

Otonom Bir Mobil Robotun Parçacık Sürü Algoritması ile Optimum Yörünge Kontrolü

Ebubekir PEKDEMİR¹, Mahit GÜNEŞ^{1*}

Kahramanmaraş Sütçü İmam Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Kahramanmaraş, Türkiye.

ÖZET: Bu çalışmada amaçlanan, mevcut bir fiziksel sistemi algılayarak Simulink eşdeğer modelini elde etmek ve elde edilen bu modele uygun parçacık sürü algoritması ile iyileştirilmiş bir pid kontrol sistemi geliştirmektir. Kontrol edilecek sistem olarak MBR – 01 mobil robotu kullanılmıştır. Sistemin modellenebilmesi için, mekanik sınırlandırmalar deneysel olarak tespit edilmiş ve bu sınırlara uygun simulink model elemanları türetilmiştir. Elde edilen Simulink modelinin, PSO'suz ve PSO'lu ayrı ayrı PID kontrolü gerçekleştirilerek, sistemin çıkış hataları karşılaştırılmıştır. Referans takip hatası %5-7 'den %0,1-0,2'ye düşürülmüştür.

Anahtar Kelimeler : Parçacık sürü algoritması, PID kontrol algoritması.

Optimum Trajectory Control of an Autonomous Mobile Robot with Particle Swarm Algorithm

ABSTRACT: What is intended in this study is to obtain an equivalent Simulink model of an existing physical system which is detected and develop appropriate control systems for the model which is obtained.* In order to model the system, mechanical limitations have been determined empirically and simulink model elements have been derived for these limits. The obtained PID control of the Simulink model with and without PSO has been compared and the output errors of the system have been compared. Reference tracking error of 5-7% is reduced to 0,1-0,2%.

Key Words : Particle Swarm Algorithm, PID Control Algorithm

1.GİRİŞ

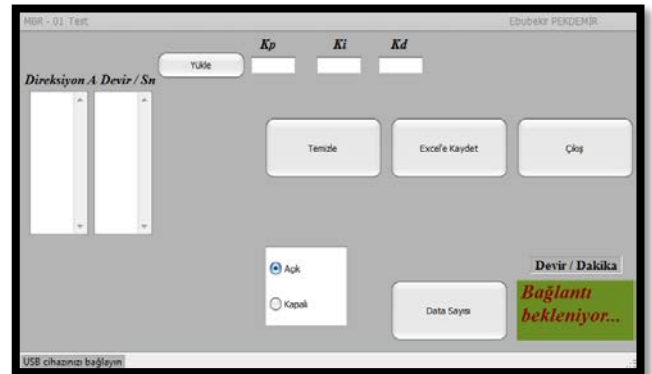
Parçacık Sürü Optimizasyonu (PSO), sürü halinde hareket eden balıklar ve böceklerden esinlenerek Kenedy ve Eberhart [1] tarafından geliştirilmiş bir optimizasyon yöntemidir. Temel olarak sürü zekâsına dayanan bir algoritmadır. Sürü halinde hareket eden hayvanların yiyecek ve güvenlik gibi durumlarda, çoğu zaman rasgele sergiledikleri hareketlerin, amaçlarına daha kolay ulaşmalarını sağladığı görülmüştür[2].

Esasında PSO, her bir elemanına “parçacık” adı verilen ve bir önceki elemanın değerine göre değer ataması yapılan matrisler yardımıyla, kontrol katsayılarına katsayı olarak getirilerek, kalıcı ve/veya geçici hal kontrol hatalarını minimize etme yöntemidir.

Bu çalışmada donanımsal ve yazılımsal olmak üzere pek çok materyal kullanılmıştır.

MBR-01 mobil robotunun modellenmesi yapılırken, sisteme haricen takılmış PIC18F4550 mikrodenetleyicili

bir haberleşme kartı ve bilgisayar(Windows tabanlı bir Tablet PC, ReederW7IC), Visual C# ve Dev Express ile hazırlanmış USB V2.0 destekli bir arayüz programı aracılığı ile sistem üzerinden gerekli parametrelerin okunması amacıyla haberleştirilmiştir.



Şekil 1 MBR-01 haberleşme ve kontrol arayüz prog.

Haberleşme kartı, MBR-01 üzerinden, aracın direksiyon açısı bilgisi, aracın tekerlek dönüş hız bilgisi, araç itiş ve direksiyon dönüş akım bilgisi okunarak USB kablo aracılığı ile tablet bilgisayara aktarılması, tablet bilgisayardan gönderilen direksiyon açısı ve tekerlek dönüş hızı taleplerini gerçekleştirme görevlerini yerine getirmektedir.

