

PAPER DETAILS

TITLE: AutoLISP KULLANILARAK ÜÇ KOLLU ROBOTUN HAREKET SIMÜLASYONU

AUTHORS: Mustafa BOZDEMİR,Kerim ÇETINKAYA

PAGES: 1-7

ORIGINAL PDF URL: <https://dergipark.org.tr/tr/download/article-file/191391>



AutoLISP KULLANILARAK ÜÇ KOLLU ROBOTUN HAREKET SİMÜLASYONU

Mustafa BOZDEMİR*, Kerim ÇETİNKAYA**

*Pamukkale Üniversitesi, Teknik Eğitim Fakültesi, Makine Bölümü, Denizli

**Gazi Üniversitesi, Teknik Eğitim Fakültesi, Makine Bölümü, Teknikokullar/Ankara

Geliş Tarihi : 19.03.1999

ÖZET

İstenilen görev ve amaca en uygun biçimde robot geliştirmek tasarımcılar için çok büyük önem taşır. Diğer bir deyişle üzerinde çok uğraşıldığı halde tasarlanan robotlar her zaman görev yerlerinde başarılı olmayabilir. Bu nedenle tasarımı düşünülen robot sistemlerin imalinden önce, bir simülasyon programı hazırlanarak, iş alanının incelenmesi, yörünge planlamasının yapılması, tasarımcıya eğer var ise yanlışlıkların daha robot yapılmadan giderilmesi imkânını sağlayabilir. Bu konuda AutoCAD programının içerisinde bulunan AutoLISP programlama dili kullanılarak, üç kollu bir robotun hareket simülasyonu yapılmıştır. Simülasyon denklemlerinde analitik ve matris çözüm metotları kullanılmıştır.

Anahtar Kelimeler : İş alanı, Yörünge planlaması

MOVEMENT SIMULATION OF THREE ARMED ROBOT BY BEING USED AutoLISP

ABSTRACT

Development of a suitable robot for desired aim and duty is very important for design engineers. On the other hand, designed robots may be unsuccessful in functioning inspite of many efforts. For this reason, before the production of considered robot systems, having prepared a simulation program, investigation of work field, getting done path plan and mistakes could be eliminated before robot production. In this study, movement simulation of a three armed robot has been realized by using AutoLIPS programing language which is supplied with AutoCAD. Analytical and matrix solution methods have been used in simulation equations.

Key Words : Work field, Path planing

1. GİRİŞ

Robot sistemler, istenilen amaca yönelik olarak hazırlanan mekanik, elektronik ve bilgisayar alt sistemlerinin belli bir hareket yönetim programı altına çalıştırılmasıyla oluşturulan sistemlerdir. Robotlar, çeşitli görevleri yapmak amacıyla değişik şekillerde programlanmış hareketlerle; nesnelerin, gereçlerin yada özel düzeneklerin taşınması için tasarlanmış çok işlevli bir manipülatörlerdir (Todd, 1986).

Elde edilen ürünlerin kalitesinin yükseltilmesi,

üretim miktarlarının artırılması konularında, bilgisayar destekli tasarım ve üretim alanlarında sürekli olarak araştırmalar yapılmaktadır. Robot sistemler, tasarlandıkları görevlerin amaçlarını çok hızlı ve hassas olarak yapabilirler. Günümüzde robot sistemlere özellikle, insan sağlığı ve güvenliğinin tehlike altında bulunduğu ortamlarda yada işlerde ihtiyaç duyulmaktadır (Bozdemir, 1996). Otomobil imal ve montajındaki her türlü kaynak işlemlerinde, ilaç sanayinde, nükleer santrallerde yakıt yüklemeinde, ısı işlemlerinde kullanılan tezgahlara iş parçası yerleştirmesinde vb. iş alanlarında, robot sistemler kullanılmaktadır (Ranky, 1985). Bu

sektörlerin kendi aralarındaki rekabetleri ve daha kaliteli ürün yapabilme çalışmaları sonucunda, robot sistemlerin kullanılmaları yaygınlaşarak devam etmektedir.

