

Endüstriyel Bir Robot İçin İnsan-Makina Arayüz Programının Geliştirilmesi

Mahir Cihan Cengiz

Bilgin Kaftanoğlu
Prof. Dr.

Makina Mühendisliği Bölümü
Orta Doğu Teknik Üniversitesi
Ankara

Bu çalışmada, bir endüstriyel robotu kontrol eden bir yazılım tanıtılmaktadır. Bahsi geçen robot altı serbestlik dereceli olup, ODTÜ Bilgisayar Destekli Tasarım, Üretim ve Robotik Merkezi'nde imal edilmiştir. Kullanıcı, bu programla, robot kolunu çalışma hacmi içinde istediği pozisyon ve yönelime gönderebilir. Bu amaç için robotun ileri ve geri kinematik analizleri yapılmıştır. Robotun üç boyutlu görünümünü sağlamak için, programa simülasyon penceresi eklenmiştir. Ayrıca program, verilen noktalar için rota oluşturabilmektedir.

Anahtar Kelimeler: Endüstriyel Robot, Kinematik analiz, Rota oluşturma.

GİRİŞ

Günümüz teknolojisinde robotların yeri çok büyüktür. Özellikle üretim ve montaj hatlarında endüstriyel robot kolları kullanılır. Robotların kullanılmasındaki en büyük etken, robotların hızlı ve hassas olmalarıdır. Ayrıca insan için tehlikeli olabilecek bazı işlemlerde de robotlar kullanılır. Bu çalışmada altı eksenli bir endüstriyel robot için hazırlanan arayüz programı açıklanmıştır.

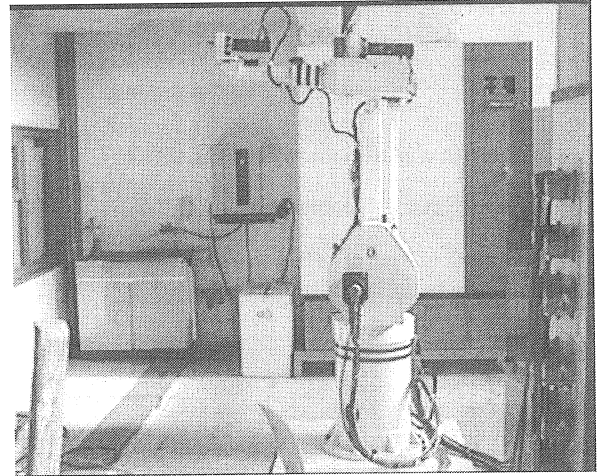
Bu çalışmada bahsi geçen robot, ODTÜ Bilgisayar Destekli Tasarım, Üretim ve Robotik Merkezinde üretilmiştir. Üretim aşamalarında tez öğrencileri görev almıştır. Robot, Şekil.1'de görülmektedir. Robot altı serbestlik derecelidir ve altı adet AC/Servo motora sahiptir. Bu motorlar Delta/Tau motor kontrol kartı ile kontrol edilmektedir.

Öncelikle robotun düz ve ters kinematik analizlerinin yapılması gerekmektedir. Bu analizler yardımıyla, uzay koordinatları ile eksen koordinatları arasındaki dönüşüm matrisleri bulunmuştur. Bu dönüşüm matrisleri yardımıyla, robotun, konum, hız ve ivme analizleri yapılmaktadır. Diğer kademe ise yörünge planlamasıdır. Robotun yapması istenen işleme göre, uç noktasının yörünge planlaması yapılacaktır. Bu planlama esnasında robotun hareketinin başlangıç ve bitiş noktalarının yanısıra, ara noktalar da verilebilmektedir. Bu sayede, robotun istenen yörüngede hareket etmesi sağlanmaktadır.

Geliştirilecek arayüz programında bu noktalar kolaylıkla belirlenebilecek ve daha sonra robot bu planlanan yörünge içinde kolaylıkla hareket ettirilebilecektir.

KİNEMATİK ANALİZLER

Bu bölümde robotun konum, hız ve ivme analizleri incelenecektir. Robotta altı adet dönel eklem bulunmaktadır ve bu eklemlerin hepsi birer serbestlik derecelidir.



Şekil 1. Robotun Genel Görünüşü

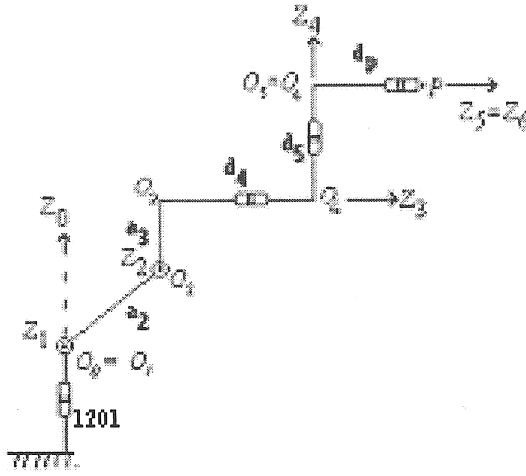
Kinematik analizler robotun eklem koordinatları ile uzay koordinatları arasındaki ilişkiyi açıklamak amacıyla yapılır. Bu analizleri düz ve ters olmak üzere iki kısma ayırabiliriz. Düz analizlerde, robotun eklem parametreleri (pozisyon, hız ve ivme) kullanılarak, robotun uç işlemcisinin uzay konum parametreleri hesaplanır. Ters analizlerde ise, bu sefer robotun uç işlemcisinin konum parametreleri kullanılarak eklem parametreleri bulunur. Robot programlama uygulamalarında ters kinematik

analizler önemlidir, çünkü robot kullanıcısı, robotun eklemlerini kontrol ederek, robotun uç işlemcisine komut etmektedir.