Tablet PC’de yer alan arayüz programı ise kullanıcının MBR-01’e hız ve yön bilgisi gönderme, USB’den veri okumaya başlama-durdurma, USB’den gelen verileri “.xlsx” uzantılı dosyalara ayrı ayrı kaydetme işlemlerini yapabileceği özelliği vardır.

2. SİSTEM ÖZELLİKLERİ

Modellemesi ve kontrolü yapılmak istenen MBR-01 mobil robotunun genel özellikleri:

- Kaportasız ağırlığı=11 kg.
- Düz yolda yaklaşık 10 kg yük taşıyabilme.
- Manuel kontrolde maksimum hız=89 metre/dakika.
- Tüm enerji gereksinimi 3 adet 6 Volt akü ile karşılanabilmekte.
- 45 derecelik açılarda dönebilen direksiyon mekanizması.
- Ön amortisman sistemi. Hareket sisteminin gerçek araçlardaki gibi arka dişli ve diferansiyel birimleri ile yapılması.

Mobil robotun ilk tasarım aşamasında gerçek bir araç ölçüleri referans alınarak 70cm uzunluğunda 30cm genişliğinde mekanik şase dizayn edilmiştir.

Tasarlanan mekanik sistemi hareket ettirecek şekilde maksimum 1600 devir/dakikaya sahip 36V DC motor kullanılmıştır[3].

Modelleme yapılmadan önce, MBR-01 mobil robotunun orijinal donanımında bulunan RF haberleşme modülü, kamera gibi bazı çevresel bileşenleri devre dışı bırakılmıştır.

MBR – 01 mobil robotunun direksiyon açısı, direksiyon mekanizmasına monte edilmiş 10 k Ω ’luk bir potansiyometre yardımıyla tespit edilmektedir.

Aracın sol ön tekerine ait döner mekanizmaya eş merkezli olarak yerleştirilmiş olan potansiyometrenin 1 ve 3 no’lu uçlarına 5V’luk DC gerilim uygulanmış ve 2 no’lu ucundan “Haberleşme Kartı”nın ADC birimine bağlantı yapılmıştır. Potansiyometrenin orta ucundan, 90°’lik dönüş açısına sahip DC servo motorun sıfır pozisyonundayken 5 k Ω ’luk bir direnç değeri ölçülmektedir. Bu durum, orta uç ile GND arasında 2,5 V gerilim meydana getirir. Ayrıca potansiyometrenin 1800°’lik dönüş yapabildiği göz önünde bulundurulduğunda, dc servo motorun dönüş açısı aralığı,

potansiyometrenin dönüş aralığını da sınırladığı kolaylıkla anlaşılabilir. O halde potansiyometrenin orta ucundan okunacak direnç aralığı 2,5 k Ω – 7,5 k Ω , gerilim aralığı ise 1,25 V – 3,75 V olacaktır. ADC biriminin dijital çıkış aralığı 256 – 768 olduğu anlaşılır. Direksiyon açısındaki 1°’lik açı değişimi, dijital dönüşüm değerinde yaklaşık 6 birimlik bir değişime denk olduğu görülmektedir.

Hızlanma donanımı olarak 1 adet DC motor, 1 adet epoksi diferansiyel dişli takımı ve bu dişli takımına eş merkezli arka dingil(aks), arka dingile bağlı ve yarıçapı 5 cm olan iki adet tekerlek mevcuttur.

Diferansiyel dişli, araç viraja girdiğinde, virajın dış kısmında kalan tekerleklerin, iç kısmında kalan tekerleğe göre, dönüş açısı ile orantılı olarak daha fazla tur atmasını sağlayarak, dönüşlerde aracın savrulmasını engeller ve daha güvenli yol tutuş imkanı verir. Aracın ani hızı ise $V_{\text{araç}} = V_{\text{rpm}} \times 2\pi \times r_{\text{tekerlek}}$ formülü ile hesaplanır. V_{rpm} , arka aksın dakikadaki dönme sayısıdır.