Robot sistem şekillerinin tasarımı yada imali yapılmadan önce, sistemin kullanılacağı yerdeki görevinin tam olarak bilinmesi gerekir. Yapılması istenilen göreve en uygun olacak şekilde robot sistem dizayn edilmelidir (Mitsubishi, 1994). Tasarımı planlanan robot sistemin imalinden önce, simülasyon programı hazırlanarak iş hacminin incelenmesi, yörünge planlamasının yaptırılması, tasarımcısına yeni görüş açıları kazandırarak belki sistemdeki yanlışlıkların daha robot yapılmadan giderilmesi mümkün olacaktır.

Robot sistem simülasyon programları, Pascal, C++, Cobol gibi programlama dilleri kullanarak hazırlanabilir. Çizim ve düzenleme fonksiyonlarının kullanılabilirliği nedeniyle AutoCAD çizim programı içerisinde bulunan, AutoLISP programlama dili kullanarak da bu şekilde bir simülasyon programı hazırlanabilir. Yapılan çalışma içerisinde, PUMA tip üç kollu robot sistemin yörünge hareket hareketinin simülasyon programı hazırlanmıştır.

Lisp programlama dili yapay zeka çalışmalarında kullanılır. AutoLISP ise Lisp'in AutoCAD ile kullanılabilecek şekilde uyarlanmasıdır. AutoLISP'in çalışması için öncelikle AutoCAD'ın çalıştırılması gerekir. AutoLISP ile hazırlanan bir simülasyon programını çalıştırabilmek için, çizim ve görüntü ayarlamalarının bulunduğu *.dwg uzantılı çizim dosyasının kullanılması gerekir (Çıkış, 1994). Kullanılan bu çizim dosyası ile programı çalıştırmak için her defa görüntü ve çizim özelliklerini ayarlamaya gerek kalmaz. Çizim dosyası içerisinde AutoCAD ekranını istenilen sayıda görüntü penceresine bölmek ve bu pencerelerin her birine değişik yönlerden bakmak mümkün olmaktadır.

Üç boyutlu olarak hazırlanan ve robot simülasyon programını içeren lisp (*.lsp) dosyası, *.dwg uzantılı bu çizim dosyası üzerinde çalıştırılabilir. Elde edilen robot simülasyonu görüntüsüne değişik yönlerden bakılabilir (Önden, üstten, perspektif vb). AutoLISP ile hazırlanan robot simülasyon programındaki tüm katman, limits, zoom ve görüntü penceresi ayarları Robot.dwg ismindeki AutoCAD çizim dosyası içerisinde saklanmaktadır. AutoLISP programı bu hazır ortamdaki ayarlamalar etkisinde çalışmaktadır (Bozdemir, 1996).

Robot sistem simülasyonu ve bilgisayar ortamında tasarım konusunda yapılan çalışmaların bir tanesi de sistemde ters kinematik çözümler kullanılarak yapılan 10 gövde ve 7 bilek mekanizması

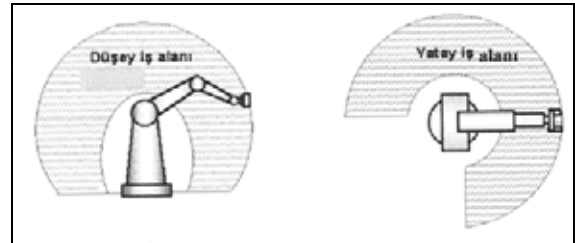
tasarlayabilen modüler robot programlarıdır (Kocabaş, 1990).

Robotların yörünge planlaması ve evrik kinematik yöntemlerle devre dışı olarak hesaplanan bilgiler sisteme girdi olarak verilerek çevrim denetimi sağlayan programlarda geliştirilebilmektedir (Dağ, 1995).

