

Bilgisayarla Bütünleşik İmalat Sistemi Tasarımı

Furkan GÜMÜŞ¹,

Marmara University, Institute of Pure and Applied Sciences, Istanbul, Turkey

^[1] furkan.gumus@muhendisbeyinler.net

Özet: Bu çalışmanın temel amacı eğitimde kullanılabilir robot manipülatör, CNC yönlendirici ve 3B yazıcı ünitelerinin yer aldığı bir bütünleşik imalat sistemi tasarımını göstermektir. Üç eksenli bir robot kol manipülatörü için kontrolcü yazılımı oluşturulmuştur. Esnek sistem tasarımı ve özelleştirilebilir arayüz yazılımı sayesinde teknik okullar, üniversiteler, küçük ve orta ölçekli işletmelerde kullanılabilecek maliyeti düşük, fonksiyon açısından endüstriyel standartlara yakın bir teknoloji sunmuş olacak. Eğitim kurumlarının deneysel çalışmalarda kullanabilmesi için sonradan değişikliğe uyarlanabilir bir tasarım yapılacaktır. Mevcut malzeme ve takımların kolay tedarik edilebilmesi hızlı ve ekonomik çözüm sağlayacaktır.

Anahtar Kelime: Bilgisayarla Bütünleşik İmalat, 3 Eksenli Robot Kol, CNC Router, 3B Yazıcı, Kontrolcü Yazılımı

Computer Integrated Manufacturing System Design

Abstract: The main objective of this work is to demonstrate an integrated manufacturing system design that includes robot manipulators, CNC routers and 3D printer units that can be used in training. Control software for a three-axis robot arm manipulator was created. Due to flexible system design and customizable interface software allows to low-cost at technical schools, universities, small and medium-sized enterprises. In terms of functions, it will be offered additional technology to industrial standards. Due to deterioration and the lifetime of the current system hasn't been used sustainably by. In this case, causing of the fail a technical training. In addition, the systems used in industry has got a lot of functions. This cause to isn't benefited sufficiently by small and medium-sized enterprises that want to mass production, but not requiring advanced equipment. Changeable a design will be made to use in experimental studies of educational institutions. Supplied in the easy way existing materials and equipment will provide a quick and economical solution.

Keywords: Computer Integrated Manufacturing, 3-Axis Robotic Arm, CNC Router, 3D printer, Controller Interface

I. GİRİŞ

Adını ilk defa 2011 yılında Almanya Hannover Fuarı'nda duyduğumuz Endüstri 4.0 terimi ile 4. Sanayi Devriminin temelleri atılmış, ülkeler ve şirketler küresel boyuttaki bu değişime ayak uydurmak için stratejilerini bu yönde planlamaya başlamıştır.

Ülkemizde otomotiv sektöründe (2017 yılı) yeni yeni uygulanmaya başlayan bu konsept ile 4. sanayi devrimine geçiş büyük bir hızla devam etmektedir. Endüstri ile bilişim teknolojileri bir araya getirilerek oluşturulan bu sistem daha güvenilir, hızlı ve verimli olmasının yanında internet bağlantılı

elektronik cihazların yer alması ile "akıllı fabrika" olma niteliğini elde etmektedir. [1]

Firmalar, bu artan rekabete ayak uydurmak için bilgisayarla bütünleşmiş imalat sistemleri kurmakta ve robotik sistemlere geçme ihtiyacı hissetmektedirler. Ancak teknolojisinin çok pahalı olduğu, ilk yatırım maliyetleri ve bakımları fazla olması nedeniyle çoğu işletmeler ve özellikle eğitim kurumları bu tür sistemleri alamamakta ve yeterli personel eğitimini sağlayamamaktadırlar.

Bu çalışmada 4. sanayi devrimine uyum sağlayabilme ve nitelikli bireylerin yetiştirilmesini sağlamak amacıyla teknik okullar ve üniversitelerde kullanılabilecek

eğitim amaçlı, ilk kurulum maliyeti düşük, ekonomik ve hızlı çözümler sağlayan bilgisayarla bütünleşik imalat sistemi tasarım ve uygulaması ele alınmıştır.

Literatür taramalarında imalat makilerinin tasarım ve uygulaması yer almaktadır. Bilgisayar ile bütünleşik imalat sistemleri için süreç analizleri yapılmış ve robotik sistemlerin uygulaması anlatılmıştır. Bu çalışma ile tüm bu unsurlar tek bir çatı altına alınarak incelenmesi yapılmıştır.

Eğitim kurumları, bu tür bir sistemi kullanarak maliyeti düşük ve hızlı çözüm sunan bir bütünleşik kontrol sistemi ve yazılım programı ile eğitim verebilirler. Yazılım olarak esnek bir yapıya sahip sistem sayesinde çoklu üniteler tek bir scada ekranından görülebilir ve kontrol edilebilir. Robot kol, CNC ve 3B yazıcıların çalışma mantığını, üzerinde yapacakları test ve geliştirme işlemlerini yapabilir olması önemli bir avantajdır. Öğrencinin cihazları bozma korkusu olmadan üzerinde deney yapabilecekleri ve sonradan değişiklik yapmaya uygun bir konsept oluşturulması amaçlanmıştır.

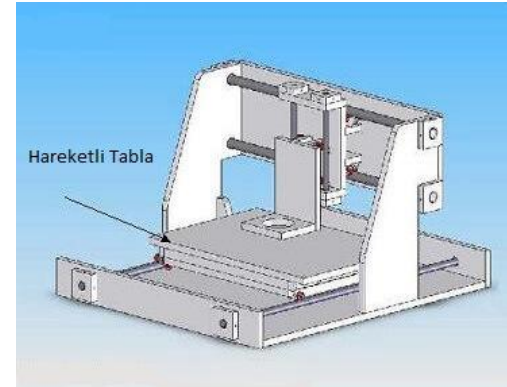
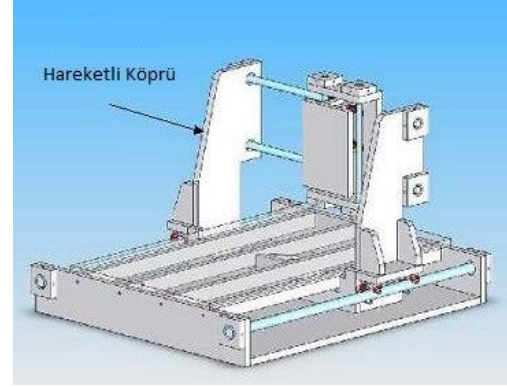
II. BİLGİSAYARLI SAYISAL YÖNETİM

Eğitim amaçlı kullanılacak cihazların nispeten ucuz ve işlevsel olması beklenmektedir. CNC freze konseptine çok benzeyen, istikrarlı ve yüksek kaliteli iş yapma kabiliyetindeki CNC Router (Bilgisayarlı Sayısal Yönetim Yönlendiricileri) bu çalışma kapsamında ele alınacaktır.

2.1. CNC Yönlendirici Tasarım Modelleri

CNC makinelerin birçok çeşidi olduğu gibi aynı sınıftaki bir makine için değişik tasarım modelleri bulunmaktadır. CNC yönlendiriciler için temelde iki farklı tasarım ön plana çıkmaktadır:

- Sabit köprü, hareketli tabla modeli
- Gezer köprü, sabit tabla modeli



Şekil 2.1. CNC router tasarım modelleri [2]

2.1.1 Sabit köprülü tasarım

CNC router üreticilerinin çok fazla tercih etmediği bir tasarım olmasına karşın sabit köprülü tasarımın da bazı üstün yanları bulunmaktadır. Bu tasarım genel olarak PCB veya gravür makineleri gibi küçük CNC makinelerin yapımında kullanılır. Bu boyutlarda bir makine için yatak tasarımı çok verimli çalışmaktadır.

Bu tasarımın en önemli avantajı yük altında esnemeyecek, sağlam bir yataklamanın çok daha kolay yapılabilir olmasıdır. Sabitlenmiş tasarım sayesinde köprü ağırlığının ve boyutunun bir önemi kalmayacaktır. Böylece köprü üzerindeki yataklamalar çok daha rahat seçilebilir; belirli boyutlara indirgeme mecburiyeti kalmaz. Gezer köprülü yapılarda ise ağırlık, lineer rulman veya kızaklar üzerinde taşınacağı için köprü ağırlığı ve boyutu çok önemli olmaktadır. Köprünün sabitlenmiş olması ise hem rijitlik hem de daha hassas parça işleme olanağı sağlamaktadır. Ayrıca daha büyük motor seçimleri ve çoklu eksenlerin yapımına izin vermektedir.

Bu tasarımın dezavantajı, x-ekseninin uzunluğu arttıkça hareketli tablanın etkisinin

daha az hale gelmesidir. Tabla ağırlığı ve taşıdığı yükün artması ile daha büyük yatak seçimi ve pahalı elemanlara ihtiyaç oluşacaktır.

Sabit köprülü hobi CNC makinelerin x yönündeki hareketi genellikle 30 ile 90 cm arasında olmaktadır. Y ve z-eksenlerinin hareket mesafesi x-eksenine göre daha azdır. Sabit köprülü konstrüksiyon daha rijit yapılı ve hafif olması sebebiyle yataklardaki radyal ve aksel yükler daha az olacaktır. Bu sayede daha ucuz ve küçük boyutlu yataklama elemanları (lineer rulmanlı yataklar veya mil yatakları) ile taşıyıcı unsurlar (kızak, mil veya ray destekleri) tercih edilebilir.