Bu analizleri yapabilmek için, bitişik uzuvlar arasındaki öteleme ve dönme parametrelerinin hesaplanması gerekmektedir. Bu amaçla Hartenberg ve Denavit (HD) tarafından ortaya çıkarılan yöntem kullanılmıştır. Bu yöntemde, matris metodu kullanılmıştır [4, 5].

HD Parametreleri ve Ön Bilgiler

Robotun HD parametrelerini bulmak için şematik model Şekil 2'ye verilmiştir.



(O0 noktasının yerden yüksekliği, mm.)

Şekil 2. Robotun Şematik Modeli

Bu modele göre, HD parametreleri aşağıdaki tablodaki gibi tanımlanabilir. (Çizelge 1)

Çizelge 1. Robotun HD Parametreleri

Eksen Numarası	a	d	α	θ
1	0	0	$\pi/2$	j.v. (Eklemler Değişkeni)
2	a_2	0	0	j.v.
3	a_3	0	$-\pi/2$	j.v.
4	0	d_4	$-\pi/2$	j.v.
5	0	d_5	$\pi/2$	j.v.
6	0	d_6	0	j.v.

Bu parametreler kullanılarak uzuvlar arasındaki dönme matrisleri şu şekilde tanımlanabilir. \tilde{C}_{i-1}^i uzuv (i-1) ve uzuv (i) arasındaki dönme matrisi olmak üzere,

$$\tilde{C}_{i-1}^i = e^{\tilde{u}_3 \cdot \theta_i} \cdot e^{\tilde{u}_1 \cdot \alpha_i} \quad (1)$$

şeklinde ifade edilebilir [4, 5].

Burada,

$$\bar{u}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \bar{u}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \bar{u}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

temel birim vektörleridir ve

$$\bar{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

gibi genel bir vektöre karşılık gelen vektörel çarpım matrisi şöyle tanımlanmıştır:

$$\tilde{n} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}$$

Böylelikle,

$$e^{\tilde{u}_1 \cdot \theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{bmatrix}$$

ve,

$$e^{\tilde{u}_3 \cdot \theta} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ifadeleri elde edilir. Yukarıdaki ifadeler kullanılarak, uzuvlar arasındaki dönme matrisleri bulunabilir.

KONUM ANALİZLERİ

i. Düz Konum Analizi

Robotun uç işlemcisinin konum ve yönelimi, robotun eksen açıları kullanılarak bulunur. Buna düz konum analizi denir. Bu analiz sembolik olarak yapılır. Bu analizden bulunan ifadeler, kinematik analizin diğer kısımlarında kullanılır. Robotik uygulamalarında genellikle düz konum analizi kullanılır, çünkü, uygulamalarda genellikle robotun uç işlemcisinin konumu ve yönelimi kontrol edilmek istenir. Bu analiz için öncelikle yönelimin bulunması gerekir, çünkü analizin devamında, yönelim bulunurken elde edilen ifadeler kullanılır.

Robotun uç işlemcisinin yönelimini bulmak için, uzuvlar arasındaki dönme matrisleri çarpılır, çünkü uzuv uzunlukları, robotun uç işlemcisinin yönelimini

etkilemez. Buna göre, robotun uç işlemcisinin yönelim matrisi şu şekilde ifade edilebilir;

$$\tilde{C} = \tilde{C}_0^6 = \tilde{C}_0^1 \cdot \tilde{C}_1^2 \cdot \tilde{C}_2^3 \cdot \tilde{C}_3^4 \cdot \tilde{C}_4^5 \cdot \tilde{C}_5^6 \quad (2)$$

Robotun uç noktasının konumu ise şu şekilde ifade edilebilir;

$$\begin{aligned} \bar{P} = & a_2 \cdot \bar{u}_1^{(2)} + a_3 \cdot \bar{u}_1^{(3)} \\ & + d_4 \cdot \bar{u}_3^{(3)} + d_5 \cdot \bar{u}_3^{(4)} + d_p \cdot \bar{u}_3^{(6)} \end{aligned} \quad (3)$$

yukarıdaki denklemlerde,

\bar{u}_1 ve \bar{u}_3 , yere bağlı eksen takımında x ve z yönündeki birim vektörler,

$\bar{u}_1^{(x)}$ ve $\bar{u}_3^{(x)}$, x uzvuna bağlı eksen takımında x ve z yönündeki birim vektörleridir.

$\bar{u}_n^{(x)}$ yere bağlı eksen takımında şu şekilde ifade edilebilir;

$$\bar{u}_n^{(x)} = C_0^x \cdot \bar{u}_1$$

Çözümü basitleştirmek için, O_5 noktasının O_0 noktasına göre konumu, bilek noktası konumu, \bar{R} olarak ifade edilebilir. Buna göre;

$$\bar{P} = \bar{R} + d_p \cdot \tilde{C} \cdot \bar{u}_3 \quad (4)$$

yukarıda \bar{R} vektörünün elemanları;

$$R_1 = A \cdot \cos(\theta_1) - d_5 \cdot \cos(\theta_4) \cdot \sin(\theta_1) \quad (5)$$

$$R_2 = A \cdot \sin(\theta_1) + d_5 \cdot \cos(\theta_4) \cdot \cos(\theta_1) \quad (6)$$

$$\begin{aligned} R_3 = & a_2 \cdot \sin(\theta_2) + a_3 \cdot \sin(\theta_{23}) \\ & + d_4 \cdot \cos(\theta_{23}) - d_5 \cdot \sin(\theta_{23}) \cdot \sin(\theta_4) \end{aligned} \quad (7)$$

yukarıdaki denklemlerde;

$$\begin{aligned} A = & a_2 \cdot \cos(\theta_2) + a_3 \cdot \cos(\theta_{23}) \\ & - d_4 \cdot \sin(\theta_{23}) - d_5 \cdot \cos(\theta_{23}) \cdot \sin(\theta_4) \end{aligned}$$

$$\theta_{23} = \theta_2 + \theta_3$$

Bu durumda, \bar{P} vektörünün elemanları ise aşağıdaki şekilde yazılabilir.

$$P_1 = R_1 + c_{13} \cdot d_p \quad (8)$$

$$P_2 = R_2 + c_{23} \cdot d_p \quad (9)$$

$$P_3 = R_3 + c_{33} \cdot d_p \quad (10)$$

Yukarıda c_{xy} , \tilde{C} matrisinin xy elemanıdır.

ii. Ters Konum Analizi

Ters konum analizi, robotun uç işlemcisinin konumu ve yönelimi verildiğinde, robotun bu konuma gelebilmesi için gereken eklem açılarının bulunmasıdır. Bu işlem için öncelikle robotun dönme ve öteleme matrisleri bulunur. İşlem basitliği için robotun uç noktasının yerine, denklemleri daha basit olan bilek noktası tercih edilir.