Puma tip robotların simülasyonu ile ilgili olarak yapılan "ROBSIMPRO" isimli grafik simülasyon programında ise üç boyutlu olarak hazırlanan model üzerinde grafik öğreti yöntemi kullanılmaktadır. Bu yöntemle off-line programlama ve yörünge planlamasının animasyonu yapılmaktadır (Konukseven, 1990)

2. SİMÜLASYONU YAPILAN ROBOTUN YAPISI

Simülasyon programında kullanılan robot kol sistemi üç kol ve bilek mekanizmasından oluşmaktadır. Bu manipülatörün kollarının yapısı, dönebilen mafsallardan oluştuğu için hareket alanı küresel bir şekildedir. İş alanının dış sınırını, robot kollarının tam açıldığı durumdaki toplam kol boyu oluşturur. Küresel iş alanı içerisinde kolların boyut farkından ve uç elemanın ulaşamadığı diğer bir küresel bölge vardır (Şekil 1).



Şekil 1. Simülasyon programında kullanılan robot sistemin çalışma alanı

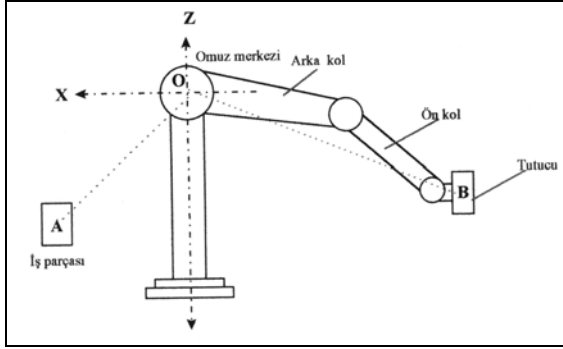
Simülasyonu yapılan robot kolunun yatay X-Y düzlemi üzerindeki hareket alanı, düşey ekseninde saatin tersi yönünde 270°'lik bir alan olarak tespit edilmiştir. Robot sistemin görevi, çalışma alanı içerisinde koordinatı verilen cisimleri konumu önceden tanımlanmış olan konveyör üzerine yerleştirmektir.

Simülasyon programının hazırlanmasında, analitik hareket denklemleri ve matris hesaplama metodları birlikte kullanılmıştır. Analitik çözümde AutoCAD'in klasik bazı düzenleme komutları kullanılarak sonuca ulaşılmıştır.

3. ANALİTİK HESAPLAMA YÖNTEMİ

Robot kol, mafsallı bağlantıları sayesinde uzay koordinat düzleminde dairesel hareketler yapmaktadır. Robot sistem kolları tarafından yapılan, hedef konuma ilerleme hareketleri sonrasında uç eleman, iş parçası olarak kabul edilen cismin koordinatlarına gelmektedir.

Şekil 2’de şematik olarak gösterilen robot kol üzerindeki B noktası uç elemanı, A noktası ise, üç boyutlu düzlem içerisindeki hedef cismi göstermektedir. A noktası, A(Csmx, Csmx, Csmz) şeklindeki koordinatta tanımlanmıştır.



Şekil 2. Analitik metotla robot kol hareketlerinin hesaplanması

Omuz merkez noktası “O”, koordinat düzleminin merkezinde (0, 0, 0) noktasında bulunmaktadır. A-O mesafesi cismin robot kol omuzuna uzaklığı olup, simülasyon programı içerisinde CİSUZ kısaltmasıyla tanımlanarak kullanılmaktadır.

Setq CİSUZ (Sqrt (+ (*Csmx Csmx)(*Csmx Csmx) (*Csmz Csmz)))

Robot sistem uç tutucusunun konum hareketlerini yönlendirme tanım noktası D(Dx, Dy, Dz) dir. D-O mesafesi açık kol boyunun omuz merkezine olan uzaklığını göstermektedir. Program içerisinde bu uzaklık KOLUZ kısaltmasıyla kullanılmaktadır.

Setq KOLUZ (Sqrt (+ (* Dx Dx) (* Dy Dy) (* Dz Dz)))

Cismin omuz merkezine olan mesafesi ile kol uç elemanının mesafesi arasındaki fark miktarı, robot ön kolunun arka kolla birleşim mafsali üzerinde saat yönüne veya ters yönde dönmesiyle eşitlenir. Bu fark miktarı Kolfark ifadesi ile tanımlanmaktadır.

Setq Kolfark (- Cisuz Koluz).