Gezer köprü tasarımlarında bilyalı vidalar kullanılırken, sabit köprülü yapı için daha ucuz olan aktarma vidalarını kullanılabilir. Ayrıca 3B yazıcılar için çok sık karşılaşılan diğer bir alternatif olan triger kayış-kasnak sistemleri de mevcuttur. Konstrüktif açıdan çok yakın olmalarına karşın freze veya matkabın talaş kaldırma sırasında alt tablaya uygulayacağı itme kuvveti, taşıma sistemlerine fazladan yük bindirmiş olacağından çoğu zaman kayış-kasnak sistemleri tercih edilememektedir. Sonuç olarak,

- Küçük boyutlu bir makine yapmak istiyorsak sabit köprülü tasarım tam bize göre olacaktır.
- Daha kompakt ve yapımı diğerlerine göre çok daha basittir.
- İskelet yapısı rijittir; esneme çok az
- Genel olarak boyuta kısıtlama getirilmeli
- Maliyeti nispeten daha düşük olmaktadır

2.1.2 Gezer köprülü tasarım

Gezer köprü tasarımı, CNC yönlendirici modellerinin %95'ini oluşturmaktadır. Tüm x-ekseni boyunca hareket edebilen köprü tasarımı sayesinde, makine boyutunun çok daha büyük yapılabilmesine olanak sağlamaktadır. Ancak boyutun artması ve ağırlığın fazla olması sebebiyle rijit köprü yapımı da zorlaşacaktır.

Gezer köprülü tasarımlarda aksel kuvvetlerin çok daha fazla olması sebebiyle aşınma miktarı en az olan ve yer yönlü kuvveti taşıyan bilyalı vidalar tercih edilir. Sürtünme kayıplarının az olması sebebiyle daha düşük güç gereksinimi olan motorlar tercih edilebilmesi bir avantaj sağlar. Bilyalı vida somunları ve yatakları özel olduğu için diğer

vidalı sistemlere göre 10 kat daha pahalı olabilmektedirler. Özellikle mil çapının artması, fiyatının da aşırı yükselmesine neden olduğundan daha ucuz motor seçme avantajımızın bir anlamı kalmamaktadır. Sonuç olarak;

- Büyük boyutlu bir makine yapılmak istenildiğinde tercih edilebilir,
- İskeletin rijit şekilde yapılabilmesi nispeten daha zordur,
- Boyutta bir sınırlama yoktur. İmkanların el verdiği ölçüde büyük yapılabilir,
- Maliyeti nispeten yüksek olmaktadır.

2.2. CNC Yönlendirici Unsurları

Eğitim amaçlı kullanacağımız makinenin daha işlevsel olması için hibrid bir tasarım mimarisi uygulayacağız. 3B Yazıcı veya CNC yönlendiriciyi kullanmak için makine üzerinde ufak değişiklikler yapılarak dönüşümlü kullanılması hedeflenmiştir. Bu sebeple tasarım modeli olarak sabit köprülü tasarım seçilmiştir. Bu kapsamında hibrid router çalışması anlatılacaktır. Gerekli tüm seçim kriterleri aşağıdaki gibi olmaktadır.

III. CNC YÖNLENDİRİCİ İMALATI

3.1. Kesme Kuvvetinin Belirlenmesi

CNC yönlendirici hesaplarını yapabilmemiz için talaş kaldırma işleminde oluşacak kesme kuvvetlerinin de bilinmesi gereklidir. Kesme kuvvetin belirlenebilmesi için bilmemiz gereken bazı terimler vardır: kesme hızı ve ilerleme. Kesme hızı değeri kesici takım imalatçı firmalarının kataloglarından alınır. Genel olarak kesme hızı HSS kesiciler için **25-50 m/dk** arasında değişmektedir [3]. Kesme hızının formülü aşağıdaki gibidir;

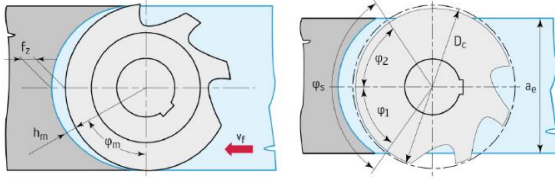
$$V = \frac{\pi D n}{1000} \left[\frac{m}{dk} \right] \quad (1)$$

Burada,

- V: Kesme hızı (m/dk)
- D: Freze kesici çapı (mm)
- n: Devir sayısı (rpm veya devir/dk)

En yüksek kesme hızı için 7 mm kesici çapına sahip parmak freze kullanılırsa,

$$V = \frac{\pi \cdot 7 \cdot 1500}{1000} \cong 33 \text{ m/dk}$$



Şekil 3.1. Parmak freze ile talaş kaldırma [4]

Ortalama talaş kalınlığını hesaplayabilmemiz için diş başı ilerleme miktarını belirlememiz gerekmektedir. İşleyeceğimiz malzemeler ahşap, sert köpük olacak ancak katalogta yer almadıkları için alüminyuma göre hesap yapacağız. Bu durumda parmak freze için f_z veya S_z değeri 0.06 olarak kabul edilmiştir. Ortalama talaş kalınlığı;

$$h_m = \frac{180 \times \sin \gamma \times a_e \times S_z}{\pi \times D \times \sin^{-1} \left(\frac{a_e}{D} \right)} \quad [mm] \quad (2)$$

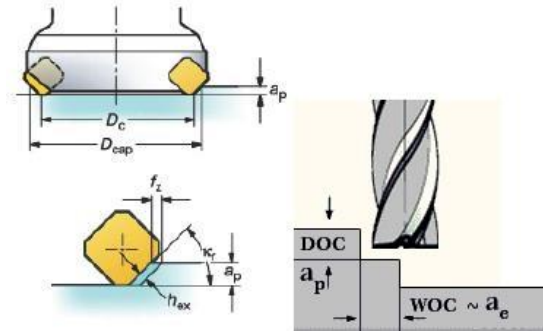
Burada,

- γ : Yaklaşma açısı (parmak freze için 90°)
- a_e : Kesme genişliği (mm) – En yüksek sonuç için 7 mm freze çapı kullanıldı
- S_z : Diş başı ilerleme (mm/diş) olmaktadır. Buna göre:

$$h_m = \frac{180 \times \sin 90 \times 7 \times 0.06}{\pi \times 7 \times \sin^{-1} \left(\frac{7}{7} \right)} = 0.038 \text{ mm}$$

İlerleme hızı (S veya V_f) ise n devir sayısı ve z frezenin diş sayısı olmak üzere aşağıdaki gibidir:

$$S \text{ (feed rate)} = z \cdot n \cdot S_z = 4 \times 1500 \times 0.06 = 360 \text{ mm/dk}$$



Şekil 3.2. Parmak freze değerleri

Frezenin en fazla 5 mm derinliğe inerek paso vereceğini kabul edersek aksel kesme derinliği $a_p = 5$ mm olmaktadır. Kesme kuvvetini hesaplamak için K_s değerini aşağıdaki tablodan 800 N/mm^2 olarak kabul

ettik. Buna göre bir dişe karşılık gelen ortalama kesme kuvveti:

$$F_{S_z} = A_s \times K_s = a_p \times h_m \times K_s \quad [N] \quad (3)$$

$$F_{S_z} = 5 \times 0.038 \times 800 = 152 \text{ N}$$

olmaktadır.

Tablo 3.1. Kolaylaştırılmış yöntemle K_s değeri [5]

Parça Malzemesi	K_s [k/mm ²]
Dökme Çelik	1500
Dökme Demir	2000
Al-Cu-Mg Alaşımı (Duraliminyum)	800

Aynı anda parça ile temasta bulunan diş sayısı ise

$$\varphi_s = 2 \sin^{-1} \left(\frac{a_e}{D} \right) = 2 \sin^{-1} \frac{7}{7} = 180^\circ \text{ ise}$$

$$Z_e = z \frac{\varphi_s}{360^\circ} = 4 \frac{180^\circ}{360^\circ} = 2 \text{ diş}$$

Buna göre ortalama kesme kuvveti,

$$F_S = Z_e \times F_{S_z} = 2 \times 152 = 304 \text{ N}$$

Kesme gücü,

$$P_S = \frac{F_S \times V}{60 \times 1000} \quad [kW] \quad (4)$$

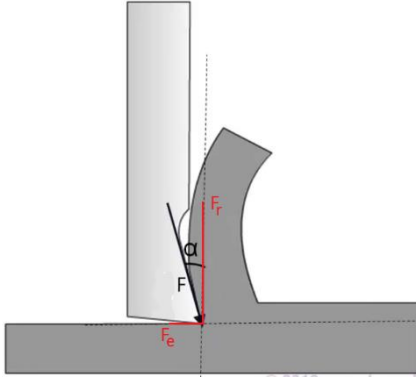
$$P_S = \frac{304 \times 33}{60 \times 1000} = 0.167 \text{ kW}$$

Mabuchi RS-775 DC motor kataloğuna göre en fazla 300 W güce sahiptir [6]. Bu durumda redüktör kullanmadan da çalışmaya uygundur. Kesme kuvvetlerinin arasında kesin bağıntılar olmamakla birlikte deneylere dayanarak oluşturulmuş ampirik bağıntılar bulunmaktadır [5]. Aynı yönde frezeleme yöntemine göre radyal ve aksel (ilerleme) kuvvetlerini hesaplırsak:

$$F_{\text{radyal}} = 0.8 \times F_S = 0.8 \times 304 = 243.2 \text{ N}$$

$$F_{\text{eksel}} = 0.9 \times F_S = 0.9 \times 304 = 273.6 \text{ N}$$

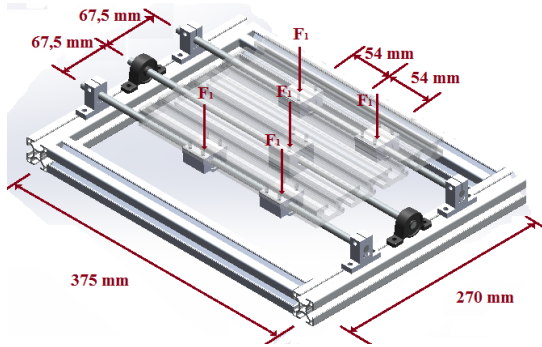
olmaktadır.



