

Çok Robotlu İstasyonlarda Çarpışma Önlemek İçin Çalışma Alanı Yönetimi

Mustafa ÇAKIR¹, Cengiz DENİZ^{2*}

¹ Kocaeli Üniversitesi, Mühendislik Fak., Elektronik ve Hab. Müh. Kocaeli
² Ford Otosan, Gölcük Fabrikası, Gövde Üretim Alan Müdürlüğü, Kocaeli

¹<https://orcid.org/0000-0003-1250-9114>

²<https://orcid.org/0000-0002-9199-7680>

*Sorumlu yazar: cdeniz1@ford.com.tr

Araştırma Makalesi

Makale Tarihiçesi:

Geliş tarihi: 1 Ekim 2019

Kabul tarihi: 10 Nisan 2020

Online Yayınlanma: 15 Haziran 2020

Anahtar Kelimeler:

Endüstriyel robot

Esnek otomasyon

Hat dışı programlama

Bölge yönetimi

Çarpışma algılama

ÖZET

Üretim tesislerinde, basit veya karmaşık tekrarlı görevleri istenen sürede yapmak için çok robotlu hücreler veya üretim hatları kurulur. Bu üretim hatlarında robotik otomasyon yazılımları operatör bağımlıdır, kaza riskleri vardır ve verimleri düşüktür. İmalat sanayisinde firmalar özel çözümler üretmek yerine güvenilirliği ve sürdürülebilirliği kanıtlanmış mevcut ticari çözümleri tercih etmektedir. Ticari hat-dışı robot programlama yazılımları ortak alan içinde çalışan robotlar için bölge paylaşımını yörünge üzerinde bütüncül olarak denetlememektedir. Çarpışma denetimi animasyon ve bilgisayar oyunlarında kullanılan bir işlemdir. Bilgisayar destekli tasarım (CAD), bilgisayar destekli imalat (CAM) programlarında kısmen mevcuttur. Teknik açıdan çarpışma algılama çözülmüş bir problemdir. Mühendislik açısından bakıldığında işlem yükü çok fazla olduğundan dolayı her platform için uygulanabilirlik sorunudur. Bu çalışmada ortak alan yönetimini verimli hale getirecek robotlar arası eş zamanlama sağlamaya elverişli hat-dışı robot programlama yazılımı konu edilmiştir. Sunulan çalışmanın özgün yönleri şu şekilde sıralanabilir: Zaman ve konuma bağlı olarak robot yörüngeleri değerlendirilerek sinyalleşme önerilmiştir. Böylelikle ortak alan daraltılmış ve robotların bekleme süreleri azaltılarak iş verimleri artırılmıştır. Robot kinematik bilgilerinden hesaplanan ön bilgilerle bütün olarak tüm parçalar için kontrol yapmak yerine yalnız riskli grupların analizi yapılmaktadır. Hesaplama yükünün azaltılmasıyla, basitlik nedeniyle tercih edilen hücre bölütleme tabanlı yöntemler yerine, yüzey kaplamalarının kesişimi değerlendirilmiş ve sıfır toleranslı çarpışma tespiti gerçekleştirilmiştir.

Workspace Management in Multi-Robotic Stations for Collision Avoidance

Research Article

Article History:

Received: 1 October 2019

Accepted: 10 April 2020

Published online: 15 June 2020

Keywords:

Industrial robot

Flexible automation

Offline programming

Zone management

Collision detection

ABSTRACT

In industry, multi robotic cell or stations are designed to complete a simple or complex job at the defined cycle time. Robotics automation software is operator dependent, risks of accidents are high and efficiency is low. Instead of custom solutions, existing commercial solutions due to reliability and sustainable are preferred by the companies in manufacturing industry. Commercial off-the-shelf robotic programming software does not have integrated approach on the trajectory to monitor the shared zones for collisions. Collision detection is a frequently used function in virtual platforms for animation and computer games. It is partly available in computer-aided design (CAD) and computer-aided manufacturing (CAM) programs. Technically collision detection is a solved problem. In engineering point of view, practicability for each platform has many problems because of huge computational burden. In this study, offline robot programming software suitable for ensuring robot-to-robot synchronization is developed. The proposed study is unique for the following aspects. Signaling has been proposed

by evaluating robot trajectories according to time and position. This narrows shared area and improves work efficiency by reducing waiting time for robots. Instead of checking all parts as a whole with the preliminary information calculated from kinematic data of the robot, only analysis of risk groups is carried out. With the reduction of computational load, instead of the cell segmentation based methods which are preferred due to simplicity, intersections of the surface mesh are impacted and zero tolerance collision detection is achieved.

To Cite: Çakır M., Deniz C. Çok Robotlu İstasyonlarda Çarpışma Önlemek İçin Çalışma Alanı Yönetimi. Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi 2020; 3(1): 1-10.

1. Giriş

Esnek otomasyon sistemlerinin yaygınlaşması ile donanım maliyetleri azaltılmaktadır. Bunun yanında programlama maliyetleri ve verim önem kazanmaktadır. Endüstride güvenilirlik esas olduğu için, akademik çevrelerde çalışmalara sıklıkla konu edilmesine karşın, otonomi, sezgisel yaklaşımlar kullanılmamaktadır. Geleneksel programlama tercih edilmektedir. Söz konusu esnek otomasyon olduğunda robotik imalat ilk akla gelen çözümdür. Robot programlama için birbirlerini tamamlayan iki yöntem vardır. Kurulum süresini kısaltmak için önce hat-dışı CAD yazılımları ile olabilirlik, verim ve erişim hesaplamaları yapılır. Robot programının çatısı oluşturulur. Kurulum sırasında ise operatörler el terminalleri aracılığı ile manipülatörün konum kontrollerini yaparlar. Hat dışı programlama yazılımlarının kullanımı giderek yaygınlaşmaktadır. Pan ve ark. [1] yayınlarında mevcut teknolojik yönelimi konu edinmişlerdir. Bu yayında hat dışı robot programlama yazılımları tablolar halinde sunulmuştur. Ayrıca programlama seçeneklerine göre üstünlüklerine ve zaaflarına yer verilmiştir.

Robot programlama yazılımları ile eskiden bu yazılımlara özel çizimler kullanılarak operatör tarafından görsel deneyimlerle programlama yapılırken, Neto ve Mendes [2] çalışmasında bir örneği tanıtılan yeni nesil yazılımlar ile tüm genel çizim programlarından veri alabilen ve bu çizimlerden yola çıkarak otomatik olarak robotlar için işlem yolu üretilmesi sağlanmaktadır. Son yıllarda geliştirilen hat dışı programlama yazılımları farklı CAD çizimleri ile çalışma bölgesi tasarımı ve robot programlama becerilerine ek olarak Baizid ve ark. [3]'de olduğu gibi en iyileme algoritmaları ile kullanıcıların daha verimli istasyonlar tasarlamasını kolaylaştırmaktadırlar.

