Izmir
Lubin
Pescara
Ribadeo
2023

Robotikte Programlama İçin Simülatör Kullanımı

ENDÜSTRİYEL ROBOTLAR

İŞBİRLİKÇİ ROBOTLAR

SİMÜLATÖR YAZILIMI

ROBOTLAR İÇİN PROGRAMLAMA DİLİ

BAKIM – ONARIM İLKELERİ

KA210VET Erasmus+ Projesi

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY

Avrupa Birliği tarafından finanse edilmektedir

INDEX

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY

Avrupa Birliği tarafından finanse edilmektedir

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

ÖNSÖZ

Endüstri 4.0 sanayi devriminde endüstriyel robotların büyük rolü var. Birçok üretim operasyonu endüstriyel robot kontrolünde gerçekleştirilir. Öğrencilerimizin endüstriyel robot sistemlerini anlamaları ve öğrenmeleri önemlidir. Öğrencilerimizin bu konudaki en büyük ihtiyaçlarından biri de ikinci kademe teknik eğitime uygun uygulamalı kaynak kitap ihtiyacıdır.

SIMPROTIC KA210VET proje ekibi olarak bu kitapçık öğrencilerimizin bu ihtiyacını karşılamak için hazırlanmıştır. Bu kitapçık öğrencilerimize rehberlik edecek ve eksiklerini fark ederek kendilerini geliştirmelerini sağlayacaktır. Bu kitapçık, endüstriyel robotlar ve bunların kontrolleri hakkındadır. Genel bir anlatım dili tercih edilmiştir. Tüm robotlara uyarlanabilir. Endüstriyel robot sistemlerinin simülasyonu için Coppeliasim Edu, RoboDK gibi ücretsiz versiyonları bulunan yazılımlar seçilmiştir. Robotik uygulamalar bir yazılım içinde inşa edilebilir veya aynı yazılım içinde simüle edilebilir. Böylece tüm deneyler bilgisayar ortamında yapılabilmekte ve mesleki eğitim sisteminde Dijital Dönüşüme bir katkı olarak görülebilmektedir.

Ayrıca Python gibi robot programlamada kullanılan yazılım dillerine de bir bölüm ayrılmıştır. Böylece endüstriyel robot sistemlerinin yazılım tarafının gösterilmesi amaçlanmaktadır.

Son bölümde ise endüstriyel robot uygulamalarında karşılaşılabilecek problemlerden bahsedilmiştir. Bu problemlerin çözümüne yönelik bilgiler bu bölümde verilmektedir.

Erasmus+ projesi KA210VET - Use a Simulator for Programming for Robotic in Robotic için bir araya gelen öğretmenlerin bilgi ve deneyimlerini paylaşmaktan mutluluk duyuyoruz. Kitabın tüm teknik personel ve teknik öğrencilere faydalı olmasını diliyoruz.

ITO VAKFI S. TASTEKIN MTAL | ZESPOL SZKOL NR.1 | IIS A. VOLTA PESCARA | APAGA GALICIA

"Erasmus+ Programı kapsamında Avrupa Komisyonu tarafından desteklenmektedir. Burada yer alan içerik yazarların görüşlerini yansıtmaktadır ve bu görüşlerden Avrupa Komisyonu ve Türkiye Ulusal Ajansı sorumlu tutulamaz."

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY

Avrupa Birliği tarafından finanse edilmektedir

ENDÜSTRİYEL ROBOTLAR

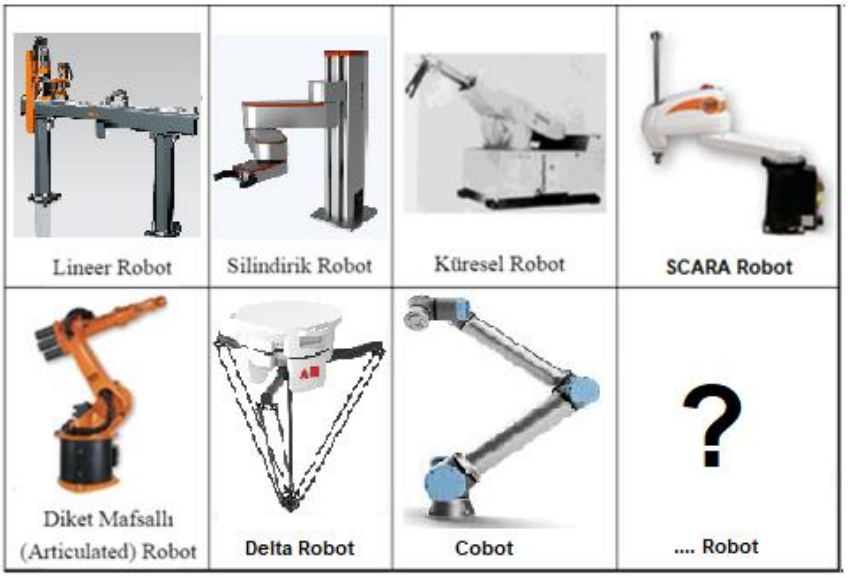
Robotik genel anlamda makina, elektrik-elektronik ve bilgisayar gibi alanların birleşiminden oluşan genel amaçlı programlanabilir makina sistemlerine yönelik çok disiplinli modem bir bilimdir. Dünyada robotiğin ilk temelleri 1136-1208 yılları arasında yaşamış olan El-Cezeri tarafından atılmıştır. Bununla birlikte robot kelimesi ilk defa 1922 yılında Çek yazar Karel Capek tarafından Slav dilinde işçi, köle, esir anlamına gelen crobota kelimesinden türetilmiştir. Robotik kelimesi ise ilk defa Isaac Asimov tarafından kullanılmıştır.

Robotların en fazla kullanıldığı alan hiç kuşku yok ki sanayidir. Sanayicilerin ana felsefesi, imalat hattının daha hızlı kaymasını sağlayarak daha fazla üretimde bulunmaktır. İmalat hattının daha hızla kayması demek; o imalat hattı boyunca çalışan birkaç işçi yerine, bir tek “mekanik robot” koymak suretiyle montaj işlemini çabuklaştırmak demektir.

Önceleri imalat hattı yanında duran ve (o hat üzerinde kayarak önüne gelen gövdelere) mekanik hareketlerle, bazı parçaları monte eden robotların geri besleme işlemi kullanılarak kapasiteleri artırılmıştır. Bu robotlar, parçayı monte ettikten sonra kontrolünü de yapar, eğer, hatalı işlem var ise, o parçayı imalat hattından çekip ayırır. Hatalı durumun giderilmesi için geri gönderir. Bu nedenle de bazı yerlerde “İmalat Hattı” (Assembly Line) yerine “Feed- Back Hattı” (Feed-Back Path) tanımlanması kullanılır.

Buradan yolan çıkarak endüstriyel robotların tanımı; Programlı hareketlerle değişik görevler için malzemeleri, aletleri ya da özel parçaları taşımak için tasarlanmış, yeniden programlanabilir, çok yönlü bir manipülatördür (kol) şeklinde yapılabilir. (Amerikan Robot Enstitüsü, RIA)

Endüstriyel robotun en kapsamlı tanımı ve robot tiplerinin sınıflandırması ISO 8373 standardında belirlenmiştir. Bu standarda göre bir robot şöyle tanımlanır. "Endüstriyel uygulamalarda kullanılan, sabit veya hareketli olabilen, üç veya daha fazla programlanabilir eksene sahip, otomatik kontrollü, yeniden programlanabilir çok amaçlı manipülatördür."