Şekil 3.3. Kesme kuvvetleri

3.2. X Ekseninin Tasarımı

Bu bölümde, cnc tablası üzerindeki bir iş parçası için en fazla ne kadar ağırlığa kadar düzenli çalışmaya uygun bunu hesaplayacağız. Maliyet açısından takım halinde tedarik etmeye en uygun seçim Çin menşeli metrik ACME T8 tipi trapezoidal vidalar olmaktadır. Vida yataklaması için UP08 serisi rulmanlı yataklar kullanılmıştır. Destekler için mil-lineer yatak sistemi kullanılacaktır. Bu amaçla trapezoidal vida uzunluğuna göre belirlediğimiz 375 mm uzunluğunda 8 mm çapında 2 adet krom kaplı indüksiyonlu taşlanmış mil kullanılmıştır. CK 55 malzemesinden imal edilmiş olup 62+/-2 HRC yüzey sertliğindedir [7]. Her mil üzerinde iki adet SC08UU serili lineer rulman kullanılacağı için mil yüzeyinin aşınmaması önem arz etmektedir. Yüzey sertliğinin düşük olması çizilmeyi, bu da rulmanın pis bir yüzeyde çalışmasına ve kısa zamanda bozulmasına sebep olmaktadır. Bu yüzden sertleştirilmiş mil kullanılmasını tercih ettik.



Şekil 3.4. X Eksenini gösterimi

Nema 17 serisi motorlar nispeten ucuz ve daha kolay bulunabilmektedir. İş parçası ağırlığı, kesme kuvveti vb. etkilerden dolayı cnc yönlendiricinin en fazla torka ihtiyaç

duyduğu kısım bu bölümdür. Bu sebeple nema 17 serilerinden en yüksek torklu motor seçilmiştir.

Başlangıç değerlerimiz

- Metrik ACME T8 Trapezoidal vida malzemesi 304 kalite paslanmaz çeliktir.
- T8 Trapezoidal vidanın akma mukavemeti: 205 Mpa [8]
- T8 Trapezoidal için pitch = 2 mm ve lead = 8 mm
- T8 Sürtünme katsayısı f = 0.08
- Nema 17 17HS8401 adım motoru çalışma torku 52 N.cm
- Nema 17 17HS8401 adım motoru frenleme torku: 2.6 N.cm

Aktarma vidası için katalogta önerilen formüle göre hesaplırsak

$$\text{Helis Açısı} = \tan^{-1}\left(\frac{\text{lead}}{\pi d}\right) \quad (9)$$

$$= \tan^{-1}\left(\frac{8}{8\pi}\right) = 17.66^\circ$$

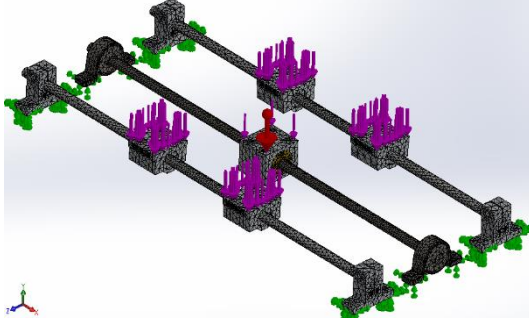
$$\text{Verim} = \frac{\tan(\text{helis açısı})}{\tan(\text{helis açısı} + \arctan f)} \times 100 \quad (10)$$

$$\delta = \frac{\tan(17.66)}{\tan(17.66 + \arctan 0.08)} \times 100 = \%77.88$$

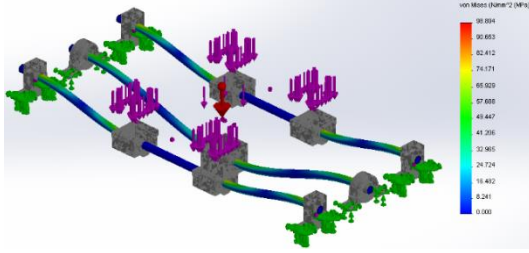
$$\text{Tork} = \frac{\text{Yük [N]} \times \text{Lead [mm]}}{2\pi\delta} \quad [\text{Nmm}] \quad (11)$$

$$\text{Yük} = \frac{2\pi \times 0.7788 \times 520}{8} = 318 \quad [\text{N}]$$

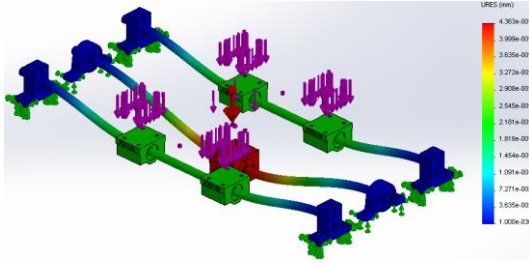
İvmesiz hareket altında Nema 17 17HS8401 model motor ile en fazla 32 kg yük taşıma kapasitesine sahiptir. Radyal kesme kuvveti (243 N) ve tabla üzerindeki ağırlıklar göz önüne alındığında güvenli olarak 7,5 kg yük rahatça işlenebilir. Ayrıca daha düşük freze çakısı çapları kullanıldığında tablanın güvenli taşıma kapasitesinin artacağı da unutulmamalıdır. Montajlanan sistemde farklı malzeme grupları olduğu için Solidworks 2013 Simulation Pro programı ile statik analiz yapılarak yük dayanımı test edilmiştir. Analizin çözümünü kolaylaştırmak için bazı kabuller yapıldı. Lineer rulmanlara düşey yönlü olarak toplamda 350 N kuvvet uygulandığı, yatak ve mil tutucuların şekil değiştirmez olarak sabitlendiği kabulü ile analiz gerçekleştirilmiştir. İterasyon işleminin uzun sürmesi ve PC belleği yetersiz kaldığı için kaba mesh işlemi uygulanabilmiştir.



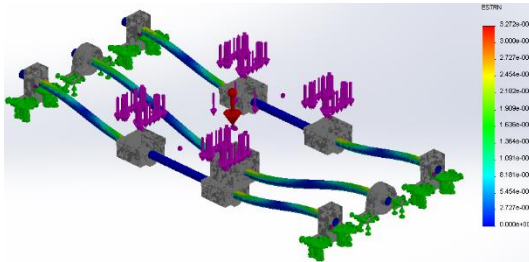
Şekil 3.5. X Eksenli statik analiz



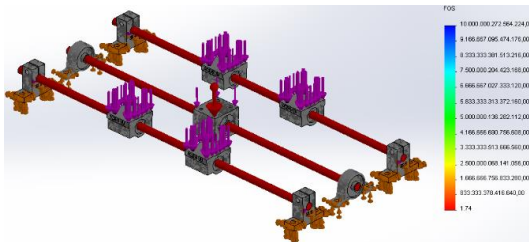
Şekil 3.6. X Eksenli von mises stresi



Şekil 3.7. X Eksenli yer değiştirme



Şekil 3.8. X Eksenli eş değer gerilme

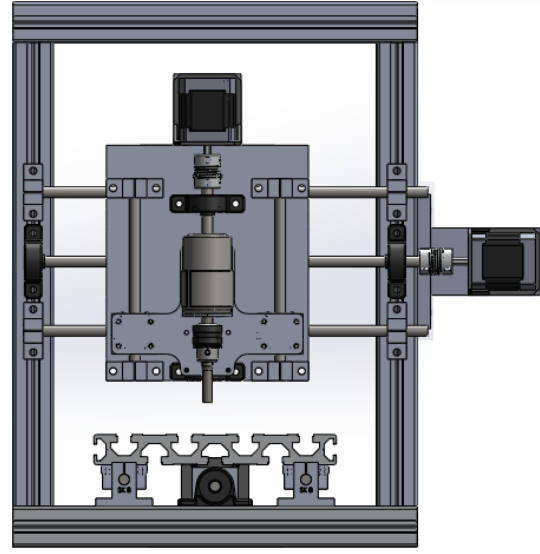


Şekil 3.9. X Eksenli güvenlik faktörü

En düşük güvenlik faktörü vida ve mil üzerinde olmasına karşın sistemi 1,74 kat emniyetli taşıyabilmektedir. En yüksek Von Mises gerilmesi (90 Mpa), trapez vidanın akma mukavemeti 205 Mpa ve krom kaplamalı mil için akma mukavemeti 550 Mpa değerlerini geçmediğinden herhangi bir kalıcı plastik deformasyonun oluşması söz konusu değildir.

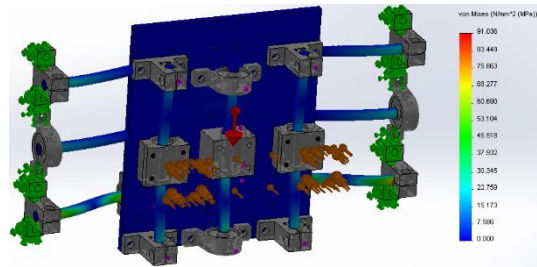
3.3. Y ve Z Ekseninin Tasarımı

Kesme kuvveti etkisiyle Y ve Z eksenlerinde iki yönlü eğilme momenti oluşacaktır.

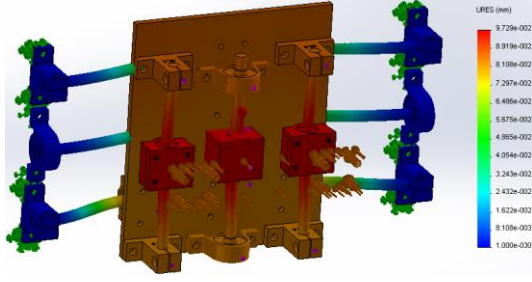


Şekil 3.10. Önden görünüş

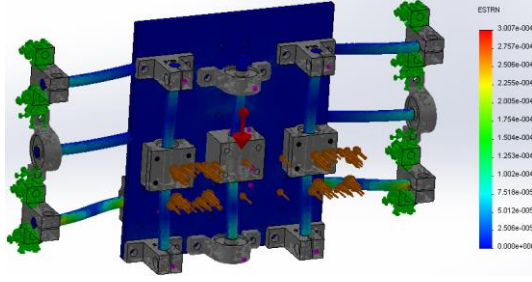
Problemi basitleştirmek kesme kuvvetini (304N) lineer rulmanlı yataklara ve vidalı mil somununa dağıttık.