Robotik istasyonlarda çoğunlukla birden fazla sayıda robot bulunur. İş akışını hızlandırmak, erişim noktalarını arttırmak için tercih edilen bu

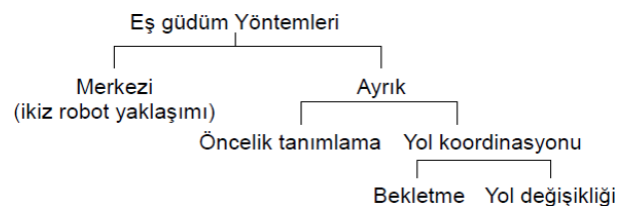
durum programlama konusunda ilave zorluklar getirmektedir. Özellikle aynı çalışma bölgesinde bulunan robotların çarpışma riskine karşı dikkatli olunmalıdır. Ortak bölgeyi paylaşan robotların çarpışmasız bir şekilde süreci devam ettirmeleri için temelde iki aşama vardır:

a)Çarpışma olasılığı olan çalışma bölgesi belirlenmeli

b)Bu bölgeye giriş çıkışlar denetlenmelidir.

Şekil 1'de bu denetlemenin ne şekilde yapılabileceğine dair genel yaklaşımlar sunulmuştur.

Çoklu robot sistemini tek bir, ancak çok eklemli robot gibi düşünen merkezi yaklaşımda hesaplama ve kontrol yükü çok fazladır. Öncelik tanıma ise kolay çözüm iken verimsizliğe neden olan boş beklemlerin en yüksek olduğu yaklaşımdır. Yol değişikliği robot programlamayı karmaşık hale getirmektedir. Curkovic ve ark. [4]'nün yayınında olduğu gibi sezgisel yaklaşımlar olmakla beraber güvenilirlikleri çok düşüktür. Paylaşılan bölge denetimi için günümüzde hemen hemen alternatifsiz kullanılan yaklaşım bekletmedir. Bölge boş olduğunda bir robotun çalışmasına izin verilirken diğer robotlar bekletilmektedir. Bu sinyalleşmede çoğunlukla programlanabilir mantık denetleçler (PLC) tarafından kontrol edilir. Operatör becerisine bırakıldığında verimsiz olabilecek bu yöntem, Flordal ve ark. [5]'nin çalışmasında olduğu gibi hat dışı programlama yazılımları ve PLC ile en düşük kayıp ile uygulanabilir. Bu yayında araştırmacılar durum makinası yaklaşımına benzer bir çözüm önermişlerdir. Çalışma uzayını bölgelere ayırarak PLC denetiminde bölge rezervasyonu ile boş beklemleri azaltmaya çalışmışlardır.



Şekil 1. Robot eş güdüm yöntemleri

Endüstriyel robotlar için çarpışma yalnız çevrelerindeki nesnelere için, Rubio ve ark. [6]'da konu edildiği gibi, söz konusu değildir. Eklemler seri robotlar erişim bölgeleri içindeki geniş bir alanda istenen noktaya çok fazla yönelme seçeneği ile ulaşabilirler. Bu kıvraklık özellikle otomotiv endüstrisinde kaynak uygulamaları için büyük avantaj kazandırır. Ancak programlama dikkatlice yapılmazsa öğretilmiş iki nokta arasında robot hareket ederken kendi katı gövdesindeki eklemler birbirlerine çarpabilirler. Literatürde çok nadir işlenmiş bu konu bir çalışmada değerlendirilmiştir [7]. Son yıllardaki eğilim ortak çalışma söz konusu olduğunda insan-robot işbirliği üzerinedir. Özellikle güvenlik ve yeni uygulamaları konu ederek bu doğrultudaki iki yüzden fazla çalışmanın irdelendiği Villani ve ark. [8]'nin çalışması konunun yakın gelecekte hangi alanlarda yoğunlaşacağını göstermektedir. Robotların birbirleri ile görev dağılımı ile çalışmaları sürerken insanlar ile de eş güdümü sürdürmesi ilave uygulama alanlarının doğmasını, süreçlerde verimin artmasını sağlamıştır. Elbette insan robot birlikteliği söz konusu olduğunda dikkat edilecek ilk konu yine çarpışma engelleme olacaktır. Ancak insan faktörü söz konusu olduğunda önceden hazırlanmış planların, çizimlerin kullanılmayacağı açıktır. Bu duruma çözüm olarak Schmidt ve Wang [9] yayınında kameralar aracılığı ile elde edilen pozisyon bilgisinin kullanılması önerilmiştir. Yazarlar bu tip iş birliğinde özellikle ortam aydınlatması nedeni ile kameranın güvenilir olamayacağını vurgulamıştır.

Çoklu robot istasyonlarında iş akışı bazen çalışma anındaki otomasyon sisteminin gönderdiği sinyallerle değiştirilebilir. Bu durumda kabaca üç çözüm vardır. Birincisi, gerçek zamanlı olarak konum algılayıcıları ile istasyondan elde edilecek bilgiler ile Schmidt ve Wang [9]'nin yayınındaki gibi kontrol yapılabilir. Endüstri için çok fazla tercih edilmeyen bir yöntemdir. Çünkü robotların anlık hızları çarpışma gerçekleşmeden önce gerekli çıkarımların yapılmasına müsaade etmeyecek kadar yüksektir. İkinci çözüm bir üst kontrolör kurulmasıdır [10]. Bu ilave kontrolör, algılayıcı bilgisi yerine iş akışını değiştiren otomasyon sinyallerini değerlendirerek robotlar arası eş zamanlamayı sağlayabilir. Bu yayında çalışma bölgesinin alt parçalara ayrılması değerlendirilmiştir. Bu yüzden paylaşılan bölgenin çok verimli yönetilemeyeceği düşünülebilir. Üçüncü çözüm sunulan bu çalışmada tercih edilen yöntemdir. Bu yöntemde güvenlik ön planda tutulduğu için çalışma anında denetim yapılması yerine başlangıçta olası tüm

yörüngeler için çarpışma denetimi benimsenmiştir. Bunu getireceği ilave iş yüklerinin nasıl aşılacağına dair öneriler yazının ilerleyen kısımlarında tanıtılmıştır.