Bilek noktasının konum vektörü (\bar{R}) bulunduktan sonra, bu vektörün ve \tilde{C} matrisinin elemanları kullanılarak, eklem açıları bulunur. Bu işlem düz konum analizini tersidir.

Genelde, ters konum analizi, tamamen analitik yolla bulunur, fakat bazı robotlarda, robotun şekli itibariyle analitik metod uygulanamaz. Bu robotta, bilek noktasının konum denklemleri dört eklem açısına bağlıdır ve dönme matrisi ise bütün eklem açılarına bağlıdır. Bu nedenle yarı-analitik metod uygulanmalıdır [5]. Bu metodda, açılardan biri (θ_1) biliniyormuş gibi kabul edilip, konum denklemlerinden diğer üç açı (θ_2 , θ_3 , θ_4), biliniyormuş kabul edilen açı cinsinden yazılır. Daha sonra dönme matrisi ve bulunan açıların fonksiyonları kullanılarak, diğer açılar bulunur [1].

Öncelikle verilen uç nokta konumu ve yönelimi kullanarak, 8-10 numaralı denklemlerden, bilek noktası konumu bulunur:

$$R_1 = P_1 - c_{13} \cdot d_p \quad (11)$$

$$R_2 = P_2 - c_{23} \cdot d_p \quad (12)$$

$$R_3 = P_3 - c_{33} \cdot d_p \quad (13)$$

Bu denklemler kullanılarak;

$$R_1 = A_1 \cdot \cos(\theta_1) - A_2 \cdot \sin(\theta_1) \quad (14)$$

$$R_2 = A_1 \cdot \sin(\theta_1) + A_2 \cdot \cos(\theta_1) \quad (15)$$

elde edilir. burada;

$$\begin{aligned} A_1 = & a_2 \cdot \cos(\theta_2) + a_3 \cdot \cos(\theta_{23}) \\ & - d_4 \cdot \sin(\theta_{23}) - d_5 \cdot \cos(\theta_{23}) \cdot \sin(\theta_4) \end{aligned} \quad (16)$$

ve

$$A_2 = d_5 \cdot \cos(\theta_4) \quad (17)$$

14 ve 15 no'lu denklemlerden,

$$A_1 = R_1 \cdot \cos(\theta_1) + R_2 \cdot \sin(\theta_1) \quad (18)$$

$$A_2 = R_2 \cdot \cos(\theta_1) - R_1 \cdot \sin(\theta_1) \quad (19)$$

elde edilir. 17 nolu denklem kullanılarak θ_4 bir işaret belirsizliğiyle bulunabilir;

$$\theta_4 = \sigma_4 \cdot \cos^{-1}(A_2 / d_5) \quad (20)$$

burada, σ_4 (-) ya da (+) olarak seçilebilir.

Park konumunda, θ_4 'ün değeri -90 derecedir. Robot hedefe sağ taraftan yaklaşması isteniyorsa, bir başka deyişle, sağ el konfigürasyonu isteniyorsa, θ_4 dar açı olmalıdır, bu nedenle σ_4 pozitif seçilir. Diğer konfigürasyon için σ_4 negatif seçilir.

7 ve 16 numaralı denklemlerden;

$$\cos(\theta_2) = \frac{A_1 - (a_3 \cdot \cos(\theta_{23}) - d_4 \cdot \sin(\theta_{23}) - d_5 \cdot \cos(\theta_{23}) \cdot \sin(\theta_4))}{a_2} \quad (21)$$

$$\sin(\theta_2) = \frac{R_3 - (a_3 \cdot \sin(\theta_{23}) + d_4 \cdot \cos(\theta_{23}) - d_5 \cdot \sin(\theta_{23}) \cdot \sin(\theta_4))}{a_2} \quad (22)$$

elde edilir ve

$$\cos^2(\theta_2) + \sin^2(\theta_2) = 1$$

olduğu göz önüne alınarak, aşağıdaki denklem oluşturulabilir.

$$B_2 \cdot \sin(\theta_{23}) + A_2 \cdot \cos(\theta_{23}) + C_2 = 0 \quad (23)$$

burada,

$$\begin{aligned} A_2 &= 2 \cdot A_1 \cdot d_5 \cdot \sin(\theta_4) - 2 \cdot R_3 \cdot d_4 - 2 \cdot A_1 \cdot a_3 \\ B_2 &= 2 \cdot A_1 \cdot d_4 + 2 \cdot R_3 \cdot d_5 \cdot \sin(\theta_4) - 2 \cdot R_3 \cdot a_3 \\ C_2 &= A_1^2 + R_3^2 - d_5^2 \cdot \cos^2(\theta_4) + d_5^2 + d_4^2 + a_3^2 \\ &\quad - a_2^2 - 2 \cdot a_3 \cdot d_5 \cdot \sin(\theta_4) \end{aligned}$$

Bu eşitliğin çözümünden ise,

$$\theta_{23} = 2 \cdot \tan^{-1}(t) \quad (24)$$

bulunur. Burada yine bir işaret belirsizliği ile;

$$t = \frac{B_2 + \sigma_2 \cdot \sqrt{B_2^2 + A_2^2 - C_2^2}}{A_2 + C_2}$$

ve σ_2 (-) ya da (+) olarak seçilebilir.