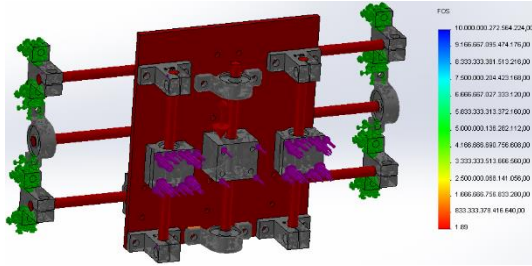
Şekil 3.11. YZ Eksenli von mises stresi



Şekil 3.12. YZ Eksenli yer değiştirme



Şekil 3.13. YZ Eksenli eş değer gerilme



Şekil 3.14. YZ Eksenli güvenlik faktörü

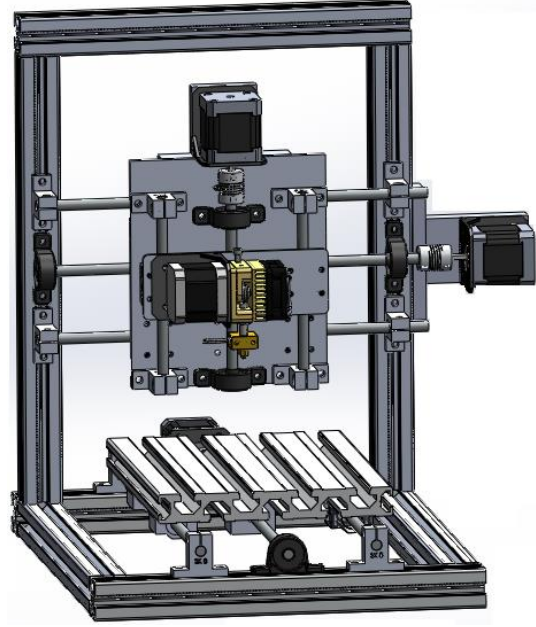
Analiz sonucunda 1,89 kat emniyetli olarak sistem çalışacaktır. X eksenine göre daha az kuvvetlere maruz kaldığından sonucun güvenli çıkması tabiidir.

IV. ÜÇ BOYUTLU BASKILAMA

Hızlı prototipleşmeye imkân verdiği için endüstriyel alanda kendisine yer bulan 3B yazıcı teknolojisi, baskı malzemesi fiyatlarının düşmesiyle birlikte artık bireysel ihtiyaçlara da cevap verebilmektedir. Mühendislik öğrencilerinin projelerini gerçek bir modele dönüştürebilmesi, üzerinde çalışabilir olması ve koşullarını sınavabilmeleri daha üretken olmalarını sağlamakla birlikte mühendislik konularını uygulamalı olarak kavrama şansı elde edebilmektedir.

Bu çalışma kapsamında imalatını gerçekleştirdiğimiz CNC yönlendiriciye ufak eklemeler yaparak en kolay yoldan şekil 4.1'deki gibi bir 3B yazıcı tasarladık. Böylece aynı tezgâh ile hem talaşlı imalat hem de

plastik model üretimi yapılabilecek bir hibrid sistem tasarlanmış oldu. CNC yönlendirici tasarımı aynen kullanılacağı için mekanik unsurlarda bir değişiklik olmamıştır. 3B Baskı teknolojisine özel kavramları incelememiz bizim için yeterlidir.



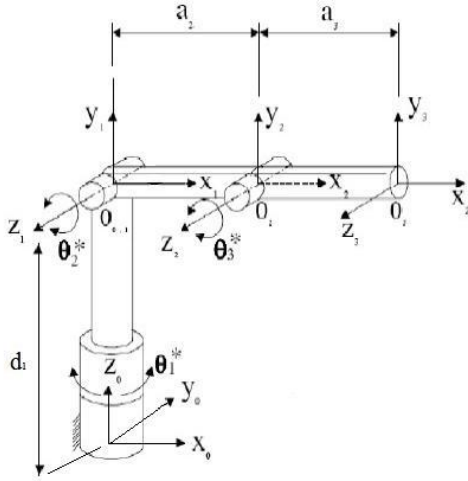
Şekil 4.1. 3B Yazıcı

V. ÜÇ EKSENLİ ROBOT KOL

Bilgisayarlı bütünleşik imalatın temel unsuru olan robot manipülatörler, endüstri 4.0 ile birlikte daha da önem kazanmıştır. Bu çalışma kapsamında eğitimde kullanılacak 3 eksenli robot kol tasarımı, hesaplaması ve imalatı ile kontrol yazılımı gerçekleştirilecektir.

5.1. Robot Kinematığı

Geometrik bir sistemin yapısını ve hareketlerini inceleyen bilim dalına kinematik denir. Bir sistemin belirli şartlar altında nasıl hareket ettiğini anlayabilmek için bu sistemin kuvvet, atalet gibi büyüklükleri yani dinamiği hakkında bilgi sahibi olmamız gerekmektedir. Robotun ileri yön kinematığı, robot bağlarının konumları, hızları ve ivmeleri arasındaki ilişkidir. Robotun eklem değişkenlerinin belirlenmesinde Denavit – Hartenberg yöntemi uygulanacaktır.



Şekil 5.1. Eksen yerleştirme

Elde edilecek transfer fonksiyonları robot manipülâtörün uç ve eklem koordinatlarını vermektedir. Yazılım bölümünde bu transfer fonksiyonları kullanılacaktır.

Tablo 5.1. Denevit – Hartenberg gösterimi

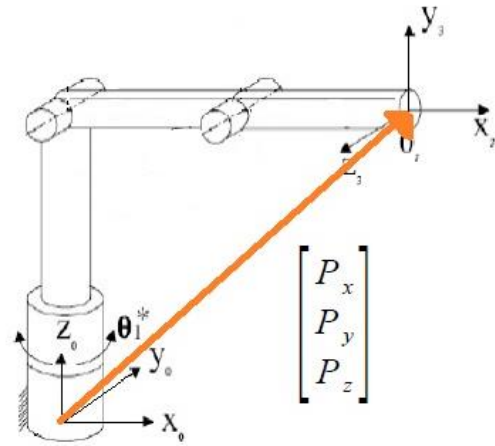
D-H Değişkenlerinin Belirlenmesi				
i	a_i	α_i	d_i	θ_i
1	0	90°	d	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

5.1.1. İleri kinematik hesaplaması

İleri kinematik hesaplamaları manipülâtörün eklem açıları girildiğinde uç noktanın koordinatlarını belirlemeye yarar. Bu koordinatları aynı zamanda kontrol yazılımında kullanacağız. Denevit – Hartenberg yönteminden faydalanarak dönüşüm matrislerini hesaplanacaktır.

Aşağıdaki matris P_x , P_y ve P_z vektörleri robot kolunun uç noktasının konum koordinatlarını vermektedir.

$${}^0_3T = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} a_2 C\theta_1 C\theta_2 + a_3 C\theta_1 C\theta_3 C\theta_2 - a_3 C\theta_1 S\theta_2 S\theta_3 \\ a_2 C\theta_2 S\theta_1 + a_3 C\theta_2 C\theta_3 S\theta_1 - a_3 S\theta_2 S\theta_3 S\theta_1 \\ d_1 + a_2 S\theta_2 + a_3 C\theta_3 S\theta_2 + a_3 C\theta_2 S\theta_3 \\ 1 \end{bmatrix}$$



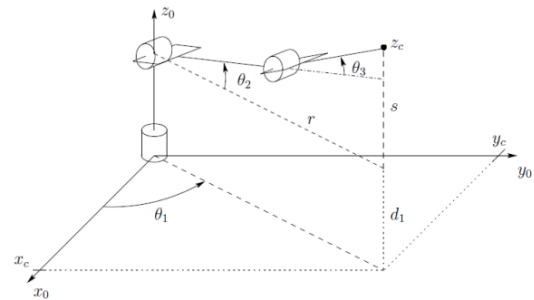
Şekil 5.2. Konum vektörü

5.1.2. Ters kinematik hesaplaması

Ters kinematik probleminin çözümü için iki farklı alternatif çözüm yolu yapılacaktır.

5.1.2.1. Birinci Yöntem

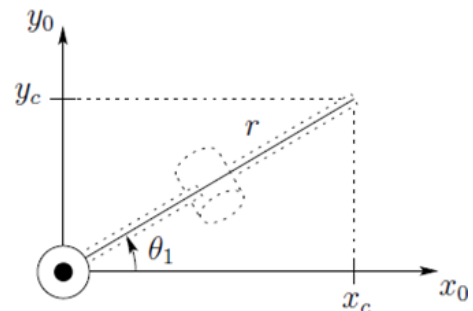
Bu yöntemde çözüm 2 alt probleme ayrılarak sonuca ulaşmaktadır.



Şekil 5.3. Ters kinematik

1. Alt Problem

Sistem diğer uzunlarından bağımsız olarak ilk önce θ_1 açısının hesabı yapılır. Bu açının referansının X_0 , Y_0 , Z_0 'a göre olduğuna dikkat ediniz.



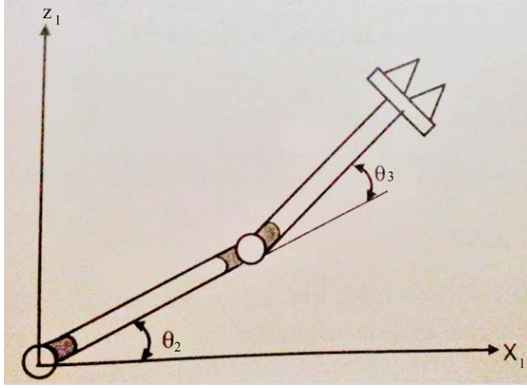
Şekil 5.4. Üstten görünüş

Buradan θ_1 değeri aşağıdaki gibi olur.

$$\theta_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right)$$

2. Alt Problem

İkinci ve üçüncü uzvu birlikte X_1, Y_1, Z_1 eksen takımına göre bağımsız analitik çözüm yapılır. X_0, Y_0, Z_0 ye göre verilen koordinatlar X_1, Y_1, Z_1 eksen koordinatlarına dönüşüm yapılarak verilir.



Şekil 5.5. X-Z Görünüş [9]

Buna göre,

$$D = \frac{(P_x / \cos(\theta_1))^2 + (P_z - d_1)^2 - a_2^2 - a_3^2}{2 \cdot a_2 \cdot a_1}$$

$$\theta_3 = \tan^{-1} \left(\frac{\pm \sqrt{1 - D^2}}{D} \right)$$

$$\theta_2 = \tan^{-1} \left(\frac{(P_z - d_1) \cdot \cos(\theta_1)}{P_x} \right)$$

$$\pm \tan^{-1} \left(\frac{\sqrt{(P_z - d_1)^2 + (P_x / \cos(\theta_1))^2 - (a_3 \cdot \cos(\theta_3) + a_2)^2}}{a_3 \cdot \cos(\theta_3) + a_2} \right)$$

Olmaktadır. Bunu örnek ile irdeleyerek daha net görebiliriz.