Kolfark ifadesinin değeri sıfırdan büyük ise ön kol, saatin tersi yönünde OB mesafesini artıracak şekilde

hareket ettirilir. Kolfark değerinin sıfırdan küçük olduğu durumda saat yönüyle aynı ve OB mesafesini azaltacak şekilde bir hareket yaptırılır.

(IF (> kolfark 0)

(WHILE (<= Koluz Cisuz)

...

(WHILE (>= Kkoluz Cisuz)

Simülasyon programda XY düzlemi üzerindeki hareketlerin gerçekleştirilmesinde AutoCAD’in “Rotate” komutu kullanılmıştır. “Rotate” komutunu kullanmadan önce nesne seçimi ve döndürme noktasının belirlenmesi için “Ssgt” ile tanımlanmış seçim penceresinin açılması gerekir. Seçim penceresinin açılmasında, seçim alanının sınırlarını oluşturacak çapraz iki köşenin koordinatı verilir. “Rotate” komutu kullanılarak seçim alanı içerisindeki işaretlenen objelerin tamamı, belirlenen Z1 noktası etrafında hesaplanan miktar kadar döndürülmesi sağlanır.

(Setq Z1(Ssgt "C " '(-45 -45) '(400 50)))

(Setq Dönnok(0,0,0))

(Command "Rotate" Z1 "" Dönnok 30)

Z ekseninde, cismin konveyöre yerleştirilmesi hareketin gerçekleştirilmesi sırasında uç eleman tarafından tutulan cismin konveyör üzerine düzgün şekilde bırakabilmesi için “Repeat” komutu kullanılmıştır. Repeat, döngü içerisinde verilen değişkeni kontrol ederek, hesaplanan miktar kadar döngünün gerçekleştirilmesini sağlar.

(Repeat (- Csmz Cz)

(Setq Cx (Sqrt(- (* Uklboy Uklboy) (* Cz Cz))))

(Setq Dx (Sqrt(- (* Kklboy Kklboy) (* Dz Dz))))

(Setq Cz (+ Cz 1))

(Setq Dz (+ Dz 1))

Yukarıdaki verilen program parçasında “Repeat” komutu içerisindeki işlem (Csmz-Cz) işlem sonucu kadar tekrar eder. İşlem sırasında Cz ve Dz değerlerinde meydana gelen yeni sonuçlar tekrar sayısını etkilemez. Simülasyon programındaki temel yaklaşım, hesaplanan temel noktalara robot sistem şekli çizilmesinden sonra yeni koordinatların hesaplanarak robot şeklinin bu noktaya tekrar çizilmesi biçimindedir. Robot kol sisteminin yeni hesaplanan konuma çizilmesi için, bir önceki robot şeklinin ekrandan silinmesi gerekir. Silme işleminde “Erase” komutu kullanılır. Bu komutun kullanılabilmesi için silinmesi gereken obje yada objelerin tanımlanması gerekmektedir (Çıkış, 1994).

(Setq Z1(Ssgt "W" '(-15 -49) '(15 -150)))

(Command "Erase" Z1"")

AutoCAD yeni şekillerin oluşturulması ve eskilerin silinmesi işlemleri sırasında ekranda oluşan bazı işaretleme noktalarının temizlenmesi gerekmektedir. Görüntü temizleme işlemini yapmak için “Redraw” komutu kullanılmaktadır. “Erase” komutu sonrasında kullanıldığında, silme işlemi sonrasında kalan noktaların temizlenmesi sağlanmaktadır.

3. MATRİS ÇÖZÜMLEME YÖNTEMİYLE SİMULASYON

Robot kol sistemlerinde; konum, hız, ivme hareketlerinin hesaplanmasında matris formül denklemleri kullanılmaktadır. Bu formüller sayesinde robot kol sistemlerine ait konum ve açısal değişimler, hız, ivme veya kuvvet hesaplamaları

yapılabilmektedir (Craig, 1981). Robot kol sistemi tarafından X, Y, Z eksenleri etrafında bir cismin döndürülmesi işlemi sonrasındaki, pozisyon koordinatlarını hesaplamak için aşağıdaki Tablo 1’deki rotasyon dönüşüm matrisleri kullanılır (Pouli, 1981).