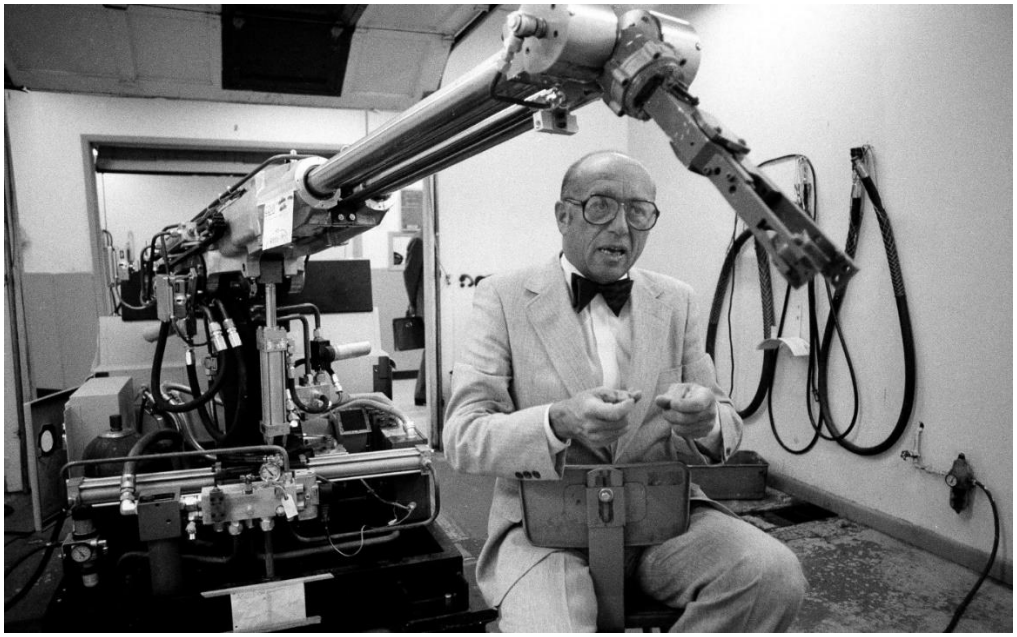


Resim 1. Endüstriyel robot örnekleri

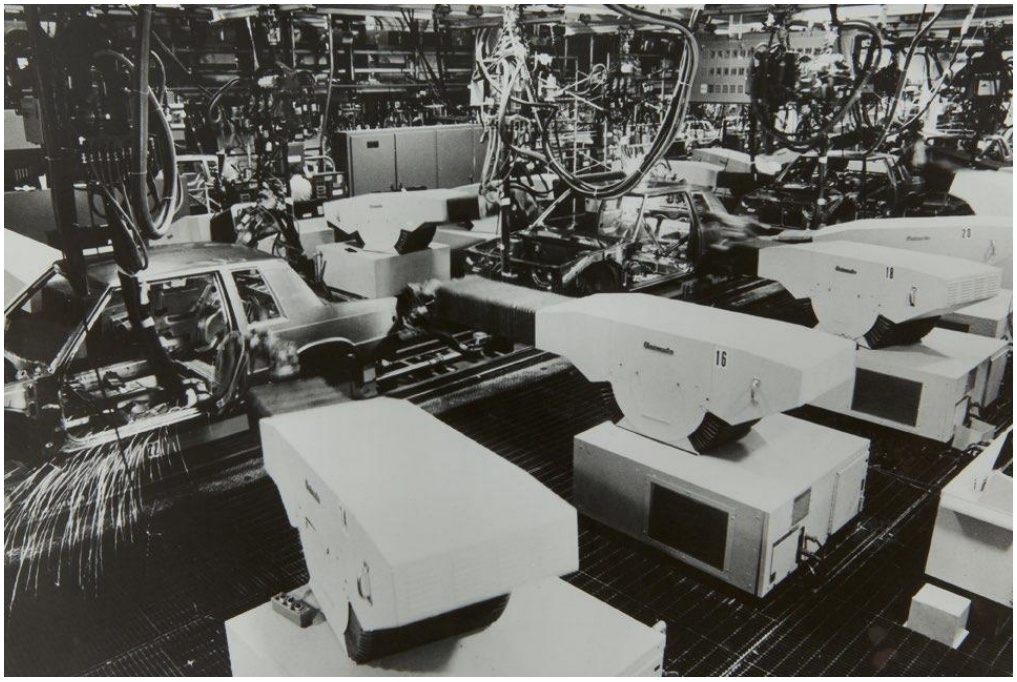
TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

1. ENDÜSTRİYEL ROBOTLARIN TARİHÇESİ

1956 yılında George Devol ve Joseph Engelberger tarafından Unimation (Universal Animation) adında bir şirket kurulmuştur. Unimation şirketi tarafından yapılan çalışmalar sonucu 1959 yılında ilk endüstriyel robot geliştirilmiştir. Dünyada Endüstriyel Robotun ilk uygulanması ise 1961 yılında Unimation şirketi tarafından General Motors'da bir konveyöre entegre edilen ve metal pres makinesinden sıcak ve ağır iş parçalarını alıp paletlere istiflemekle görevli Unimate olmuştur. O zamanki teknoloji ile endüstriyel robotun programı manyetik bir tambur üzerine kaydedilmiştir.



Resim 1.1. İlk endüstriyel robot (Unimate, 1961) - Joseph F. Engelberger



Resim 1.2. The Unimate, ABD'deki bir araba fabrikasında punta kaynağı yapıyor.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

2. ENDÜSTRİYEL ROBOTLARIN AVANTAJ VE DEZAVANTAJLARI

Endüstriyel Robotların kullanılmasının sağladığı avantajlar şu şekilde sıralanabilir:

- İnsanların fiziksel özellikleri zorlayan ağır ve büyük işlerde çalışabilirler,
- İnsan sağlığı için elverişsiz ve tehlikeli koşullarda çalışabildiklerinden işyeri güvenliğini adımlar,
- Sahip oldukları yüksek hassasiyet ve tekrarlanabilirlik sayesinde ürün kalitesinde standartları korurlar,
- Bozuk üretim miktarı azaltılarak, hammadde israfı engellenir ve üretim maliyetini düşürürler,
- Yeniden programlama sayesinde yeni bir işe kolayca adapte edilebilirler,
- Monoton, sıkıcı ve yorucu işlerde hızlı bir şekilde verimi ve ürün kalitesini düşürmeden sürekli çalışarak daha çok iş yapabilirler,
- Uzaktan erişim, yönetim, kontrol edilebilme özelliklerine sahiptirler,
- Aynı ortamda birlikte sorunsuz ve hızlı bir şekilde çalışabilmektedirler,
- İşyeri güvenliği, sağlık, eğitim, sigorta vb. giderlerin azaltılmasıyla birlikte ucuz işgücü saklarlar,

Endüstriyel robotların kullanımında sağladığı avantajların yanında birtakım dezavantajlar da olabilmektedir. Bunlar ise şu şekilde sıralanabilir:

- İşgücünün ucuzlaması nedeniyle işsizlik problemlerine yol açabilmektedirler,
- Programlama sorunlarında istenmeyen zararlı sonuçlara neden olabilmektedirler,
- Tekrarlı işlerde yapılan bir hesaplama hatası tüm üretilen ürünlere yansıyabilmektedir.

Yukarıda ifade edilen işsizlik problemlerine yönelik olarak; aslında teknolojik gelişmeler, bilgi toplumu olma yolunda insanların bedenini değil, zihnini zorlayan işlerle uğraşmasına neden olmaktadır. Böylelikle de ihtiyaç sebebiyle yeni robotların tasarlanması, geliştirilmesi ve programlanmasının yanı sıra robot ve çevre ekipmanların bakım-onarım ve tamiri, robot üretim tesislerinin kurulumu vb. yeni iş imkânlarının oluşmasına yol açmaktadır.

3. ENDÜSTRİYEL ROBOTLARIN UYGULAMA ALANLARI

Sanayide Endüstriyel Robotların kullanımı başta elleçleme (Handling: tutma, taşıma ve bırakma), kaynak-lehimleme, montaj-sökme, boyama, kesme olmak üzere birçok alanda gün geçtikçe yaygınlaşmaktadır.

Sahip olduğu avantajlar nedeniyle endüstriyel robotların özellikle başta otomotiv, elektrik- elektronik, kimya, plastik makine, metal, yiyecek-içecek sektörleri olmak üzere kullanım alanları oldukça yaygındır.Endüstriyel Robotların temel kullanım alanlarının belli başlı olanları şu şekilde sıralanabilir:

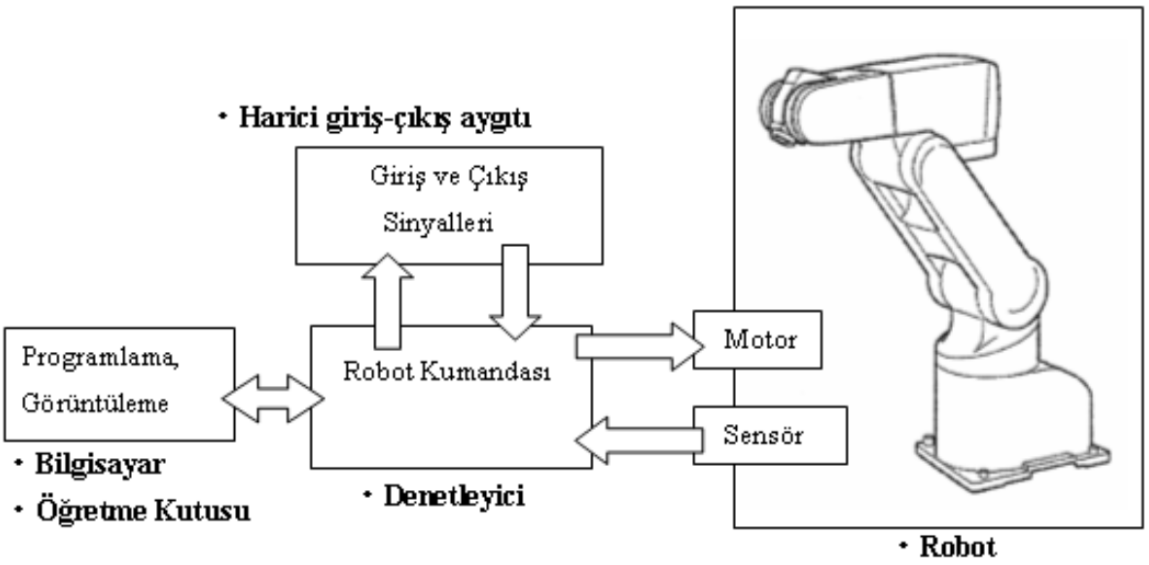
- Elleçleme (Handling) (Malzeme seçme, taşıma, sıralama, yerleştirme vb.) uygulamalarında,
- Montaj (Mounting) ve Sökme uygulamalarında,
- Nokta Kaynak (Spot Welding), Ark Kaynak ve Rotalı Kaynak uygulamalarında,
- Yapıştırma/Sızdırmazlık maddeleri uygulamalarında
- Malzeme işleme (Frezeleme vb.) uygulamalarında,
- Çapak temizleme, parlatma, boyama vb. uygulamalarda,
- Paketleme, stok ve yükleme uygulamalarında,
- Döküm, presleme, dövme vb. uygulamalarda,
- Ölçüm ve kontrol uygulamalarında

Resim 3.1. Bir araba fabrikasındaki endüstriyel robotlar.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

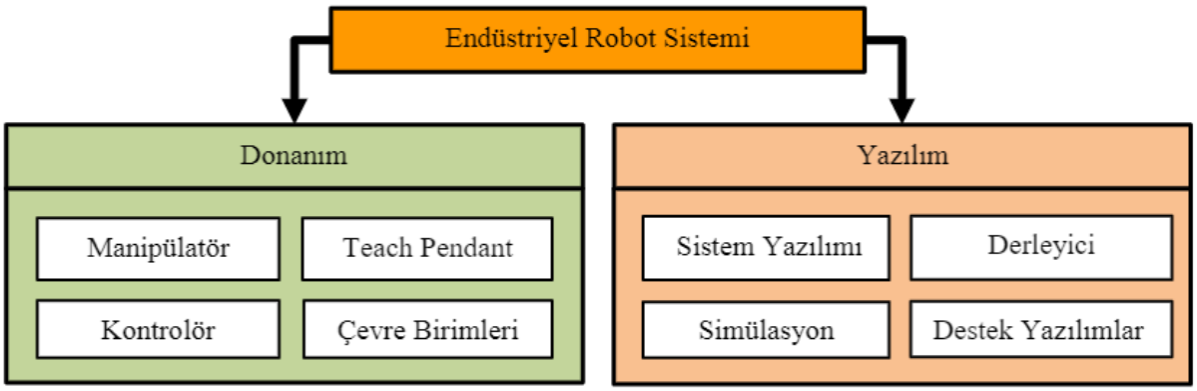
4. ENDÜSTRİYEL ROBOTLARIN YAPISI

Bir robotun ana bölümleri manipülatör, güç kaynağı ve kontrolördür. Manipülatör; parçaları, maddeleri ve üretim için gerekli araçları kaldırmada kullanılır. Manipülatörü hareket ettirmek için güç kaynağı kullanılır. Kontrolör ise güç kaynağını kontrol eder. Böylece manipülatör kendi görevini düzenlemiş olur.



