde etmek için LQR denetleyicisine döndürmektedir.



Şekil 28. LQG kontrolörlü robotun lineer olmayan dinamiklerinin simulink blok modeli.

Sistemi kontrol etmek için kullanılan parametreler aşağıdaki şekilde seçilmiştir :

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 300000 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

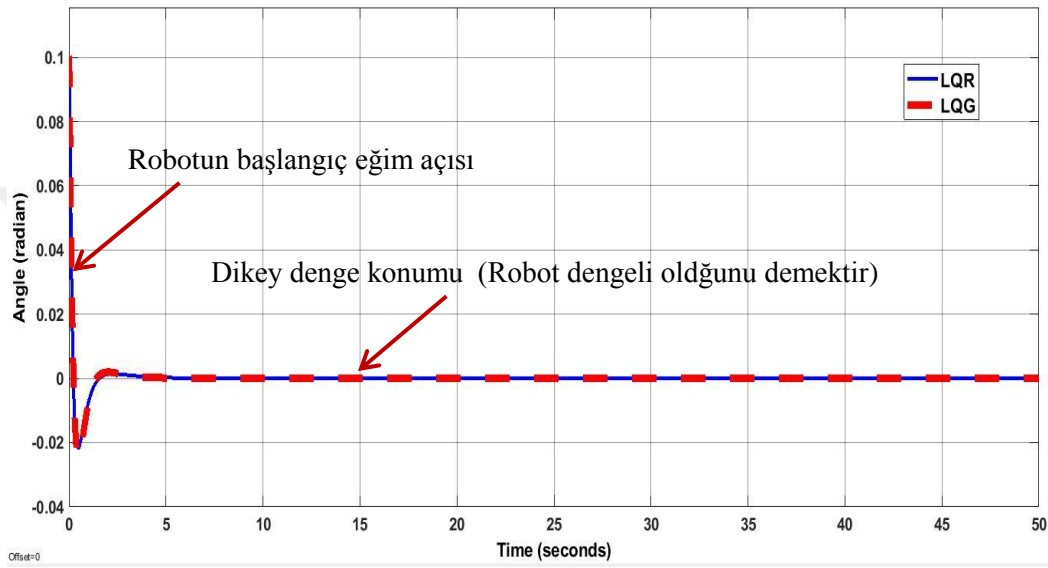
$$R = 1$$

$$k_r = [31.6 \quad -55.6 \quad 1732 \quad 165]$$

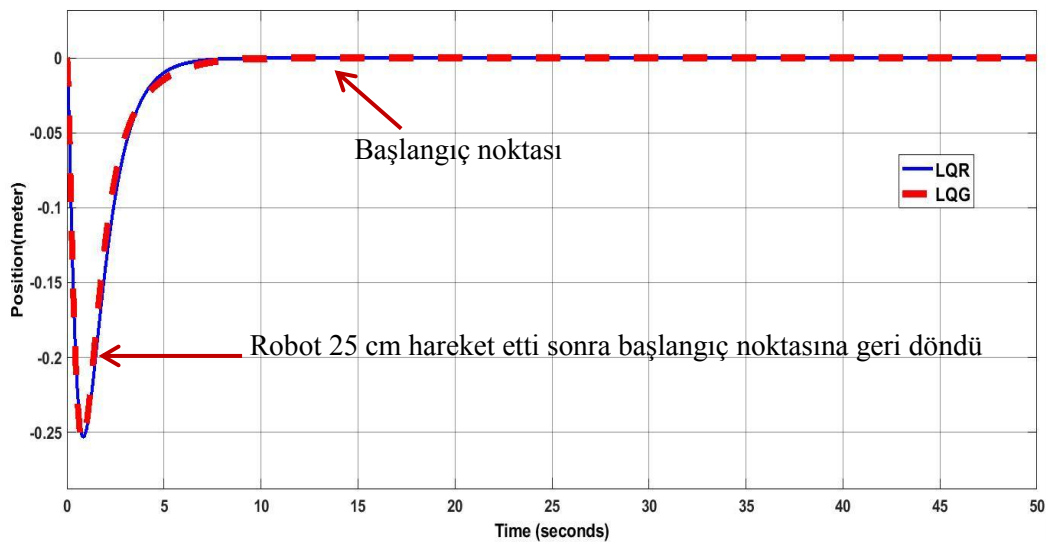
$$k_f = \begin{bmatrix} 25 \\ 313 \\ 387 \\ 4833 \end{bmatrix}$$

2.2.1.1. Dik Kendini Dengeleyen Durum

Bu durumda, her iki kontrolör de robotu düşmeden dik pozisyonda dengeleyebilir. Şekil 29ve Şekil 30’ da, bu uygulanmıştır bozuklukların üstesinden gelmek için 25 cm kadar hareket etmesi robotun dengeli olduğunu göstermektedir. Ardından robotun dengeli pozisyonunu koruyarak orijinal noktasına geri döndü.

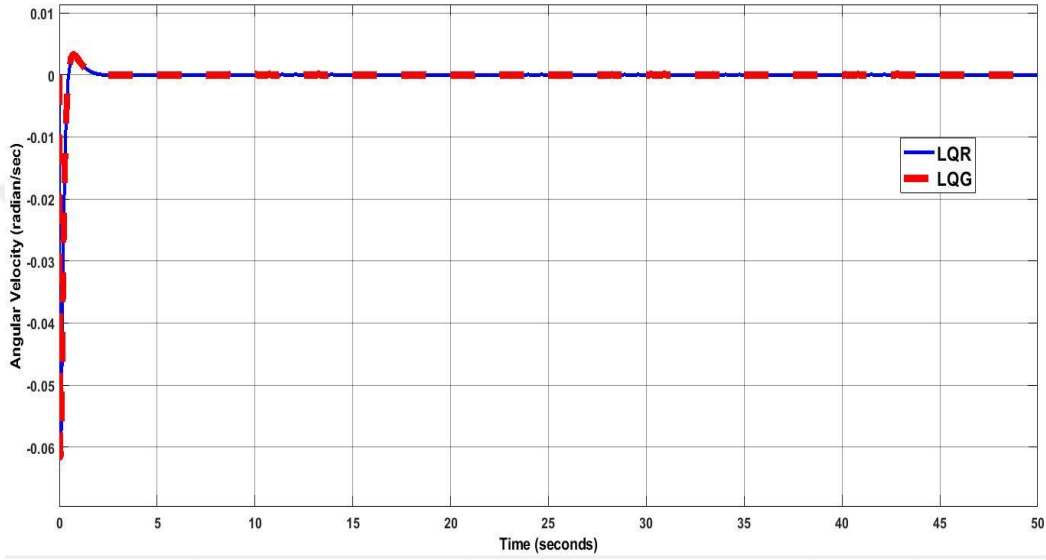


Şekil 29. LQR ve LQG denetleyicileri kullanan robotun açısı.

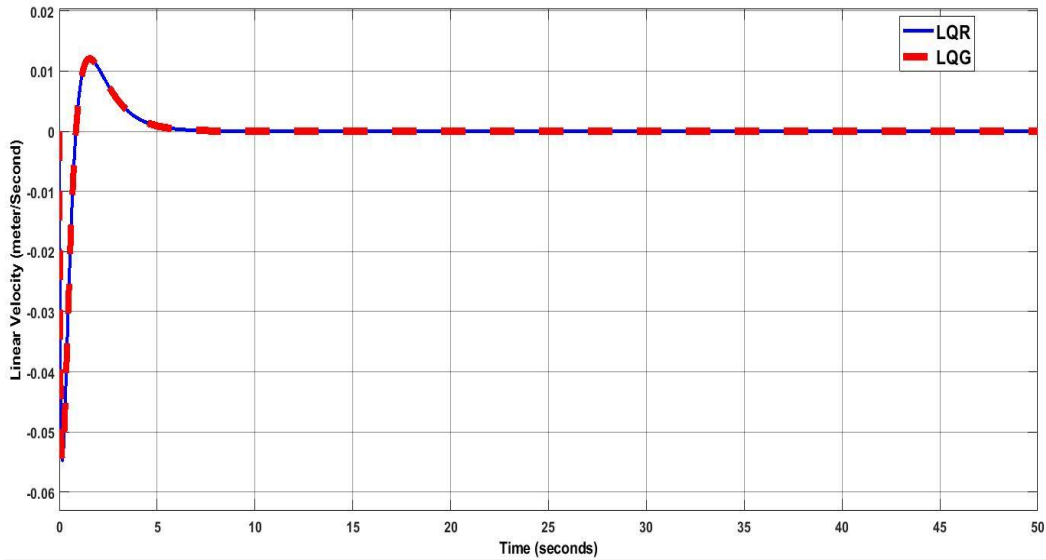


Şekil 30. LQR ve LQG denetleyicileri kullanarak olan robotun konumu.

Şekil 29'den görüldüğü gibi başlangıç değeri, robotun şasisine 0.1 Radyan = 5.7 derece olacak şekilde uygulanmıştır. Denetleyicileri, bu bozukluğun üstesinden gelmek ve açısı = 0 olan dengeli noktaya geri dönmüştür. Şekil 30'dan, uygulanan bozucu üstesinden gelmek için robotun geriye doğru 25 cm hareket ettiği görülmektedir. -0.1 bozulma uygulanmış ise robot ileri doğru hareket eder ve robot düşmeden orjinal noktasına geri dönmektedir.



Şekil 31. LQR ve LQG denetleyicileri kullanan robotun açısal hızı.