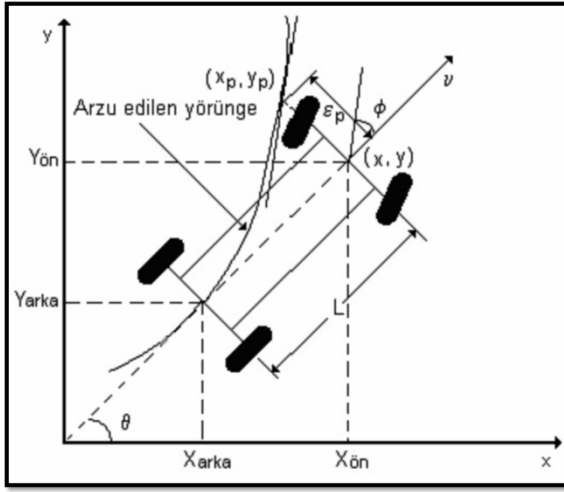
Elde edilen veriler doğrultusunda, MATLAB Simulink eşdeğer modelinin elde edilebilmesi için sisteme ait başlangıç koşulları ve kinematik denklemlerinin tespit edilmesi gereklidir:

n_m Hız motorunun maksimum devir sayısı	1600 d/dk
Z_m Hız motor milinin diş sayısı	8
Z_d Diferansiyel dişlisinin diş sayısı	90
n_a Maksimum aks devir sayısı	284 d/dk
r_t Tekerlek yarıçapı	10 cm
R_d Kendi eksenine etrafında dönme yarıçapı	298 cm
K Sürtünme katsayısı	0,37
J Hız motorunun atalet momenti	0,01 kg m ²
b Hız motoru mekanik iç sürtünme sabiti	0.1 Nm(sn)
K_e Hız motoru elektromotor kuvvet sabiti	0,01
K_t Hız motoru tork sabiti	0,01
R Hız motoru sargı direnci	0,88 Ω
L Hız motoru sargı endüktansı	0,65 H

3. KİNEMATİK DENKLEMLER

Mobil robotların kontrol işlemlerinde kinematik ve hareket denklemleri kullanmak karmaşık bir çalışmayı gerektirmektedir. Fakat aracın belirli bir zeminde düzgün hareketi söz konusu ise 2 boyutlu basitleştirilmiş bir model türetilir.

Tekerlek yapısı Şekil – 2’de gösterilen mobil robotun kinematik modelini elde etmek için ileri hız ve dönme hızlarını ve bunların türevlerini katarak (x,y, θ, v_i, v_d) dinamik konfigürasyon kavramlarından faydalanılabilir[5].



Şekil 2 Mobil robotun koordinat düzlemindeki konumu

Şekil 2’de gösterildiği gibi mobil robotun arka ve ön tekerleklerin merkezlerini baz alarak uzunluğu L ile ifade edilmiştir. Gerçek yörünge ile mobil robotun o anki yörüngesi arasındaki açı (θ), ön tekerleğin dönme açısını (ϕ), mobil robotun ileri hızı (v_i), dönme hızı (v_d) ve pozisyon hatası (ϵ_p) olarak ifade edilmiştir. Mobil robotun yörüngeden sapmadığı kabul edilerek, hareket denklemi Denklem 3.1 ile ifade edilir:

$$x' \cdot \sin(\theta) - y' \cdot \cos(\theta) = 0 \quad (3.1)$$

Anlık oluşan pozisyon hata denklemini ise Denklem 3.1 ile ifade edilebilir:

$$\epsilon_p = \sqrt{(X_p - X)^2 - (Y_p - Y)^2} \quad (3.2)$$

Buna göre x ve y hareket denklemleri:

$$x' = v_i \cdot \cos(\theta + \phi) = v_x \quad (3.3)$$

$$y' = v_i \cdot \sin(\theta + \phi) = v_y \quad [6] \quad (3.4)$$

$$\theta' = \frac{v_i \tan \phi}{L} \quad (3.5)$$

$$v_i \cos(\theta + \phi) = v_i \cos \theta \cdot \cos \phi + v_i \sin \theta \cdot \sin \phi \quad (3.6)$$

$$v_i \sin(\theta + \phi) = v_i \sin \theta \cdot \cos \phi + v_i \cos \theta \cdot \sin \phi \quad (3.7)$$

Yukarıdaki denklemler kullanılarak mobil robotun hareket denklemleri, hızı (v_i) ve dönme açısı (ϕ) parametrelerine bağlı olarak durum uzayında veya direkt integraller alınarak çözülebilir[3].

$$x = x_0 + \int_{t_0}^t [v \cos(\theta) \cdot \cos(\phi) + v \sin(\theta) \cdot \sin(\phi)] dt \quad (3.9)$$

$$y = y_0 + \int_{t_0}^t [v \sin(\theta) \cdot \cos(\phi) + v \cos(\theta) \cdot \sin(\phi)] dt \quad (3.10)$$