Katı cisimlerin birbirlerine temasının algılanması animasyon ve oyun yazılımlarında sıklıkla kullanılmaktadır. Modeller bilindiği üzere bilgisayar grafiklerinde köşe noktalarına bağlanmış yüzey yamaları ile görselleştirilirler. Bu yamalar farklı poligon yüzeyleri ile oluşturulabildiği gibi modelin yüzey topoğrafyasına daha iyi uyum sağlayabilmesi için üçgenlerden oluşturulması tercih edilmektedir. Çok genellemeci bir tavrıyla, kaba grafiklerde yüzlerce üçgenden model oluşturulurken orta seviyede grafiklerde dahi on binlerce yama bulunduğu söylenebilir. İki cismin temasının algılanması için tüm bu yamaların birbirleri ile kesişim testlerinin yapılması yüksek işlem yükü getirir. Üçgen yamalardan oluşan iki yüzeyin kesişiminin algılanması bilinen geometrik bir işlemdir. Binlerce yüzey örgüsünün test edilmesi araştırmacıları bu işlevi hızlandırıcı yöntemler geliştirmeye günümüzde dahi sevk etmektedir. Örneğin direk köşelerle sınırlandırılmış yüzeylerin testinin yapılmasındansa bu yüzeyi daire ile sınırlandırarak test edilmesinin daha verimli olduğu gösterilmiştir [11]. Çarpışma algılama işleminde temelde yapılan hesaplama basittir. Öncelikle hızlı bir test ile sınırlar değerlendirilerek karar verilebilir. Eğer bu test geçerse yüzey paralelligi sorgulanır ve gerekirse detaylı kesişim incelenebilir. Bu sürecin iş yükü, aynı işlevin çok sayıda tekrar edilmesinden kaynaklanmaktadır. Bu görevler paralel çalıştırılabilen işlemcilerin en verimli olduğu tip problemlerdir. Bu motivasyonla yapılan çalışmalarda [12-13] ve benzeri literatürde fazlasıyla yer alan çalışmalarda grafik işlemci üniteleri (GPU) ile hızlandırma yöntemleri önerilmiştir. Binlerce yüzey örgüsünün test edilmesini gerektiren çarpışma algılama için akademik literatürde bu algoritmaların hızlandırılması ile ilgili çalışmalar devam etmektedir.

Sunulan bu yazıda çok robotlu istasyonlar için ortak alan yönetimi konu edilmiştir. Eş güdüm için "öncelik tanıma" çözümü gereksiz beklemelelere sebep olacağı için tercih edilmemiştir. "yol değişikliği" ise endüstrinin temel kıstası olan güvenilirlik ilkesine aykırı bulunmuştur. Bu sebeple ortak alan paylaşımının "bekletme sinyalleri" ile yapılması düşünülmektedir. Gerçekçi robot modeli yerine işlem yükünü hafifletmek için robot eklemleri

kanonik geometriler ile modellenmesi işlem avantajını sağlarken gereksiz bekleme neden olabilir. Sunulan çalışmada çarpışma kontrolü nihai aşamada katı cisim modellerini oluşturan üçgen yamaların kesişimleri test edilerek yapılacaktır.

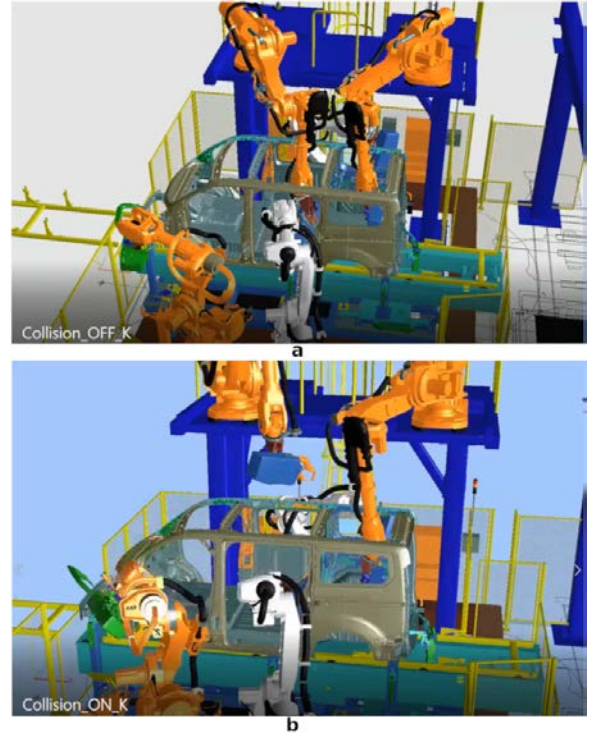
1.1 Problemin Tanımı

Zamanla değişen imalat ihtiyaçlarına yeni yatırımlara gerek kalmadan uyarlanabilmeleri sebebi ile esnek otomasyon sistemleri günümüzde imalat sürecinin olmazsa olmaz parçalarındandır. Donanım maliyetlerinin esnek çözümlerle düşürülmesine karşılık, üretim şartlarındaki değişiklik nedeni ile otomasyon ekipmanlarının yeniden programlanması ilave masraf getirmektedir. Karmaşık süreçlerde programlama, deneme-gözleme-düzenleme döngü adımlarını içeren imalat sahası çalışmasını içermektedir. Saha çalışmasını minimum düzeye indirerek olası maliyet, kaza ve zaman kaybı gibi unsurlarda işletme kaynaklarını verimli kullanabilmek için masa başı çalışmanın önemi büyüktür. Bu durumda otomasyon ekibinin modelleme yazılımlarını kullanması gerekir. Esnek, akıllı otomasyon sistemleri denince ilk akla gelen endüstriyel robotlu (IRB) imalattır. IRB'lerin programlanmasında hareket komutlarının yanında mantık, giriş-çıkış kontrolünü içeren kodlar bulunmaktadır. Birden fazla IRB'nin aynı çalışma bölgesini paylaşması durumunda aralarındaki etkileşimi yönetecek bağlantı fonksiyonlarının da hazırlanması gerekir. Çoklu robot istasyonları için ortak çalışma bölgesinin belirlenmesi, çarpışma bekleme noktalarının tespiti robot hareketlerini programlayan operatör tarafından, ihtiyacı karşılayacak hat-dışı modelleme programı olmadığı için, sahada yapılmaktadır. Operatör görüşünden dolayı yanlış ve verimsiz programlamaya açık bu yöntem robotik iş istasyonlarında hasarlı çarpışmalara, boş bekleme süresi kaynaklı verimsiz kaynak kullanımına sebep olmaktadır. Bu yazının oluşmasına kaynaklık eden Ford Otosan fabrikasında Şekil 2'de gösterilen gövde kaynak istasyonudur.