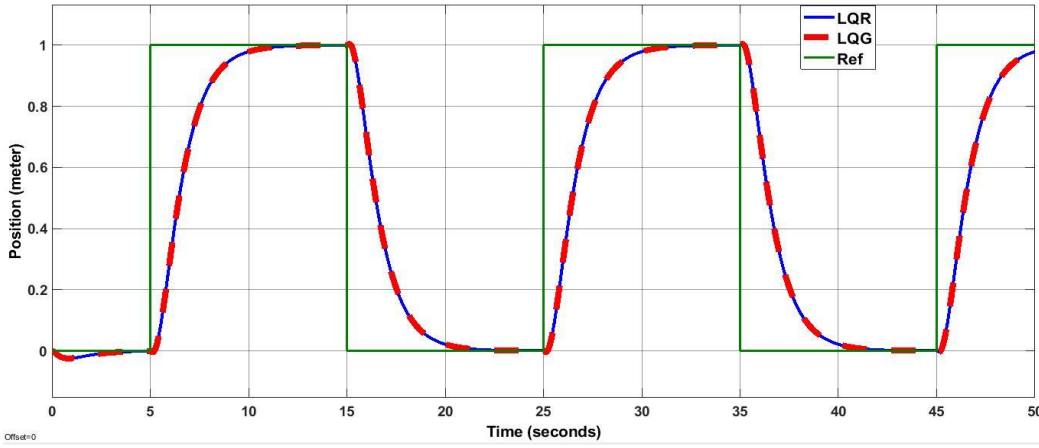


Şekil 32. LQR ve LQG kontrolörleri kullanan robotun doğrusal hızı.

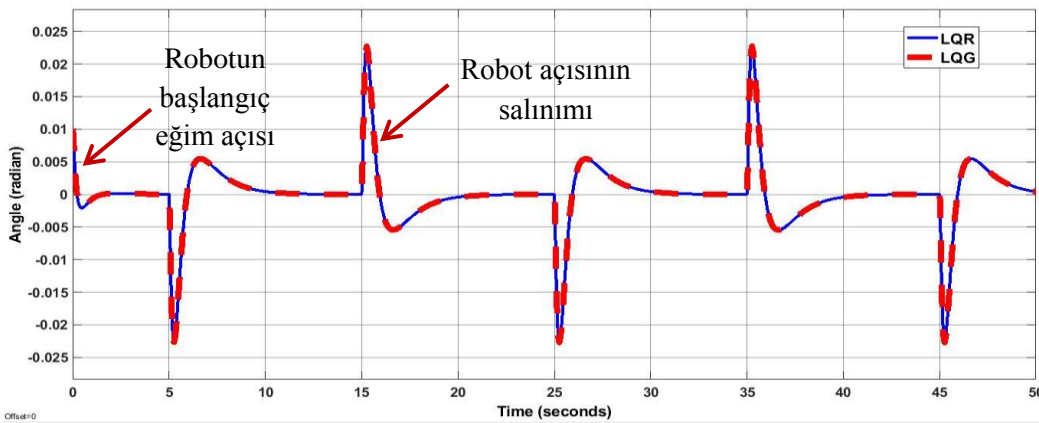
Şekil 31'den ve Şekil 32'den robotun açısal hızının ve lineer hızının negatif olduğu, yani robotun şasisinin saat yönünün tersine döndüğü ve robot tekerleklerinin geriye doğru hareket ettiği anlamına gelmektedir.

2.2.1.2. Referans İzleme Durumu

Bu durumda, her iki kontrol algoritmasının da dengeyi düşük hızlarda tutarken istenen pozisyona hareket edebildiği ve orijinal pozisyonuna dönebildiği gösterilmiştir. Şekil 33 ve Şekil 34, robot istenen metreye 1 metre hareket ettiğinden ve düşmeden başlangıç noktasına geri döndüğünden robotun hareketini ve dengelemesini göstermektedir.



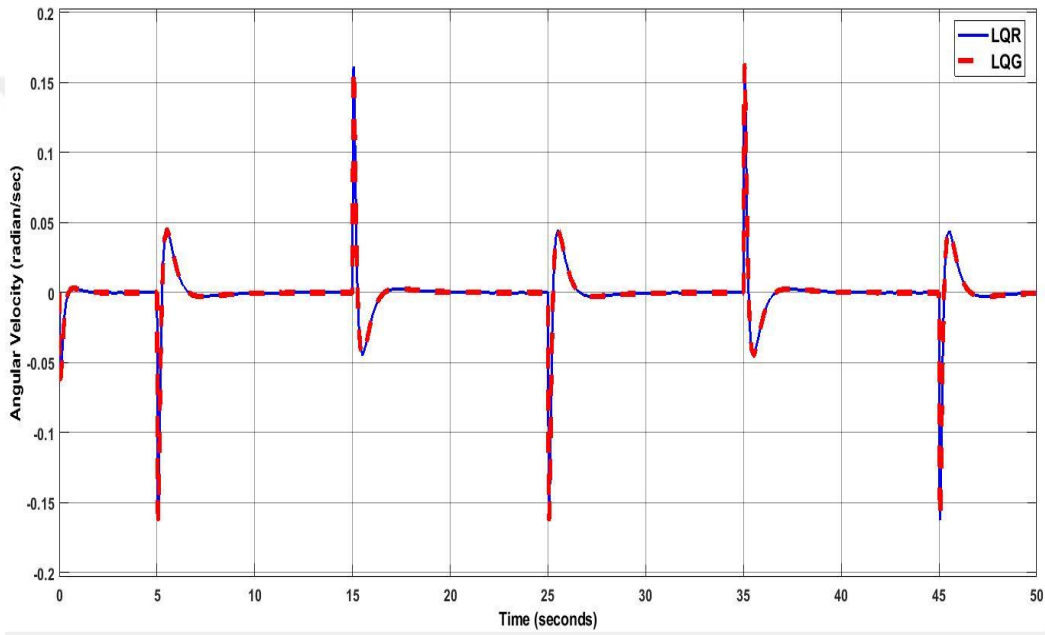
Şekil 33. Referans izleme durumunda LQR ve LQG denetleyicileri kullanan robotun konumu.



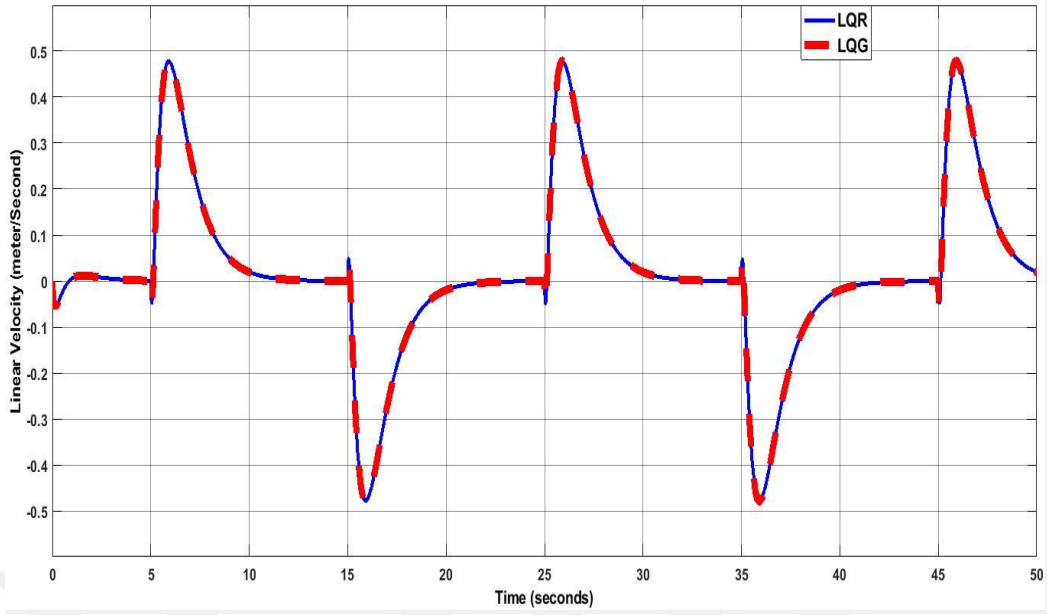
Şekil 34. Referans izleme durumunda LQR ve LQG denetleyicileri kullanan robotun açısı.

Şekil 33'dan hareketin başlangıcında, robotun başlangıç açısının üstesinden gelmek için biraz hareket ettiği görülmüştür. Bundan sonra, robot istenen noktaya taşınmıştır ve düşmeden orjinal noktaya geri dönmüştür.

Şekil 34' da robot açısının salınımının hareket sırasında $0.02 \text{ Rad} = 1.146 \text{ Dereceyi}$ aşmadığı görülmüştür. Robot ileri doğru hareket ettiğinde, robotun şasisinin açısı negatiftir, bu da saatin tersi yönde döndürüldüğü anlamına gelmektedir. Robot geriye doğru hareket ederse robotun şasisinin açısı pozitifdir bu robotun saat yönünde eğileceği anlamına gelmektedir.



Şekil 35. Referans izleme durumunda LQR ve LQG kontrolörleri kullanan robotun doğrusal hızı.