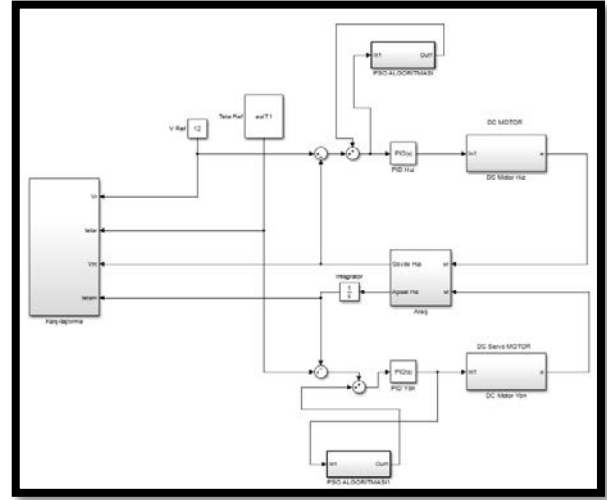
$$\theta = \theta_0 + \int_{t_0}^t \frac{v_i}{L} \tan(\phi) dt \quad (3.11)$$

4. SIMULINK MODELİNİN OLUŞTURULMASI

İlk olarak başlangıç koşullarını içeren “baslangic” isimli bir m-file oluşturulmuştur. Daha sonra da simülasyonda uygulanmak üzere, referans hız bilgisi

içeren, “referans” isimli bir matris tanımlanmıştır. Bu ön hazırlıkların ardından sistemin MATLAB Simulink modellemesine geçilmiştir.

Şekil – 3’te, tasarlanan ve kontrol edilen sistemin genel yapı eşdeğeri gösterilmiştir. Bu modelde MBR – 01’e ait 3 blok, referans girişlerine ait 2 blok, kontrolöre ait 4 blok ve ölçümlerin yapıldığı 1 blok olmak üzere toplam 10 blok bulunmaktadır.



Şekil 3 Sistemin MATLAB Simulink Modeli[4]

5. PARÇACIK SÜRÜ ALGORİTMASI (PSA)’NİN UYGULANMASI

Parçacığın elde ettiği en iyi çözümü sağlayan koordinatlar pbest, popülasyonda tüm parçacıklar için o ana kadar elde edilen en iyi çözümü sağlayan koordinatlar ise gbest olarak adlandırılır. Her iterasyon için pbest ve gbest bulunduktan sonra parçacığın konumu ve hızı (3.11.) ve (3.12.)’ye göre güncellenir[4].

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot \text{rand}_1^k \cdot (pbest_i^k - x_i^k) + c_2 \cdot \text{rand}_2^k (gbest^k - x_i^k) \quad (3.11.)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (3.12.)$$

“c₁” ve “c₂”; öğrenme faktörleri olup, parçacığın hız ve konum koordinatını en iyi değerlere (pbest ve gbest) yaklaştırır.

“rand₁” ve “rand₂” ise 1’den küçük pozitif rasgele katsayılarıdır.

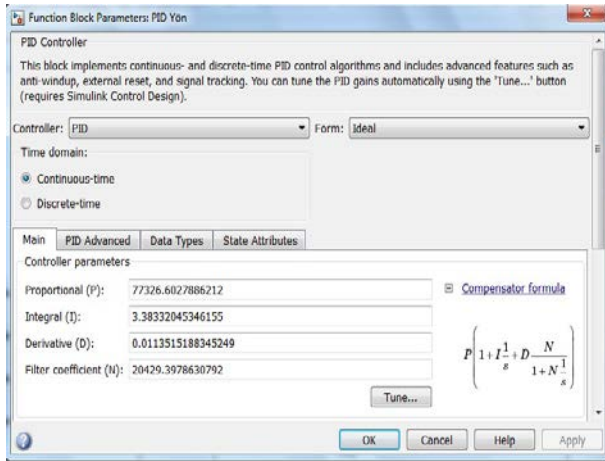
“k”; iterasyon sayısını simgeler. Diğer bir deyişle, en uygun parçacığın koordinatını bulma arayışının kaç tur

devam edeceğini göstermektedir.

