

Ters Kinematik Problemine Sezgisel bir Yaklaşım

Amir Rahmanparast¹, Mir Mohammad Reza Alavi Milani²

¹ Karadeniz Teknik Üniversitesi, Makine Mühendisliği Bölümü, Trabzon

² Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, Trabzon

rahmanparast@ktu.edu.tr, milani@ktu.edu.tr

Özet: Ters kinematik çözümü robotik araştırma alanında büyük bir sorun olduğunu ve son yıllarda çok önemli bir araştırma alanı haline gelmiştir. Ayrıca, Genetik Algoritma (GA), Parçacık Sürü Optimizasyonu (PSO) ve Harmony Arama (HS) gibi Herustic yöntemlerinin kullanımı birçok optimizasyon problemi ve çözümü doğrusal olmayan problemleri ele alarak giderek artmaktadır. Bu kağıt PUMA 560 robot manipülatör için PSO dayalı bir ters kinematik çözüm yöntemi sunuyor. Bu sorun, bir optimizasyon problemi haline ve sonra global optimizasyon özelliği yararlanarak altı eklemlili robot manipülatör optimal bir değer elde etmek için PSO algoritması kullanılmıştır. Simülasyon sonuçları önerilen yaklaşım pozisyon doğruluğu ve çözüm süresi açısından da üstün performanslı olduğunu göstermektedir

Anahtar Sözcükler: Ters Kinematik, Sezgisel Yöntem, optimizasyon, robot manipülatör.

A Heuristic Approach for Inverse Kinematics

Abstract: The inverse kinematics solution is a major problem in robotic research area and has become very important research area in recent decades. Also, the use of Heuristic methods such as Genetic Algorithm(GA), Particle Swarm Optimization (PSO) and Harmony Search (HS) has been increasing to handle many optimization problem and solving nonlinear problems. This paper presents an inverse kinematics solution method based on PSO for PUMA 560 robot manipulator. This problem have transformed into an optimization problem and then used the PSO algorithm to obtain an optimal value of six jointed robot manipulator by taking advantage of the global optimization property. The simulation results show that proposed approach has been performane in terms of both position accuracy and the solution time.

Keywords: Inverse Kinematic, Heuristic Method, Optimization Problem, Robot Manipulator.

1. Giriş

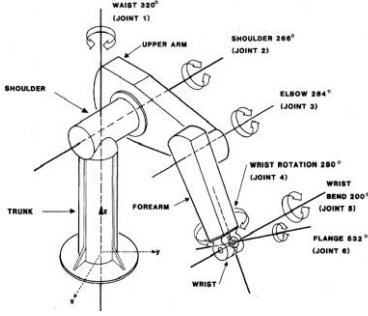
Günümüze kadar büyük aşama kaydeden robotlar, madde taşınması (konveyör sistemler), boyacılık, kanak endüstrisi tıp vs. gibi birçok farklı sektörde kullanılmaktadır. Robotik bilimi birçok bilim dalını içine alan disiplinler arası bir terbiyedir bunların birçoğu robot kinematiği dahil tartışılabilir. Kinematik çalışmaları (Kinesiyoloji) robot hareketi genellikle geometri argüman ifade eder. Ancak, çeşitli yöntemlerle bu konu için geliştirilmiştir. Sorunları çözmek için bu tür yöntemler arasında, yakın zamanlarda kabul edilen algoritmalar ve yapay zeka teknikleri dayalı yöntemler gibi bilgisayar bilimine bağlı kullanılır[1,2,3,5,6,7]. Son zamanlarda

robotik biliminde Hızlı ve doğru kinematik hesaplamaları çözüme optimize optimizasyon yöntemleri olarak sürü zeka yöntemleriyle yaygın olmuş.[4,8,9,10]

PUMA 560 genel robot türü hakkında kısaca bu yazının ikinci bölümünde tanıdık özelliklerini tanıtacağız. Üçüncü bölümde, optimizasyon teknikleri kolektif zeka ile ilgili olarak PSO yöntemini tartışacağız, ve görüşecek dördüncü bölümde *PUMA 560* robot için kinematiği kontrol edeceğiz ve beşinci bölümde *PUMA 560* robot için ters kinematiği bulmayı önerilen yeni yöntem analizi edilecektir. Son olarak, pratik sonuçları ile altıncı bölümde yaklaşım tartışılacak.