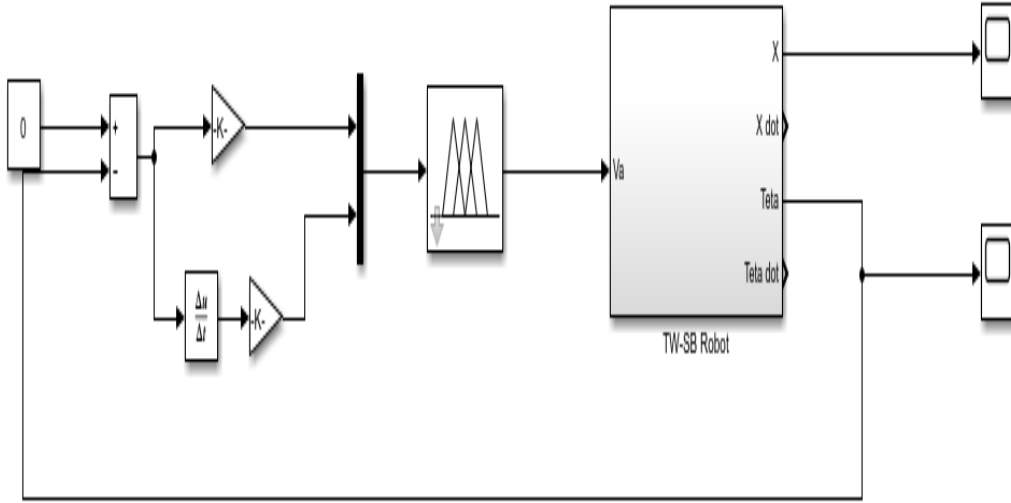


Şekil 36. Referans izleme durumunda LQR ve LQG denetleyicileri kullanarak olan robotun açısal hızı.

Her iki kontrol cihazı da aynı sonuçlara sahiptir, bu LQG kontrol cihazındaki Kalman filtresinin diğer tüm durumları doğru şekilde tahmin edebildiği anlamına gelmektedir.

2.3. Bulanık Mantık Denetleyicisi

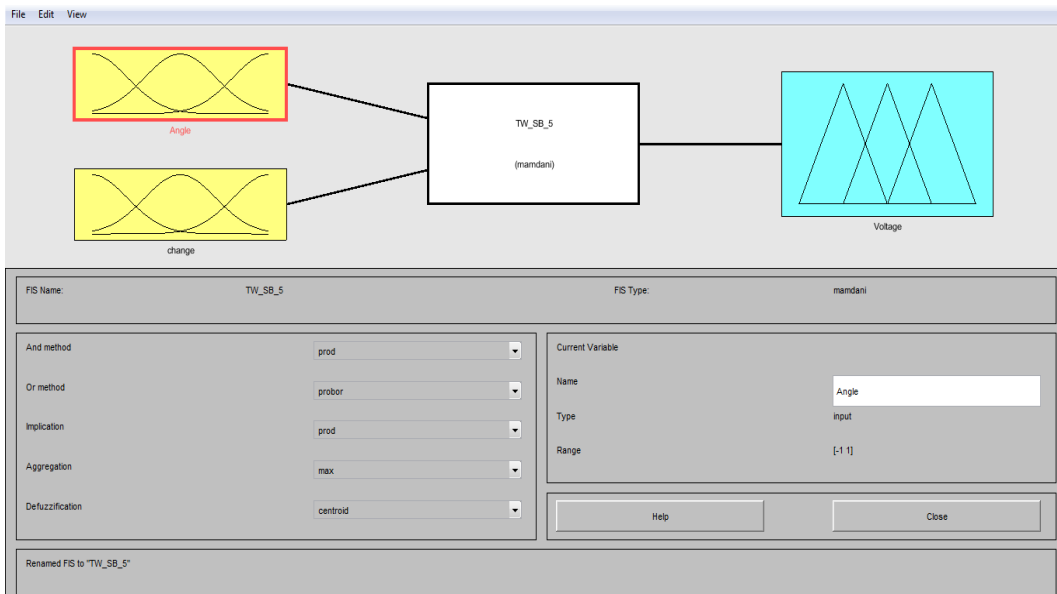
Bulanık mantık denetleyicisinde robotun sadece açısını kontrol etmek için inşa edilmiştir. Denetleyiciyi tasarlamamanın amacı robotu düşük hızlarda hareket ederken robotu dengede tutmaktır. Bulanık mantık denetleyici ile doğrusal olmayan iki tekerlekli Kendinden Dengeleme robotunun Simulink modeli Şekil 37'te gösterilmiştir. Mamdani tipi FIS kullanılarak bulanık mantık denetleyicisini tasarlanmıştır. Denetleyicide iki giriş vardır: Hata ve hatanın değiştirilmesi. Kontrol cihazının çıkışında ise DC motorlara uygulanan voltajdır. Önerilen kontrolörün performansı kendini dik dengeleyen durumda incelenmiştir.



Şekil 37. Bulanık mantık Denetleyicisi ile robotun doğrusal olmayan dinamiklerinin Simulink blok modeli

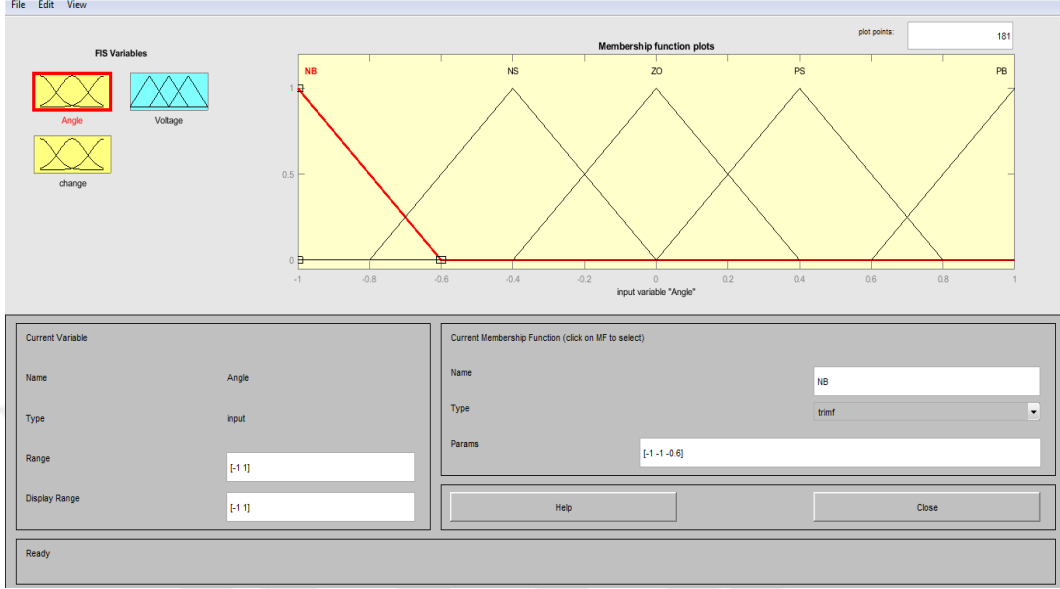
2.3.1.1. Matlab'daki Bulanık Mantık Denetleyicisinin Tasarımı

Bulanık mantık kontrolörü Şekil 38 'de gösterildiği gibi doğrusal olmayan iki tekerlekli kendini dengeleyen robot açısını kontrol etmek için Mamdani FIS arayüzünü kullanarak Matlab'da tasarlanmıştır.