Matris çözümleme tekniği simülasyon sırasında robotu oluşturan bütün temel noktalara uygulanmalıdır. Konum pozisyon koordinatları elde edilen bütün noktalar, simülasyon programı içerisinde kullanılarak robot kol sistemin hareketi sağlanmaktadır.

Sistemin çözümüne omuz ekleminden başlanır. Çözüm sırasında kullanılacak her bir kola ait çözüm değişkenleri Tablo 2’de gösterilmiştir.

Tablo 1. Rotasyon Dönüşüm Formüllerinin X, Y, Z Örnek Noktalarına Uygulanışı (Bozdemir, 1996; Pouli, 1981)

	Rotasyon Matrisleri	Rotasyon Matrislerinin A,B,C Örnek Noktalarına Uygulanması	A, B, C Örnek Noktalarına Ait Pozisyon Koordinatları
X EKSENİ	$\text{Rot}(x, \phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) & 0 \\ 0 & \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\text{Rot}(x, \phi) * A = \begin{bmatrix} ax \\ ((ay.\cos(\phi)) + (az.(-\sin(\phi)))) \\ ((ay.\sin(\phi)) + (az.\cos(\phi))) \\ 1 \end{bmatrix}$	$\begin{aligned} Ax1 &= ax \\ Ay1 &= ((ay.\cos(\phi)) + (az.(-\sin(\phi)))) \\ Az1 &= ((ay.\sin(\phi)) + (az.\cos(\phi))) \\ A &(Ax1, Ay1, Az1) \end{aligned}$
Y EKSENİ	$\text{Ro}(y, \gamma) = \begin{bmatrix} \cos(\gamma) & 0 & \sin(\gamma) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\gamma) & 0 & \cos(\gamma) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\text{Rot}(y, \gamma) * B = \begin{bmatrix} ((bx.\cos(\gamma)) + (bz.\sin(\gamma))) \\ by \\ (((bx.(-\sin(\gamma)) + (bz.\cos(\gamma)))) \\ 1 \end{bmatrix}$	$\begin{aligned} Bx1 &= ((bx.\cos(\gamma)) + (bz.\sin(\gamma))) \\ By1 &= by \\ Bz1 &= ((bx.(-\sin(\gamma)) + (bz.\cos(\gamma))) \\ B &(Bx1, By1, Bz1) \end{aligned}$
Z EKSENİ	$\text{Ro}(z, \phi) = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 & 0 \\ \sin(\phi) & \cos(\phi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\text{Rot}(z, \phi) * C = \begin{bmatrix} ((cx.\cos(\phi)) + (cy.(-\sin(\phi)))) \\ ((cx.\sin(\phi)) + (cy.\cos(\phi))) \\ cz \\ 1 \end{bmatrix}$	$\begin{aligned} Cx1 &= ((cx.\cos(\phi)) + (cy.(-\sin(\phi)))) \\ Cy1 &= ((cx.\sin(\phi)) + (cy.\cos(\phi))) \\ Cz1 &= cz \\ C &(Cx1, Cy1, Cz1) \end{aligned}$

Tablo 2. Robot Kol Sistem Kollarına ait Çözüm Değişkenleri (Bozdemir, 1996; Craig,1981)

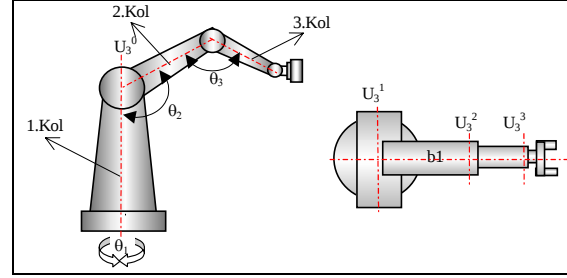
Kol No	Eklem değişkeni	Burulma açısı β_k	dk	b _k	Cos β_k	Sin β_k
1	θ_1	90°	0	0	0	1
2	θ_2	0	0	B1	1	0
3	θ_3	0	0	B2	1	0