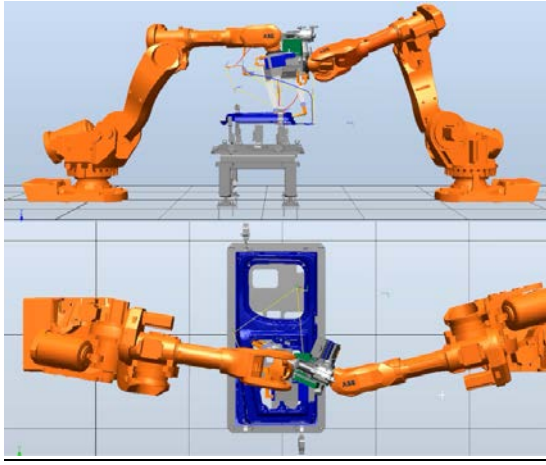
Şekil 2. Çok robotlu nokta kaynak istasyonu – Ford Otosan Gölcük Fabrikası

Bu istasyon Tablo 1'de yer bulan RobCad yazılımı ile ilk olarak masa başında kurgulanmıştır. Şekil 3'de RobCad yazılımı ile istasyonun tasarımı görülmektedir. 5 tane robot aynı araç üzerinde çalıştırılarak çevrim süresinin kısaltılması amaçlanmıştır. Ancak bu durumda IRB'ler ortak çalışma bölgelerinde birbirlerini beklemek zorunda kalmışlardır. Şekil 3a'da çarpışma kontrolü bulunmamaktadır. Dikkatli bakılırsa IRB'lerin birbirlerine çarptıkları görülebilir. RobCad programında otomatik çarpma kontrolü olmadığı için Şekil 3b'de gösterilen bekleme sinyalli benzetim, programcının anlık öngörüsü ile hazırlanmıştır. Bu öngörüye rağmen hat operatörleri tarafından işletme sahasında ince ayar yapılması zorunlu olmuştur. Şekil 2, 3 ve 4'de verilen bağlantıdaki videolardan anlık görüntüler alınarak elde edilmiştir [14].



Şekil 3. Robotik gövde kaynak istasyonu RobCad tasarımı a. alan yönetimli b. alan yönetimsiz

ABB-Robot Studio programı ile Ford Gölcük fabrikasında kapı hatları için uygulanmış bir örnek Şekil 4'de gösterilmiştir. RobCad programına benzer şekilde yazılım çarpışma durumu için operatöre uyarı yapmamaktadır. Buraya kadar olan incelememizde mevcut ticari programların çoklu robotik istasyonlarda verimliliğin ve iş riskinin en büyük nedeni olan çarpışma kontrolünü yapmadığını tespit ettik. Akademik süreçlerde ise geliştirilen yazılımlar ya bütünlük taşımamakta ya da güvenilirlikleri düşüktür. Endüstrinin ihtiyacını karşılamaktan uzaktırlar.

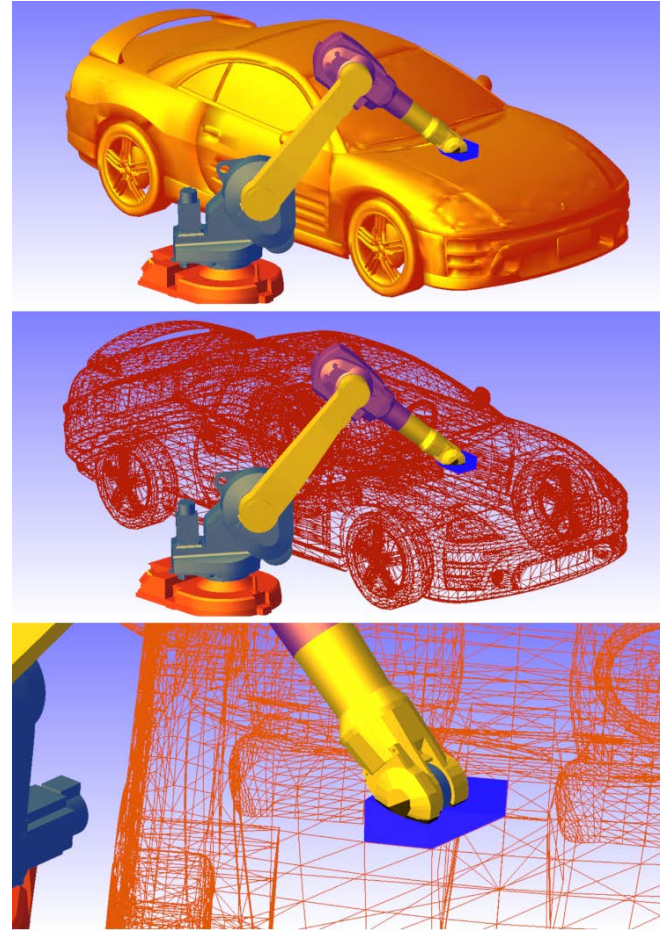


Şekil 4. ABB Robot Studio hat dışı programlama örneği

2. Materyal ve Metot

Çarpma algılamada işlem yüküne çok fazla ihtiyaç duymaktadır. Araştırmacılar sınırlandırılmış hacimler yaklaşımları ile modelleri basitleştirme eğilimindedir. Yüz binlerce üçgen yamadan oluşan oyun grafiklerinde uygulanan bu yöntem katı hesaplamaların gerektiği robotik imalatta tercih edilmemektedir. Oyun ve benzeri grafik canlandırmalara kıyasla endüstriyel robot uygulamalarında çalışma bölgesi daha sadedir. Öncelikle mevcut durumu değerlendirmek amacıyla standart yöntem geliştirdiğimiz yazılım ile Şekil 5’de gösterilen ortamda test edilmiştir.

İşlem parçası olarak değerlendirilen araba modeli 83991 adet üçgen yüzey örgüsünden oluşmaktadır. IRB modeli imalatçının sağladığı, robotik CAD programlarında kullandığı çizim dosyalarından alınmıştır. Taban kaidesi dahil her bir eklemün örgü sayısı Tablo 1’de sunulmuştur. IRB toplam 182736 üçgen yamadan oluşmaktadır. İşlem parçası ile tüm robot gövdesinin çarpışmanın denetlenmesi için yaklaşık 15.000.000.000 adet üçgen kesişim testi uygulanması gerekir. Tablo 1’de her bir eklem için gereken hesaplama süresi de verilmiştir. Bu süreleri vermekteki amaç yalnızca orantının anlaşılması içindir. Süreler IRB duruşuna göre büyük farklılıklar gösterebilmektedir. Çünkü üçgenlerin tam kesişimlerinin kontrolü yapılmadan önce basit bir test ile bazı kaplamaların kesişme ihtimallerinin olmadığı anlaşılabilir. Süre kısaltılabilmektedir.