2. PUMA 560 SİSTEMİ MODELİ

PUMA 560 robot kolu 6 serbestlik derecesine sahiptir ve altı karmaşık eklemleri vardır. Oysa ilk üç eklemler uç işlevci kısmına yerleştirilmiştir, aşağıdaki üç eklemler uç işlevci yönelimi sağlamak içindir. PUMA 560 kontrolü şekil.1 de gösterilmiştir.



Şekil.1. PUMA560 Robot kolu

J. NO	a _i (meters)	d _i (meters)	RANGE Degree)
1	0	0	-160o +160o
2	0	0	-245o +45o
3	0.4318	0.1491	-45o +225o
4	-0.0203	0.4331	-110o +170o
5	0	0	-100o +100o
6	0	0	-266o +266o

Tablo 1. Puma 560 Parametre değerleri

Denavit-Hartenberg (D- H) parametreleri PUMA 560 için tablo.1 de verilmiştir. Denklem 1 çarpma dönüşümlerin bütün komşu çerçevelerini verir. Bu denklem önerilen yaklaşımın uygunluk fonksiyonu hesaplamak için kullanılacak

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (1)$$

${}^i T_{i+1}$, i. bağlantı donuşum matrisidir. ${}^0 T_6$ matrisi her hangi altı eklem açıları için Kartezyen koordinat üretir. Önerilen yaklaşımın maliyet fonksiyonu¹ elde edilmiş ve hedef noktaları arasındaki Kartezyen uzayda Öklid mesafesi olduğundan, ${}^0 T_6$ maliyet fonksiyonu olarak elde edilen noktası Kartezyen koordinat hesaplamak için de kullanılabilir Denklem 1 sonucu eşitliğinde verilen bir 4x4 donuşumu, deklemler 2 ve denklem 3 te verilmiştir.

$${}^0T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Öyle ki

$$\begin{aligned} r_{11} &= s_1 [c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] + s_1(s_4c_5c_6 + c_4s_6), \\ r_{21} &= s_1 [c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1(s_4c_5c_6 + c_4s_6), \\ r_{31} &= -s_{23}(c_4c_5c_6 - s_4c_6) - c_{23}s_5c_6, \\ r_{12} &= c_1 [c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] + s_1(c_4c_6 - s_4c_5s_6), \\ r_{22} &= s_1 [c_{23}(-c_4c_5s_6 - s_4c_6) + s_{23}s_5s_6] - c_1(c_4c_6 - s_4c_5s_6), \\ r_{32} &= -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6, \\ r_{13} &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5, \\ r_{23} &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5, \\ r_{33} &= s_{23}c_4s_5 - c_{23}c_5, \\ p_x &= c_1[a_2c_2 + a_3c_{23} - d_4s_{23}] - d_3s_1, \\ p_y &= s_1[a_2c_2 + a_3c_{23} - d_4s_{23}] + d_3c_1, \\ p_z &= -a_3s_{23} - a_2s_2 - d_4c_{23}. \end{aligned} \quad (3)$$

3. Parçacık sürü optimizasyon algoritması

PSO algoritması parçaları kolektif davranışı için Kennedy tarafından sunulan sosyal bir arama algoritması ve 1995 yılında ilk kez bir optimizasyon yöntemi [11] olarak tanımlandı. Bu yöntem guruplarında yiyecek ararken kuşların toplu hareketi esinlenilmiştir. Sınıflandırma doğruluğu ve seçilen öznelik vektörlerin uzunlukları parçacık hareketleri için değerlendirme kriterleri olarak kabul edilmektedir .PSO yöntemde ,akış parçacıklar arama alanı kendi pozisyonunda bir değişiklik onların deneyim ve bilgi ve komşuları tarafından etkilenir, bu nedenle diğer sürüsü parçacıkların konumu diğer parçacıkların arama etkiler. Bu sosyal davranış modelleme sonucunda parçacıkların başarılı alanları doğru hareket bir arama işlemidir. Sürüsü parçacıklar birbirlerine öğrenmek ve elde edilen bilgiye dayalı en iyi komşuları gidin. PSO'nun temelinde her an her parçacık zaten yerleştirilmiş olan en iyi konumda ve bütün komşuları iyi pozisyona göre arama uzaydaki konumunu ayarlar olmasdır.