θ = Eksen etrafında dönme
 bk = Kol uzunluğu
 β = İki eklem arasındaki burulma açısı
 dk = İki kol arasındaki orta normal farkı

$$A_n = \begin{vmatrix} \cos \theta_k & -\sin \theta_k \cos \beta_k & \sin \theta_k \sin \beta_k & bk \cos \theta_k \\ \sin \theta_k & \cos \theta_k \cos \beta_k & -\cos \theta_k \sin \beta_k & bk \sin \theta_k \\ 0 & \sin \beta_k & \cos \beta_k & dk \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (1)$$

Tablo 2’de gösterilen robotun kollarına ait değişkenler, 1 numaralı genel dönüşüm matrisinde kendi yerlerine yerleştirilerek kolların dönüşüm matrisleri bulunur (Craig, 1981). 1, 2 ve 3 numaralı kollar (Şekil 3) ait değişkenlerin genel dönüşüm

matrisi denklemindeki yerlerine konularak yapılan işlemlerin sonuçları Tablo 3’de gösterilmiştir.



Şekil 3. Simulasyonda kullanılan Robot Kol Sisteminin, konum matris denklemlerinde kullanılan değişken elemanları

Tablo 3. Robot Kollarına Ait Genel Dönüşüm Denklemi Sonuçları (Craig,1981; Pouli,1981)

1. Kol eklemine ait Dönüşüm matrisi	2. Kol eklemine ait dönüşüm matrisi	3. Kol eklemine ait dönüşüm matrisi
$A_1 = \begin{vmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$	$A_2 = \begin{vmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & b1 \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & b1 \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$	$A_3 = \begin{vmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & b2 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & b2 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

Robot kolları için bulunan dönüşüm matrislerinin çarpımından 3 numaralı kolun ucunda bulunan uç eleman bağlantı noktasının pozisyon matrisi elde edilir. Robot sistemin herhangi bir mafsalında meydana gelen açısal konum değişmesi sonucundaki, serbest kol ucunun konum koordinatı 2 numaralı denklemle hesaplanır (Pouli,1981).

$$T_3^0 = A_1 \times A_2 \times A_3 \quad (2)$$

Robot sistem serbest kol ucunun, diğer kolların hareketlerindeki değişime bağlı olarak hesaplanan konum pozisyonu matrisi 3 numaralı denklemde görülmektedir (Craig,1981).

$$T_3^0 = \begin{vmatrix} C1C2C3 - S2S3C1 & -S3C1C2 - S2C1C3 & S1 & C1C2C3b2 - C1S2S3b2 + C1C2b1 \\ S1C2C3 - S1S2S3 & -S1C2S3 - S1S2C3 & -C1 & S1C2b2C3 - S1S2S3b2 + S1C2b1 \\ S2C3 + C2S3 & C2C3 - S2S3 & 0 & S2b2C3 + b2S3C2 + b1S1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3)$$