Resim 4.1. Robotik sistem örneği

Robot Hücresi (Robot Cell) olarak da bilinen Endüstriyel Robot Sistemi Şekil 4.2'de görüldüğü gibi temelde donanım ve yazılım tabanlı bir sistemdir.



Resim 4.2. Endüstriyel Robot Sistemi

4.1. Mekanik Yapı

Ana gövdeyi ya da sütunu, mekanik kolları ve içine yerleştirilen aletleri içine alır.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

4.1.1. Manipülatör

Genel anlamda manipülatör deyince, asıl mekanik düzeni oluşturan robot kolu akla gelmektedir. Manipülatör, robotun kinematik zincirini oluşturan ve sahip olduğu eksenleri doğrultusunda hareket etmeyi sağlayan birbirine bağlı çok sayıda hareketli aksam ile birlikte mekanik ve elektronik aksamın oluşturduğu robot kolu olarak ifade edilebilmektedir.

4.1.2. Aletler/Efektörler

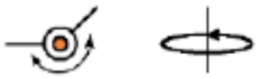
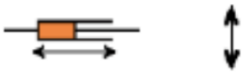
Robot teknolojisinde kullanılan kavrayıcılar, ölçüm araçları, aletler ve programa göre çalışma alanında hareket eden ve robotun ortamını manipüle etmek için robota hizmet eden diğer işlem elemanları uç efektörler olarak tanımlanır.

4.1.3. Endüstriyel Robotların Çalışma Alanı Zarfları

Çalışma Alanı Zarfı (WorkingEnvelope - WorkArea - Çalışma Hacmi - Erişim Uzayı), manipülatörün mekanik hareket yeteneğine bağlı olarak çevresinde erişebileceği tüm noktaları kapsayan uzayı anlatmaktadır. Robotun tasarlanmasında eksenler ve serbestlik derecelerine bağlı olarak manipülatörün Çalışma Alanı Zarfı değişmektedir. Bir robotun Çalışma Alanı Zarfı, diğer makine ve sistemlerle etkileşimleri açısından oldukça önemlidir.

Endüstriyel robotun çalışma alanının belirlenmesinde eklemler büyük rol oynamaktadırlar. Eklemler sayesinde robota çok yönlü hareket etme kabiliyeti kazandırılmaktadır. Dolayısıyla hareket kabiliyeti robotun çalışma alanının belirlenmesiyle doğrudan ilgili olmaktadır. Endüstriyel robotların tasarlanmasında genellikle Revolute (Dönel) ve Prismatic (Prizmatik) olmak üzere 2 temel eklem tipi kullanılmaktadır. Ayrıca endüstride silindirik, küresel, vida vb. çeşitlilikte eklem tipleri de mevcuttur.

Bir robotun her eklemi sınırlı bir hareket aralığına sahiptir. Endüstriyel Robotun ilk 3 ekleminin oluşturduğu ve bilek pozisyonunun belirlenmesini sağlayan eksen Major Eksen, sonraki 3 eklemin oluşturduğu ve elin yönünü belirlenmesini sağlayan eksen ise Minör Eksen olarak adlandırılmaktadır. Endüstriyel Robotun Majör eksen elden yapısına bağlı olarak çalışma zarfı belirlenir.

Eklem Tipi	**Sembolik Gösterim**		**Eklem Tanımı**
	Harfsel	Şekilsel	
Rotational – Revolute (Dönel)	R		Eksen etrafında dönem hareketi
Prismatic – Translational (Prizmatik – Ötelemeli)	P (veya T)		Eksen boyunca lineer hareket

Tablo 4.1. Robot eklem tipleri

4.1.4. Eklem Sayıları ve Özellikleri

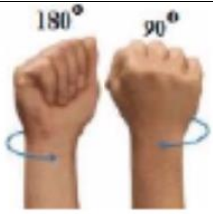
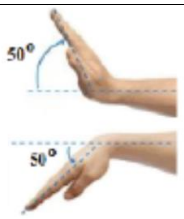
Endüstriyel robot sisteminde, bir manipülatörün sahip olduğu eksen sayısı ve özelliğine bağlı olarak manipülatörün hareket kabiliyeti değişmektedir. Tablo 4.1'de manipülatör eksen tipleri ve özellikleri görülmektedir.

Eksen	Tip	Özellik	Açıklama
1-3	Major eksen	Bilek pozisyonu belirleme	Endüstriyel robotun çalışma alanı zarfı belirlenir
3-6	Minör eksen	Alet yönü belirleme	3D uzayda manipülatör ucundaki aletin yönlendirilmesi sağlanır.
7-n	Redundant eksen	Engellerden sakınma	Manipülatörün istenilmeyen alanlardan kaçınılması veya çalışma uzayındaki engellerin etrafından erişim sağlanır.

Tablo 4.2. Endüstriyel robot eksenleri

Robot, aynen çalışan bir insan gibi kendisine verilen görevi yerine getirmek durumundadır. Bu görevi ifa için insanın kemiklerindeki manivela ve adalelerindeki eğilip bükülme sistemlerine sahip olması gerekir.

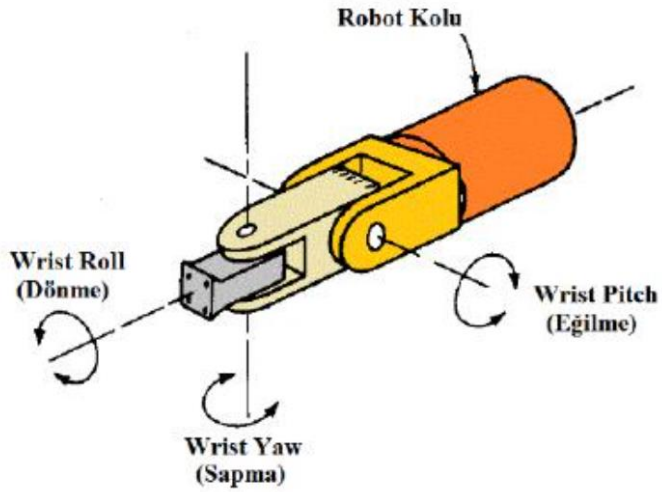
Endüstriyel robotların yönü Roll, Pitch ve Yaw ile ölçülmektedir. İnsan ve robot bileği benzer hareket kabiliyetlerine sahiptir. Kişiden kişiye değişmekle birlikte, insan bileğinin hareket kabiliyetleri Tablo 4.3'de görülmektedir. Resim 4.1'de ise bir endüstriyel robotun bileğine (minör eksen) yönelik Row (Dönme), Pitch (Eğilme) ve Yaw (Sapma) hareketleri görülmektedir.

Hareket Şekli	Açıklama	Örnek kabiliyet
ROLL (Dönme)	Sağ kol 0° konumunda düz ve avuç içi aşağı şekilde iken, bileğin saat yönünde ve saat tersi yönde hareket ettirilmesi	 Dönme=180°+90°
PITCH (Eğilme)	Sağ kol 0° konumunda düz ve avuç içi aşağı şekilde iken, bileğin aşağı ve yukarı hareket ettirilmesi.	 Eğilme=50°+50°

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

YAW (Sapma)	Sağ kol 0° konumunda düz ve avuç içi aşağı şekilde iken, bileğin sola ve sağa hareket ettirilmesi	 Sapma=20°+45°

Tablo 4.3. İnsan elinin hareket kabiliyetleri



Resim 4.3. Robot bilek hareketleri (minör eksen)

İnsan kolunun ikinci bölümünde üç serbestlik dereceli iki bağlantı vardır; iki serbestlik dereceli omuz ve bir serbestlik dereceli dirsek. Fakat robotun tek serbestlik dereceli bir omzu vardır. Robotun bel kısmı insan omzunun ikinci hareketini üstlenir.

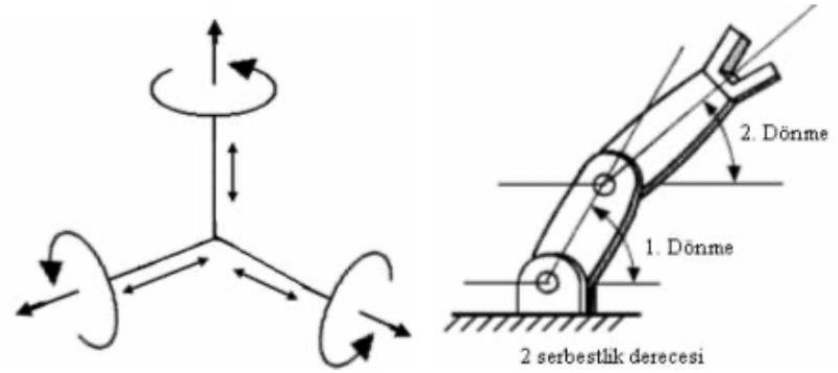
Kolun sonunda beş parmak vardır. Parmaklar bir cismi sıkacak olursa, parmak bağlantıları bağımsız değillerdir ve tutulan cismin konumuna ve yönüne etki etmezler. Parmaklar tekil ve bağımsız kullanılırsa hareketlilik sağlar.