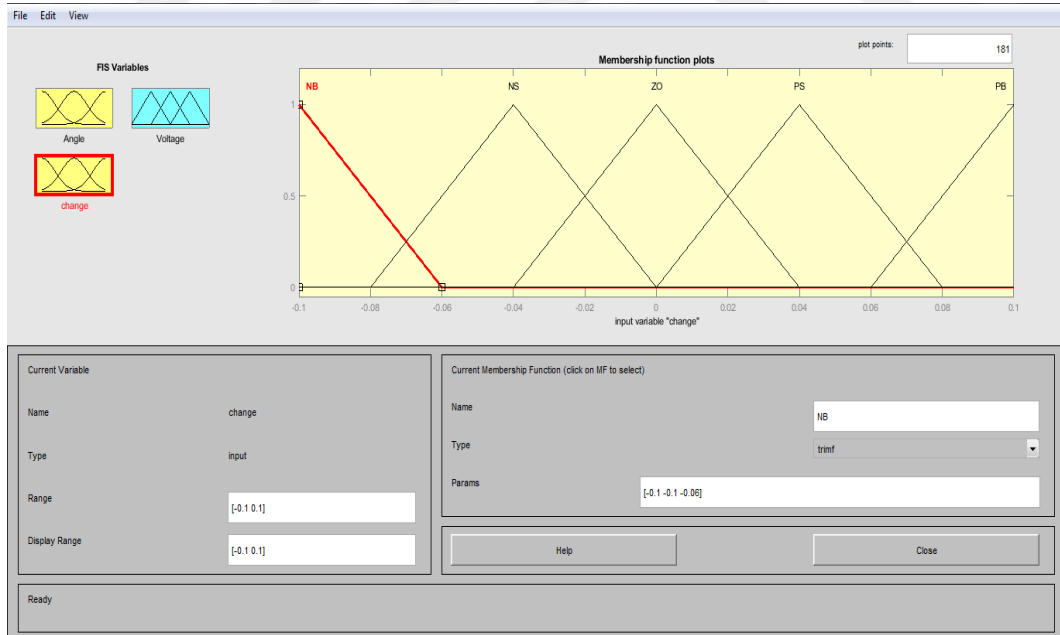


Şekil 38. Matlab'da Mamdani FIS arayüzü

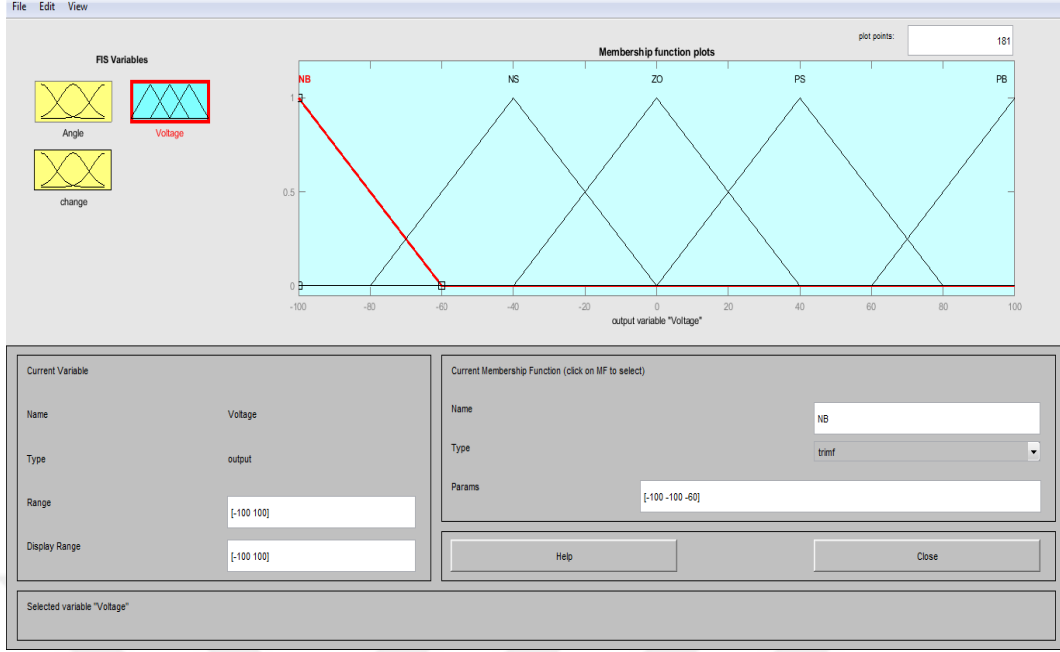
üyelik fonksiyonları her giriş ve çıkış için 5 alt kümeden oluşmuştur. Üçgen üyelik fonksiyonları alt kümeleri oluşturmak için seçilmiştir.



Şekil 39. bulanık mantık denetleyicisinin ilk girişi



Şekil 40. bulanık mantık denetleyicisinin ikinci girişi



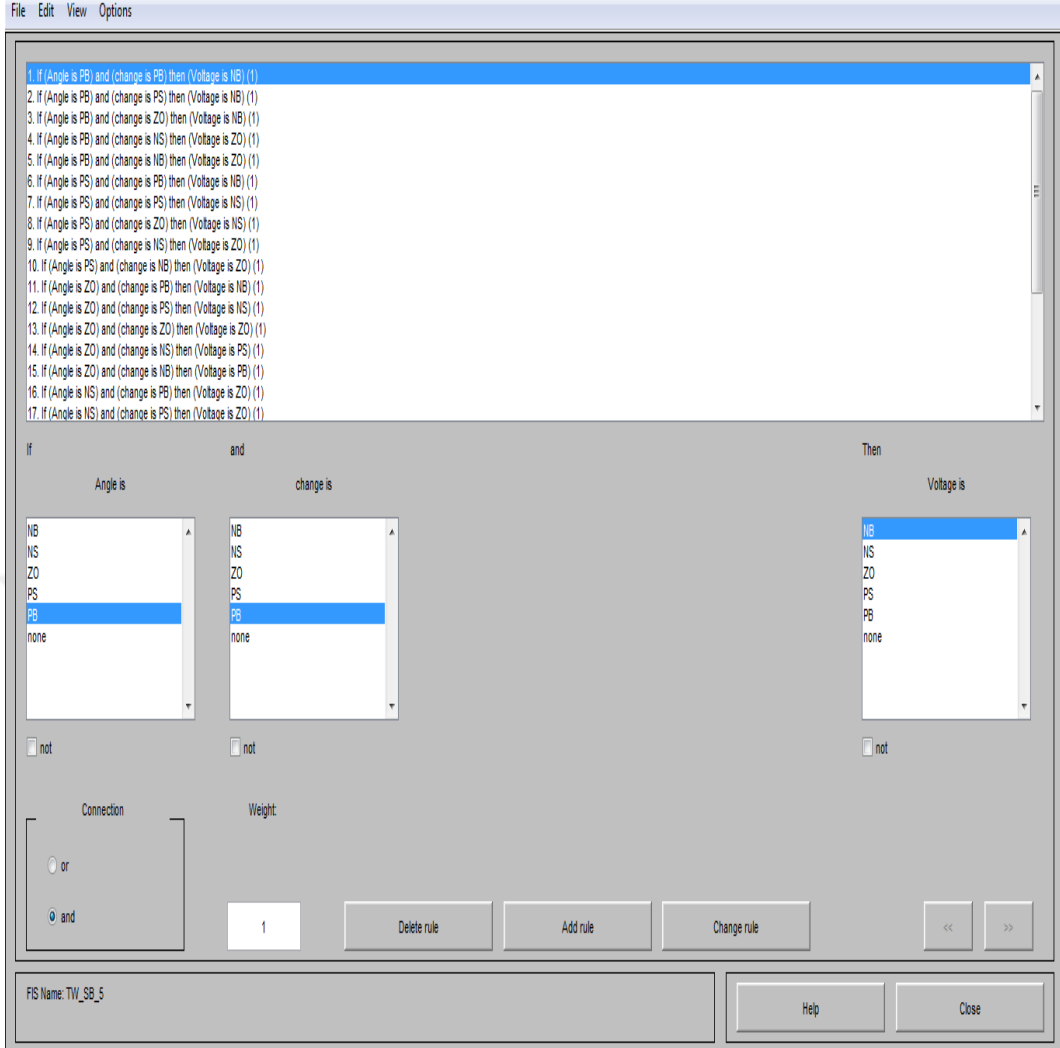
Şekil 41. bulanık mantık denetleyicisinin çıkışı

2.3.1.2. Bulanık Denetleyicinin Kuralları

Denetleyicinin her biri için beş üyelik fonksiyonuna sahip iki girişi olduğundan bulanık kurallar tablosu Şekil 42'te gösterildiği gibi 25 kuraldan oluşmuştur.

Dilsel değişkenler aşağıdaki gibi tanımlanmıştır:

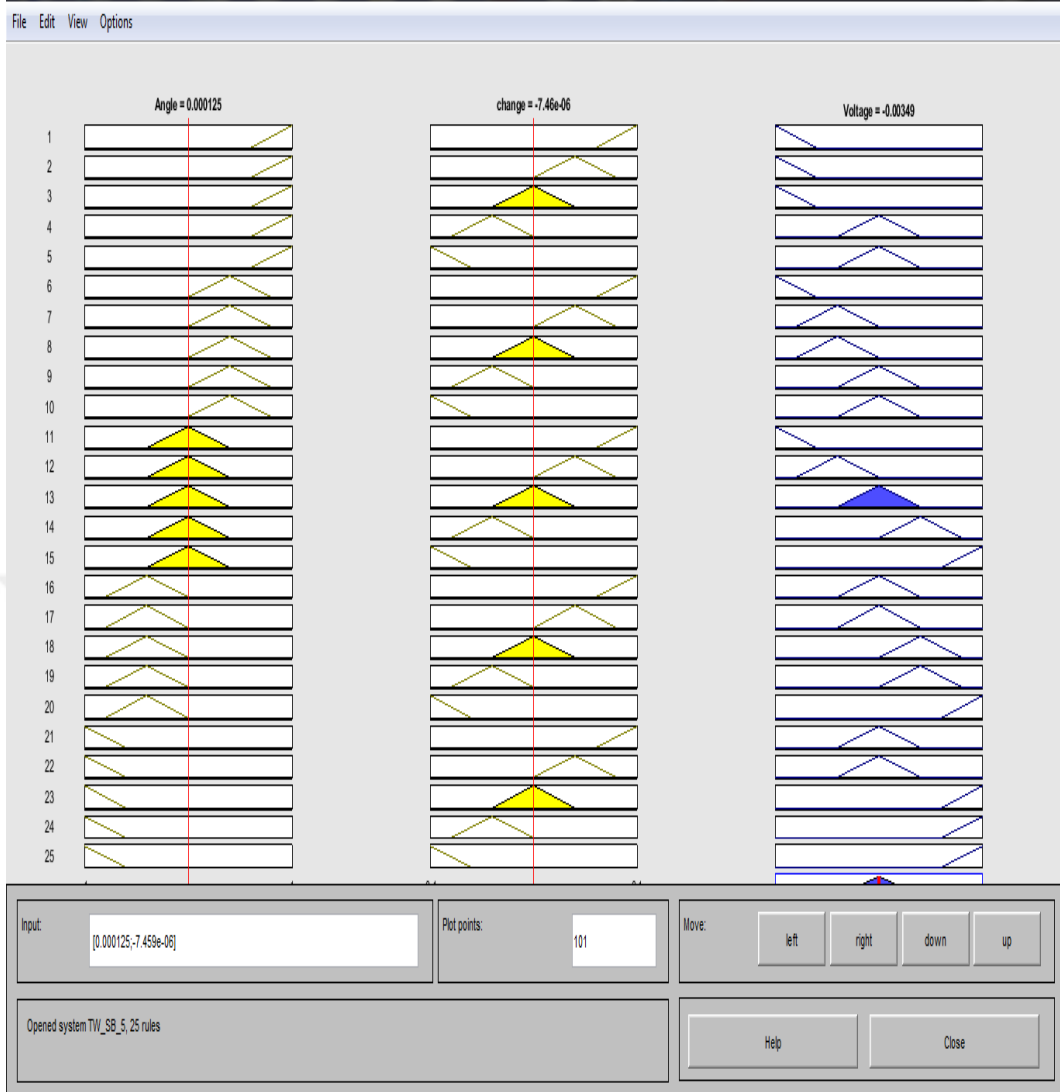
- NB: Negatif Büyük
- NS: Negatif Küçük
- ZE: Sıfır
- PS: Pozitif Küçük
- PB: Pozitif Büyük