Eğer σ_2 pozitif alınırsa, dirsek yukarı konfigürasyon seçilmiş olur. σ_2 negatif olursa, çözüm dirsek aşağı konfigürasyon için bulunmuş olur.

θ_{23} ($\theta_{23} = \theta_2 + \theta_3$) hesaplandıktan sonra, θ_2 , 21 ve 22 nolu denklemler kullanılarak şu şekilde bulunabilir;

$$\theta_2 = a \tan 2(\cos(\theta_2), \sin(\theta_2)) \quad (25)$$

$$\theta_3 = \theta_{23} - \theta_2 \quad (26)$$

θ_2 , θ_3 , θ_4 , θ_1 cinsinden bulunduktan sonra, θ_1 'in doğru değerini bulmak için dönme matrisi \tilde{C} kullanılır. 2 no'lu denklemden;

$$(\tilde{C}_0^1 \cdot \tilde{C}_1^2 \cdot \tilde{C}_2^3 \cdot \tilde{C}_3^4)^{-1} \cdot \tilde{C} = \tilde{C}_4^5 \cdot \tilde{C}_5^6 = \tilde{B} \quad (27)$$

elde edilir.

Matris işlemleri yapıldıktan sonra, matrislerin son elemanlarının eşitliğinden;

$$\begin{aligned} &\sin(\theta_1) \cdot (-c_{13} \cdot \cos(\theta_4) - c_{23} \cdot \sin(\theta_4) \cdot \cos(\theta_{23})) \\ &+ \cos(\theta_1) \cdot (c_{23} \cdot \cos(\theta_4) - c_{13} \cdot \sin(\theta_4) \cdot \cos(\theta_{23})) \\ &- c_{33} \cdot \sin(\theta_4) \cdot \sin(\theta_{23}) = 0 \end{aligned}$$

bulunur. Yukarıdaki denklemde bütün açılar θ_1 'in cinsinden yazılabildiğinden. Yukarıdaki fonksiyonun kökü, yani θ_1 'in doğru değeri, uygun bir sayısal yöntemle bulunabilir.

Bu işlemden sonra, θ_5 ve θ_6 , denklem 27'deki matrisin diğer elemanlarının eşitliğinden aşağıdaki şekilde elde edilir:

$$\theta_5 = a \tan 2(-b_{23}, b_{13})$$

$$\theta_6 = a \tan 2(b_{32}, b_{31})$$

Bu denklemlerde, b_{xy} 'ler \tilde{B} matrisinin elemanlarıdır.

iii. Konfigürasyon Kontrolü ve Tekil Konumlar

Ters konum analizindeki işaret belirsizlikleri nedeni ile robotun çözümünde birden fazla konfigürasyon vardır. Bu çözümde iki adet işaret belirsizliği vardır, bunlar;

- Sağ/Sol el konfigürasyonu,
- Dirsek yukarı/aşağı konfigürasyonu.

Sağ/sol el konfigürasyon seçimi, arayüz programında kullanıcıya bırakılmıştır. Bu konfigürasyon seçimi, çözümdeki 20 no'lu denklemde gösterilmiştir. Diğer konfigürasyon ise robotun tasarımcısı tarafından belirlenmiştir. Yapı ve fiziksel nedenlerden dolayı robot dirsek yukarı konumunda çalışmalıdır. Bu seçim ise denklem 24'te görülebilir.

Robotun konum analizinin bazı tekil konumları da vardır [1]. Birinci tekil konum ile robotun O_6 noktası ile Z_0 eksenini çakışırsa karşılaşılır. Bu durumda θ_1 açısı belirlenemez. İkinci tekil konum ile, robottaki O_6 noktası ile O_1 noktası çakışırsa karşılaşılır. Bu durumda ise θ_{32} açısı belirlenemez. Bu iki tekil konuma da, robotun genel çalışma alanı içerisinde ulaşamayacağından, bu tekil konumlar robot çalışırken bir sorun çıkarmaz.

Robotun en önemli tekil konumu, $\theta_5=0$ veya $\theta_5=\pm 180$ iken ortaya çıkar. Bu durumda θ_4 ve θ_5 ayrı ayrı çözülemez ve sadece $\theta_4+\theta_5$ değeri bulunabilir. Böyle bir belirsizliğin ortaya çıkması için hareket planlaması yapılırken $\theta_5=0$ olmamasına dikkat edilmelidir.

HIZ VE İVME ANALİZLERİ

Jacobian Matrisi

Temel olarak bu matris, uzay koordinatlarındaki hız vektörü ile eklem koordinatlarındaki hız vektörü arasındaki dönüşüm matrisidir. Jacobian matrisi denklem 28'da gösterilmiştir. J_{AX} vektörleri, robotun dönme (\tilde{C}) matrisi kullanılarak bulunur. J_{PX} vektörleri ise \bar{P} vektörü kullanılarak bulunur J_P matrisi bulunduktan sonra hız ve ivme analizlerinin her ikisinde de kullanılır [4].

$$J_P = \begin{pmatrix} J_{P1} & J_{P2} & J_{P3} & J_{P4} & J_{P5} & J_{P6} \\ J_{A1} & J_{A2} & J_{A3} & J_{A4} & J_{A5} & J_{A6} \end{pmatrix} \quad (28)$$

Burada, J_{Pn} ve J_{An} ,

$$J_{Pn} = \frac{\delta}{\delta q_n} P \quad (29)$$

$$J_{An} = \text{col} \left[\left(\frac{\delta}{\delta q_n} C \right) \cdot C^T \right] \quad (30)$$

formülleriyle elde edilir [4].