Taşıma işlemi yapan robotlarda (pick and place tipi) başparmak, birinci ve orta parmak kullanılır.

Kol yapısının önemli özelliklerinden biri de üst kolun ön kola oranıdır. Bu 1:1 ila 1:2 civarındadır. Bu oran, robotun ön kolunun üst kola eşit ya da daha kısa olduğu anlamına gelir. Bu oran karşılanmazsa robotun hareketinde bir düzensizlik oluşacaktır. Mekanik olarak değerlendirildiğinde insan kolu doğrusal ve doğrusal olmayan elemanlardan oluşan hiyerarşik bir yapı olduğu görülür.

Serbestlik Derecesi (Degree of Freedom: DOF): Bir nesnenin yapabileceği bağımsız hareketlerin sayısı serbestlik derecesi sayısıdır. Serbest bir cisim uzayda serbest olarak hareket ettiği zaman altı serbestlik derecesine sahiptir. Bunların üçü “yer” diğer üçü ise “yönelim” içindir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

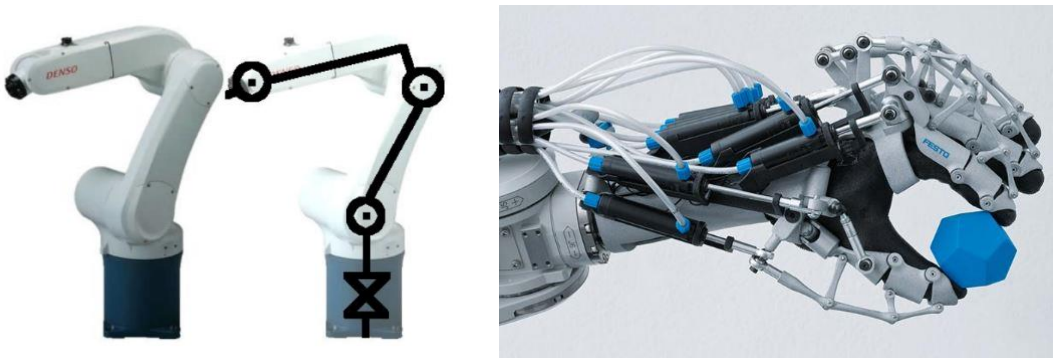


Resim 4.4. Serbestlik derecesi

Koldaki eklem sayısı azaldıkça kolun çalışma uzayı, bağlantı parçalarının fiziksel boyutları aynı kalsa bile yine de hacim olarak küçülür ve kolun bu uzaydaki herhangi bir noktaya erişebilmesindeki esneklik de azalır.

Bazı işlemler bu esnekliğin yüksek olmasını gerektirdiğinde kolun serbestlik derecesinin de yüksek seçilmesi zorunluluğu doğar. Böyle durumlarda altıdan da fazla, dokuz ya da on eklemli kol yapıları kullanılmaktadır. Serbestlik derecesini artırmak robot kol maliyetini arttıracaktır. Yine de olağan yapıda insanın bel, omuz, dirsek, bilek ve parmaklarındaki hareketlerin benzerlerini robot kolların eklemlerindeki hareketlerde bulmak olanaklıdır.

Ayrıca altıdan fazla mafsala sahip olan robotlarda randıman artsa bile koordinat dönüşümlerinin hesaplanmasında programlama zorluklarına yol açmaktadır.



Resim 4.5. DOF and harici hava besleme ünitesi örneği

4.2. Kontrol Sistemi

Sayısal elektronik devreler topluluğundan oluşur.

4.2.1. Enerji Besleme Yöntemleri

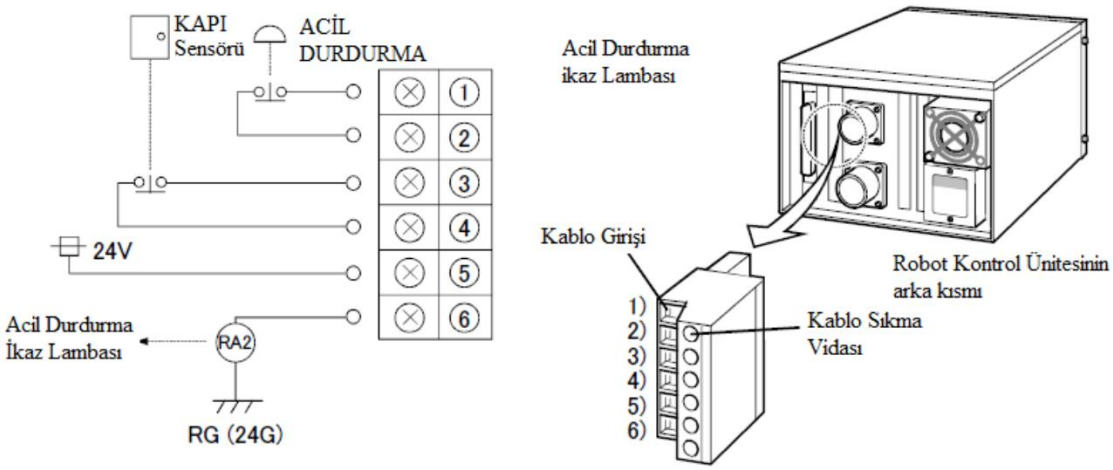
Robotlara gerekli enerjinin sağlanması ve sinyallerin iletilmesinde genel olarak iki yöntem kullanılır:

- Harici enerji beslemesi; Bu yöntemde gerekli olan enerji, robottan bağımsız olarak kollar veya benzer tertibatlar üzerinden hortum veya kablo paketleri vasıtasıyla sağlanır. Robot hareketine bağlı olarak hortum veya kablo paketlerinin zarar görmemesi için düzgün bağlantı yapılmasına özen gösterilmelidir.
- Robota entegre enerji beslemesi; Bu yöntemde gerekli olan enerji, robot yapı grupları içerisinden veya robot yapı grupları boyunca iletilerek sağlanmaktadır.

4.2.2. Giriş çıkış sayıları ve özellikleri

Acil Durdurma Butonu giriş ve çıkış noktası, Robot kol muhafaza kabin kapısı sensörünün giriş noktası ve paralel giriş çıkış birimi olmak üzere üç tip giriş çıkış bağlantısı bulunmaktadır.

Paralel Giriş Çıkış bağlantı noktasındaki giriş çıkış sayısı kullanılan robotun tipine göre değişmektedir. Bazı robotlarda 16 giriş 16 çıkış, bazılarında ise 32 giriş 32 çıkış bulunmaktadır.



Şekil 4.6. Robot giriş çıkış bağlantıları (KUKA)

4.2.3. Çevre Birimleri Bağlantıları

Endüstriyel robot ve çevre birimleri arasında iletişimin kurulması için çeşitli yöntemler kullanılır. Bunlar;

- Entegre girişler/çıkışlar
- Veriyolu (bus) sistemleri
- Ethernet'tir.

4.3. Güç Ünitesi

Robot manipülatör eklem hareketleri için gerekli gücün sağlanmasında kullanılan sürücü sistemleri şunlardır:

• Elektriksel Sürücü Sistemleri

• Hidrolik Sürücü Sistemleri

• Pnömatik Sürücü Sistemleri

Hidrolik sürücü sistemlerine sahip endüstriyel robotlar genellikle ağır endüstride kullanılmakla birlikte erimiş çelik işleme, otomobil parçası vb. büyük yükler için yüksek hız ve dayanım sağlamaktadırlar. Robotun taban bağlantısının yapılması gerekmektedir. Hidrolik sürücüler iller büyük ve hantal olmakla birlikte gürültülü, yağ sızdırma ve temizlik sorunlarına sebep olabilmektedirler. Düşük güçte yüksek tork üretebilmelerine rağmen, performansları doğrusal olmadığından kontrolleri zordur.

Günümüzde çoğu robot manipülatörleri için DC servo motor ve step motor sürücüleri kullanılmaktadır. Elektrik sürücü sistemleri temiz yapılı olmakla birlikte hassasiyet ve tekrarlanabilirlik olarak daha iyidirler. Ancak elektrik sürücü sistemleri, hidrolik sürücülere göre hem daha yavaş hem de düşük güçlüdürler. Robotun taban bağlantısının yapılması gerekmektedir. Elektrik motorları, redüktörler yardımıyla daha güçlü ve hassas bir hale getirilebilmektedir. DC servo motorların düşük güçte yüksek tork üretmeleri önemli bir tercih sebebi olmuştur. Step Motor ise genellikle yüksek tork ihtiyacı gerektirmeyen tutma, taşıma ve yerleştirme gibi daha basit uygulamalarda kullanılmaktadır.

Pnömatik sürücü sistemleri, özellikle birkaç serbestlik dereceli (DOF) küçük robotlar için kullanılmaktadır. Genellikle Tutma-Bırakma gibi basit işlemlerin hızlı bir şekilde yapılmasını saklarlar. Pnömatik sürücü sistemlerinin enerji verimliliği daha iyi olmakla birlikte, geri besleme (feedback) kontrolü zordur. Ayrıca hareketli robot pistonlarının ataletini hızlı bir şekilde ortadan kaldıracak hava basıncı tertibatı sıkıntısından dolayı da kontrol işlemi zorlaşmaktadır. Bu nedenle de genellikle basit uygulamalarda tercih edilmektedirler. Genellikle manipülatör efektörleri pnömatik yapıda olmaktadır.

4.4. Sensörler

Robotlardaki algılayıcılar çok geniş alanlarda kullanılırlar. Bunlar genel olarak;

- Dokunmatik sensörler,
- Yaklaşım ve oran sensörleri,
- Muhtelif sensörler ve temel sensör sistemleri,
- Otomatik görme sistemleri, olarak gruplandırılırlar.