5.1.2.2. İkinci Yöntem

Bu yöntem ile eksenlerin yerini değiştirmeden çözümü vermektedir. Buna göre;

$$\theta_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right)$$

$$F = \frac{P_x^2 + P_y^2 + (P_z - d_1)^2 + a_2^2 - a_3^2}{2 \cdot a_2 \cdot \sqrt{P_x^2 + P_y^2 + (P_z - d_1)^2}}$$

$$\theta_2 = \tan^{-1} \left(\frac{P_z - d_1}{\sqrt{P_x^2 + P_y^2}} \right) + \tan^{-1} \left(\frac{\pm \sqrt{1 - F^2}}{F} \right)$$

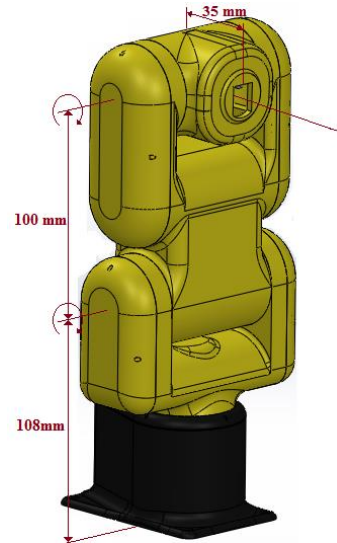
$$G = \frac{P_x^2 + P_y^2 + (P_z - d_1)^2 - (a_2^2 + a_3^2)}{2 \cdot a_2 \cdot a_3}$$

$$\theta_3 = \pm \tan^{-1} \left(\frac{\sqrt{1 - G^2}}{G} \right)$$

Her iki yöntemin karşılaştırmasını yaparsak, son çözüm yolumuzda her bir denklem içerisinde x, y ve z konumları mutlaka bulunmaktadır. Ancak ilk yöntemde z koordinatı yer almadığı için kontrol gerçekleştirilirken fazladan fonksiyon kullanılması gerekli kalmaktadır. Bu sebeple yazılım kontrolü yapıldığında son çözüm yöntemi temel alınarak fonksiyon yazılmasına dikkat edilmiştir.

5.2. Manipülatör Tasarımı

Üç eksenli robot manipülatörün tasarımını belirleyecek en önemli unsur taşıyacağı ağırlıktır. Kendi ağırlığı, hareket sırasındaki dinamik yükler ve motor ataleti göz önünde bulundurularak boyutlandırma ve motor seçimi yapılması gerekmektedir. Sonraki bölümde ele alınacak kontrolcü yazılımının gerçekleştirilebilmesi için rc servo tip motor kullanılmıştır. Motor seçimini yapmadan önce taslak bir model hazırlamamız gerekmektedir. Bu modeli referans olarak başlangıç kütle ve mafsallar arası uzunlukları bir başlangıç koşulu olarak kullanacağız.

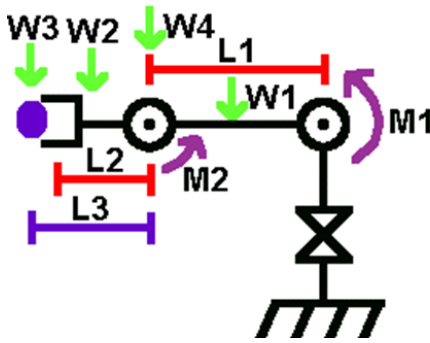


Şekil 5.6. Üç eksenli manipülatör modeli

Ucuza imal etmek için Tower Pro SG90 RC Mini Servo kütle ve tork değerlerine göre optimum tasarım yapılmıştır.

5.2.1. Statik kuvvet hesabı

Eklemlere binecek kuvvetleri ve oluşturacakları momentlerin tespit edilmesi sayesinde minimal motor değerlerimizi tespit edebiliriz. Motorlar sadece robot kolunun kendi ağırlığını taşımakla kalmaz, aynı zamanda tutulacak cisminde hedefe güvenli bir şekilde ulaştırılmasını sağlayan önemli bir unsurdur. Taşınacak cismin ağırlığı ve atalet değerleri robotun hassasiyetliğinin ölçütünü belirleyecektir.



Şekil 5.7. Üç eksenli manipülör için serbest cisim diyagramı

İlk adım olarak en fazla uzunluğu gösterecek şekilde robot kolumuzun şekil 5.7 'deki gibi bir serbest cisim diyagramı çizmek ile başlıyoruz. Sonraki adımda aşağıdaki gibi bir moment kolu hesabı yapıyoruz.

0. Eklem için statik moment:

$$M_0 = 0$$

1. Eklem için statik moment:

$$M_1 = (KMU_o \times W_1) + (L_1 \times W_4) + (KMU_d \times W_2) + (L_1 + L_3) \times W_3 \quad [kg.cm]$$

$$W_1 = 0,06226 \text{ kg}, W_2 = 0,0144 \text{ ve}$$

$$W_3 = 0,01 \text{ kg}, W_4 = 0,009 \text{ kg ise}$$

$$M_1 = 4,7 \times 0,06226 + 10 \times 0,009 + 11 \times 0,0144 + (10 + 3,5) \times 0,01$$

$$M_1 = 0,67 \quad [kg.cm]$$

2. Eklem için statik moment:

$$M_2 = (KMU_d \times W_2) + (L_3 \times W_3) \quad [kg.cm]$$

$$M_2 = ((11 - 10) \times 0,0144) + (3,5 \times 0,01) \quad [kg.cm]$$

$$M_2 = 0,0494 \quad [kg.cm]$$

5.2.2. Hız ve hareket planlaması

Uç noktanın hız değerinin hesaplanması matematiksel olarak biraz karmaşıktır. Ancak biz burada RC Servo motor kullandığımız için motorun en fazla zorlanacağı konuma göre maksimum hızı bulmaya çalışacağız. Kolun en hızlı olacağı pozisyon kolların merkezden en uzak olacağı konum olacağından basitçe $\vartheta = \omega \cdot r$ formülünü kullanarak dairesel hızını bulabiliriz. J_0 eklemının devir sayısı ve uç noktanın merkeze göre olan uzaklığını bilmemiz yeterli olacaktır. Eğer diğer eklemlerde dönüş olsaydı bu vakit hızımız daha farklı olacak ve hesaplaması zorlaşacaktı bu yüzden sadece tek bir eklemi hareketli olduğunu ve merkez noktaya göre dönüşündeki hızının analizini buluyoruz:

$$Uç \text{ Noktanın Hızı} = \omega \cdot r = (2\pi r) \times rpm$$

$$rpm = \frac{\text{devir sayısı}}{\text{dakika}}$$

RC Servo motorların çoğu en fazla 180° açı değerine kadar dönebilmektedir. Genellikle katalog üzerinde belirtilen açısal hız değerleri $\frac{1}{\omega} = \frac{\text{Saniye}}{60^\circ}$ şeklinde olmaktadır. SG 90 Micro Servo Modeli bir motoru 5 cm'lik bir uzaklıkta 180° açı yol aldığıında maksimum uç nokta hız değeri aşağıdaki gibi olacaktır.

(SG 90 Micro Servo için işletme hızı $0.12s/60^\circ$)

$$V = \frac{\omega \cdot r}{180^\circ} = \frac{60^\circ \times \pi \times 5cm}{180^\circ \times 0.12s} = 43,63 \frac{cm}{s}$$

$$= 0.4363 \text{ m/s olacaktır.}$$

5.2.3. Dinamik kuvvet hesabı

Bu bölüme kadar olan kısımda robot kolun statik yani her zaman için durağan koşullarda üzerine etkiyen kuvvet değerlerine göre hesap yaptık. Ancak hareketli cisimlerin dönmeye karşı kütle ve yüzey şeklini değiştirmeye çalışan kuvvetlere karşı da bir tepkisi olacaktır. İşte bu tepkiye atalet momenti demekteyiz. Kısaca atalet momenti, rijit cisimlerin kendi dönme hareketlerindeki değişime karşı gösterdiği eylemsizlik tepkisidir. Atalet kuvvetleri göz önünde

bulundurarak dinamiksel tork ifadesi oluşturmamız gereklidir.

Kolaylık olması açısından robot kol uzuvlarını L uzunluğunda ve m kütlesinde ince çubuk olduğunu kabul ettik. Çubuğu merkez noktasından döndürmek istediğimizde oluşacak atalet momenti:

$$I_{\text{merkez}} = \frac{mL^2}{12}$$

Dönme eksenini çubuğun sonunda olduğundaki atalet momenti ise,

$$I_{\text{son}} = \frac{mL^2}{3} \text{ olmaktadır.}$$

Robotumuzun J_0 eklemi için I_{merkez} J_1 , J_2 için ise I_{son} atalet momentlerinin değerleri kullanılır. Bu ifadelere göre dinamiksel tork ifademizi tanımlarsak;

$$\text{Tork} = \text{Atalet Momenti} \times \text{Açısal İvme}$$

Veya $\tau = I \cdot \alpha$ olmaktadır. İfadeyi daha açarsak;

$$\text{Tork} = \text{Kütle} \times \text{uzaklık}^2 \times \left(\frac{\text{Açısal Hızdaki Değişim}}{\text{Zamandaki Değişim}} \right)$$

Burada,

Açısal Hızdaki Değişim

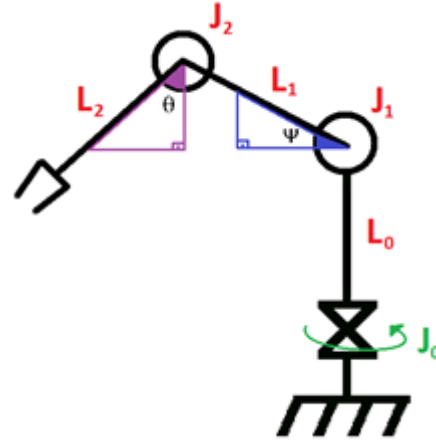
$$= \text{Son Açısal Hız} \\ - \text{İlk Açısal Hız} = \omega_1 - \omega_0$$

$$\text{Açısal ivme}(\alpha) = \frac{\omega_1 - \omega_0}{\Delta t} \text{ olmaktadır.}$$