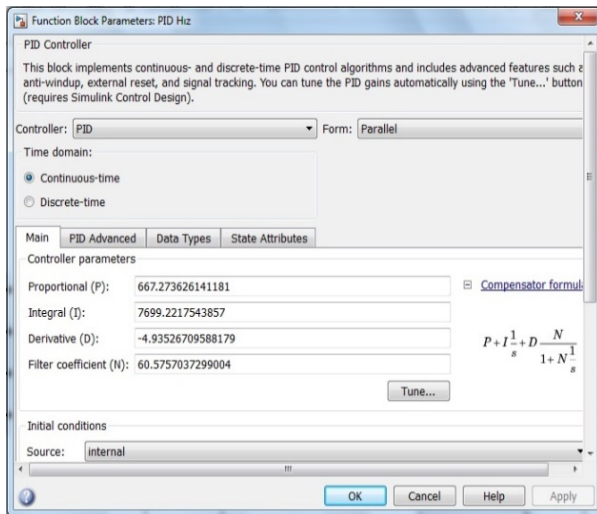
“w”; eylemizlik ağırlığıdır. Seçilecek “w” değeri 1’den küçük ve pozitif olacak şekilde her iterasyonda, iterasyon sayısına bağlı olarak azaltılmalıdır. Bu sayede sürüdeki bir sonraki parçacığın hızı, önceki elemanın hız değerinden giderek daha az etkilenecek ve rasgele katsayılar olan $rand_1$ ve $rand_2$ ’nin rolü önem kazanacaktır.

Simülasyon sonunda, sistemin giriş/çıkış grafikleri çizdirilerek PSO öncesi ve sonrası durumları karşılaştırılmış, olumlu sonuç alınması neticesinde de PSA ile optimize haldeki K_p , K_d ve K_i katsayıları not edilmiştir.

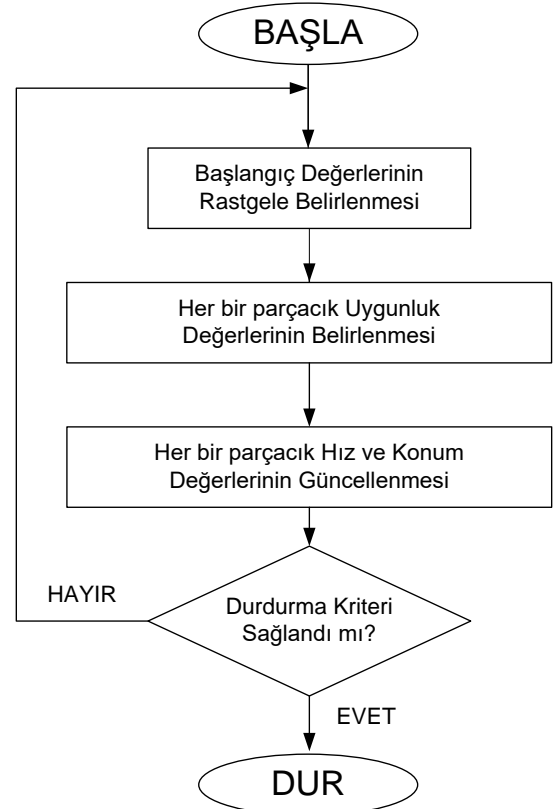
Matlab Simulink deneyleri ile elde edilmiş en ideal PID katsayıları, “MBR-01 Haberleşme ve Kontrol Arayüzü” yazılımındaki PID algoritmalarına, ilgili metin kutucuklarına yazmak suretiyle ilave edilmiştir.



Şekil 4. PSA Öncesi Hızlanma PID Parametreleri



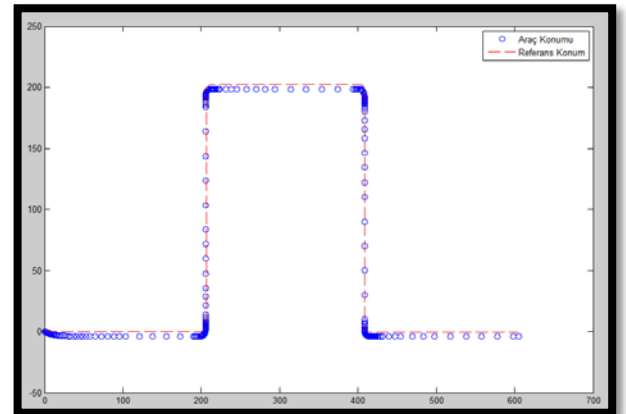
Şekil 5. PSA Öncesi Yön PID Parametreleri