Şekil 42. Bulanık mantık denetleyicisinin kuralları

Tablo 4. Bulanık denetleyicinin kuralları

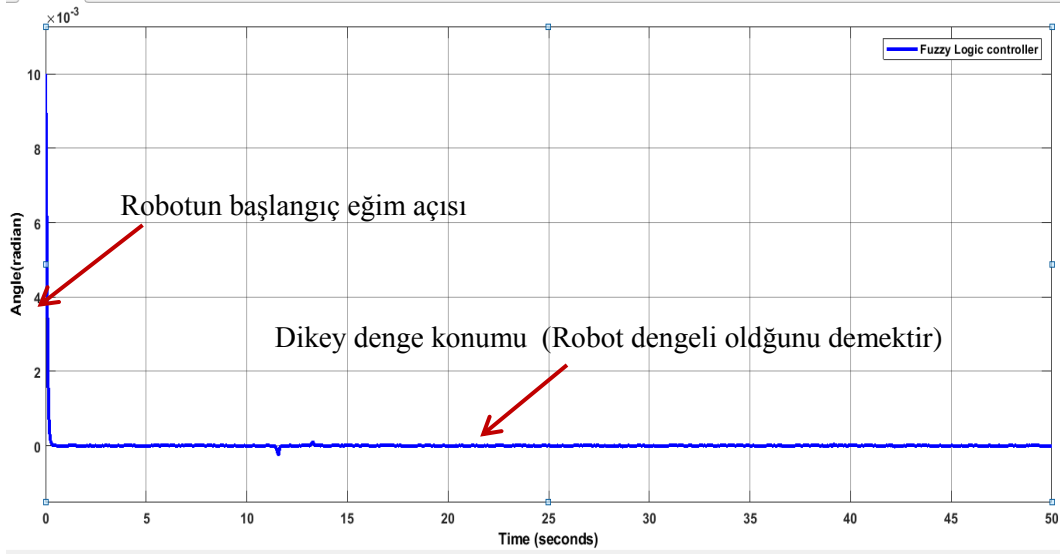
	<i>Açı hatasının türevi</i>					
		PB	PS	ZE	NS	NB
<i>Açı hatası</i>	PB	NB	NB	NM	ZE	ZE
	PS	NB	NS	NS	ZE	ZE
	ZE	NM	NS	ZE	PS	PB
	NS	ZE	ZE	PS	PS	PB
	NB	ZE	ZE	PB	PB	PB



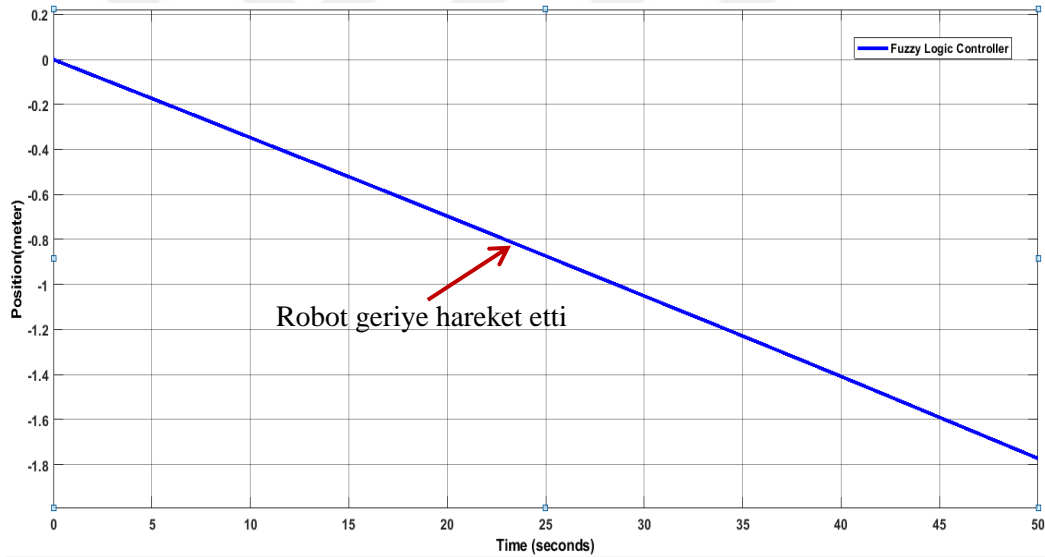
Şekil 43. FIS Matlab arayüzünde kural inceleyici

2.3.1.3. Kendini Dik Dengeleyen Durum

Bu durumda, kontrolör düşmeden robotu dik bir konumda dengeleyebilir . Şekil 44 ve Şekil 45, robotun tekerlekleri hareket etmeye devam ederken robotun dengeli olduğunu gösterir. Ayrıca önerilen kontrolör uygulanan bozukluğun üstesinden gelmiştir.

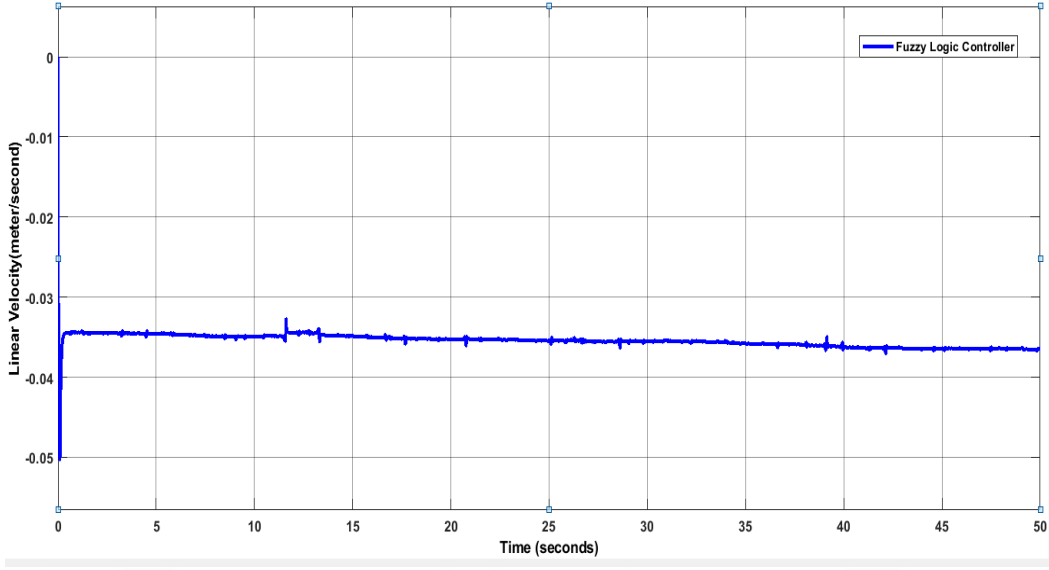


Şekil 44. Bulanık mantık denetleyicisi'ni kullanarak olan robotun açısı

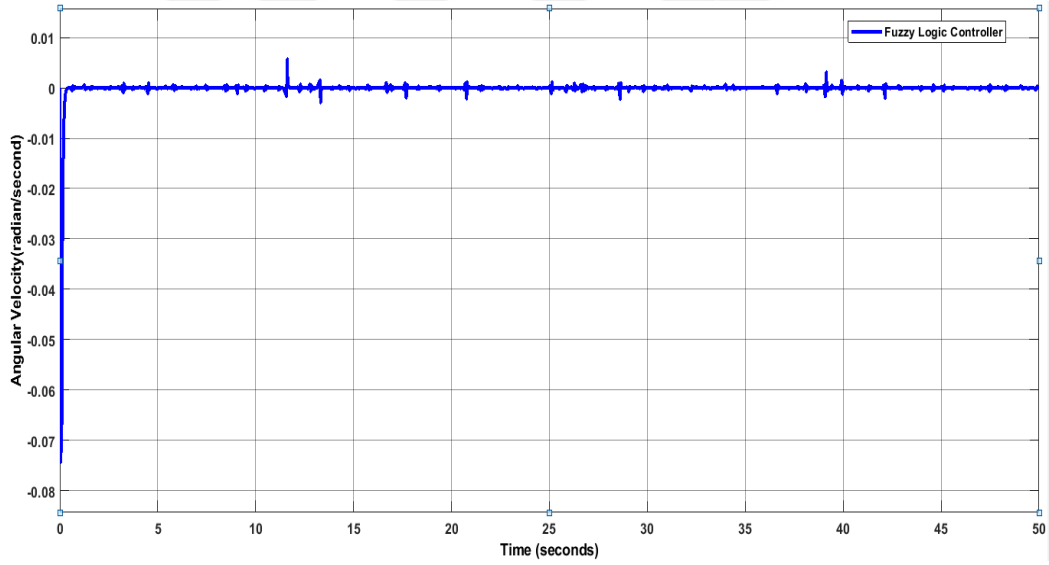


Şekil 45. Bulanık mantık denetleyicisi'ni kullanarak olan robotun konumu

Şekil 44'te görüldüğü gibi, bozukluk robotun şasisine $0.01 = 0.57$ dereceye eşit olarak uygulanır, kontrolör bu bozukluğun üstesinden gelir ve 0 açısında dik dengeli konuma geri döner. Robotun geriye doğru hareket etmesini sağlamak için robotun şasisi Şekil 44 ve Şekil 45'da gösterildiği gibi pozitif yönde (saat yönünde) yatırılmalıdır.