4.4.1. Dokunmatik Sensörler

Dokunmatik sensörler bazı katı cisimlerin ve bunların kendileri arasındaki bağlantıyı gösteren cihazlardır. Bu sensörleri iki sınıfa ayırabiliriz: Bunlar temas ve kuvvet sensörleridir. Temas sensörleri, cisimler arasında oluşan veya oluşmayan bağlantıları ikili çıkış sinyalleri olarak verir. Kuvvet sensörleri (Bazen gerilim sensörleri olarak da adlandırılır) sadece cisimler arasındaki bağlantı kuvvetinin büyüklüğünü de gösterir.

<u>Temas Sensörleri</u>

Temas sensörleri bağlantı kuvvetinin büyüklüğüne bakmaksızın iki cisim arasındaki bağlantının olup olmadığını göstermekte kullanılır. Bu kategoride limit anahtarlar gibi basit cihazlar, mikro anahtarlar vb. yer alır. Temas sensörleri periyodik olarak robotların kenetlenme sistemlerinde kullanılır. Örneğin

konveyörler boyunca bir noktanın hızlanmasının gösterilmesinde kullanılabilir. Resim 44'de temas sensörleri bulunan bir robot sistemi görülmektedir.
Temasla algılamanın kullanıldığı bir başka yer de, malzeme yüzeyini araştırma kontrolleridir. Serbestlik derecesi 6 olan bir robot parça yüzeylerine ulaşabilme kapasitesine sahiptir. Üç eksenli koordinatlarda makine ölçümleri zordur.

Kuvvet Sensörleri

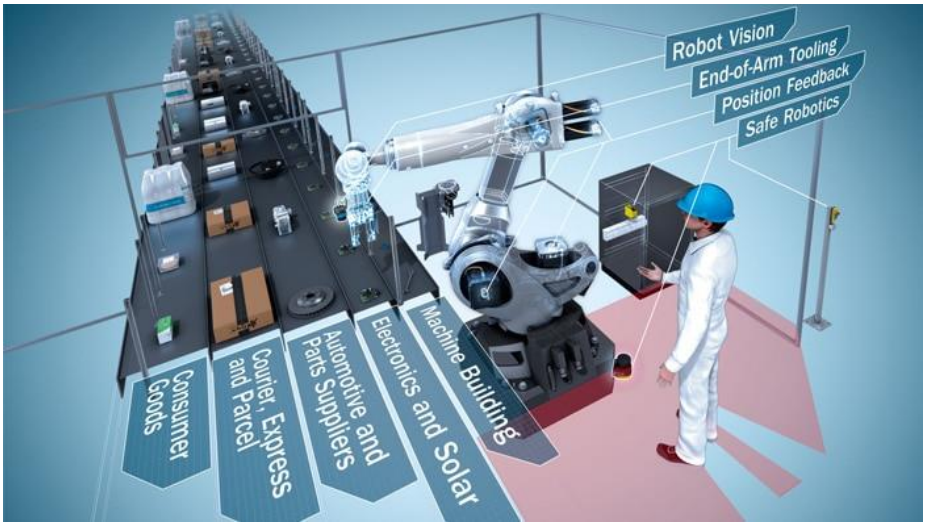
Ölçülebilen kuvvet kapasitesi robota yaptırılan görev sayısının izin verdiği kadardır. Bu kapasite değişik ebatlardaki maddelerin kullanılması, makine yüklenmesi, işlerin kurulması, uygun seviyede kuvvet uygulanması gibi şeyler içerir.
Robotlarda kuvvet algılanması birçok yoldan sağlanabilir. Genel olarak kullanılan teknik, kuvvet algılama krankpini tekniğidir. Bu, bilek ve kavrayıcı arasına monte edilmiş bir yük hücresinden oluşur. Başka bir teknik ise her bir bağlantıda oluşan momentlerin ölçülmesi tekniğidir. Bu genellikle bütün motor bağlantılarındaki motor akımlarının algılanması ile sağlanır.

4.4.2. Yaklaşım ve Oran Sensörleri

Yaklaşım sensörleri bir cisim diğer bir cismi kapattığı zaman devreye girer. Cismin nasıl kapatıldığını inceler ve onun kendine has çalışmasını sağlar. Algılama mesafesi birkaç milimetre ile 15-20 cm arasında olabilir. Bu sensörlerin bazıları cisim ile sensörler arsındaki uzaklığı ölçer. Bunlara da oran sensörleri adı verilir. Yaklaşım ve oran sensörleri robotun hareketli bölümleri olan el ve son etkileyicilere yerleştirilmiştir. Robotlarda yaklaşım sensörlerinin pratikte bir kullanımı ise, iş parçası veya diğer cisimlerin varlığını veya yokluğunu tespit etmekte kullanılır. Oran sensörleri robotlarla ilişkili olan cismin konumunu belirlemede kullanılır. İleri teknoloji yaklaşım ve oran sensörlerinin yapımı için uygundur. Bunlar; optik malzemeler, akustik, elektriksel alan ve diğerleridir.

4.4.3. Muhtelif Sensörler ve Temel Sensör Sistemleri

Bu muhtelif kategoriler sensörlerin ve transdüserlerin diğer tiplerini içerir. Bunlar robotlardaki çalışma bölgelerinin duyma yeteneğindeki değişimleri algılayabilen cihazlardır. Bu duyma yetenekleri ise sıcaklık, kuvvet, akışkan akışı ve elektriksi özellikler içerir.



Resim 4.7. Sensör Uygulamaları

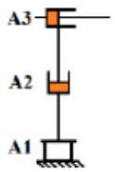
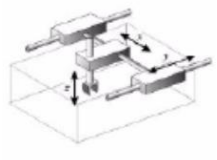
5. ENDÜSTRİYEL ROBOTLARIN SINIFLANDIRILMASI

Endüstriyel robotlar; geçmişten günümüze değişik yapı ve özellikle teknolojiye sahip olmakla birlikte genel anlamda şu şekilde sınıflandırılabilirler:

- Kartezyen Robotlar
- Silindirik Robotlar
- Küresel Robotlar
- SCARA Robotlar
- Mafsallı Robotlar
- Delta Robotlar

5.1. Kartezyen Robotlar

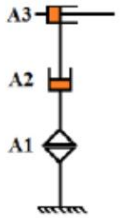
Kartezyen Robot, 3 majör ekseninin hepsi prizmatik olan (PPP) en basit robottur. Bütün robot hareketleri birbirine dik açılı bir şekilde gerçekleşmektedir. Kartezyen robotlarda hareketli kısımlar X, Y ve Z kartezyen koordinat sistemi eksenlerine paralel olarak hareket etmektedirler. Kartezyen robotlar, en kısıtlı hareket serbestliğine sahip robot tasarımına sahiptirler. Kartezyen Robotların Çalışma Alanı Zarfı dikdörtgen prizması şeklindedir. Kartezyen robotlara yönelik özellikler ve uygulama örneği Resim 5.1'de görülmektedir. Kartezyen robotlar, malzeme taşıma veya yüzeysel çalışma yapmak için yere veya tavana monte edilebilmektedirler. Kartezyen robotlar özellikle mermer, cam, ahşap gibi malzemelerin montajı, taşınması ve işlenmesi işlemlerinde kullanılmaktadır.

Major Eksen Tipi	Kinematik Yapı	Çalışma Alanı	Uygulama Örneği
P-P-P (3P)	 Kartezik Koordinatları	 Prizmatik	

Resim 5.1. Kartezyen Robotlar

5.2. Silindirik Robotlar

Silindirik Robot, birinci eklemi Revolute (R), ikinci eklemi Prismatic (P) olan (RPP) robottur. Silindirik robotlarda, robot kolu silindir veya silindir parçası şeklinde hareket etmektedir. Silindirik Robotların Çalışma Alanı Zarfı silindir parçası şeklindedir. Silindirik robotlara yönelik özellikler ve uygulama örneği Resim 5.2'de görülmektedir.

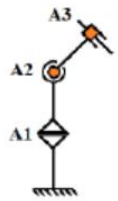
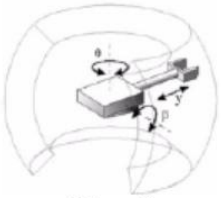
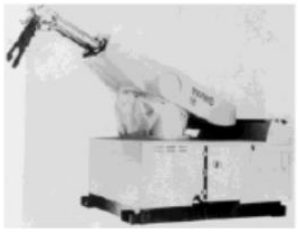
Major Eksen Tipi	Kinematik Yapı	Çalışma Alanı	Uygulama Örneği
R-P-P (R2P)	 Silindirik Koordinatlar	 Silindirik	

Resim 5.2. Silindirik Robotlar

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

5.3. Küresel Robotlar

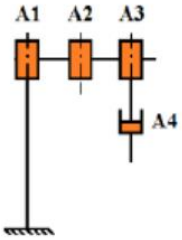
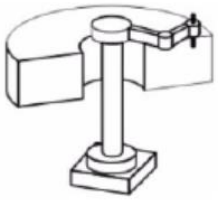
Küresel Robot, ilk 2 eklemi Revolute (R) ve 3. eklemi Prismatic (P) olan (RRP) robottur. Küresel Robotların Çalışma Zarfı küre şeklindedir. Küresel robotlara yönelik özellikler ve uygulama örneği Resim 5.3'de görülmektedir.

Major Eksen Tipi	Kinematik Yapı	Çalışma Alanı	Uygulama Örneği
R-R-P (2RP)	 Küre Koordinatları	 Küresel	

Resim 5.3. Küresel Robotlar

5.4. SCARA Robotlar

SCARA (Selective Compliance Assembly Robot Ann) Robot, Küresel Robot gibi ilk 2 eklemi Revolute (R) ve 3. eklemi Prismatic (P) olan (RRP) veya ilk 3 ekseni Revolute (R) ve 4. Ekseni Prismatic (P) olan (RRRP) robottur. SCARA Robotun, Revolute (R) eklemleri yatay hareket etmektedir. SCARA robotlar doğruluk, yüksek hız ve kolay montaj açısından önemli özelliklere sahiptir. SCARA robotlara yönelik özellikler ve uygulama örneği Resim 5.4'te görülmektedir.