Şekil 6: PSO Algoritması Akış Diyagramı

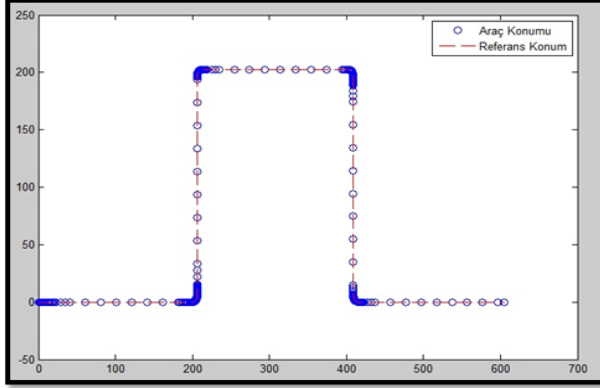
Aşağıda verilen şekillerde, parçacık sürü algoritması uygulanmasından önce ve sonra, sistemin referans konuma göre hareketi ve anlık hız ve yön değişimi incelenmektedir.

Şekil 7’de, MBR – 01’in, parçacık sürü algoritması uygulanmadan önceki PID kontrol algoritmasıyla referans konum takibi görülmektedir.



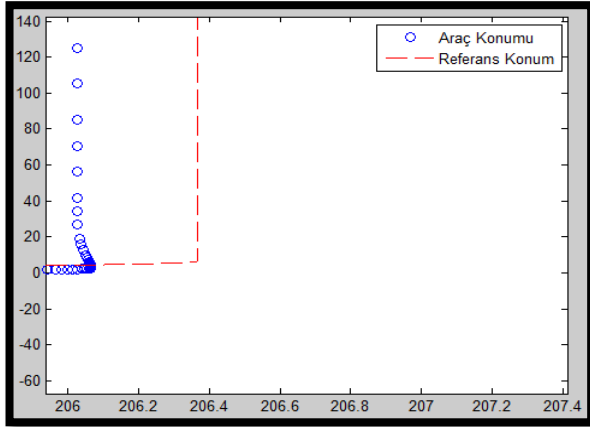
Şekil 7 PSA Öncesi Referans Konum Takibi

Şekil 8’de, parçacık sürü algoritması uygulandıktan sonra gerçekleştirilen konum takibi yer almaktadır.

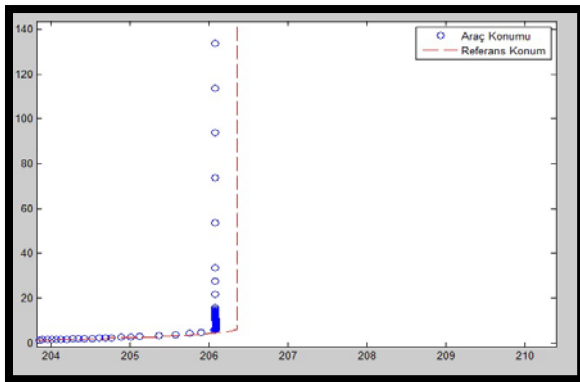


Şekil 8 PSA Sonrası Referans Konum Takibi

Şekil 9 ve Şekil 10’da, aracın PSO uygulanmasıyla birlikte ilk virajdaki konum takibindeki iyileşme gözlemlenmektedir.