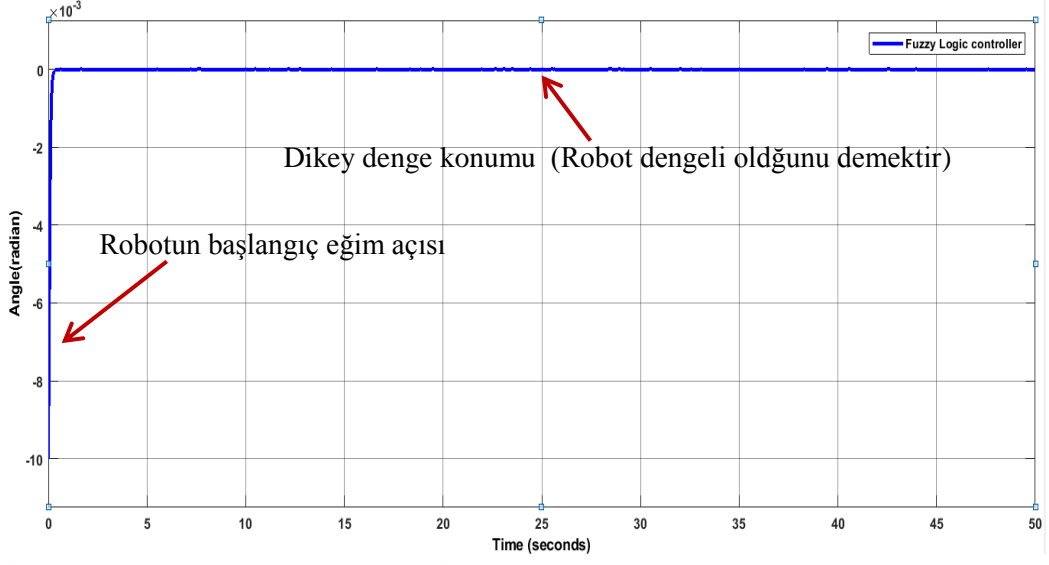


Şekil 46. Bulanık mantık denetleyicisi'ni kullanarak olan robotun doğrusal hızı

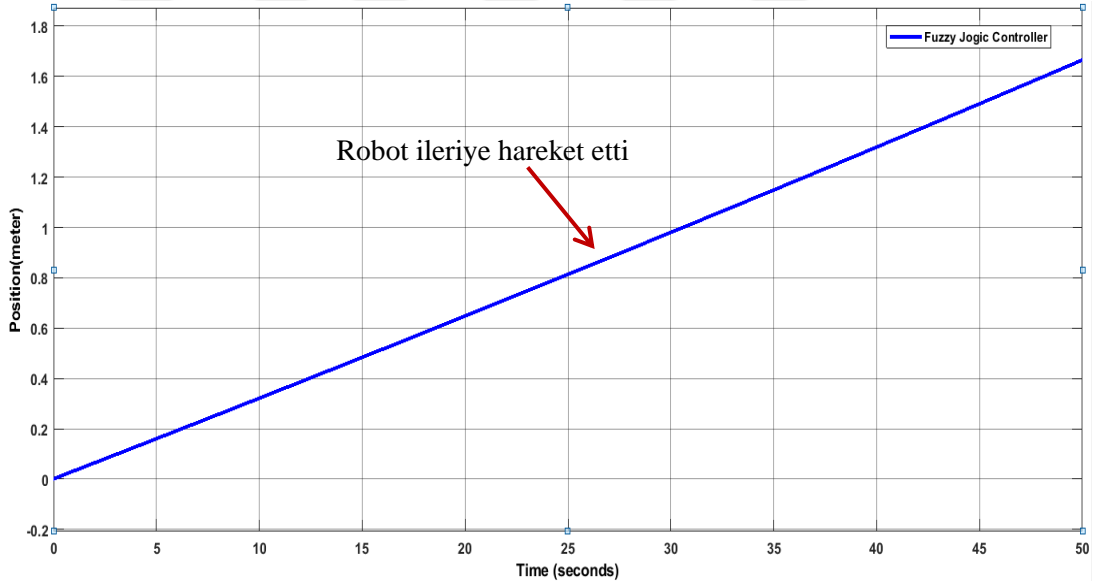


Şekil 47. Bulanık mantık denetleyicisi kullanılarak olan robot şasisinin açısız hızı

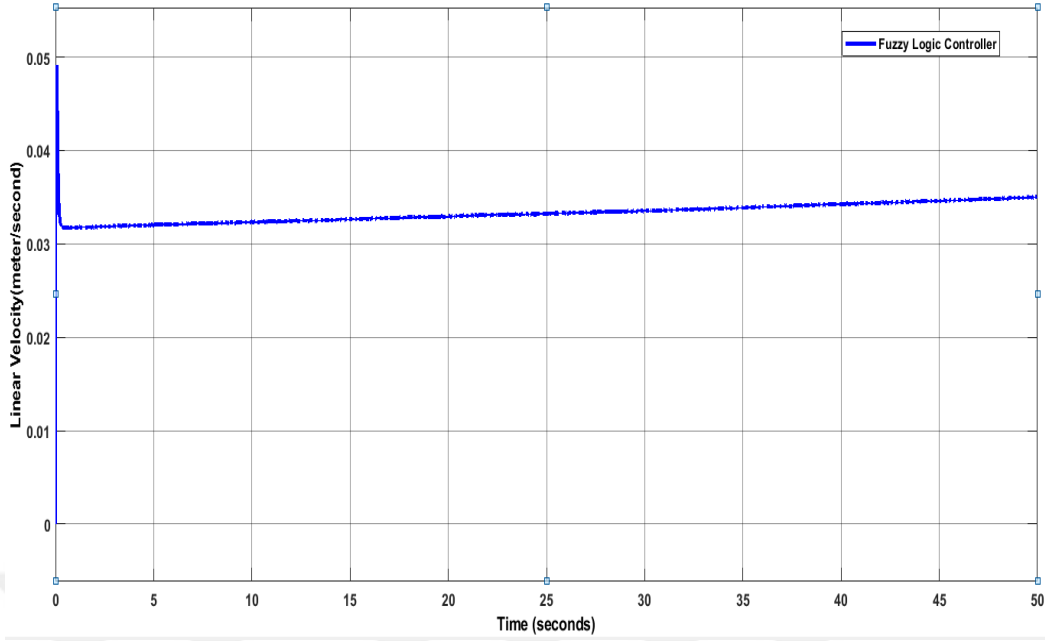
Şekil 46 ve Şekil 47'dan robotun şasisinin açısız hızının ve robotun tekerleklerinin doğrusal hızının negatif olduğu gösterilmiştir, bu da robotun şasisinin saat yönünün tersine döndüğü ve robotun tekerleklerinin geriye doğru hareket ettiği anlamına gelir. Öte yandan, robotun ileri hareket etmesi için robotun şasisi, Şekil 48 ve Şekil 49'da gösterildiği gibi negatif bir yönde (saat yönünün tersine) döndürülmelidir.



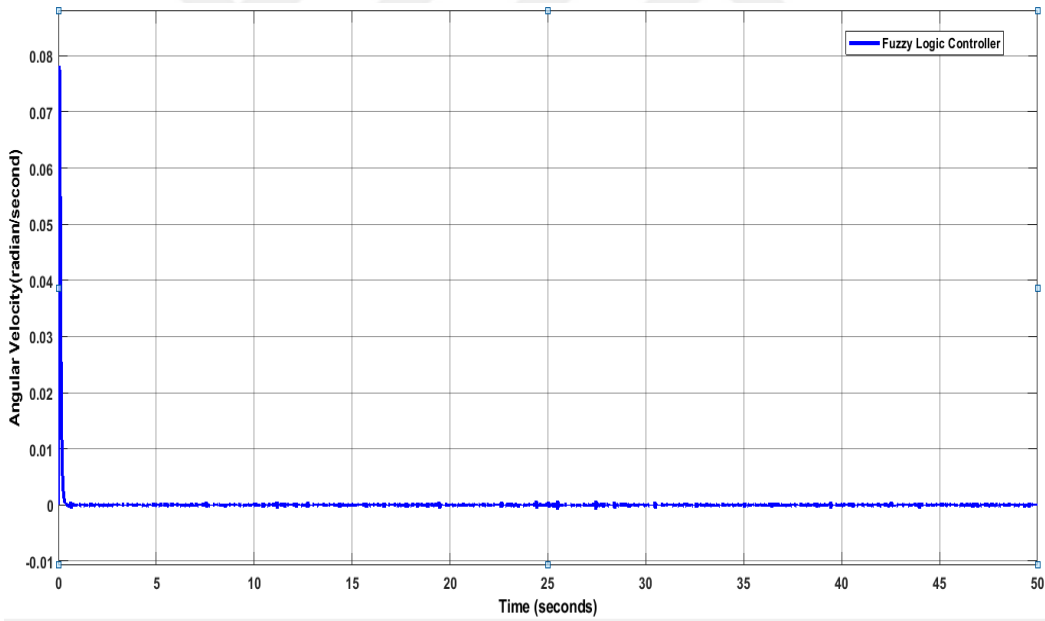
Şekil 48. Bulanık mantık denetleyicisi'ni kullanarak olan robotun açısı