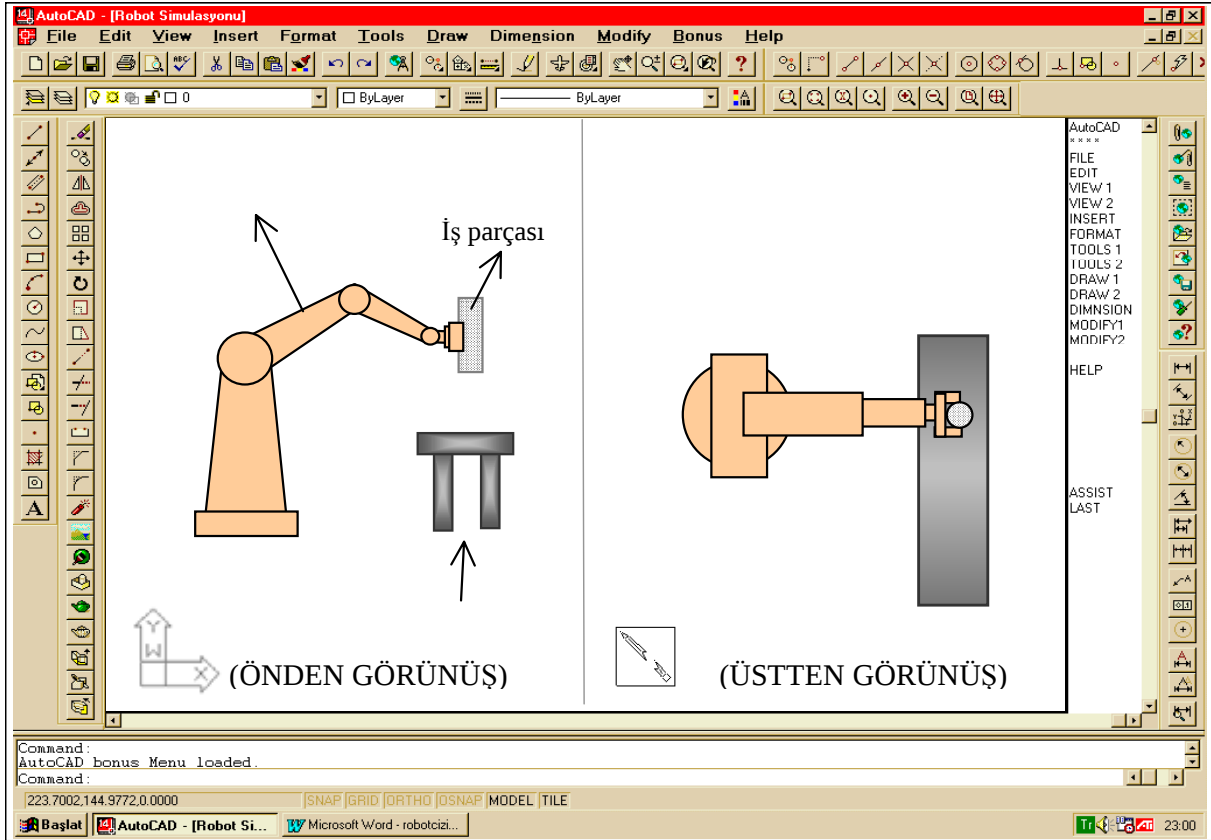
4. ROBOT SİMULASYON PROGRAMI

AutoLisp’te hazırlanan robot simulasyon programını çalıştırabilmek için, AutoCAD çizim ortamının saklandığı *.dwg uzantılı dosyasının kullanılması gerekir. Çizim dosyası AutoCAD ortamına yüklendikten sonra, FILE menüsü içerisindeki “Applications” başlığı kullanılarak robot simulasyon programını içeren lisp dosyası yüklenmelidir

(Akkurt, 1994). Program yüklendikten sonra güncel görüntü penceresi olarak nesne seçiminin tam olarak yapılabildiği pencerenin kullanılması gerekir. Anlatılan şekilde hazırlanmış ve bilgisayara yüklenmiş olan bir simulasyon programının ekranda görüntüsü Şekil 4’te görülmektedir. Elde edilen bu görüntüde, robot kol sistemi iş parçası olan cismi tutarak konveyör üzerindeki yerleştirme noktasına hareket etmektedir.

Simulasyon sırasında robot kol sistemine değişik bakış açılarından bakabilmek mümkündür. AutoCAD ekranı, örneğin 5 farklı pencereye ayrılarak robotun ön, yan, üst, alt ve perspektif şekillerde görüntülenmesi sağlanabilir (Şekil 4).

Simulasyonu yapılan robot kolunun herhangi bir eksen etrafında dönmesini sağlayabilmek için, robotu oluşturan bütün noktaların bu eksen etrafında hareketinin sağlanması gerekmektedir. Bu nedenle her nokta için matris olarak hesaplanmış koordinat denklemleri yazmak gerekir. Simulasyon programında kullanılan örnek robotun yaklaşık 30 kadar temel çizim noktası vardır. Bu noktaların hareketlerinin sürekliliği sağlanması halinde simulasyon programı doğru biçimde çalışacaktır. Robot kol üzerindeki bazı noktaların, pozisyon koordinatları program içerisinde şu şekilde hesaplanmaktadır.