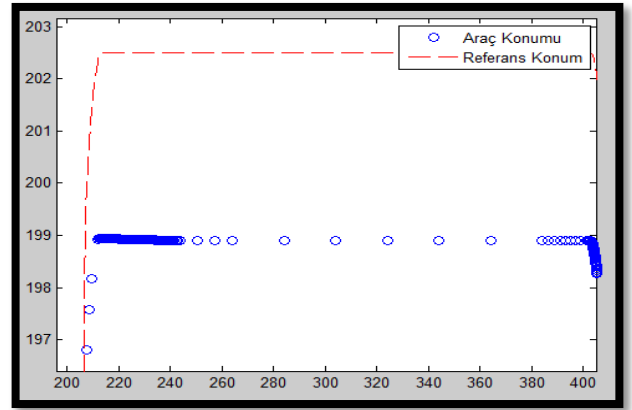
Şekil 9 PSA Öncesi İlk Viraj Alma



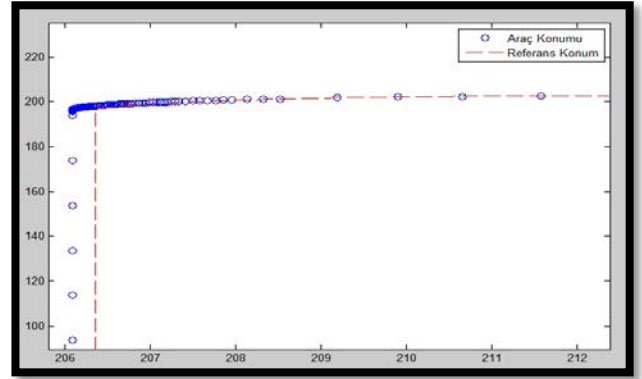
Şekil 10 PSA Sonrası İlk Viraj Alma

Buna göre MBR – 01 mobil robotu, parçacık sürü algoritması uygulandıktan sonra, referans girişlere daha hızlı ve aşımı daha düşük tepki vermekte olduğu, ayrıca kararlı hal hatasının daha da azaltıldığı sonucuna ulaşılır.

Şekil 11 ve Şekil 12’de ise, bu sonucun, kontrol evresinin her aşamasında gerçekleştiği kanıtlanmaktadır



Şekil 11 PSA Öncesi İkinci Viraj Alma



Şekil 12 PSA Sonrası İkinci Viraj Alma

Tablo 1 Anlık Referans Girişlerine Göre PSA'sız ve PSA'lı Sistem Çıktıları						
	R_x	A_{x1}	A_{x2}	R_y	A_{y1}	A_{y2}
Ölçüm-1	200,0027	203,6634	199,7135	0	0	0
Ölçüm-2	206,3667	210,0182	206,0683	8,4085	7,6679	8,3963
Ölçüm-3	206,3668	210,0183	206,0684	14,9074	14,0840	14,8858
Ölçüm-4	212,9885	216,6304	212,6805	202,6188	199,4024	202,3258
Ölçüm-5	214,4554	218,0952	214,1453	202,6188	199,4024	202,3258
Ölçüm-6	359,0935	362,5242	358,5743	202,4447	199,4024	202,3258
Ölçüm-7	404,1288	407,4944	403,5445	194,5072	191,3942	194,226
Ölçüm-8	409,1031	412,4615	408,5116	194,5072	191,3942	194,226
Ölçüm-9	457,4150	453,4651	457,4150	0,1420	-0,4932	0,1418

Tablo 2-a ve Tablo 2-b'de ise, sisteme verilen referans konum girişlerine göre, sistemin ürettiği X ve Y eksenlerine göre ürettiği konum hatalarını ve bu hataların referans konuma göre yüzdeleri ifade edilmiştir.

Tablo 2-a PSA'lı ve PSA'sız Sistem Çıktılarının Hata ve Hata Yüzdeleri (-X Düzlemi)					
	R_x	Hata x₁	Hata x₂	% Hata x₁	% Hata x₂
Ölçüm-1	200,0027	-3,6607	0,2892	-1,830325291	0,144598048
Ölçüm-2	206,3667	-3,6515	0,2984	-1,769423071	0,144596972
Ölçüm-3	206,3668	-3,6515	0,2984	-1,769422213	0,144596902
Ölçüm-4	212,9885	-3,6419	0,3080	-1,709904525	0,144608746
Ölçüm-5	214,4554	-3,6398	0,3101	-1,697229354	0,14459883

<i>Ölçüm-6</i>	359,0935	-3,4307	0,5192	-0,955377917	0,144586299
<i>Ölçüm-7</i>	404,1288	-3,3656	0,5843	-0,832803799	0,144582618
<i>Ölçüm-8</i>	409,1031	-3,3584	0,5915	-0,820917759	0,14458458
<i>Ölçüm-9</i>	457,4150	3,9499	0	0,863526557	0

$$\text{Hata } x_1 = R_x - A_{x_1},$$

$$\text{Hata } x_2 = R_x - A_{x_2},$$

$$\% \text{ Hata } x_1 = (\text{Hata } x_1 / R_x) * 100$$

$$\% \text{ Hata } x_2 = (\text{Hata } x_2 / R_x) * 100$$

Tablo 2-b PSA'lı ve PSA'sız Sistem Çıktılarının Hata ve Hata Yüzdeleri (-Y Düzlemi)					
	Ry	Hata y₁	Hata y₂	% Hata y₁	% Hata y₂
<i>Ölçüm-1</i>	0,0000	0,0000	0,0000	0	0
<i>Ölçüm-2</i>	8,4085	0,7406	0,0122	8,807754	0,159105
<i>Ölçüm-3</i>	14,9074	0,8234	0,0216	5,523431	0,153366
<i>Ölçüm-4</i>	202,6188	3,2164	0,2930	1,587414	0,146939
<i>Ölçüm-5</i>	202,6188	3,2164	0,2930	1,587414	0,146939
<i>Ölçüm-6</i>	202,4447	3,0423	0,1189	1,502781	0,059628
<i>Ölçüm-7</i>	194,5072	3,1130	0,2812	1,600455	0,146922
<i>Ölçüm-8</i>	194,5072	3,1130	0,2812	1,600455	0,146922
<i>Ölçüm-9</i>	0,1420	0,6352	0,0002	447,3239	-0,04055