Hız Analizleri

Verilen eklem konum ve hızları kullanılarak, robotun uç noktasının hızı kolaylıkla bulunabilir.

$$\begin{pmatrix} V_P \\ w_P \end{pmatrix} = J_P \cdot \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{pmatrix} = J_P \cdot \dot{Q} \quad (31)$$

Bu denklemde,

V_P robotun uç noktasının çizgisel hız vektörü,
 w_P robotun uç işlemcisinin açısal hız vektörü,
ve
 q_X x'nci eklem değişkenidir.

J_P matrisi düz ve ters hız analizlerinde kullanılır. Bu denklemler aşağıdaki gibidir;

$$V = J_P \cdot \dot{Q} \quad (32)$$

$$\dot{Q} = J_P^{-1} \cdot V$$

Bu denklem $\det(J_P) \neq 0$ olduğu sürece geçerlidir; $\det(J_P) = 0$ olması ise hız ve ivme ile ilintili tekil konumları belirler. Dolayısıyla, konum analizinde olduğu gibi, hız analizinde de bu konumların önceden belirlenmesi, hareket planlamasında bu konumlardan kaçınabilmek için gereklidir. Bu tekil konumlardan en sık rastlanılabilecek olanı, konum analizinde de görülen $\theta_5=0$ konumudur. Diğer tekil konumlara ise bu robotun öngörülen çalışma bölgesi içinde rastlanılamayacağı, $\det(J_P) = 0$ denkleminin çözümlerine bakılarak görülebilir.

İvme Analizleri

Robotun verilen eklem konum, hız ve ivme değerleri kullanılarak, robotun uç işlemcisinin ivme vektörü aşağıdaki denklemler aracılığıyla bulunabilir [4].

$$J_P \cdot \ddot{Q} + \dot{J}_P \cdot \dot{Q} = \ddot{V} \quad (33)$$

$$\ddot{Q} = J_P^{-1} \cdot (\ddot{V} - \dot{J}_P \cdot \dot{Q})$$

YÖRÜNGE PLANLAMA

Robotik uygulamalarında, genel olarak robotun belli bir rota üzerinde gitmesi istenir. Boyama, parlatma ve kaynak gibi bazı özel uygulamalarda, robotun hareketinin daha hassas olması gerekmektedir. Bütün uygulamalarda, robotun geçmesi veya geçmemesi gereken bazı ara noktalar vardır. Bu hareketleri robota yaptırmak için rota planlaması yapılmalıdır.

Rota planlaması yapmak için programcının iki seçeneği vardır. Planlamayı uzay koordinatlarında yada eklem koordinatlarında yapabilir. Endüstriyel robotlar için, eklem koordinatlarında planlama yapmak daha verimlidir, çünkü bu işlem daha hızlı bir işlemdir ve robot daha sarsıntısız ilerler.

Genel Hususlar

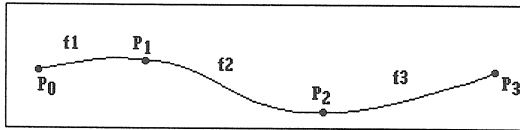
Rota planlaması için, bütün eklem açılarının, konum, hız ve ivmelerinin tanımlanması gerekmektedir. Genel yaklaşım olarak, planlanan fonksiyonun her aralık (iki nokta arası), için güncellenmesidir. Bu planlanan fonksiyon, bütün zaman aralığı için akıcı olmalıdır. Bunun garantilenmesi için, fonksiyonun birinci ve ikinci türevlerinin, ara noktalarda eşit olması gerekmektedir.

Eğer rota üzerinde (n+1) nokta varsa, rota planlaması için (n) tane fonksiyon gerekmektedir. (Şekil 3) kullanıcı, başlangıç noktası ve bitiş noktaları için konum, hız ve ivme tanımlayabilmelidir. Bu, başlangıç ve bitiş noktaları için üçer denklem getirir. Ara noktalar için kullanıcının konum vermesi yeterlidir, fakat akıcılık için her iki taraftaki fonksiyonun hız ve ivme değerlerinin eşit olması gereklidir. Bu da her ara nokta için (4n-4) denklem daha getirir. Bu nedenle, toplam olarak (n) nokta için (4n+2) denklemin sağlanması gerekir.

Bu koşulların sağlanabilmesi için ve nokta sayısı arttıkça gelecek denklemlerin çözülebilmesi için, noktaların arasına uydurulacak en iyi fonksiyon sırası, yani yörünge, 4-3-3-4 yörünge sırasıdır. Bu sırada, ilk ve son aralığa 4. derece fonksiyonlar, aralara ise 3. derece fonksiyonlar uydurulur [3].

4-3-4 Yörünge Sırası

Yörünge planlaması, asıl olarak, her eklem için ayrı ayrı yörünge planlamasıdır. Yörüngeyi planlamadan önce, her eklem, her noktadaki konumları ile, başlangıç ve bitiş noktalarındaki hız ve ivmelerinin bulunması gerekmektedir.



Şekil 3. 4 nokta için rota planlaması

Geçilmesi istenen n nokta olduğunu düşünülün. İlk (P₀) ve ikinci (P₁) nokta arasındaki fonksiyon parçasının denklemi [3];

$$f_1(t) = A_1 + A_2t + A_3t^2 + A_4t^3 + A_5t^4 \quad (34)$$

Son (P_n) ve sondan bir önceki (P_{n-1}) noktanın arasındaki fonksiyon parçasının denklemi;

$$f_{n-1}(t) = Z_1 + Z_2t + Z_3t^2 + Z_4t^3 + Z_5t^4 \quad (35)$$

Ara noktaların (P_k, P_{k+1}) arasındaki fonksiyon parçalarının denklemi ise;

$$f_k(t) = B_1 + B_2t + B_3t^2 + B_4t^3 \quad (36)$$

olsun. Noktalardaki konum değerleri, şu şekilde verildiğini düşünersek;

$$f_1(0) = P_1, \quad (37)$$

$$f_1(t_1) = P_2, \quad (38)$$

$$f_2(0) = P_2, \quad (39)$$

$$\dots$$

$$f_{n-1}(t_{n-1}) = P_n. \quad (40)$$

burada, t₁, başlangıç noktasından birinci noktaya gitme süresi ve t_x ise P_{x+1} noktasından P_x noktasına gitme süresidir.