Robotumuzu J_0 ekleminden 180° açı ile Sg 90 model servo motor ile 0,3 saniyede alacağı dinamiksel tork değerini bulmak isteyelim. Başlangıç anındaki ($t_0 = 0$) açısal hızı $\omega_0 = 0$ olarak kabul edersek (yani ilk hızsız olarak harekete başlarsa) 0.3 saniye sonundaki açısal hız ve ivme değerleri aşağıdaki gibi olmaktadır. (Sg90 model 180° 'yi 0.3s almaktadır)

$$\text{Açısal Hız} (\omega) = \frac{\theta_1 - \theta_0}{\Delta t} = \frac{\pi}{t} = \frac{\pi}{0.3} = 10.47 \frac{\text{Rad}}{\text{s}}$$

$$\text{Açısal ivme}(\alpha) = \frac{\omega_1 - \omega_0}{\Delta t} = \frac{10.47}{0.3} = 34.9 \frac{\text{Rad}}{\text{s}^2} \text{ olmaktadır.}$$



Şekil 5.8. Üç eksenli manipülatörün dinamik analizi

Her bir kolün kütlesi sırasıyla m_0 , m_1 , m_2 ve her bir motorun kütlesi de j_0 , j_1 , j_2 olmak üzere kolları gidebileceği en uzak konuma kadar açarak atalet Momentlerini hesaplırsak;

$$I_0 = \frac{(m_1 + m_2 + j_1 + j_2)(L_1 + L_2)^2}{3}$$

$$I_1 = \frac{(m_1 + m_2 + j_2)(L_1 + L_2)^2}{3}$$

$$I_2 = \frac{(m_2)(L_2)^2}{3}$$

Olmaktadır. Atalet değerleri dinamiksel tork formülünde yerine konarak hareketli bir cisim döndürecek tork değerini bulmuş olduk. Bazı durumlarda dinamik değerler statik değerlerden fazla olmaktadır. Bu durumda en büyük tork değeri tercih edilmelidir. Daha emniyetli bir sonuca ulaşılmak isteniyorsa ve değerleriniz ufak ise tork değerlerinin toplamı tercih edilir.

$$\tau_{\text{motor}} = \tau_{\text{statik}} + \tau_{\text{dinamik}}$$

Buradaki formüller yazılım kısmında kullanılarak bir hesap makinesi yapılacaktır. Bulunan değerler kullanılarak seçtiğimiz motorun uygunluğu tespit edilecektir.

VI. YAZILIM

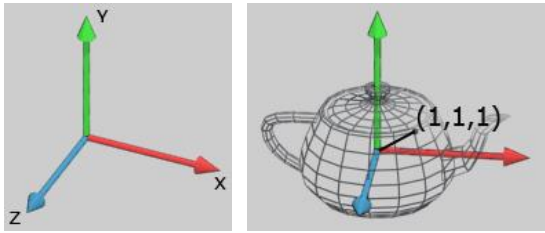
Bu bölümde bir robot kolun simülasyonu, gerçek zamanlı kontrol işlemi, hesaplama ve sonuçların çıktısını almayı gösteren bir yazılım anlatılacaktır.

C# nesneye yönelik programlama dili olmasına karşın 3B modellerin hareketi, konumu gibi önemli parametrelerin kendi içerisinde tanımlanmaya ihtiyacı

bulunmaktadır. Bilgisayar grafiğinde bazı kavramları ilerleyen bölümlerde sık kullanacağımız için tanımlama yapmakla başlayalım.

6.1. Bilgisayar Grafiği

Bir bilgisayar ekranındaki pikseller iki boyuta sahiptir ancak görüntülemek istediğimiz nesne eğer üç boyuta sahipse bunu ancak derinlik algısı yaratarak oluşturabiliriz. Bu algıyı oluşturmak için ışıklandırma, konum için ise vektör uzayı kavramlarını bilmemiz gerekmektedir. Ayrıca nesnenin hareketini yani pikseller arası geçişlerin matematiğini bizlere matris çözümlerini verecektir.



Şekil 6.1. Vektör tabanı [10]

Matris dönüşüm zincirleri bir 3B nesnenin 2B ekran üzerinde temsil edilmesine izin vermeyi sağlar. Bir vektör uzayı ise lineer olan bağımsız vektörlerin sayısına göre tanımlanan matematiksel bir yapıdır. Buna vektör tabanı da denilmektedir. Örnek olarak üç boyutlu bir uzay, üç taban vektörüne sahip olur. Bu taban vektörleri uzayda tüm diğer vektörleri elde etmek için ölçeklenebilir ve birbirine eklenebilir. Vektör uzayı aslında çok geniş bir konudur ve bizim amacımız tüm bu başlıkları açıklamak değildir. Bizim hedefimiz gerçek bir özel vektör uzayındaki modelimizi, model uzayı ismi altında incelemek ve kabul edilmiş 3B koordinat sistemi ile temsil etmektir.

Şekil 6.1 ele alırsak tüm köşeler model uzayın orijinine göre yapılmıştır. Bu yüzden eğer biz model uzayda (1,1,1) koordinatında bir noktaya sahipsek, nerede olduğumuzu tam olarak biliriz. Her model kendi model uzayında gerçek ve hareketlidir, eğer istersek bu modelleri herhangi bir mekânsal ilişki içerisinde sokabiliriz (Çaydanlığı masa üstüne koymak gibi).

Bunları yapmak için ortak bir uzaya (genellikle World uzayı olarak isimlendirilir) onları dönüştürmeniz gereklidir. Bir vektörün, sadece koordinat sistemi içinde anlamı

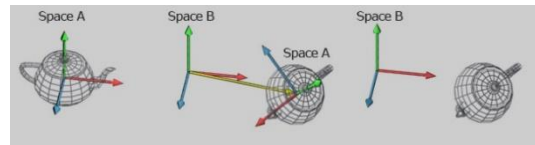
olduğunu anlamanız önemlidir. Biz uzay belirtmiyorsak herhangi bir noktayı tanımlayamayız. Modeller dışa aktarıldıktan (export) sonra tüm köşeler model uzayında tanımlanmış olur. Eğer bir objeyi içeri almak (import) istersek, onu istenilen konuma taşımamız gerekecektir. Bunu da nesnenin içerisindeki world uzayına yerleştireceğiz. Taşıma, döndürme ya da ölçeklendirmeyi (scaling) bir nesnenin transformasyonu olarak isimlendiriyoruz. Tüm nesnelere ortak bir uzay (World Uzay) içerisinde dönüştürülmektedir. Onların köşeleri kendi world uzayına bağlı olacaktır.

6.1.1. Dönüşüm kavramı - transformation

Herhangi bir değişiklik için, vektör uzayındaki dönüşümü basitçe başka bir uzaydan görebiliriz. Vektör dönüşümlerinin en zor kısmı bitleridir (parçaları) bu yüzden her birini açık ve anlaşılır yapmaya çalışacağız.

Bir 3B vektör uzayında şekil 6.1'deki gibi 3 ortogonal eksen olduğunu hayal edeceğiz. Daima "aktif" bir uzaya sahip olmamız gerekiyor. Böylece bu uzayı her şeyin referansı gibi kullanabiliriz (Ya da hem geometri hem de diğer uzaylar için). Eğer her birisi kendi Model uzayında olmak koşuluyla 2 modelimiz varsa, onları ortak bir "aktif" uzayda tanımlayana kadar her ikisini çizemeyiz.

Şimdi aktif bir uzay kavramı ile başlayalım ve bu uzaya A diyelim; bu da bir çaydanlık içeriyor olsun. Şimdi biz A uzayında her şeyi yeni bir konum içerisine taşıyan bir dönüşüm uygulamak istiyoruz diyelim. Ancak eğer taşırsak bizim A uzayını, sonradan dönüştürülmüş A uzayını temsil edecek yeni bir aktif uzay tanımlamamız gerekecek. İşte bu yeni aktif uzayımıza B uzayı olarak isim veriyoruz.



Şekil 6.2. Vektör uzayı [10]

Dönüşümden önce, A uzayındaki herhangi bir nokta tanımlıdır. A uzayının orijini başlangıç konumuna göredir (şekil 6.2'de en soldaki). Dönüşüm uygulandıktan sonra tüm noktalar şimdi yeni aktif uzaydaki durumuna bağlıdır: B Uzayı (sağdaki). A uzayındaki herhangi bir işlemde B uzayında yeniden tanımlanır ve bu

bir dönüşüm olur. Nasıl olduğuna dikkat edin, dönüşümden sonra A uzayı B uzayı içerisinde kayboluyor ya da daha kesin olarak onu B uzayında yeniden haritalandırıyor. Böylece herhangi diğer dönüşümleri uygulamamız için bizim başka yolumuz olmayacaktır. (Dönüşümleri geri almadıkça ve A uzayını "aktif" uzay yapmadıkça tekrar)

Bunu görmenin diğer yolu, tüm bunları bir uzayda taban vektörleri ile hareket ettiğini hayal edin ve A uzayını B uzayı üstünde mükemmel bir şekilde çakıştırılmış olduğunu düşünün. Dönüşümü yaptığımız zaman bizim A uzayı B uzayından ayrılıyor ve A uzayındaki tüm şeyler onun ile hareketleniyor. Tüm köşeler taşındıktan sonra bizim B uzayımızdaki konumlara göre tanımlı oluyor ve dönüşümü tamamlamış oluyoruz.