İlk olarak, bileşenler bir dizi rastgele sorununa muhtemel cevap olarak oluşturulur, daha sonra daha iyi durumlarda grup hareketleri, belirli bir desen ile bileşen kayma birleştirerek içine taşımak için elde edilir. Bir durum bu pozisyon değişikliği iki faktöre dayanır:

¹ Cost function

1. Önceki hareketlerin karşılaşılan herhangi bir bileşenin en iyi durumu;

2. Nüfus içinde en iyi bileşenin durumu.

Arama işleminin sonunda, tüm parçacıkların bir optimum noktada birleştirilsin. Her parçacık problemi ve bu parametreler ile belirlenen arama uzayının boyutu gibi birçok parametre vardır. Her parametre için parametre sayısının artırılması, bir boyut boşluk aramak eklenecektir. Sorunun, arama alanını varsayılarak n-boyutlu, nüfusun i 'inci molekülün durum, bir n-boyutlu bir vektör $X_i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n})$ ile gösterilebilir ayrıca, parçacık ait olan vites $V_i = (v_{i,1}, v_{i,2}, v_{i,3}, \dots, v_{i,n})$ ve parçacığın en ziyaret edilen durum, bir n-boyutlu bir vektör P ile verilebilir ve $P = (p_{i,1}, p_{i,2}, p_{i,3}, \dots, p_{i,n})$. Bu nedenle, bir $n+1$ aşamasında parçacık hareketi eşittir (denklem 4):

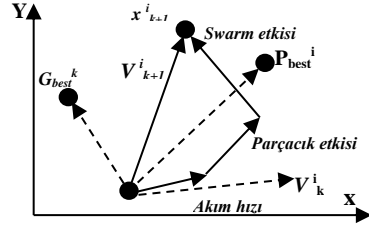
$$V_{i,j}^{(n+1)} = wV_{i,j}^{(n)} + C_1u_1(P_{best,i,j} - X_{i,j}^{(n)}) + C_2u_2(g_{best,j} - X_{i,j}^{(n)}) \quad (4)$$

g_{best} nüfusu en iyi parçacık ve değeri herhangi bir aşamasında ortaya nerede, $i = 1, 2 \dots m$ parçacıkların sayısı m , $j = 1, 2, \dots, n$, n sorunun boyutu olduğu, W ağırlık faktörü C_1 ve C_2 ivme sabitleri, x_i , $j(n)$ vardır, eylemsizlik ağırlığı faktörü olan bir parçacığın güncel pozisyonunuz, u_1 ve u_2 0 ile 1 arasında rastgele değerler vardır. Ve bu aşamada bir parçacığın yeni durumu denklem 5 gibidir.

$$X_{i,j}^{(n+1)} = x_{i,j}^{(n)} + v_{i,j}^{(n+1)} \quad (5)$$

Bilişsel ve sosyal faktörler olarak bilinen Katsayıları C_1 ve C_2 , bir şekilde bir sorun dayalı seçilebilir $C_1 + C_2 < 4$. Uygun bu değerler seçilmesi, hızlı ve uygun yaklaşmalara yardımcı olabilir, ancak olmalı Yanlış seçimler etkileri olabileceğini kaydetti. C_2 daha katsayısı C_1 büyük görülürse Örneğin, algoritma Tepesi Tırmanma algoritması benzer olacaktır. Ve eğer C_2 'i arttırsak erken yakınsamayı elde ederiz. Genel olarak eğer C_1 büyükse parçacıklar bağımsız ve dağınık aranmalı, eğer C_2 büyükse parçacıkların kendi kolektif durumu tutulur. Bu ikisi arasında bir denge oluşturmak için çalışılmalıdır. Pozisyon güncellemeleri için hız vektörü diyagramı

Şekil 2 de gösterilmiştir.



Şekil 2. Pozisyon güncellemeleri hız vektörü gösterimi

4. Robot kinematığının çözümlemesi

Kinematik robotu çözmekten hedefimiz robotun konumu ve bitiş noktalarını bulmaktır. Bu makalede ters kinematik ve ileri kinematığın PUMA 560 robotunda kontrol edeceğiz.

4.1. İleri kinematığının çözümü

Bu yöntemde açıları ve eklemlerin koordinatlarına sahip olduğumuzdan, robotun bitiş noktasını uzayda hangi noktada olduğunu belirliyoruz.

Genelde bu yöntemde robotun her bir eklemi için dönüşüm matrisi bulduktan sonra eklemin dönüşüm matrisine çarpılırız ve toplam dönüşüm matrisi buluruz.