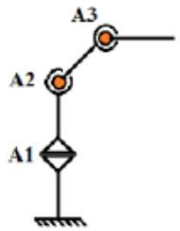
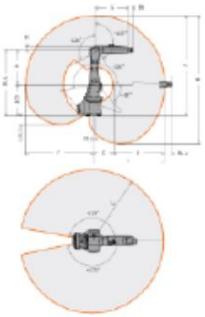
Major Eksen Tipi	Kinematik Yapı	Çalışma Alanı	Uygulama Örneği
R-R-R-P (3RP)	 Mafsal Koordinatları (Yatay)	 Döndürme Kollu Robot (SCARA)	

Resim 5.4. SCARA Robotlar

5.5. Dikey Mafsallı (Articulated) Robotlar

Dikey Mafsallı (Artlenlated) Robotlar, insan kolu anatomisine benzeyen majör eklemlerinin her üçü de Revolute (R) olan (RRR) robotlardır. Dikey Mafsallı robotlara aynı zamanda Antropomorfik veya Revolute Robot adı da verilmektedir. Dikey Mafsallı Robotların Çalışma Alanı tam karaya benzer şekildedir. Mafsallı robotlar, hareket yeteneklerinden dolayı daha kabiliyetli robotlardır. Mafsallı robotlar, özellikle kaynak ve boyama alanlarında yaygın olarak kullanılmaktadır. Dikey Mafsallı (Articulated) robotlara yönelik özellikler ve uygulama örneği Resim 5.5'te görülmektedir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

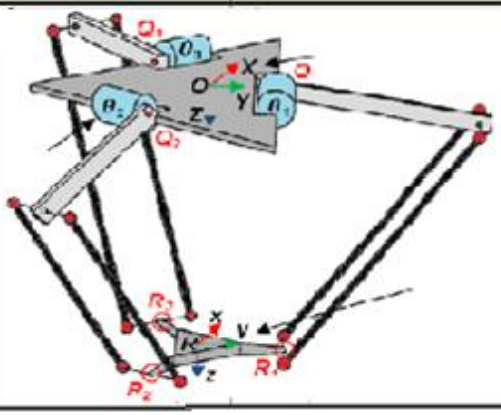
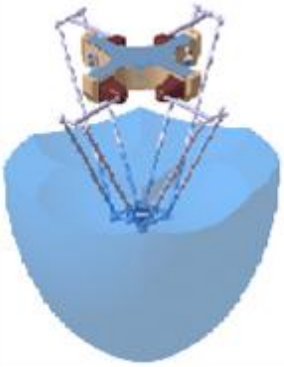
Major Eksen Tipi	Kinematik Yapı	Çalışma Alanı	Uygulama Örneği
R-R-R (3R)	 Mafsal Koordinatları (Dikey)		

Resim 5.5. Dikey Mafsallı Robotlar

5.6. Delta Robotlar

Bir delta robot, tabandaki üniversal mafsallara bağlı üç koldan oluşan bir tür paralel robottur. Anahtar tasarım özelliği, kollarda uç efektörün yönünü koruyan paralelkenarların kullanılmasıdır.

Delta robotları, oldukça hızlı olabildikleri ve bazıları dakikada 300'e kadar toplama gerçekleştirebildikleri için fabrikalarda toplama ve paketlemede popüler bir kullanıma sahiptir. Özellikle ilaç ve biskivü fabrikalarında.

Major Eksen Tipi	Kinematik Yapı	Çalışma Alanı	Uygulama Örneği
			

Resim 5.6. Delta Robotlar

6. ENDÜSTRİYEL ROBOTLARIN PERFORMANS ÖLÇÜTLERİ

Endüstriyel robotların performanslarının belirlenmesinde hassasiyet, hız, yük taşıma kapasitesi, tepkime süresi, kararlılık gibi çeşitli faktörler ön plana çıkmaktadır.

6.1. Hassasiyet

Hassasiyet (Precision) ölçülebilecek en küçük değişim miktarı olarak tanımlanmaktadır. Endüstriyel robotların hareket hassasiyeti, robotun en önemli performans göstergelerinden biridir. Robotun performans ölçütlerinden hassasiyet; doğruluk, çözünürlük ve tekrarlanabilirlik özelliklerinin bir fonksiyonu olarak tanımlanmaktadır.

6.2. Doğruluk

Doğruluk, yapılan ölçümlerin gerçek değere göre ne kadar yakın olduğu ile ifade edilmektedir. Bu açıdan doğruluğun ifade edilmesinde ölçme ve ölçme hatası önemli bir rol oynamaktadır. Ölçme, bilinmeyen bir büyüklüğün kendi cinsinden bilinen ve birim olarak kabul edilen büyüklükle karşılaştırılması işlemidir. Ölçme hatası ise, ölçüm sonucu elde edilen değer ile gerçek değer arasındaki farktır. Bir robotun doğruluğu ise, robotun TCP'sini çalışma alanı zarfı içerisinde herhangi bir noktaya konumlandırma mesafesi yeteneğidir.

6.3. Çözünürlük

Çözünürlük, çıkış değerinde gözlenebilir bir değişikliği üreten en küçük giriş değişim aralık değeridir. Endüstriyel robotlarda çözünürlük ise, eksenlerin hareket adım aralığı ile ilişkilidir. Bu şekilde eksen hareket adım aralığı azaldıkça, robotun çözünürlüğü ters orantılı olarak artmaktadır.

6.4. Tekrarlanabilirlik

Tekrarlanabilirlik, aynı koşullarda aynı giriş değerinin tekrarlanan uygulamalarında aynı çıkışı vermesi kabiliyeti olarak tanımlanmaktadır. Endüstriyel robotun tekrarlanabilirliği ise, çalışma alanı zarfı içerisinde robota daha önceden öğretilen bir noktaya robotun TCP'ni tekrar tekrar konumlandırma yeteneği olarak tanımlanmaktadır. Bu şekilde bir endüstriyel robotun tekrarlanabilirliği, robotun tekrarlanan hareketleri sonucunda, robot TCP'si ile öğretilen nokta arasındaki maksimum hata miktarı ile belirlenmektedir.

6.5. Tepkime Süresi

Tepkime Süresi (Response Time) yani Cevaplama Süresi, sistemin girişine verilen değişikliğe karşılık olarak çıkışında fark edilebilir bir değişikliğin elde edilebilmesi için gereken süredir. Endüstriyel Robotlar için Tepkime süresi ise, robotun hareket hızıyla ilişkili olarak kısa süre içerisinde bir sonraki duruma geçme yeteneği olarak ifade edilmektedir. Endüstriyel robotların tercihen hızlı bir tepkime süresine sahip olması istenmektedir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

6.6. Kararlılık

Kararlılık (Stability), sabit bir girişi ölçmek için bir zaman periyotu süresince aynı çıkışı verebilme kabiliyetidir. Endüstriyel Robotlar için kararlılık ise, genellikle bir pozisyondan diğer pozisyona hareketin gerçekleştirilmesi esnasında robot kolunda meydana gelen salınımların ölçüsü olarak ifade edilmektedir. Kararlılığı iyi olan bir endüstriyel robotun, hareket esnasında hiç salınım göstermemesi gerekmektedir.

6.7. Yük Taşıma Kabiliyeti ve Hız

Endüstriyel robotların yük taşıma ve hız kapasiteleri; robotların sistem tasarımı, büyüklüğü, koordinat ve sürücü sistemleri gibi teknolojik unsurlarına bağlı olduğu kadar taşınan malzemelerin boyut ve şekillerine bağlı olarak da değişiklik göstermektedir. Endüstriyel Robotlar için genellikle Maksimum ve Nominal yük taşıma kapasiteleri ön plana çıkmaktadır:

Maksimum Yük Taşıma Kapasitesi; Bir robotun minimum hızda iken tekrarlanabilirlik sınırları içerisinde taşıyabileceği maksimum yük değeri olarak ifade edilmektedir.

Nominal Yük Taşıma Kapasitesi: Bir robotun maksimum hızda iken tekrarlanabilirlik sınırları içerisinde taşıyabileceği maksimum yük değeri olarak ifade edilmektedir.

Endüstriyel Robot hızı, söz konusu bir iş çevriminin tamamlanması için geçen süreyi anlatmaktadır. Endüstriyel Robotun hızlı olması demek, yapılması istenen işin daha kısa sürede yapılması anlamına gelmektedir.

! Bir sonraki bölümde, son yıllarda tanıtılan işbirlikçi robotları anlatacağız. Haydi hazırlık için aralarındaki farklara bir göz atalım.

Resim 6.1. Robot – Cobot karşılaştırma tablosu

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

COBOTLAR (İŞBİRLİKÇİ ROBOTLAR)

7. COBOT NEDİR?

İşbirlikçi robot veya diğer adıyla Cobot, ortak çalışılan bir alanda insanlarla yan yana etkileşime girerek güvenli bir şekilde çalışabilmeyi sağlayan robotlardır. İlk kez 1999 yılında ortaya çıkan cobot kelimesi, İngilizce iş birliği (collaboration) ve robot kelimelerinin birleşmesinden oluşmuştur.

Cobotları robotlardan ayıran en önemli ve ayırt edici özellik ihtiyaç halinde insanlarla yan yana etkileşim ve güven içerisinde çalışabilmesidir. Kısaca Cobot, herhangi bir güvenlik bariyerine ihtiyaç olmadan yan yana çalışabilen endüstriyel robot koludur.