Bunlara ek olarak, ilk ve son noktaların hız ve ivmeleri de şu şekilde verilmiş olsun;

$$f_1'(0) = P_1' \quad (41)$$

$$f_1''(0) = P_1'' \quad (42)$$

$$f_1'(t_{n-1}) = P_n' \quad (43)$$

$$f_1''(t_{n-1}) = P_n'' \quad (44)$$

Bütün bu koşulları ve verilenleri birleştirirsek, sonucu matris formunda gösterebiliriz;

$$M \times C_f = V \quad (45)$$

Burada, M matrisi, dönüşüm matrisidir. C_f, fonksiyonların katsayı vektörüdür ve V de konum, hız ve ivmelerin vektörüdür.

HAREKET KONTROLÜ

Robot üzerinde, daha önce de belirtildiği gibi, altı adet motor vardır. Bu motorlar AC servo motorlardır. Bu motorları sürmek için, servo sürücüler ve transformatör içeren bir elektrik panosu vardır. Bilgisayar ve elektrik panosu arasında bulunan hareket kontrol kartı, Delta-Tau marka motor kontrol kartıdır.

Kart, Delta Tau firmasının Pmac2 hareket kontrol kartıdır. Bu kart ile aynı anda sekiz adet motor simultane bir biçimde sürülebilir. Pmac2 kartı,

başlı başına da kullanılabilir, fakat bu projede bilgisayar yardımı ile motorlar sürülecektir.

Motor Kontrol Kartının Kullanımı

Motor kontrol kartını kullanmadan önce, bazı parametrelerin girilmesi ve ayarlanması gerekmektedir. Bu parametreler, motorun ön yükleme parametreleri ve kontrol parametreleridir. Ayrıca her motorun hız ve ivme için limit parametreleri de vardır. Bütün kontrol ve güvenlik parametreleri girildikten sonra, motorlar kontrol için hazırdır.

Kullanıcı, motorları kullanmak için, ekrandan hareket komutlarını girebilir, ya da hareket programını bir matris dosyasına yazarak motor kontrol kartına gönderebilir. Ayrıca kullanıcı, kendi koordinat eksenini de belirleyebilir.

Hareket programlarını yazmak, bilgisayar programı derleyicisine kod yazmaktan daha kolaydır. Aşağıda örnek bir hareket programı vardır. (Şekil 4). Satırların açıklamaları sağ tarafındadır [7, 8].

OPEN PROG 1	; Program 1 başlangıcı
CLEAR	; Hafızadaki bilgilerin sil
LINEAR	; Hareketler lineer modda olsun
TA500	; İvme zamanının 500 ms. olarak ayarlanması
F5000	; Hızın 5000 birim / sn. Olarak ayarla
X10000	; X-Eksenini 1000 noktaya ilerlet
DWELL500	; O konumda ½ sn. dur
X0	; X-Eksenini 0 noktasına gönder
CLOSE	; Programın sonu

Şekil 4. Örnek Hareket Programı

Hareket Kontrol Kartına C++ Programı üzerinden Erişmek

Pmac hareket kontrol kartına C++ veya Visual Basic gibi programlarla da erişmek mümkündür. Bu amaç için bir DLL (Dynamic Link Library) kütüphane dosyası mevcuttur. Bu kütüphane dosyası değişik fonksiyonlar içerir. Bu kartın dosyası olan Pmac.dll dosyası içinde yaklaşık 800 fonksiyon vardır. Bu fonksiyonların listesi, bu dll dosyası yardımıyla oluşturulabilecek olan Pmac.def dosyası içinde görülebilir [6][9].

Bu erişim işlemini yapmak için, dll dosyasını önceden bilgisayarın erişimine açan, aşağıdaki satırları (Şekil 5), programın başına eklemek gerekir. Buradaki fonksiyon, kartı kullanıma açar. Fonksiyonun içindeki "dwnum" parametresi, kartın

sıra numarasıdır ve bir kart durumunda bu değer hep "0"dır.

```
HINSTANCE hinstDLL = LoadLibrary ("PMAC.DLL");
void (FAR *lpfnOpenPmacDevice)(DWORD dwDevice);
FARPROC lpfnOpenPmacDevice=GetProcAddress(hinstDLL, "OpenPmacDevice");
```

Şekil 5. Fonksiyonları Tanımlamak için Örnek Kod

Bu kodu yazdıktan sonra, bu fonksiyon, programın içinde herhangi bir yerde, aşağıdaki şekildeki gibi (Şekil 6) çağırılabilir

```
LpfnOpenPmacDevice(0);
```

Şekil 6. Örnek Fonksiyon Çağırma Kodu

Bu kod Pmac kartına bir kanal açar. Kanal açıldıktan sonra diğer fonksiyonlar yardımıyla, robot istenen konuma getirilebilir. Eğer rota planlanmışsa, bu rota kartın anlayabileceği dile çevrilerek bir metin dosyasına, hareket programı olarak yazılır ve karta yüklenir. Bu işlem sonucunda robot harekete başlar ve istenen rotayı takip eder.