Şekil 49. Bulanık mantık denetleyicisi'ni kullanarak olan robotun konumu



Şekil 50. Bulanık mantık denetleyicisi'ni kullanarak olan robotun doğrusal hızı

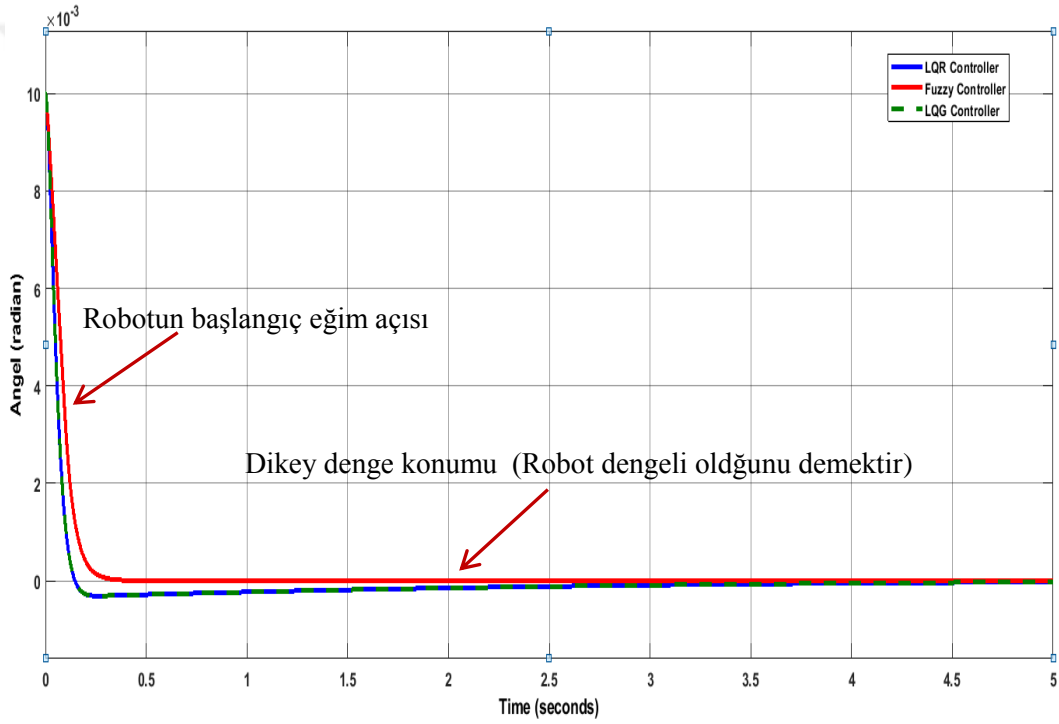


Şekil 51. Bulanık mantık denetleyicisi kullanılarak olan robot şasisinin açısız hızı

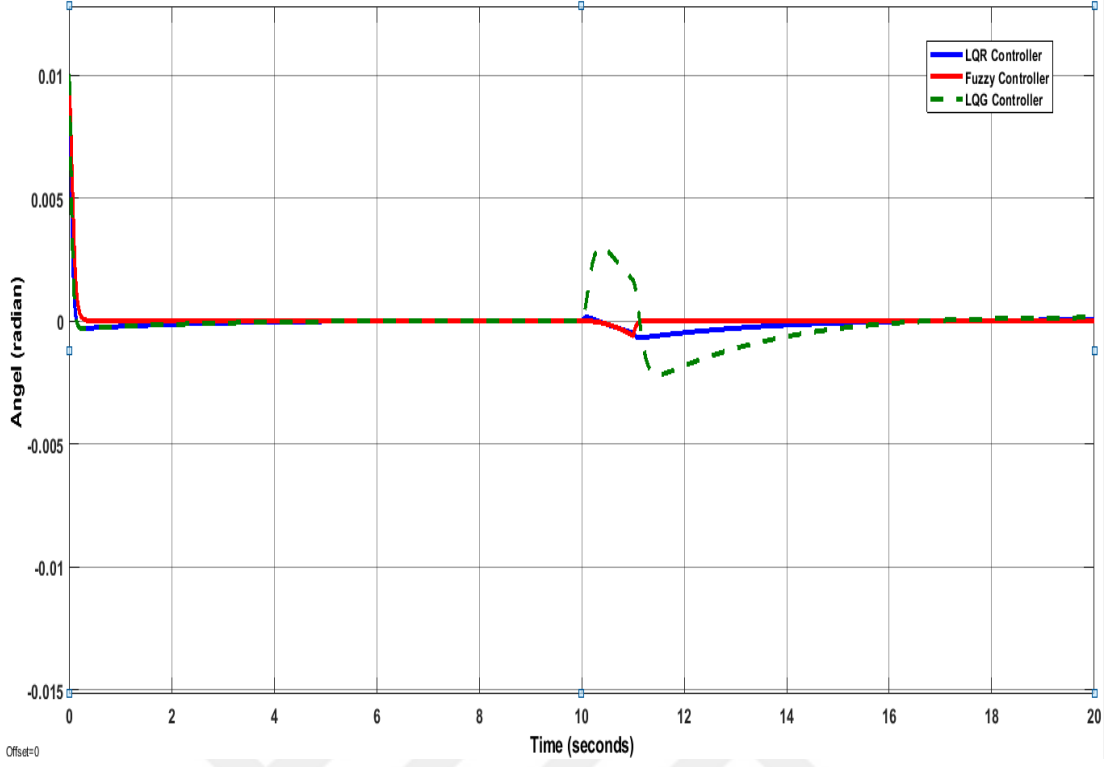
Şekil 50 ve Şekil 51'den robotun şasisinin açısız hızının ve robotun tekerleklerinin doğrusal hızının pozitif olduğu gösterilmektedir, bu da robotun şasisinin saat yönünde döndüğü ve robotun tekerleklerinin ileri doğru hareket ettiği anlamına gelir.

2.4. LQG, LQR Ve Bulanık Mantık Denetleyicisi Arasında Karşılaştırılması

Bu kısımda, bulanık mantık, LQR ve LQG denetleyicileri robotun açısını kontrol etmek için karşılaştırılmıştır. Şekilde 52 görüldüğü gibi, her kontrolör robotun açısını dengeleyebilir. Ancak bulanık mantık denetleyicisi LQR ve LQG denetleyicilerinden daha hızlıdır. Bundan sonra, bir bozucu 4 ve 5 saniye arasında robotun şasisine uygulanmıştır. Şekil 53'den görüldüğü gibi, bulanık kontrolör küçük salınım ile uygulanan bozucunu üstesinden gelmiştir, LQG kontrolörü büyük salınım ile uygulanan bozucunu üstesinden gelmiştir.



Şekil 52. LQG ,LQR ve bulanık mantık denetleyicileri kullanan robotun açısı



Şekil 53. Robot şasisinin açısını kontrol etmek için bulanık mantık, LQR ve LQG kontrolörleri arasındaki karşılaştırma

3. SONUÇLAR

Bu tezin amacı, Matlab / Simulink'teki robotun doğrusal olmayan dinamik modeline bağlı iki tekerlekli kendi kendini dengeleyen robotun dengelenmesini sağlayarak başarılı bir şekilde gerçekleştirilmiştir. Robot için denge kontrolü sorununun üstesinden gelmek için iki farklı kontrol yöntemi kullanılmıştır.

İlk stratejide, sistemi dengelemek için LQG kontrol algoritması kullanılmıştır. Bu kontrol yönteminde, doğrusal kuadratik regülatör (LQR) ve doğrusal kuadratik tahmin ediciyi (LQE) tasarlamak için doğrusallaştırılmış matematiksel model tanımlanmıştır. Kontrolör, robotun Matlab / Simulink'teki açısını ve konumunu kontrol etmek için robotun doğrusal olmayan matematiksel modeline uygulanmıştır.

İkinci stratejide, robotun dengelenmesini sağlamak için Bulanık mantık denetleyicisi kullanılmıştır. Ayrıca bulanık mantık denetleyicisi, Matlab / Simulink'teki robotun doğrusal olmayan dinamiklerine uygulanmıştır.

Simülasyon sonuçları, her iki kontrolörün düşük hızlarda da robotu dengeleyebildiğini ve bozuklukların üstesinden gelebildiğini göstermektedir.

4. ÖNERİLER

Bu tez, iki tekerlekli kendi kendini dengeleyen bir robotun geliştirilmesi ve robotun dengelenmesinin geliştirilmesi üzerine gelecekteki arařtırmaların temelini oluřturmaktadır. Zaman sınırlamaları nedeniyle bu arařtırmanın gelecekteki alıřmalarında iyileřtirilebilecek ve geniřletilebilecek birok alanı vardır.