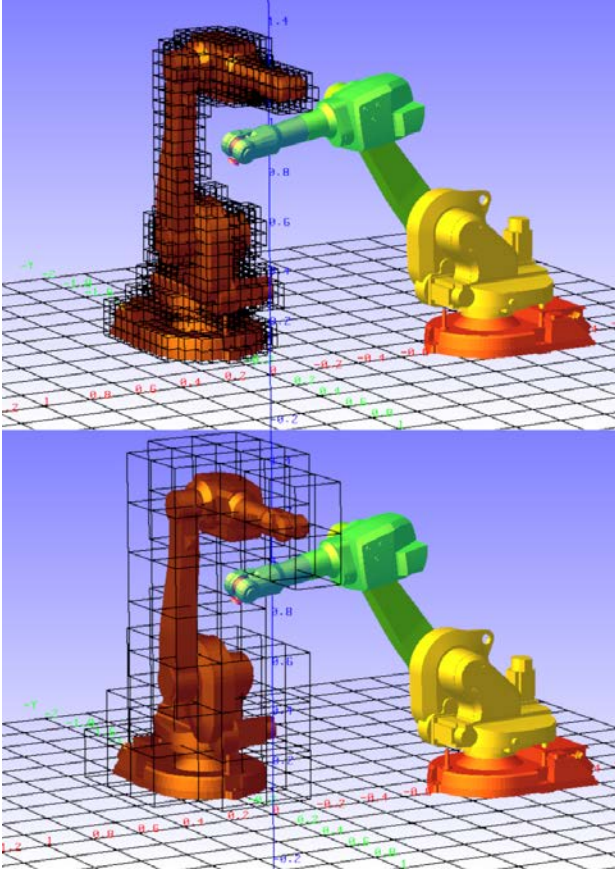
Şekil 5. IRB işlem parçası arasında çarpışma algılama

Tablo 1. Robot eklemleri yama sayıları ve çarpışma algılama süreleri (IRB links patch counts and elapsed time for collision detection)

Eklem numarası	Kaplama sayısı	Algılama süresi, s
Taban	19.086	4.53
1	83.247	19.78
2	13.146	3.14
3	36.480	8.65
4	11.727	2.79
5	14.172	5.96
6	4.878	1.15

Şekil 6’da gerçek model, küplerden oluşturulan hacim içerisinde gösterilmiştir. Hücre boyutu azaltılarak boş alanlar daraltılabilir. İşlem yükü göz önüne alınarak çok geniş hacimler gerçek model yerine kullanılırsa Şekil 2’deki gibi gerçek uygulamalarda robotların serbestçe hareket edebileceği bölge olmayacaktır. Çoklu robot istasyonları süreç hızlandırma avantajını yitirecek, yalnız sıralı işlem yapar hale gelecektir. Hücre boyutu çok azaltıldığında ise hedeflenen işlem yükü hafifletilmesinden vaz geçilmiş olur.

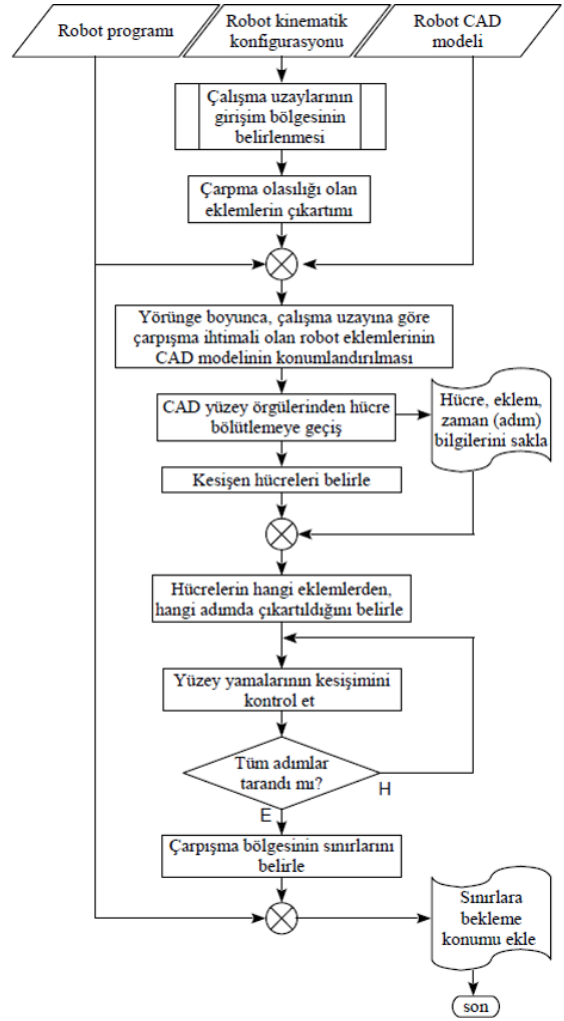
Sunulan çalışmada bu yöntem tek çözüm olarak tercih edilmemiştir.



Şekil 6. Hücre boyutlarının uygunsuz seçimine bağlı hatalı çarpışma algılama

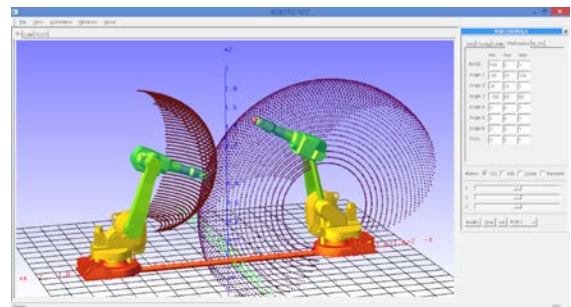
Önerdiğimiz yöntemde ait akış şeması Şekil 7'de verilmiştir. Yöntemdeki hedefler şunlardır. İşlem yükü azaltılmalı, çarpışma algılama sıfır pay ile gerçekleştirilmeli, anlık görüntüler yerine yörünge üzerindeki akışta değerlendirilme yapılmalı. Süreç robot yörüngelerinin programa yüklenmesi ile başlamaktadır. Sürecin sonundaki çıktı ise her bir robotun kendi yörüngesi üzerinde diğerleri ile kesişimin başladığı ve bittiği noktaların belirlenmesidir.

Hesap yükünü azaltmak için ilk önce gereksiz bölgelerdeki hesaplamalardan kaçınılmalıdır. Bu sebeple istasyon yerleşiminde robotların ortak olarak bulunabileceği hacmin belirlenmesi gerekir. Şekil 8'de hazırlanan program ile bu aşama görselleştirilmiştir. Eklemlili robot kollarının çalışma bölgeleri temel geometrilerle tanımlanamaz.