Uluslararası Robotik Federasyonu (IFR) dört tip iş birliğine dayalı üretim uygulaması tanımlar :

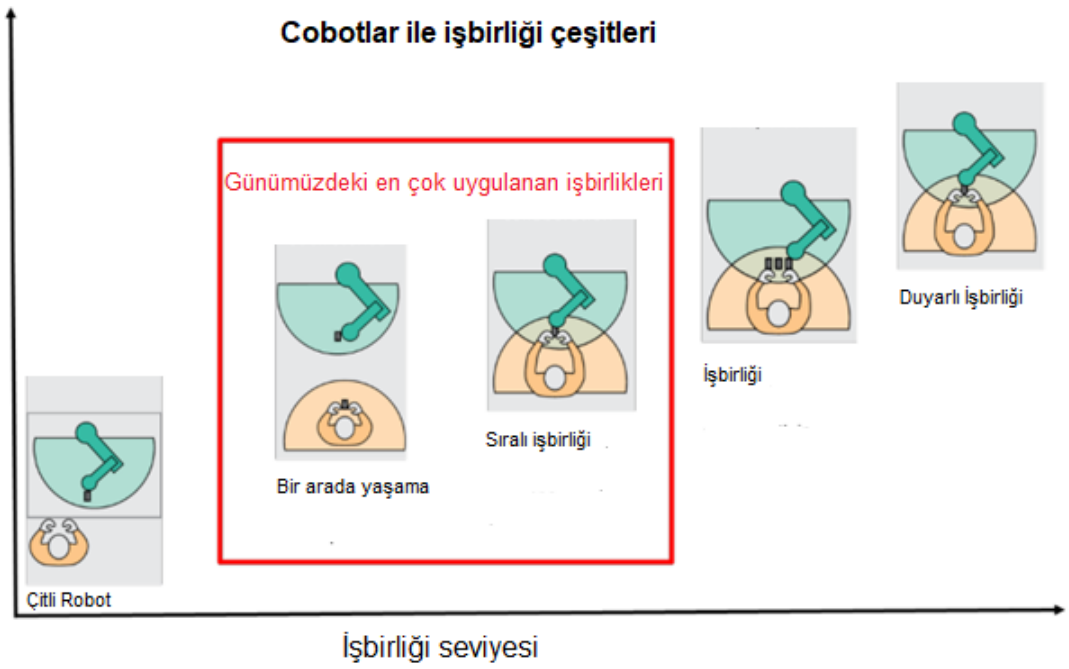
Bir arada yaşama: İnsan ve robot ortak bir çalışma alanı olmadan birlikte çalışır.

Sıralı iş birliği: İnsan ve robot, bir çalışma alanının tamamını veya bir kısmını paylaşır, ancak aynı anda bir parça veya makinede çalışmaz.

İş birliği: Robot ve insan aynı parça veya makinede aynı anda çalışıyor ve ikisi de hareket halinde.

Duyarlı iş birliği: Robot, çalışanın hareketine gerçek zamanlı olarak yanıt verir.

Günümüzde cobot'ların çoğu endüstriyel uygulamalarda insan ile aynı alanı paylaşır, ancak görevleri bağımsız veya sırayla tamamlar (bir arada yaşama veya sıralı iş birliği). İş birliği veya duyarlı iş birliği şu anda daha az yaygındır. Uluslararası Robotik Federasyonu'nun (IFR) 2018 yılında yayınladığı yayında bu durum aşağıdaki şekilde gösterilmiştir.

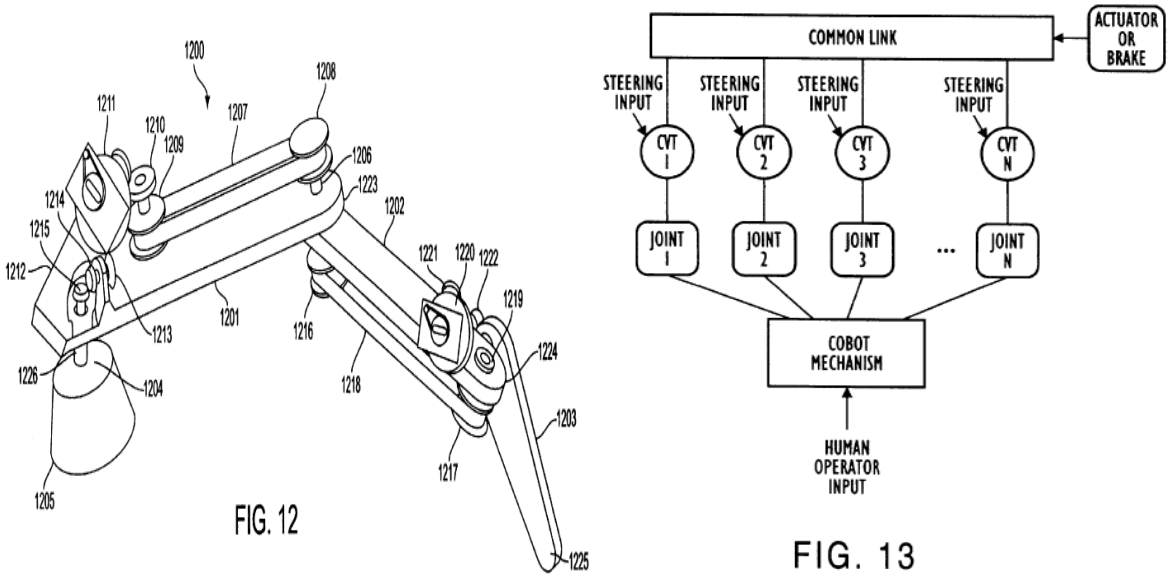


Resim 7.1. İşbirlikleri Seviye Grafiği

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

8. COBOTLARIN TARİHÇESİ

Cobotlar 1996'da Northwestern Üniversitesi'nde profesörler olan J. Edward Colgate ve Michael Peshkin tarafından icat edildi. Bu iki profesör 1997 yılında 5,952,796 sayılı ABD Patent belgesini almışlardır. Bu patente göre Cobot kelimesi tanımlanmıştır. Bu profesörler tarafından kurulan Cobotics firması 2003 yılına kadar çeşitli cobotlar üretmişlerdir.



Resim 8.1. 1999 yılında ilk cobotun patent başvurusundaki mafsal kol ve mimari diyagramları.

KUKA firması ilk cobotu 2004 yılında piyasaya sürmüş, Universal Robots firması ise ilk cobotunu 2008 yılında piyasaya sürmüştür. FANUC ve ABB firmaları ilk cobotlarını 2015 yılında üretmişlerdir.

Grand View Research firması araştırmasına göre cobot piyasasında olan firmalar şöyledir;

ABB Group DENSO Robotics Epson Robots EnergidTechnologies Corporation F&P Robotics AG Fanuc Corporation KUKA AG MRK-Systeme GmbH Precise Automation, Inc Rethink Robotics, Inc Robert Bosch GmbH Universal Robots A/S Yaskawa Electric Corporation MABI Robotic AG Techman Robot Inc. Franks Emika Gmbh AUBO Robotics Comau S.p.A.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

9. COBOTLARIN ÖZELLİKLERİ

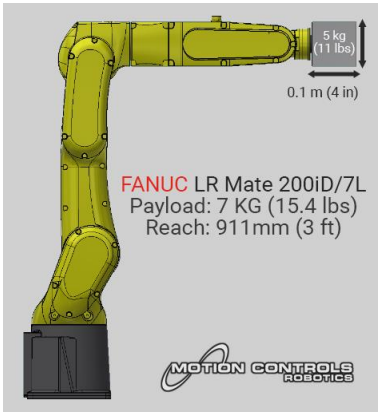
9.1. Teknik Özellikleri

Genel olarak bakılacak olursa cobotlar taşıma kapasitelerine oranla son derece hafiftirler. 10 kg taşıyan bir robotun ağırlığı 150 kg civarlarındadır. Ama aynı taşıma kapasitesindeki cobot yalnızca 30 kg ağırlığındadır. Besleme voltajı ev kullanıcı şehir şebeke sistemidir. Bu da Avrupa için AC 220 volt demektir.

Cobotlar 6 eksenlidir ve tüm eksenlerinde 360 derece dönüş özelliğine sahiptirler. Cobotların bazı önemli teknik özellikleri aşağıda verilmiştir.

Payload (Faydalı Yük): Kolaboratif robotun taşıyabileceği maksimum yüktür. 0,5 kg 'dan başlar, 35 kg'a kadar çıkmaktadır.

Reach (Maksimum erişim mesafesi): Kolaboratif robotun kollarıyla erişebileceği en uzak mesafedir. 250 mm'den başlar, 1813 mm'ye kadar çıkmaktadır.



Resim 9.1. Fanuc marka cobotun faydalı yük ve maksimum erişim mesafesi verileri.

DoF – Degree of Freedom (Serbestlik Derecesi): Kolaboratif robotun kaç eksende dönme ve/veya öteleme hareketi yapabildiğini gösterir. Bu arttıkça robot daha esnek hareketler yapabilir, ulaşması zor noktalara ulaşma kabiliyeti artar. 4'ten başlar, 14'e kadar çıkmaktadır.