Şekil 4. Simulasyon programda AutoCAD ekranı değişik görüntü pencerelerine bölünmesiyle, önden ve üstten bakışın sağlandığı ekran görüntüsü

;B1

```
(Setq BP1x(car BP1))
(Setq BP1y(cadr BP1))
(Setq xx(-(* c BP1x)(* s BP1y)))
(Setq xy(+(* s BP1x)(* c BP1y)))
(Setq xz(caddr BP1))
(Setq BP1(List xx xy xz ))
```

;C1

```
(Setq CP1x(car CP1))
(Setq CP1y(cadr CP1))
(Setq xx(-(* c CP1x)(* s CP1y)))
(Setq xy(+(* s CP1x)(* c CP1y)))
(Setq xz(caddr CP1))
(Setq CP1(List xx xy xz ))
```

5. SONUÇ

Tasarlanmak istenilen robot kol sistemleri için, simulasyon programları kullanılarak robotun imali yapılmadan iş hacminin, yörünge planlamasının, çalışma durumunun incelenmesi, sistem tasarımcılarına büyük avantajlar sağlamaktadır. Robot kol sisteminin bilgisayarın sanal ortamında kullanılması ve robotun yapacağı işin benzerinin

simulasyonunun yapılması, robotu tasarlayan üreticilere bazı yeni fikirler yada bakış açıları kazandırabilmektedir.

Robot simulasyonu konusunda Pascal, C++ ve Cobol gibi diller kullanılarak yapılmış değişik programlar vardır. Bu çalışmada Autolisp kullanılarak da robot simulasyon programının yapılabileceği gösterilmiş ve AutoCAD çizim programının yaygın olarak kullanılmaya başlandığı ülkemizde, AutoLISP' e yeni bir kullanım alanı daha ortaya konulmuştur.

AutoLISP programlama dili kullanılarak 3 kollu PUMA tipi bir robotun, analitik çözüm ve matris hesaplama metotlarıyla hareket simulasyonu yapılmıştır. Her iki çözüm yöntemi de, simulasyonun gerçekleşmesini sağlayabilmektedir. Fakat analitik olarak hesaplama yapmak üç boyutlu uzay içerisinde hesaplama karmaşıklığına sebep olmaktadır. AutoCAD'in "Rotate" ve bazı düzenleme komutları kullanılarak pozisyon koordinatlarının çözümündeki karmaşıklık engellenebilmektedir.

Matris çözümleme yöntemiyle yapılan simulasyon, robot kol sistemi oluşturan temel çizim noktaların

koordinatlarının çok hassas hesaplamasından dolayı tercih edilir. Matris hesaplama metodunda robot kol sistemini oluşturan bütün noktalar için hareket denklemleri yazmak gerekir. Buna rağmen, matris çözüm uygulanarak yapılan simülasyon, analitik olarak yapılan simülasyondan daha hızlı gerçekleşmektedir.

6. KAYNAKLAR

Akkurt, M. 1994. AutoCAD R14, Birsan Yayınevi.

Bozdemir, M. 1996. Robot Sistem Elemanları ve Hareket Analizleri, Y. Lisans Tezi, Gazi Üniversitesi.

Craig, J. 1981. Introduction To Robotic, Mechanics And Control.

Çıkış, E. 1994. AutoLISP, Türkmen Kitapevi.

Dağ, B. 1995. Üstel Dönme Matrisleri Kullanılarak Robot Kollara ait Kinematik Formüllerin

Çıkarılması ve Bilgisayara Uygulanması, ODTÜ, 739.

Kocabaş, H. 1990. Bilgisayar Yardımıyla Modüler Robottasarımı, 4. Ulusal Makine Sempozyumu, ODTÜ, 225.

Konukseven, E. İ. 1990. Robsimpro: Grafiksel Robot Simülasyonu ve Puma Tipi Bir Robotun Programlanması, 4. Ulusal Makine Sempozyumu, ODTÜ, 213.

Mitsubishi Industrial Micro-Robot System, 1994, Model Rv-M1 Instruction Manual, 9-

Pouli, R. 1981. Robot Manipulators Mathematics, Programming And Control.

Ranky, P. G. 1985. Robot Modelling, Control And Applications With Software, IFS (Publications) Ltd. UK, Springer-Verlag

Todd, D. J. 1986. Fundamentals of Robot Technology, Kogan Page, London.