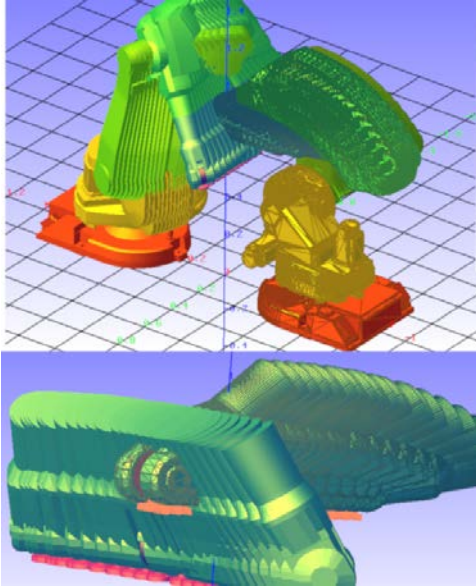
Şekil 7. Robot programının çarpışma bölgesi için düzenlenmesine ait akış şeması

Hatırlatmak gerekir ki eklemlili robotlarda eklemler tam 360 derece hareket etmezler. Bu grafikte robot uç noktasının düz kinematik bağıntılar ile hesaplanıp görselleştirilmesi ile oluşturulmuştur. Önerilen yöntemde yalnız uç noktası değil tüm bağlantılar için hacim belirlenmiştir. Her bir eklemin hangi hacim içerisinde hareket ettiği hesaplanmaktadır. Robot bir bütün olarak ele alınmadan, eklemlerin kesişip kesişmediğinin kontrolü ile işlem yükü azaltılmıştır.



Şekil 8. Paylaşılan çalışma alanının tespit edilmesi

Şekil 8'de gösterildiği gibi robot yerleşimlerine göre çalışma uzayı belirlendikten sonra, girişim yapması olası eklemlerin belirlenmesi aşamasına kadar robot programı göz önüne alınmamıştır. Bundan sonraki aşamada robot hareket komutları ile Şekil 9 deki gibi tüm yörünge değerlendirilir. Robot hareket komutlarında temelde hedef pozisyonun konum ve rotasyona ait bilgileri yer alır. Bu hedefe kullanılan komuta göre doğrusal, dairesel ya da eksen hareketlerle robot yönlendirilir. Şekil 9 bu hareketler sırasında belirli adımlarda ara noktalar belirlenerek elde edilmiştir. Robot programında ve detaylı çarpışma testlerinde değerlendirilmek üzere ilgili robot programı satırı ve ara noktanın konum ve rotasyon bilgileri depolanır. Bir önceki adımda yalnız çarpışma olasılığı olan eklemler çıkartılmıştı. Şimdiki aşamada Şekil 6 de tanıtilan hücreleme ile kaba hatlarla kesişen eklemler tespit edilir.



Şekil 9. Her bir IRB için ortak alanda bulunan eklemlerinin yörünge boyunca canlandırılması

Yüzeyleri oluşturan üçgen yamalarının testi Möller [15]'in yayınında bahsedilen yöntemle benzer şekilde yapılmıştır. Bu yayında kesişim algılaması için önerilen yöntem şudur.

T_1, T_2 üçgenleri sırası ile $(V_0^1, V_1^1, V_2^1), (V_0^2, V_1^2, V_2^2)$ köşe noktaları ile belirlenmiş olsun. T_2 için yüzey denklemi X bu yüzey üzerinde herhangi bir nokta, \vec{N}_2 yüzey normali olmak üzere,

$$\vec{N}_2 X + d_2 = 0 \quad (1)$$

olarak bulunur.

Yüzey normali \vec{N}_2 ve d_2 denklem (3) ve (4) ile hesaplanabilir.

$$\vec{N}_2 = (V_1^2 - V_0^2) \times (V_2^2 - V_0^2) \quad (2)$$

$$d_2 = -\vec{N}_2 \cdot V_0^2 \quad (3)$$

T_1 üçgeninin köşelerinin T_2 üçgeninin belirlediği düzleme olan mesafesi

$$d_{V_i^1} = \vec{N}_2 \cdot V_i^1 + d_2, \quad i = 0, 1, 2 \quad (4)$$

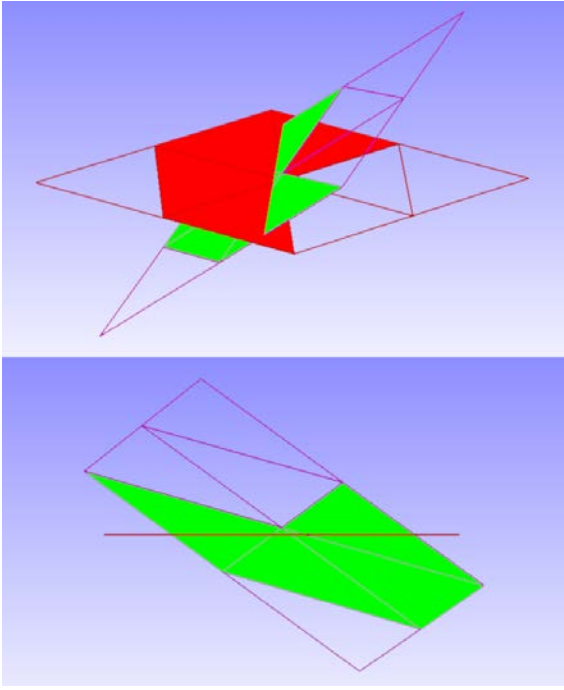
ile bulunur. Eğer bütün $d_{V_i^1}$ değerleri sıfırdan farklı ve aynı işaretli ise T_1 üçgeninin köşeleri T_2 üçgeni ile bulunan düzlemin aynı tarafındadır ve kesişim yoktur. Aksi durumda her iki düzlemin kesişim hattı

$$D = \vec{N}_1 \times \vec{N}_2 \quad (5)$$

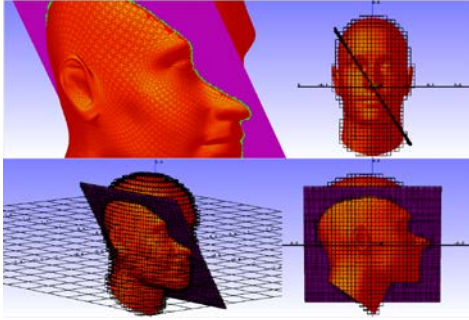
ile verilen doğru üzerindedir. $(V_1^2 - V_0^2) \cap D$ ve $(V_2^2 - V_0^2) \cap D$ ile T_2 üzerindeki kesişim hattı parçası bulunur.