Örnek: Eğer A ekleminin dönüşüm matrisi T_A ile ve B ekleminin dönüşüm matrisi T_B ile elde edilirse AB ekleminin dönüşüm matrisi denklem 6 da bulunur

$$T_{AB} = T_A \cdot T_B \quad (6)$$

Şekil 3. örnek olarak iki eklemli robot kolunun dönüşüm matrislerin göstermektedir.

$$T_A = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$T_B = \begin{bmatrix} r'_{11} & r'_{12} & r'_{13} \\ r'_{21} & r'_{22} & r'_{23} \\ r'_{31} & r'_{32} & r'_{33} \end{bmatrix}$$

$$T_{AB} = T_A \cdot T_B = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

Şekil 3. Örnek dönüşüm matrisler

4.2. TERS KİNEMATİĞİN ÇÖZÜMÜ

Bu yöntem konum koordinatlarını sahip olduğumuzdan robotun bitiş noktasını dönüş yaptığında eklemlerin koordinatları ve açılarını bulmak içindir. Ters kinematiği çözmek için farklı yöntemler bulunmaktadır.

Ters kinematiğin çözümü ileri kinematiğinden zor olduğu açık bir gerçektir. Bu makalede suru zeka optimizasyon yöntemini kullanarak PUMA 560 robotunda istediğimiz değerleri (eklemlerin açıları) çok hızlı ve net bir şekilde tasarım yapabiliriz.

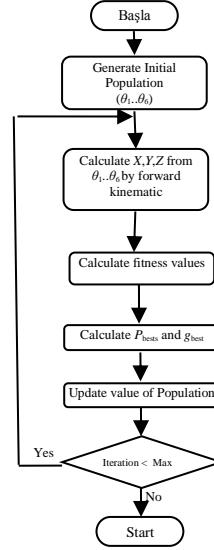
5 - Önerilen yöntem,

5-1 Önerilen Algoritma

Önceki bölümde belirtildiğine göre $\theta_1, \dots, \theta_6$ açıları için optimal değerlerini bulmak için PUMA 560 robotu için ters kinematik problemini çözmek için Optimizasyon PSO yöntemini kullanırız. Bu unutulmamalıdır ki sorunu çözmek için PSO yöntemini ve uygunluk fonksiyonunu (fitness) kullanmak gerektirir. Bu durumda, ileri kinematik yöntemini kullanarak istediğimiz konuma uzak veya yakın olmasını uygunluk fonksiyonunu (fitness) kriter olarak bulunur. Bu fonksiyonla ilgili daha fazla bilgi gelecek bölümlerde verilecektir. Belirlediğimiz yöntemde öncelikle, bir ilk nüfus $\theta_1, \theta_2, \dots, \theta_6$ değerleri öğretildi, başlangıçta yapılması için rastgele seçilmiş imal edilebilir. Bununla beraber, her bir açı θ_1 rasgele seçme minimum ve maksimum değerleri için kriter tablosunda (tablo 1) dikkate alınmalıdır. Örneğin, θ_3 rasgele değerler $[-450^\circ +225^\circ]$ bir aralık içinde oluşturulması Tablo [1] içinde verilmektedir. Bu yüzden başlangıç nüfus değerleri θ_1 den θ_6 ya kadar öğretilir ve bu nüfusun sayısı soruyla düzetilebilir ve bu sayıyı arttıktça sorunun net bir çözüme ulaşılma ihtimali yükselir ama sorunu çözmek fazla zaman alır. İlk nüfusu oluşturduktan sonra, her bir parçacığın θ_i değerlerini uygulama fonksiyonunda (fitness) giriyoruz ve bu değerlerin sonucunu PSO algoritmasında kullanıyoruz ve her bir parçacığın uygunluk değerleri (fitness) P_{best} (parçacığın en iyi konum noktası) ,nüfusun bütün parçacıklarına göre elde edilir. Ayrıca

bulduğumuz değerleri kullanarak g_{best} (nüfusun içindeki en iyi konumdaki parçacık) hesaplayabiliriz. Gerekli değerleri hesapladıktan sonra, denklem 4 ve 5 kullanarak nüfusun bir sonraki adımı ortaya çıkar. Bu işlemi en uygun değeri (fitness) bulana kadar tekrarlayacağız.

Önerilen akış diyagramı algoritması şekil 4 de verilmektedir



Şekil 4. Önerilen yöntemin akış diyagramı

Algoritmanın sonunda, g_{best} (nüfusun en iyi parçacığı)değerlerini her bir parçacıkta θ_1 den θ_6 değerlerine optimal çözüm olarak kullanılır.

5-2 uygunluk fonksiyonu (Fitness)

Aslında bu fonksiyon belirlediğimiz noktanın mesafesini ve θ_i değerleriyle robot hareket etikten sonraki konum yerini belirler.