Bu durumda A uzayındaki işlemleri tekrar tersine dönüşüm yapmamız (Inverse Transformation) ile B uzayını elde etmemiz de mümkündür. Böylece B uzayı yeniden haritalanabiliyor olacak A uzayı içerisinde yeniden (ve bu noktaları biz "kaybederiz" B uzayında). Eğer hem dönüşümünü hem de onların tersini (inverse) biliyorsak daima yeniden 2 uzayı, birini diğerinden yeniden haritalandırabiliriz. Vektör uzayındaki dönüşümleri ölçeklendirme (scale), öteleme (translation), döndürme (rotation) için kullanırız. Burada önemli bir nokta, tüm dönüşümler daima orijine göre tanımlıdır. Eğer 90° sol ve sonra öteleme yaparsak farklı bir şey eğer ilk öteleme sonra 90° döndürme yaparsak farklı şeyler yapmış oluruz.

6.1.2. Dönüşüm matrisi – transformation matrix

Biliyoruz ki bir dönüşüm bir uzaydan diğerine olan değişimdir ve bunu matematiksel olarak yapabiliriz. Eğer 3B uzayı olan bir dönüşümü diğerine temsil etmek istersek 4x4'lük bir matrise ihtiyacımız olacak. OpenGL'de olduğu gibi, bir sütun vektör notasyonu üzerinden bunu anlatmaya çalışalım. Eğer satır vektörleri içerisindeyseniz sadece devrik (transpose) matrise ve çarpılabilir vektörlere ihtiyacınız olacaktır. Dönüşümleri uygulamak amacıyla tüm vektörleri çarpılabilir yapmamız gereklidir. Eğer A uzayındaki vektörler ve dönüşümler A uzayının yeni konumu B uzayına göre tanımlanabiliyorsa, çarpmadan sonra tüm vektörler B uzayında sonradan tanımlanabilir olacaktır.

6.1.2.1. Öteleme matrisi – translation matrix

Öteleme bir 3B vektörü uzayımızda taşımak istediğimiz bir konumu temsil etmektedir. Bir öteleme matrisi, aktif uzayda tam olarak döndürülmüş tüm eksenleri bırakır.

$$\begin{bmatrix} 1 & 0 & 0 & Translation.x \\ 0 & 1 & 0 & Translation.y \\ 0 & 0 & 1 & Translation.z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6.1.2.2. Ölçekleme matrisi – scale matrix

Her bir eksen boyunca ölçeği tanımlayan bir 3D vektördür. İlk sütunu okursanız yeni X eksenin nasıl olduğunu görebilirsiniz. Hala aynı yöne bakacak şekilde duruyor ancak Scale.x tarafından ölçeklendirilmiş şekildedir. Bunun yanı sıra tüm diğer eksenlerde de aynı şeyler olur. Ayrıca translation sütunlarının (en sağdaki siyah olan sütün) bu sefer 0 olduğuna dikkat edin, bunun anlamı öteleme yapılmadığını gösteriyor.

$$\begin{bmatrix} Scale.x & 0 & 0 & 0 \\ 0 & Scale.y & 0 & 0 \\ 0 & 0 & Scale.z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

6.1.2.3. Dönme matrisi – rotation matrix

Teta, bizim dönüş yapmak için kullanmak istediğimiz açıdır. X eksenini etrafında dönüş yapıyor olacağımızdan ilk sütünün asla değişmeyecektir. Ayrıca tetanın 90°'lik değişikliğinin nasıl Y eksenini Z eksenine içerisinde ve Z eksenini de -Y eksenine içerisinde yeniden haritalandığını görebiliyoruz.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

X eksenini dönme matrisi

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Y eksenini dönme matrisi

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Z eksenini dönme matrisi

Z ve Y eksenini için dönme matrisleri X eksenini etrafındaki dönme matrisiyle aynı davranışlara sahiptir. Biz burada sadece en çok kullanılan bazı matrisleri anlattık. Birbiri ardına matrisleri çarpım şekline getirerek onları zincir dönüşümü şeklinde yapabilirsiniz. Sonuç tam bir dönüşümü şifreleyen tek bir matris olacaktır. Dönüşüm bölümünde gördüğümüz gibi transformasyonları uygulamalı olarak kullanabilmemiz çok önemlidir. Matematikteki çarpımda yer değişim özelliği burada biraz farklıdır çünkü [öteleme x dönme] ile [dönme x öteleme] birbirinden farklıdır.

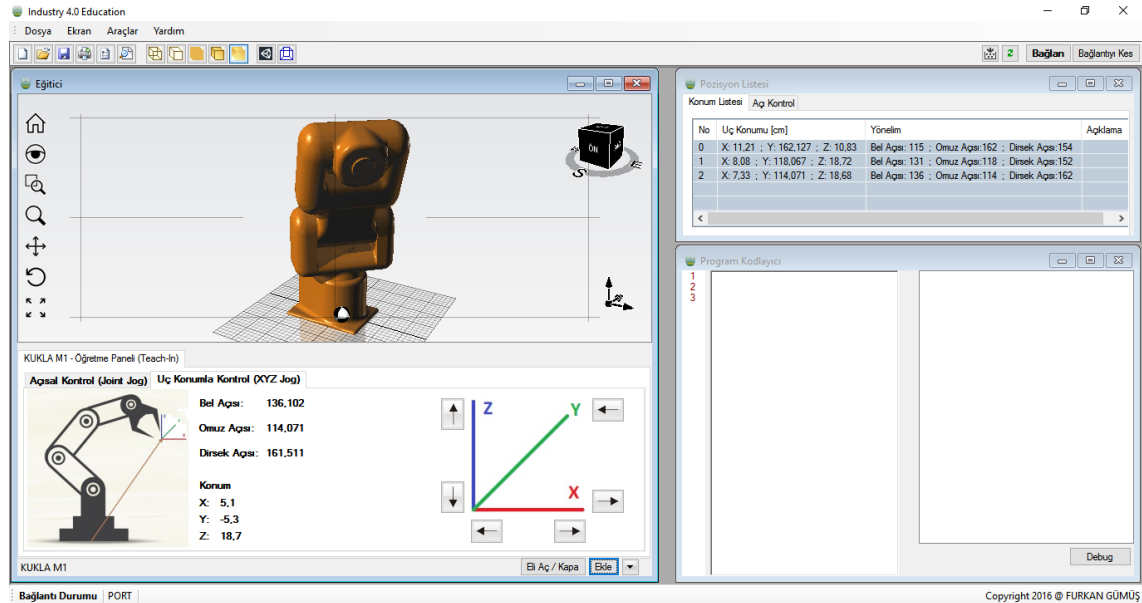
Bu bölümdeki kavramları kullanarak simülasyondaki robot manipülâtörün hareketlerini matrisler cinsinden ifade edeceğiz. Hesaplamış olduğumuz ileri ve ters kinematik matris yapıları ile birlikte kullanılarak C# yazılımında simüle edilebilirlik kazandırılmıştır.

6.2. Kontrolcü Yazılımının Oluşturulması

Bütünleşik imalat sistemlerini bir çatı altında toplayan ve öğrencilerin temel robot programlama bilgisini oluşturma amacıyla aşağıdaki gibi bir program hazırlanmıştır. Gerçek zamanlı kontrol, simülasyon ve elektronik kartlar ile seri haberleşebilme ve hatta robot programlama dili ile hazırlanmış (move master command) metni debug ve compiler edebilmektedir.

Kullandığımız rc servo motor Arduino prototip geliştirme kartı ile sürülmektedir. Kart ve program arasında haberleşmenin sağlanması için Visual Studio'da açtığımız bir winform içerisindeki araçlar menüsünden SerialPort eklendi

Servo motorların sürülebilmesi için hazır kütüphaneye sahip olan Arduino sayesinde bilmemiz gerekli olan tek şey motorun konumudur. Sahip olduğumuz motor 0-180° açı değerlerinde çalışabilmektedir. Kart ile yazılım arasında sadece int (tam sayı) değerinde bir karakter gönderilmesi pozisyonlama için yeterlidir. Ancak birden fazla motor olduğu için hangi motora hangi açı değeri gönderilecek bunu ayırt edecek bir şifreleme metodu oluşturmamız gereklidir. Bu amaçla gönderilecek koddan önce her seferinde bir char tipinde karakterin önceden gelmesini istiyoruz. Algoritmanın bu mantık ile kurulması, char tipinde şifreli bir kodun gönderilip başka bir ortamda çözülmesine benzer. Bu özellik farklı ortamlarda da işlevsel olarak kullanılabilir. Ancak programı yazan kişinin ne tür bir şifre göndereceğini karşı tarafa da söylemiş olmalıdır. Bu algoritma sayesinde şifreli ve güvenli bir haberleşme sağlanmış olmaktadır. Şekil 6.6'da görüldüğü gibi robotu istediğimiz pozisyona ulaştırmak için motorun açı değerini kullanıyoruz. Bu değer direk seri haberleşme üzerinden karta gönderip gerçek zamanlı hareket yapmasını



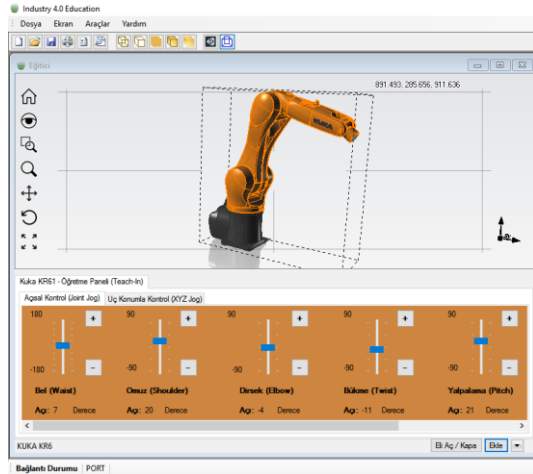
Şekil 6.4. Üç eksenli robot kontrolü

Kontrolcü algoritması ve manipülâtörün hareketini oluşturan temel unsur servo motorların çalışma mantığına bağlıdır.

sağlayabiliriz veya pozisyonları kaydederek daha sonra belirli sıralarda çağırabiliriz. Açı değerini kullanarak manipülâtörün uç

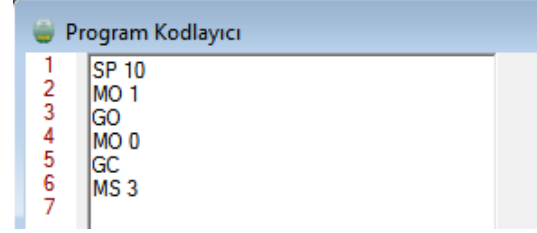
konumunun koordinatlarını ileri kinematik hesaplama ile daha önceki bölümde nasıl bulunacağını göstermiştik. Şekil 6.4'de ise tam tersi gösterilmektedir. Uç noktanın koordinatları x, y ve z'de bir birim ilerlenerek değişmektedir. Bu hareketler sırasında motorların alacağı açı değerlerinin hesaplanması ters kinematik ile olmaktadır. 5. Bölümdeki ters kinematik işlemler kullanılarak bu açı değerlerini saptadık. Bunu hareketlerin simülasyon ekranında 3 boyutlu olarak gösterilmesini sağlamak için önceki bölümde anlatılan transfer matrisleri kullanılmıştır.