Benzer şekilde T_1 için bulunan doğru parçası birbirine girişmiş durumda ise kesişim tespit edilmiş olur. $d_{V_i^1}$ değerleri sıfıra eşit ise her iki üçgen düzlemi paraleldir. Bu durumda bir üçgenin köşelerinin diğerinin içinde olup olmadığının testinin yapılması ile basitçe kesişim varlığı kontrol edilebilir. Şekil 10'da bu test ile gerçekleştirilmiş kesişim algılama gösterilmiştir. Dikkat edilirse görülecektir ki pek çok yama için detaylı test gerektirmeden (4) ile elde edilecek sonuç değerlendirilerek kesişim olmadığı anlaşılabilir.

Tüm yamaların bu şekilde test edilmesi her ne kadar detaylı analiz gerektirmeden sonuca ulaşılabilir de çok zaman alacaktır. Bu sebeple Şekil 11'de gösterilen yaklaşım bu çalışmada önerilmektedir. Bu öneride öncelikle modeller kübik hücrelere ayrılır. Hücrelerin kesişim testi üçgen üçgen kesişim testine göre basit ve hızlıdır. Yalnızca çakışan hücreler içinde yer alan yamaların kontrolü ile sıfır paylı çarpışma olup olmadığı anlaşılır.



Şekil 10. Yüzey örgülerinin kesişimlerinin tespiti



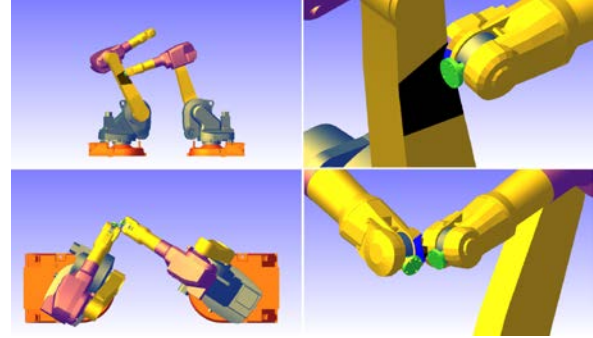
Şekil 11. Hızlandırma için hücre temelli kesişim testi

Şekil 11'de kullanılan modeller anlaşılabilirliği arttırmak için tercih edilmiştir. Çok robotlu istasyonlar söz konusu olduğunda bu işlemin şekil 9'da verilen sahne için yapılacağı bilinmelidir. Gerçek robot modelleri kullanılarak gerçekleştirilen çarpışma kontrolüne ait grafik Şekil 12'de verilmiştir.

3. Tartışma

Ortak çalışma alanı yönetimi özellikle otomotiv endüstrisinde robotik hatlarda çok önemlidir. Yanlış kurgulandığında mekanik hasarlara yol açabilen el ile yapılan düzenlemeler, özenli yapıldığında dahi verimsizliklere yol açmaktadır. Bu yazıda çoklu robot istasyonları için çevrim süresini hızlandırma, çarpışma kaynaklı mekanik hasarların oluşmasını engelleme, operatör yükünü azaltmak amacıyla, hazırlanmış robot

programlarını işleyip çarpışma kontrolü sinyalleşmesini düzenleyen yazılım tanıtılmıştır.



Şekil 12. Önerilen yöntemle gerçekleştirilmiş farklı IRB duruşları için çarpışma kontrol sonuçları

Çarpışma durumunda gerçek koşullarda tek geçerli tepki robotları en kısa biçimde durdurmaktır. Bu sebeple robot programında çarpışmanın gerçekleşmemesi için gerekli sinyalleşme düzenlenmiştir. Geliştirilen yazılımda sezgisel yöntemler kullanılmadan işletme, süreç güvenilirliğini arttırmak için IRB CAD modelini oluşturan üçgen örgülerin kesişim kontrolü ile çarpışma algılaması yapılmıştır. Konvansiyonel yaklaşımda bu işlem ya robot çizgi modeli üzerinde yapılmakta ya da hücre bölütleme gibi sınırlandırılmış hacim yaklaşımları ile gerçekleştirilmektedir. Önerdiğimiz yöntemden uygulamacıların temel sakınma nedeni robot kaplamasının her bir yaması için ilgili ekleme kadar düz kinematik modelin tekrar tekrar çalıştırılması neticesindeki işlem gücü yetersizliğidir. Kesişim kontrolü literatür çalışmalarında genelde tüm üçgen yamalar için ayrı ayrı uygulanırken bu çalışmada kinematik ön bilgilerden yola çıkarak yalnız çarpma olasılığı olan yamalar üzerinde yapılmıştır.

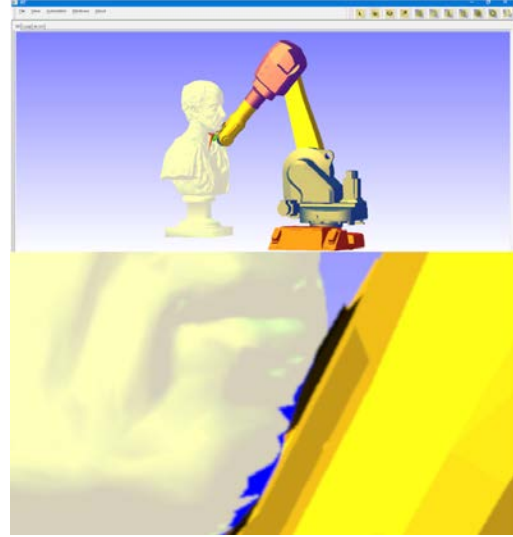
Geliştirilen yazılım aşağıdaki özelliklere sahiptir.

- Seçilmiş IRB manipülatörü görselleştirirken robot imalatçılarının CAD modelini kullanmaktadır.
- Çalışma bölgesi hacmini görselleştirebilmektedir. Manipülatör kinematik bilgilerini anlık olarak ekrana yansıtılabilmektedir. (program LibNoDave Kütüphanesi ve ABB RobSDK aracılığı ile eklem açılarını alabilmekte ve ekranda bu duruştaki robotu görselleştirebilmektedir.)
- Hat dışı programlamada çarpışma kontrolü yapabilmektedir.

Çarpışma algılama için kullanılan üçgen-üçgen kesişim tespitinde öncelikle üçgen yamanın düzlem denklemleri bulunur. Test edilecek diğer üçgenin köşeleri bu düzlemin aynı tarafında ise

test yapmaya gerek kalmadan kesişim yok kararı verilir. Eğer bu basit test ile kesişim olmadığı tespit edilebilir ise işlem süresi çok kısa olur. Dolayısı ile çarpışma algılama süresinin IRB duruşuna çok bağlı olduğu söylenebilir. Bu sebeple sunulan makalede kesin sayısal süreler vermekten kaçınırsak da bu işlem süresinin kullanılan donanıma bağlı olarak değişebileceği bilinmektedir. Bizce önemli olan test yapılması gereken üçgenlerin sayılarının azaltılmasıdır. Robot kinematik bilgilerinden yola çıkarak eklemelerin kesişim bölgesi içinde olup olmadığının tespiti ile öncelikle takip eden işlemler azaltılmıştır.