Gelecekteki alıřmalar için bazı öneriler ařağıdaki gibi özetlenmiřtir:

- Önerilen kontrol algoritmalarını gerek prototipte uygulamak ve sonucu elde edilen simülasyon sonuçları ile karřılařtırmak.
- Önerilen kontrol yöntemleri, düşük hızlarda hareket sırasında robotun dengelenmesini kontrol etme verimliliğini göstermiřtir. Bu nedenle robot için uyarlanabilir bir bulanık denetleyici geliřtirmek Yüksek hızda gelecekteki alıřmalar için ok önemlidir.
- Genetik algoritmalar gibi sezgisel yöntemlerden bazılarını kullanarak LQG ve Bulanık denetleyicilerin parametrelerini ayarlanabilir.
- Sinir ağı gibi robotun kontrolörünü geliřtirmek için diđer akıllı kontrol algoritmaları düşünülebilir.

5. KAYNAKLAR

1. F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: a mobile, inverted pendulum," IEEE Trans. Ind. Electron., vol. 49, no. 1, 2002, pp. 107–114.
2. <http://www.segway.com>, D. Kamen, Apr-2015.
3. Rich Chi Ooi, "Balancing a Two-Wheeled Autonomous Robot," Univ. West.Aust., Degree in Mechatronics Engineering, 2003.
4. M. P. M. Baloh, "Modeling and model verification of an intelligent self-balancing two-wheeled vehicle for an autonomous urban transportation system," Proc. Conf. Comp. Intelligence Robot. Autonom. Syst., Dec. 2003.
5. W. An and Y. Li, "Simulation and control of a two-wheeled self-balancing robot," in 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), Shenzhen, China, Dec. 2013 - Dec. 2013, pp. 456–461.
6. V. Tjandra and E. F. Sinaga, "Simulating the behaviour of PID controlled small-scale transformer robot," in 2016 International Conference on Instrumentation, Control and Automation (ICA), Bandung, Indonesia, Aug. 2016 - Aug. 2016, pp. 137–142.
7. J. A. B. Valencia, J. J. R. Pasaye, and R. G. Bernai, "Instrumentation and wireless control for the self-balancing mobile robot on two wheels," in 2014 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), Ixtapa, Mexico, Nov. 2014 - Nov. 2014, pp. 1–5.
8. M. I. Ali and M. M. Hossen, "A two-wheeled self-balancing robot with dynamics model," in 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Sep. 2017 - Sep. 2017, pp. 271–275.
9. Chhotray, D. R. Parhi, R. Kundu, P. B. Kumar, and K. K. Pandey, "Modelling and Stability Analysis of a TWMM Using Kalman Filter and PID Controller," in 2018 International Electrical Engineering Congress (iEECON), Krabi, Thailand, Mar. 2018 - Mar. 2018, pp. 1–4.
10. B.-B. Li, J.-H. Zhu, and H.-Q. Min, "BMP: A self-balancing mobile platform," in 2012 International Conference on Machine Learning and Cybernetics, Xian, Shaanxi, China, Jul. 2012 - Jul. 2012, pp. 868–874.
11. Y. Gong, X. Wu, and H. Ma, "Research on Control Strategy of Two-Wheeled Self-Balancing Robot," in 2015 International Conference on Computer Science and Mechanical Automation (CSMA), Hangzhou, China, Oct. 2015 - Oct. 2015, pp. 281–284.

12. M. Engin, "Embedded LQR controller design for self-balancing robot," in 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva, Jun. 2018 - Jun. 2018, pp. 1–4.
13. L. J. Butler and G. Bright, "Feedback control of a self-balancing materials handling robot," in 2008 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, Dec. 2008 - Dec. 2008, pp. 274–278.
14. C.-C. Tsai and C.-H. Tsai, "Adaptive robust motion control using fuzzy wavelet neural networks for uncertain electric two-wheeled robotic vehicles," in 2013 International Conference on System Science and Engineering (ICSSE), Budapest, Hungary, Jul. 2013 - Jul. 2013, pp. 229–234.
15. H. M. Omar, A. M. Elalawy, and H. H. Ammar, "Two-wheeled Self balancing robot Modeling and Control using Artificial Neural Networks (ANN)," in 2019 Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, Oct. 2019 - Oct. 2019, pp. 196–200.
16. L. Wang and J. Lei, "Research on Self-Balancing Control for Upright Vehicle with Two Wheels Based on Sliding Mode Control," in 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, Dec. 2017 - Dec. 2017, pp. 1192–1195.
17. C.-C. Tsai, S.-Y. Ju, and S.-M. Hsieh, "Trajectory tracking of a self-balancing two-wheeled robot using backstepping sliding-mode control and fuzzy basis function networks," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Oct. 2010 - Oct. 2010, pp. 3943–3948.
18. J. Qiu et al., "Two-wheeled self-balancing robot modeling and nonlinear control," in 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Jun. 2017 - Jul. 2017, pp. 734–739.
19. L. Pupek and R. Dubay, "Velocity and position trajectory tracking through sliding mode control of two-wheeled self-balancing mobile robot," in 2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Apr. 2018 - Apr. 2018, pp. 1–5.
20. F. Dai et al., "Development of a coaxial self-balancing robot based on sliding mode control," in 2012 IEEE International Conference on Mechatronics and Automation, Chengdu, China, Aug. 2012 - Aug. 2012, pp. 1241–1246.
21. N. Uddin, "Adaptive Control System Design for Two-Wheeled Robot Stabilization," in 2018 12th South East Asian Technical University Consortium (SEATUC), Yogyakarta, Indonesia, Mar. 2018 - Mar. 2018, pp. 1–5.
22. C.-C. Tsai, S.-C. Lin, and B.-C. Lin, "Intelligent adaptive motion control using fuzzy basis function networks for self-balancing two-wheeled transporters," in International Conference on Fuzzy Systems, Barcelona, Spain, Jul. 2010 - Jul. 2010, pp. 1–6.

23. I. Jmel, H. Dimassi, S. Hadj-Said, and F. Msahli, "An adaptive observer for two wheeled self-balancing robot with a varying center of mass," in 2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, Mar. 2019 - Mar. 2019, pp. 467–472.
24. L. Jiang, H. Qiu, Z. Wu, and J. He, "Active disturbance rejection control based on adaptive differential evolution for two-wheeled self-balancing robot," in 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, May. 2016 - May. 2016, pp. 6761–6766.
25. X. Chen, H. Zhao, H. Sun, and S. Zhen, "Adaptive Robust Control Based on Moore-Penrose Generalized Inverse for Underactuated Mechanical Systems," IEEE Access, vol. 7, 2019, pp. 157136–157144.
26. S. Jung and S. S. Kim, "Control Experiment of a Wheel-Driven Mobile Inverted Pendulum Using Neural Network," IEEE Trans. Contr. Syst. Technol., vol. 16, no. 2, 2008, pp. 297–303.
27. C.-H. Chiu and C.-C. Chang, "Design and Development of Mamdani-Like Fuzzy Control Algorithm for a Wheeled Human-Conveyance Vehicle Control," IEEE Trans. Ind. Electron., vol. 59, no. 12, 2012, pp. 4774–4783.
28. C. Sun, T. Lu, and K. Yuan, "Balance control of two-wheeled self-balancing robot based on Linear Quadratic Regulator and Neural Network," 2013 Fourth International Conference, 2013, pp. 862–867 .
29. S.-C. Lin and C.-C. Tsai, "Development of a Self-Balancing Human Transportation Vehicle for the Teaching of Feedback Control," IEEE Trans. Educ., vol. 52, no. 1, 2009, pp. 157–168.
30. H.-S. Juang and K.-Y. Lurrr, "Design and control of a two-wheel self-balancing robot using the arduino microcontroller board," in 2013 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, Jun. 2013 - Jun. 2013, pp. 634–639.
31. C.-C. Tsai, H.-C. Huang, and S.-C. Lin, "Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter," IEEE Trans. Ind. Electron., vol. 57, no. 4, 2010, pp. 1420–1428.
32. T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control – Taming Nonlinear Dynamics and Turbulence*. Cham: Springer International Publishing, 2017.
33. S. Skogestad and I. Postlethwaite, *Multivariable feedback control- Analysis and design*, 2nd ed. Chichester: Wiley, 2005.
34. H. Saadat, *Computational aids in control systems using MATLAB*. New York, London: McGraw-Hill, 1993.

35. I. H. Altas, Fuzzy logic control in energy systems With design applications in MATLAB /Simulink. Stevenage: Institution of Engineering and Technology, 2017.
36. I.H. Altas and A.M. Sharaf, "A Generalized Direct Approach for Designing Fuzzy Logic Controllers in Matlab/Simulink GUI Environment", ", International Journal of Information Technology and Intelligent Computing, Int. J. IT&IC no.4 vol.1, 2007.
37. https://www.st.com/content/st_com/en.html, STMicroelectronics.
38. S. L. Brunton and J. N. Kutz, Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge: Cambridge University Press, 2019.
39. R. F. Stengel, Optimal control and estimation. New York: Dover Publications; London Constable, 1994.
40. J.C. Doyle, "Guaranteed margins for LQG regulators," IEEE Transactions on Automatic Control, 1978.
41. M. Margolis, Make an Arduino-controlled robot. Beijing: O'Reilly, 2012.

ÖZGEÇMİŞ

Cemil DABBAH 1992 yılında Halep şehrinde doğdu. İlkokul, ortaokul ve lise eğitimini Halep'te tamamladı. Halep Üniversitesi Elektrik-Elektronik Mühendisliği Kontrol ve Otomasyon Bölümü'nden 2014 yılında mezun oldu. 2017 yılında Karadeniz Teknik Üniversitesi Elektrik-Elektronik Mühendisliği Bölümünde yüksek lisans eğitimine başladı. İngilizce, Türkçe ve Arapça bilmektedir.