Burada kullandığımız her bir model Solidworks ortamında çizilmiştir. Kinematik denklemlerini çıkartabileceğimiz her modeli buraya yükleyebilir ve kontrol edebiliriz. Bu program bizlere esnek ve geliştirilebilir bir ortam sunmaktadır.



Şekil 6.6. KUKA KR6 model manipülatörün kontrolü

Robotun hareketini açısal kontrol (joint jog) veya uç konum (XYZ jog) hareketiyle sağlayabilmekteyiz. Şekil 6.4'de görüldüğü gibi her hareket sonrasında kaydet butonuna basarak konum ve yönelim bilgisini pozisyon bilgisi ekranında saklayabiliyoruz. Daha sonra bu verilen konumları kullanarak robotumuzu eğitebilir ve algoritma kurarak döngü içerisinde çalıştırılmasını sağlayabiliriz. Bu amaçla geliştirilmiş Move Master, Melfa Basic gibi robot programlama dilleri mevcuttur. Eğitim amaçlı bir uygulama olacağı için bu dilleri kullanabileceğimiz kütüphanelerin oluşturulması gerekmektedir. Move Master komutlarından bazılarını kullanarak bunu biraz açalım.



Şekil 6.7. Move master kodlaması

Örnek bir programlama ekranında, SP (speed) komutu robotun hareket hızını, MO (move) komutu kaçınıcı sıradaki pozisyona gidileceğini, MS (move straight) istenen konuma doğrusal hareket yapılacağını, GO (gripper open) ve GC (gripper close) ise elin açılıp kapatılacağı anlamına gelmektedir. Programın işleyişi yukarıdan aşağı doğru satır halinde okuyarak gitmektedir. Ancak bu ekrana verdiğimiz kodların dışında yanlış bir karakter girilmesi, fazla veya düşük sayı verilmesi veya eksik kod yazılması durumunda sistemin bu hatayı belirlemesi gerekmektedir. Bu amaçla düzenli ifadeler (regex) kavramını kullanacağız.

Visual studio'da

System.Text.RegularExpressions kütüphanesi eklenerek satırdaki ifadelerin düzenli olup olmadığının kontrolünü yapabiliriz. Örnek olarak,

```
Regex move = new Regex("^MO\\s*[0-9]*\\s*$");
Regex movestraight = new Regex("^MS\\s*[0-9]*\\s*$");
Regex equal = new Regex("^EQ\\s*[0-9]*\\s*,\\s*[0-9]*\\s*$");
Regex gripperopen = new Regex("^GO\\s*$");
Regex gripperclose = new Regex("^GC\\s*$");
```

Burada ^ ifadesi MO ile başlayan kelimeyi tut demektir. \\s* ise çoklu satır aralığı ve [0-9]* ifadesi ise sıfır ile 9 rakamları arasında çoklu basamaklar halinde olabilecek karakterlerin eşleşmesini aramaktadır. \$ ifadesi ile aradığımız cümle veya kelime bütünü sonlandığını söylüyoruz.

```
if
(move.Match( richTextBox1.Lines[i] ).Success)
{
```

```
foreach (Match isim in
Regex.Matches( richTextBox1.Lines[i],
"^MO\\s*[0-9]*\\s*$" ))
```

```
MessageBox.Show(Convert.ToString(isim.Val
ue)); string[] dizi =
Regex.Split(richTextBox1.Lines[j], " ");
```

```
for (; j <= dizi.Length - 1; j++)
    listBox1.Items.Add(dizi[j]);
}
```

Satır içerisinde bu veya buna benzer regex ifadelerini bulamaz ise o satır geçilmektedir. Geçilen her bir satır aslında debug edilmiş olduğundan hata analizi için benzer bir kodlama yapılmıştır. Yukarıdaki fonksiyonda eşleşme yapılan her bir kodun parçalarına ayrılıp diziler içerisinde saklandığı gösterilmektedir. Bu fonksiyon içerisinde motorun nasıl bir hareket yapması gerektiğini anlatan kütüphaneler yazılmıştır.

Bölüm 5'de robot kinematığını anlattığımızda hesaplamaların yazılım içerisinde yer alacağından bahsetmiştik. Program içerisine 3 eksenli robot için statik, dinamik ve kinematik analizlerin yapılabileceği bir kodlama yapılmıştır. Şekil 6.8'deki değerler 5.bölümde tasarımını yaptığımız manipülatöre aittir.

halinde bağlaması ile hem daha az maliyet (profil fiyatları arasında ciddi fark vardır) hem de sisteme daha az yük binmesi sağlanabilir. Böylece daha düşük torka sahip ucuz step motor kullanılabilir.

Spindle yerine kullanılan Mabuchi RS-775 model motor gayet yüksek performans göstermiştir. Tahta ve sert köpük işleme düşük ilerleme hızlarında başarılıdır. Daha iyi freze takımı ile yüksek ilerleme hızlarında etkili sonuç alabilmeyi vaat etmektedir. Mandlen takımı için daha uygun bağlantı şekilleri yapılarak iyileştirilemeye ihtiyaç bulunmaktadır. 12V DC güç kaynağı kullanıldığı için katalogdaki nominal 18V değerinde çalışılma koşulları sınınamamıştır. Motor 20V (en fazla 24V) çalışabilme kapasitesine sahiptir. Voltaj regülatörlü ve PWM kontrollü motor sürücü devresi kullanılarak torktan kayıp olmayacak şekilde verimli talaş kaldırma ve delme işlemleri yapılmaya müsaittir. Redüktör kullanılarak tork daha da yükseltilebilir ve uygun tasarım yapılarak mandlen takımı daha verimli yerleştirilebilir.

UNSUR	Ağırlık (kg)	Uzunluk (cm)	K.M.U
Bel (L0)	0,04168	10,8	8,15
Omuz (L1)	0,06226	10	4,7
Dirsek (L2)	0,0144	3,5	11

UNSUR	Ağırlık (kg)	Verim
0.Motor (J0)	0,009	90
1.Motor (J1)	0,009	90
2.Motor (J2)	0,009	90
Nesne	0,01	

İleri Kinematik	Ters Kinematik		
Eklemler	Açı	Eklemin x Konumu	Eklemin z Konumu
J0	0	0	10,8
J1	0	10	10,8
J2	0	13,5	10,8

Unsur	Ağırlık [kg]	Uzunluk [cm]	Kütle Merkezi Uzaklığı [cm]	Statik Moment [Nm]	Dinamik Moment [Nm]	Çizgisel Hızı [m/s]	Merkezil İvme [m/s ²]	Atalet Momenti [kg.m ²]
Bel [L0]	0,04168	10,8	8,15	0	0,0221939383809178	1,41371669411...	14,804406601634	0,0006358095
Omuz [L1]	0,06226	10	4,7	0,0663177582	0,020285420843862	1,41371669411...	14,804406601634	0,0005811345
Dirsek [L2]	0,0144	3,5	11	0,00484614	0,000347785942280...	0,36651914291...	3,83817948931253	9,96333333333333E-06

Şekil 6.8. Üç eksenli manipülatör için hesap makinesi

VII. SONUÇLAR

Hibrid yönlendiricide bağlama tablası için kullanılan 22.5x180'lik yüzey kaplama profili ağırlığından dolayı trapez vidaya fazladan yük bindirmiştir. Aynı işlevi 15x120'lik yüzey kaplama profili yan şekilde çoklu üniteler

Simülasyonda kullanılan modellerin uzantılarına göre yüklenme süreleri değişken olduğu gözlemlendi. Stl uzantısına sahip çizimler daha hızlı açılırken step uzantılı dosyaların yüklenmesi yaklaşık 130 saniye daha uzun sürmektedir. Step uzantılı modellerin görümleri daha canlı ve gerçekçi olurken stl

uzantılı modellerde bazı görünüm özellikleri eksik çıkmaktadır. Yaptığımız modellerin çıktısını alırken global bir koordinat tanımlamamız gerekmektedir. Aksi halde çizim uzayındaki yerel koordinatlar simülasyon uzayımızda farklı doğrultulara düşebilmektedir.

KAYNAKLAR / REFERENCES

- [1] Endüstri 4.0, <https://goo.gl/wGkcqx> (06.01.2017)
- [2] CNC Router Tasarım Modelleri, <https://goo.gl/Eroork> (06.01.2017)
- [3] CNC Operatörünün El Kitabı, Makine Yüksek Mühendisi Hamit ARSLAN
- [4] Freze Çakısı Ürün Kataloğu, Engineering Supply Company Ltd., İskoçya, 2017
- [5] Akkurt, A., (1985) Talas Kaldırma Yöntemleri ve Takım Tezgâhları, Birsen Yayınevi, İstanbul
- [6] Mabuchi RS-775 Motor Kataloğu, http://61.114.190.123/cgi-bin/catalog/e_catalog.cgi?CAT_ID=rs_775vcwc (09.01.2017)
- [7] Krom Kaplı Milin Özellikleri, <https://goo.gl/dqTG8I> (09.01.2017)
- [8] 304 Kalite Paslanmaz Çeliğin Özellikleri, <https://goo.gl/trJ3nH> (09.01.2017)
- [9] Bingül, Z., Küçük, S., (2005) Robot Tekniği I, Birsen Yayınevi, ISBN: 975-511-424-6
- [10] Bilgisayar Grafiği Kavramı, <https://www.muhendisbeyinler.net/bilgisayar-grafiği-nedir-ve-matris-kavrami> (09.01.2017)