Mevcut ticari yazılımlarda çalışma bölgesi içinde her nesne sabit iken operatör tek bir manipülatörü gezdirerek çarpışma olasılığını değerlendirmektedir. Daha sonra sabit tutulan nesneler bir sonraki konumlarına getirilir ve tekrar hedef manipülatör bir önceki aşamada olduğu gibi yörüngesinde hareket ettirilir. Sunulan çalışmanın özgün yanlarından biri de her hareketli nesnenin tüm yörüngeleri boyunca hareketlerini bir kerede işlemedir. Bu özellikte hiç bir yazılım şu an için mevcut değildir. Önerdiğimiz yöntemde tüm yörünge üzerindeki hareket bir çerçevede toplanmıştır. Bu toplama işlemi yalnız çarpışma ihtimali olan eklemeler için yapılmıştır. Sonraki aşamada hücre bölütleme yöntemi ile kesişen hücreler tespit edilmiştir. Yalnız kesişen hücreler içindeki üçgen yamalar test edilerek süreç iyice kısaltılmıştır. Çarpışma daha önceden de değinildiği üzere yalnız IRB-IRB için değil aynı zamanda IRB işlem parçası için de söz konusu olabilir. Şekil 13'de robotik frezeleme uygulamasına ait resim verilmiştir. Bu yazıda önerdiğimiz yöntemler ile çarpışma tespiti bu tür uygulamalar için de kullanılabilir. Güncel yazılımlarda frezeleme işlem için robotun uç noktasının işlem hattı belirlenebilmesine karşın çarpışma kontrolü yapılmamaktadır.



Şekil 13. Robotik frezeleme uygulaması için çarpışma algılama

4. Sonuçlar

Bu çalışmada üçgen üçgen kesişim algılama için yeni bir önerimiz yoktur. Araştırmacılar günümüzde algoritmanın geliştirilmesi yerine iş akışının düzenlenmesi üzerine yoğunlaşmaktadırlar. İşlem yükünü hafifletmek için, tek komut çok veri yaklaşımına uyan hesaplamaları GPU üzerinde gerçekleştiren çalışmalar devam etmektedir. Gelecek çalışmalarda bu yazıda önerdiğimiz yöntem ile azaltılmış test edilecek üçgen yama sayısı avantajı, uygun hafıza ve işlem organizasyonu ile değerlendirilerek GPU'lar üzerinde programın daha da hızlandırılması sağlanabilir.

Not: Bu çalışma, 25-27 Nisan 2019 tarihleri arasında Antalya/Türkiye'de düzenlenen 4. Uluslararası Akdeniz Bilim ve Mühendislik Kongresi'nde (IMSEC 2019) sunulmuştur.

Kaynakça

- [1] Pan Z., Polden J., Larkin N., Van Duin S., Norrish J. Recent progress on programming methods for industrial robots, *Robotics and Computer-Integrated Manufacturing* 2012; 28(2): 87-94, doi: 10.1016/j.rcim.2011.08.004
- [2] Neto P., Mendes N. Direct off-line robot programming via a common CAD package, *Robotics and Autonomous Systems* 2013; 61, 896-910. doi: 10.1016/j.robot.2013.02.005
- [3] Baizid K., Cukovic S., Iqbal J., Yousnadj A., Chellali R., Meddahi A., Devedzic G., Ghionea I. IRoSim: Industrial robotics simulation design planning and optimization platform based on CAD and knowledge ware technologies, *Robotics and Computer-Integrated Manufacturing* 2016; 42, 121-134. doi: 10.1016/j.rcim.2016.06.003

- [4] Curkovic P., Jerbic B., Stipancic T. Co-evolutionary algorithm for motion planning of two industrial robots with overlapping workspaces, *International Journal of Advanced Robotic Systems* 2013; 10(55): 1-11. doi:10.5772/54991
- [5] Flordal H., Fabian M., Akesson K., Spensieri D. Automatic model generation and PLC-code implementation for interlocking policies in industrial robot cells, *Control Engineering Practice* 2007; 15(11): 1416-1426. doi: 10.1016/j.conengprac.2006.11.001
- [6] Rubio F., Llopis-Albert C., Valero F., Suer JL. Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory, *Robotics and Autonomous Systems* 2016; 86, 106-112. doi:10.1016/j.robot.2016.09.008
- [7] Peidro A., Reinoso O., Gil A., Marin JM., Paya L. A method based on the vanishing of self-motion manifolds to determine the collision-free workspace of redundant robots, *Mechanism and Machine Theory* 2018; 128, 84-109.
- [8] Villani V., Pini F., Leali F., Secchi C. Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications, *Mechatronics* 2018; 55, 248-266. doi:10.1016/j.mechatronics.2018.02.009
- [9] Schmidt B., Wang L. Depth camera based collision avoidance via active robot control, *Journal of Manufacturing Systems* 2014; 33, 711-718.
- [10] Zhou J., Aiyama Y. On-line collision avoidance system for two PTP command-based manipulators with distributed controller, *Advanced Robotics* 2015; 29(4): 239-251. doi:10.1080/01691864.2014.985610
- [11] Danaei B., Karbasizadeh N., Masouleh MT. A general approach on collision-free workspace determination via triangle-to-triangle intersection test, *Robotics and Computer-Integrated Manufacturing* 2017; 44, 230-241. doi: 10.1016/j.rcim.2016.08.013
- [12] Ruiz de Miras J., Salazar M. GPU inclusion test for triangular meshes, *Journal of Parallel Distributed Computing* 2018; 120, 170-181. doi: 10.1016/j.jpdc.2018.06.003
- [13] Ling-yu W. A faster triangle-to-triangle intersection test algorithm, *Computer Animation Virtual Worlds* 2014; 25, 553-559. doi:10.1002/cav.1558
- [14] Çakır M. Ders notu. Kocaeli Üniversitesi, Mühendislik Fak. Elektronik ve Hab. Müh. http://ehm.kocaeli.edu.tr/dersnotlari_data/mca_kir/Mak_Collision/2010.
- [15] Möller T. A fast triangle-triangle intersection test, *Journal of Graphic Tools* 1997; 2(2): 25-30. doi: 10.1080/10867651.1997.10487472