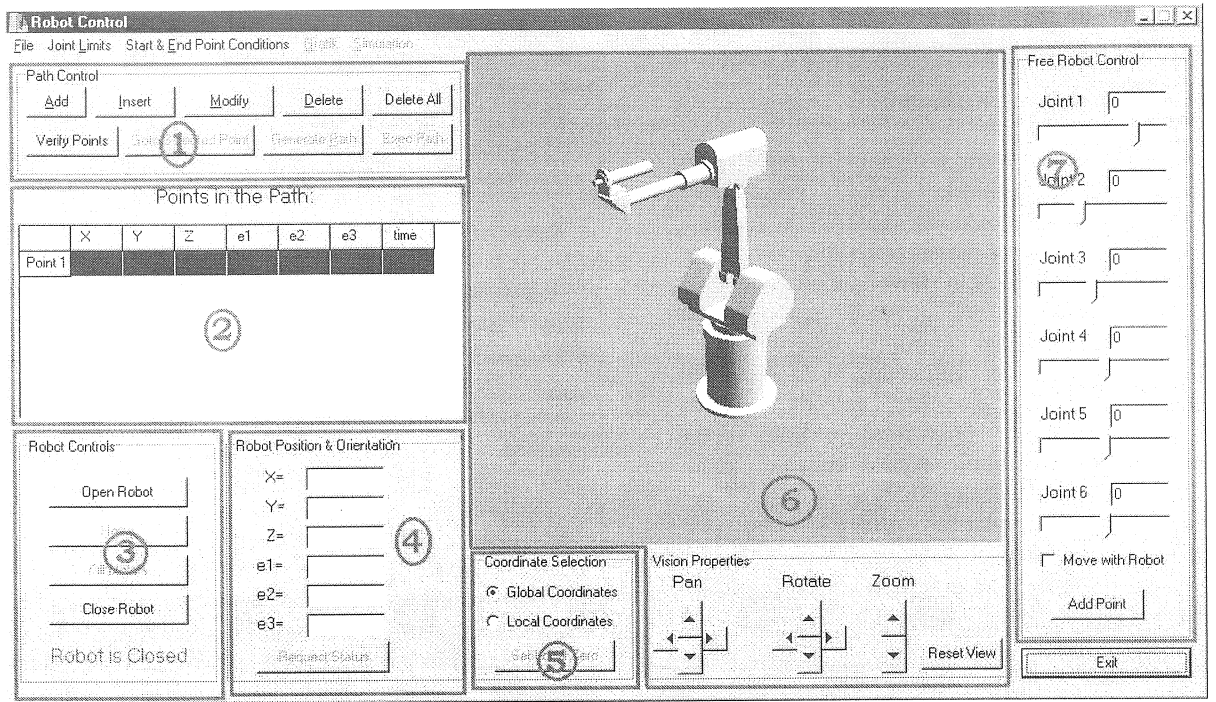
ARAYÜZ PROGRAMI

Bu kısımda, arayüz programının uygulaması ve kullanımı anlatılmıştır. Program, robotun simulasyonu ve kontrolünü içermektedir. Bu program sayesinde, robot noktadan noktaya, ya da rota belirlemek suretiyle hareket ettirilebilir [2].

Program

Programın asıl amacı, robotu kontrol etmektir. Robot, eklemleri tek tek hareket ettirilerek, ya da uç noktanın konum ve yönelimi verilerek kontrol edilebilir. Her eklem için ayrı ayrı açılarını kontrol etmek suretiyle robotu kontrol etmek, programın bir kısmıdır. Bu tür kontrolde, robot anında hareket ettirilebilir. Kullanıcı robotun eklemlerini ekranda hareket ettirirken, aynı zamanda ekrandaki robot simulasyonu hareket eder. Eğer kullanıcı, robotu da aynı anda hareket ettirmek isterse, ekrandaki "Move with Robot" (Robotla hareket et) seçeneğini işaretlemesi yeterlidir.

Programın ekran görüntüsü aşağıda verilmiştir. (Şekil 7). ekran üzerinde yedi kısım ve ayrıca altı adet de menu seçeneği vardır (Şekil 8).



Şekil 7. Programın Ekran Görüntüsü

Ekran Kontrolleri

1) Nokta listesindeki noktalar, bu kısımdan ayarlanır. Ayrıca robot istenen noktaya, ya da rotaya buradan gönderilir.

2) Noktalar bu kısımda listelenir.

3) Robotla iletişim bu kısımdan başlatılır, başka bir deyişle, robot buradan açılır ve kapanır. Ayrıca robot park konumuna buradan gönderilir.

4) Uç noktanın gitmesi gereken konum ve yönelim buradan takip edilir, ayrıca robotun o andaki konumu buradan sorgulanır.

5) Bu kısımdan, yerel/evrensel koordinat sistemi seçilir.

6) Bu kısımda robotun üç boyutlu simülasyonu vardır. Robotun o anki konumu ve gideceği noktalar buradan takip edilebilir.

7) Robotun eklem açıları değiştirilerek kontrolü buradan yapılır. Her eklem açısı ayrı ayrı kontrol edilir. İçerisindeki "Add Point" (Nokta Ekle) düğmesi vasıtasıyla, robotun konumu nokta listesine eklenebilir.

Menu Seçenekleri



Şekil 8. Menu Seçenekleri

1) **File (Dosya)**: Buradan, dosya içerisindeki noktalar, nokta listesine aktarılabilir.

2) **Robot Parameters (Robot Parametreleri)**: Robot Eklem uzunlukları ve konfigürasyon buradan kontrol edilebilir. Ayrıca robotun ucuna bir aparat takıldığında uzunluğu buradan girilir.

3) **Joint Limits (Eklem Limitleri)**: Buradan, her eklem için, konum, hız ve ivme limitleri belirlenebilir.

4) **Start & End Point Conditions (Başlangıç ve Bitiş Şartları)**: Bu kısım yörünge planlaması içindir. Yörünge başlangıç ve bitiş noktalarında hız ve/veya ivme isteniyorsa, buradan ayarlanabilir.