Dolayısıyla bu fonksiyonun geri sayısalı değeri mesafeyi ve istenilen parçacın değerini belirler. Tabi ki bu değer her ne kadar az olursa daha iyidir ve istenilen noktaya yakındır. Problemi çözmek için en yakın noktayı optimum noktası olarak kullanırız.

Ve bu fonksiyonda θ_1 den θ_6 (istenilen parçacın açıları) ve (x_1, y_1, z_1) istenilen nokta olarak k_i robot istenilen noktaya gitmek zorundadır.

Başlangıçta dönüşüm matrislerinin her bir

eklemlerinden toplam dönüşüm matrisini PUMA 560 robotu için buluruz ve fonksiyon bu şekilde çalışır.(denklem 1 gibi)

Sonra her bir parçacığın θ değerlerini robotun dönüşüm matrisinde yerleştiririz ve aynı ileri kinematik gibi robotun gidecek konum yerini bu θ değerlerini verdikten sonra belli olur.

Örnek: eğer (x, y, z) robotun ileri kinematikten sonra konum yeri olduğunu verirse bu durumda konum yerinin mesafesi istediğimiz yerden (x, y, z) denklem 7 den elde edilir.

$$distance = ((x-x')^2 + (y-y')^2 + (z-z')^2)^{1/2} \quad (7)$$

$distance$ değerini hesapladıktan sonra uygunluk fonksiyonunun değerini denklem 8 den hesaplaya biliriz.

$$Fitness_i = \frac{1}{1 + distance_i} \quad (8)$$

6.Sonuçlar

Bu makalede gelişmiş PSO kullanarak ters kinematiği hesaplayıp ve PUMA 560'ın kollarının açıları optimal değerlere arıyoruz ve sürü zekasını optimal olarak kullanıp ve bir hızlı ve net değerler için çalışmışız. Arama alanı bu konuda bir 6 boyutlu alandır ve eklem hareket açıklığı bu alanda her boyutta tablo 1 de verilmiştir. Bu konuyu çözmek için sonuçları Harmony Search [10] yöntemiyle tablo 2 'ile karşılaştırmışız.

	Önerilen yöntem	HS yöntemi
Hata	3.32E08	2.43E04
Zaman	0.0361	0.376

Tablo 2.Önerilen yöntem ve HS yöntemi

Tablo 2 yapılan yöntem HS yönteminden daha uygun ve daha iyi olduğunu gösteriyor

8. Kaynaklar

[1] S. Küçük and Z. Bingül, "The inverse kinematics solutions of fundamental robot manipulators with offset wrist", in *Proceedings of 2005 IEEE International Conference on Mechatronics*, Taipei,Taiwan, pp. 197-205.

[2] S. Tejomurtula and S. Kak, "Inverse kinematics in robotics using neural networks", *Information Sciences*, vol. 116, pp. 147-164, 1999.

[3] S. Alavandar and M. J. Nigam, "Inverse kinematics solution of 3DOF planar robot using ANFIS", in *Proceedings of 2008 International Conference on*

Computers, Communications and Control, vol. 3, Oradea, Romania, pp. 150-155.

[4] P. Kalra, P. B. Mahapatra, and D. K. Aggarwal, "On the solution of multimodal robot inverse kinematic functions using real-coded genetic algorithms", in *Proceedings of 2003 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, Washington DC, USA, pp. 1840-1845.

[5] C.S.G.Lee. "Robot arm kinematics", *dynamics and control, IEEE Trans. Comput*, 15(12)(1982) 62-80

[6] R.P. Paul, B. Shimpano, and G. E. Meyer. "Kinematic control equations for simple manipulators." *IEEE Trans, System Man, Cybern, SMC-11 (6)(1981) 449-455*

[7] R. Manseur, K.L. Doty. "A fast algorithm for invers kinematics analysis of robot manipulators". *International Journal of Robotics Research*, 7(3)(1988) 52-63

[8] L. T. Wang, C. C. Chen. "A combined optimization method for solving the invers kinematics problem of mechanical manipulators". *IEEE Transactions on Robotics and Automation*, 7(4)(1991) 489-499

[9] K. Y. Lee, M. A. El-Sharkawi, "Modern heuristic optimization techniques theory and applications to power systems". *IEEE Press, 445 Hoes Lane Piscataway, NJ 08854, (2008)*

[10] B. Durmuş, H. Temurtaş. A. Gün. "An inverse kinematics solution using particle swarm optimization". *6th International Advanced Technologies Symposium IATS'11, Elazığ, Turkey. (2011)*

[11] Russell Eberhart. "A New Optimizer Using Particle Swarm Theory". *Sixth International Symposium on Micro Machine and Human Science (1995)*