$$\begin{aligned} \text{Hata } y_1 &= R_y - A_{y_1}, \\ \text{Hata } y_2 &= R_y - A_{y_2}, \end{aligned}$$

$$\begin{aligned} \% \text{ Hata } y_1 &= (\text{Hata } y_1 / R_y) * 100 \\ \% \text{ Hata } y_2 &= (\text{Hata } y_2 / R_y) * 100 \end{aligned}$$

Yapılan simülasyon sonrasında elde edilen sisteme ait kontrol parametreleri Tablo 3'te verilmiştir:

Tablo 3 Sistemin Hız ve Yön Kontrolüne Ait PID Katsayıları				
	Kp	Ki	Kd	N (D.F.)
PID Hız	1221,0295	8753,8947	-32,6654	21,4812
PID Yön	6183,61855	0,8871	0,2504	17377,7365

6. SONUÇ

Haberleşme Kartı'ndaki mikrodenetleyicinin ADC modülü dönüştürme hızı daha da artırıldığında, dönüştürülecek analog sinyalin ölçümü sırasında hatalar meydana gelmekte ve hatalı dönüşüm yapmakta olduğu gözlemlenmiştir.

Araç ağırlığının düşürülmesiyle, hızlanma işlevini yapan dc motorun yükü azaltılarak daha hızlı tepki vermesi sağlanacaktır.

Maliyetinin düşük olması sebebiyle pvc epoksi dişliler kullanılmıştır. Bu nedenle duruş/kalkış anlarında esnemeler meydana gelmektedir. Bunun yerine metal dişliler –tornada işlenmesinin kolay olması nedeniyle özellikle pirinç dişliler- kullanarak esneme miktarı azaltılır, referans takibinde tepki hızı artırılabilir.

Mikrodenetleyicileri tetikleme elemanı olarak 4 MHz'lik kristal osilatör kullanılmıştır. Mikrodenetleyicinin tüm çalışma hızı bu osilatörün hızıyla sınırlıdır. 4 MHz'lik yerine 20 MHz'lik bir osilatör kullanarak tüm ölçüm ve haberleşme işlemleri daha da hızlandırılarak, sistemin referans girişine cevabı da hızlanır.

Ayrıca ölçüm alınan kritik nokta sayısı artırılarak da daha detaylı hata analizi yapmak ve bu hataya göre kontrol katsayısı üretmek mümkündür. Ancak bu kez işlem sayısı arttığı için simülasyon süresinde uzamalar ve fiziksel kontrol esnasında ölçümlerin uzun sürmesine bağlı olarak kontrol işaretinin vaktinde gönderilememesinden kaynaklanan konum hataları meydana gelmektedir.

Simulink ortamında “Problem Solver” seçimi de katsayı üretiminde bir diğer faktördür.

7. KAYNAKLAR

- [1] Kennedy, J.; Eberhart, R. C., “Particle Swarm Optimization”, Proc. of the IEEE Int. Conference on Neural Networks, 4, 1942-1948, 1995.
- [2] M. Yasin ÖZSAĞLAM, Mehmet ÇUNKAŞ / POLİTEKNİK DERGİSİ, CİLT 11, SAYI 4, 2008 Yıldız Teknik Üniversitesi
- [3] Değişken Bir Yörünge Üzerinde Hız ve Konum Kontrolü, Mahit GÜNEŞ Doktora Tezi, MARMARA ÜNİVERSİTESİ / İSTANBUL – 2005.
- [4] Ozan Der, Revna Acar Vural, Tülay Yıldırım Parçacık Sürü Optimizasyonu Tabanlı Evirici Tasarımı Elektronik ve Haberleşme Müh. Bölümü
- [5] Kim Dong Hwa and Park Jin Ill: Intelligent PID Controller Tuning of AVR System using GA and PSO: Springer-Verlag Berlin Heidelberg: ICIC 2005, Part II.
- [6] Liu, K.; Lewis, F.L.: “Fuzzy Logic-Based Navigation Controller for an Autonomous Mobile Robot”, IEEE Transactions on Control Systems Technology, (1994).