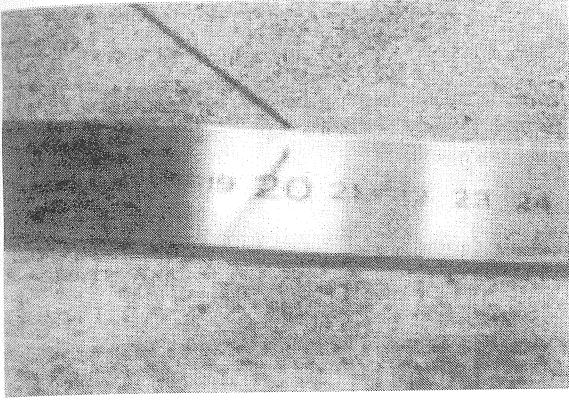
5) **Graphics (Grafik)**: Bu kısım yörünge planlandıktan sonra aktif olur. Bütün eklemlerin konum, hız ve ivme grafikleri ile birlikte, uç noktanın konum, hız ve ivme grafikleri takip edilebilir.

6) **Simulation (Simülasyon)**: Yörünge planlandıktan sonra, robota göndermeden önce kontrol edilmek istenirse, bu düğme sayesinde yörünge ekranda takip edilebilir.

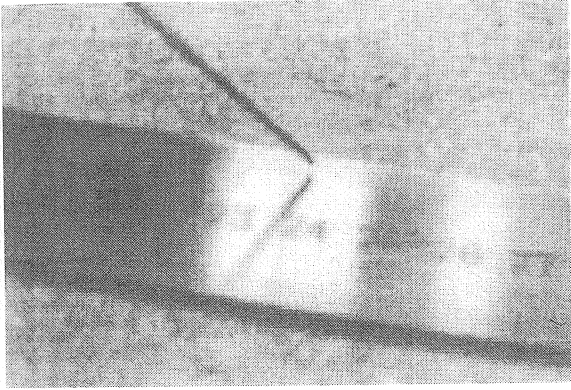
PERFORMANS ve DEĞERLENDİRME

Program geliştirildikten sonra, robotun ucuna, gittiği noktayı göstermesi amacıyla, ucunda 1 mm çapında tel takılı bulunan bir aparat takılmıştır. Bu aparat sayesinde, robotun uç noktasının hareketi hassas olarak gözlenebilmiştir. Daha sonra robota, bir noktadan diğer noktaya gitmesi için komut verilmiştir. Cetvel üzerinde yapılan bu denemede,

robotun, aparatın hassasiyeti olan 1 mm'nin altında bir doğruluk ile hareket ettiği gözlenmiştir. Aşağıda, robota verilen 4 cm'lik hareket komutunun başlangıcı ve bitiş pozisyonları verilmiştir.



Şekil 9. Hareketin Başlangıcı



Şekil 10. Hareketin Sonu

SONUÇ VE ÖNERİLER

Bu projede, altı serbestlik dereceli bir endüstriyel robot prototipi ve yazılımı, tümüyle yerli olanaklarla geliştirilmiştir. Robot uzay koordinatları kullanılarak, istenen rota için programlanabilmektedir. Bu sayede fabrika otomasyon ortamında, esnek üretim sistemlerinde, tezgah yükleme ve boşaltma işlemlerinde kullanılabilir. Ayrıca taşlama, parlatma gibi uygulamalarda da yararlı olabilecektir. Kaynak ve boya işlemlerini de gerekli teçhizat kullanıldığında yerine getirebilecektir.

Robotun akıllı şekilde çalışabilmesi için kamera aracılığı ile görme tanıma algoritmaları geliştirilmektedir. Bu sayede kendisi gerekli parçaları görerek ve tanıyarak, uzay koordinatları verilmeden işlemler yapma yeteneğine sahip olacaktır.

Ayrıca robot ile beraber çalışacak el, konveyör ve döner tabla tasarım ve imalat çalışmalarını da devam ettirmektedir.

SOFTWARE DEVELOPMENT FOR MAN-MACHINE INTERFACE FOR AN INDUSTRIAL ROBOT

In this study, a robotic software, which controls a robot, is developed. The robot considered is a six degree of freedom robot and it is designed and manufactured in METU. User can send the robot anywhere in space within its workspace, in any orientation. Forward and inverse kinematics can be executed according to the needs.

Simulation framework is embedded into the software for the 3D visualisation of the robot. Any movements can be simulated on the screen.

Software also generates the path for the given points. Then generated path is simulated on the screen. All position, velocity and acceleration graphics of joints can be examined for the generated path.

Keywords: Industrial robot, kinematic analysis, 3D simulation, path generation.

TEŞEKKÜR

Yazarlar, Bilgisayar Destekli Tasarım, İmalat ve Robotik (BİLTİR) Merkezine, alet ve cihaz kullanımı için, ve 08-04DPT2001K120980 kod numaralı proje kapsamında DPT ve devam eden YUUP desteği için, teşekkürlerini sunarlar.

KAYNAKÇA

1. Balkan T., Özgören M. K., Arıkan M. A. S., Baykurt H. M., "A Method of Inverse Kinematics Solution Including Singular and Multiple Configurations for a Class of Robotic Manipulators", Mechanism and Machine Theory, pp.1221-1237, 2000.
2. Cengiz, M. C., "Software Development for Man-Machine Interface for an Industrial Robot", METU Mechanical Engineering Master of Science Thesis, 2003.
3. Konukseven E. İ., "Graphical Simulation and Programming of Robots", METU Mechanical Engineering Master of Science Thesis, 1989.
4. Özgören M. K., "ME 522 Lecture Notes on Principle of Robotics", METU, unpublished 2001.
5. Özgören M. K., "Topological Analysis of 6-joint Serial Manipulators and Their Inverse Kinematic Solutions", Mechanism and Machine Theory, vol.37, No.5, pp.511-548, 2002.
6. "Pcomm32, PMAC 32 Bit Driver" manual, Delta Tau Systems, 2000
7. "Pewin32, PMAC Executive for Windows" manual, Delta Tau Systems, 1995.
8. "PMAC Users Manual", Delta Tau Systems Inc, 1991.
9. Valley, S., "ObjectWindows : Programming Guide", Borland International, 1992.