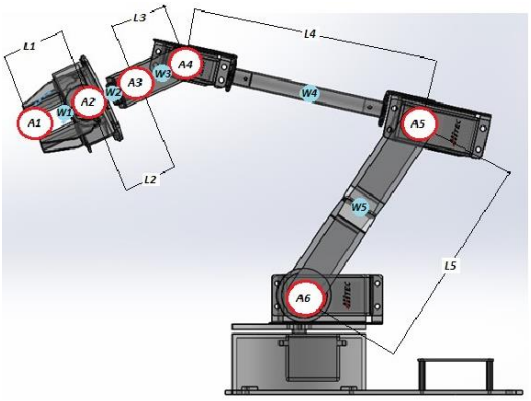
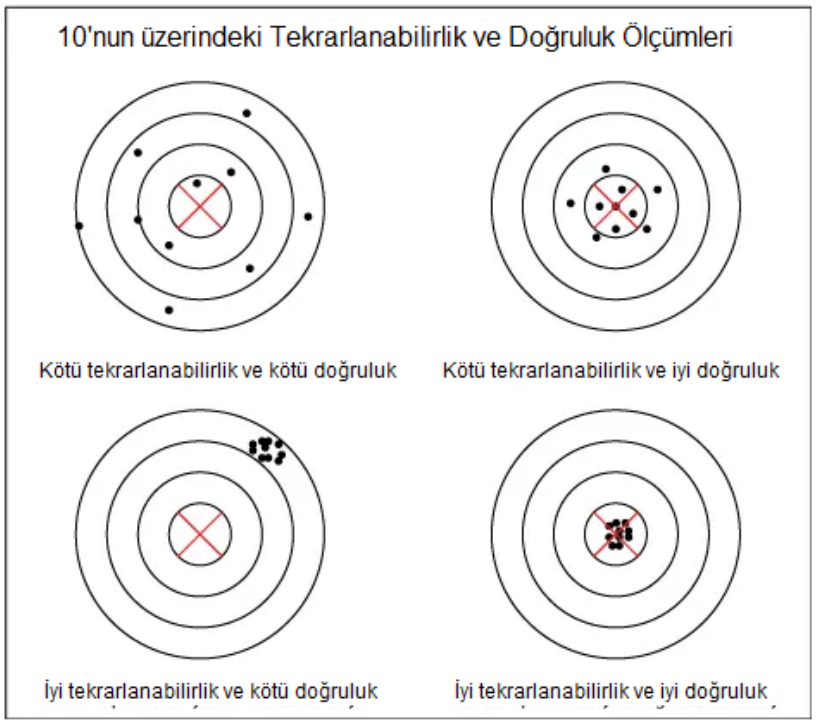


Figure 1:

Resim 9.2. 5 DoF cobotik kolun tasarımı

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Tekrarlanabilirlik : Kolaboratif robotun tekrarlanan işi, ne kadar doğrulukla yaptığının bir göstergesidir. Örneğin sürekli, dart tahtasının tam orta noktasını boyamasını istiyoruz. Robot sürekli tam olarak o noktaya giderse tekrarlanabilirliği kusursuzdur. Ancak bu neredeyse imkansızdır. Dolayısıyla cobot/robot, tam gitmesi gereken noktadan 0.01 mm ya da 0.1 mm gibi aşağı yukarı sağa sola gidebilir. Bu maks. sapma değeri cobotun/robotun operasyonu ne kadar aynı şekilde (tekrarlanabilir) yapabildiğini gösterir. Değer arttıkça tekrarlanabilirliği azalır ve bu durum operasyona bağlı olarak çoğunlukla arzu edilmez. Bu değer günümüzde 0.01 mm'dir.



Resim 9.3. Codiac for Co-Packers marka cobot tekrarlanabilirlik ve doğruluk testi

9.2. Kullanım Özellikleri

Cobotların kullanım özellikleri kısaca şöyledir;

Üreticilere geleneksel robotlarda olmayan kolay programlama

Esnek ve hızlı kurulum süreci

Çok çeşitli alanlarda kullanım

Küçük ölçekli üretime montaj edilebilmesi

İşbirliğine dayalı güvenli çalışma

Özel korumalı iş hücrelerinin olmaması

Az yer kaplaması, düşük enerji tüketimi

Maliyetlerinin düşük olması

10. COBOT UYGULAMA ALANLARI

Üretimde, dikkat gerektiren ve sık tekrarlanan iş modelleri ya da üretim birimleri, insanlar için bir yandan hata yapma ihtimalini ve iş kazalarını artırırken, diğer yandan da verimi düşürüyor. İnsanla iş birliğine dayalı 'Cobot', bu sorunların üstesinden gelerek iş güvenliğini de en üst seviyelere çıkararak üretimde verimliliği artırıyor.

Bugün dünyanın 50'den fazla ülkesinde kullanılan cobot'lar, yüksek hassasiyete sahip kolları sayesinde oldukça kolay bir programlama ile;

çapak alma-polisaj,

makina besleme,

kalite kontrol,

taşıma,

montaj,

yapıştırma-dağıtma,

cıvatalama,

zımparalama,

alma-bırakma,

paketleme-paletleme,

etiketleme,

enjeksiyon kalıplama,

CNC,

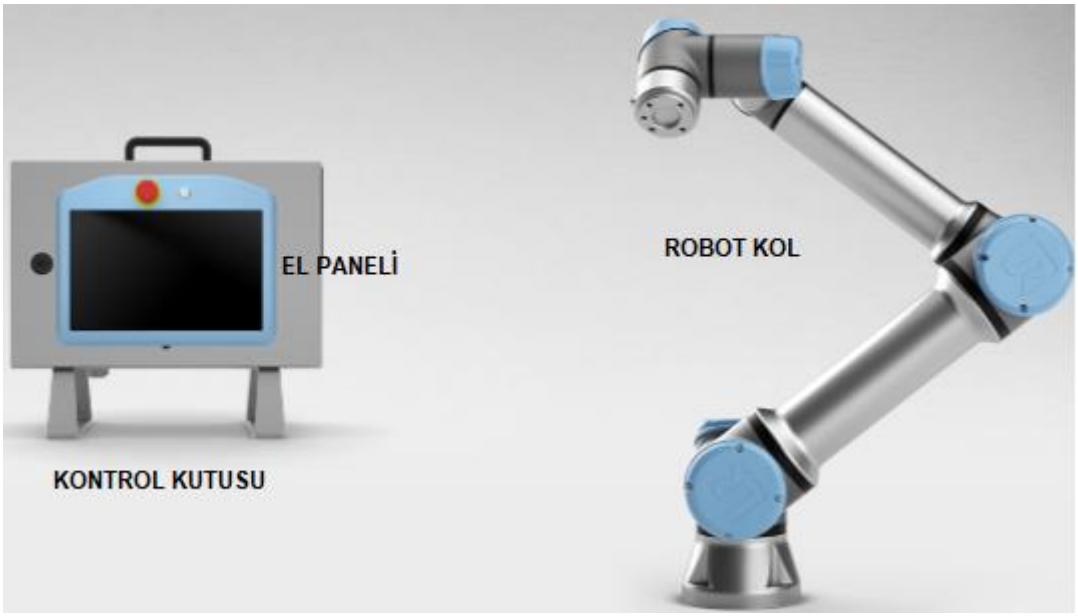
kaynak,

laboratuvar analiz,

test ve örnek alma gibi birçok proseslerde de kullanılabiliyor.

11. COBOT SİSTEM YAPISI

Cobotlar, işbirlikçi çalışmaya yönelik teknolojisiyle, insanı üretim sürecinin merkezine alan yaklaşımıyla ve her türlü üretim tesisine ve her uygulamaya entegre olacak şekilde tasarlanmasıyla ön plana çıkmaktadırlar. Cobotlar Tak Çalıştır ve Üretime Başla konseptiyle hazırlanmıştır. Bu yüzden kurulumları, çalıştırılmaları ve programları son derece kolaydır. Aşağıda standart bir cobot sistem yapısı gösterilmektedir.



Resim 11.1. UR Cobot sistemi

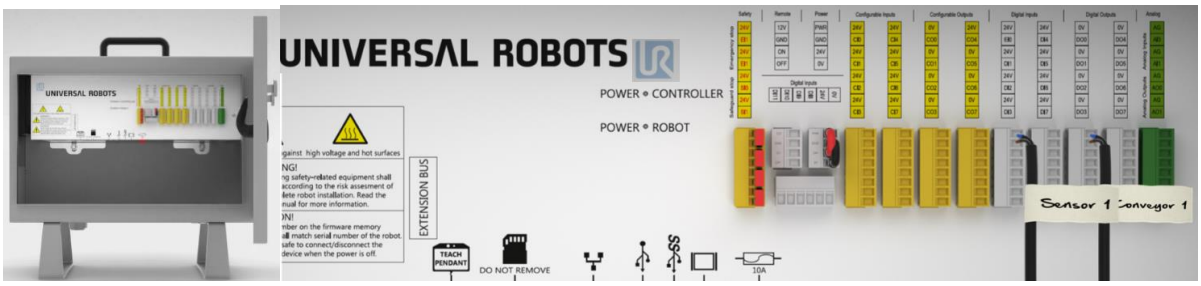
Cobot sistemi içerisinde öncelikle 6 eksenli bir robot kol vardır. 6 eksenin tamamında 360 derece dönüş mümkündür. Tanımlamaları aşağıdaki resimde verilmiştir.

Resim 11.2. UR Cobot eksen tanımlamaları

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Cobot siteminde robot kolun yanısıra bir “Kontrol Kutusu” yer almaktadır. Bu kontrol kutusunda anakart, hafıza kartı ve güvenlik kontrol kartları yer almaktadır. Tüm giriş ve çıkışlar buraya bağlanır. Tüm çevre ekipmanlar ile bağlantıyı bu kontrol kutusu sağlamaktadır. Örneğin El Kumandası Paneli, anahtarlar, sensörler, konveyorlar, kameralar vb gibi.

Kontrol kutusu Linux tabanlı bir mikroişlemcili sistemdir. Her cobot firmasının kendine ait özel yazılımları ile iletişim kurar. İçeriğindeki devre elemanları genellikle tak-çalıştır tarzı bağlantı noktalarına sahiptir.



Resim 11.3. UR Cobot kontrol kutusu

El kumandası paneli cobotun programlanması için gereklidir. Cobotun çalıştırılması ve durdurulması bu panel aracılığıyla yapılmaktadır. Aynı zamanda bu panel aracılığıyla hareket noktaları, hareket şekilleri, bekleme süreleri, giriş ve çıkış işlemleri gibi programlama işlevleri yerine getirilir. Güvenlik uyarıları yine bu el kumandası panelinden takip edilir.

Resim 11.4. UR Cobot el kontrol paneli (Teach Pendant)

Her üreticinin ürettiği cobot sistemi farklıdır. Farklı robot kol, farklı kontrol paneli ve farklı el kumandası paneli içermektedir. Kullanılan yazılımlar birbirinden farklı olabilir. Fakat temel sistem yapısı ve temel programlama mantığı hepsinde aynıdır.

! Bir sonraki bölümde tüm cobot markalarını programlayabileceğiniz simülasyon yazılımlarını anlatacağız. Hazırlanmak için bu yazılımlara bir göz atalım.

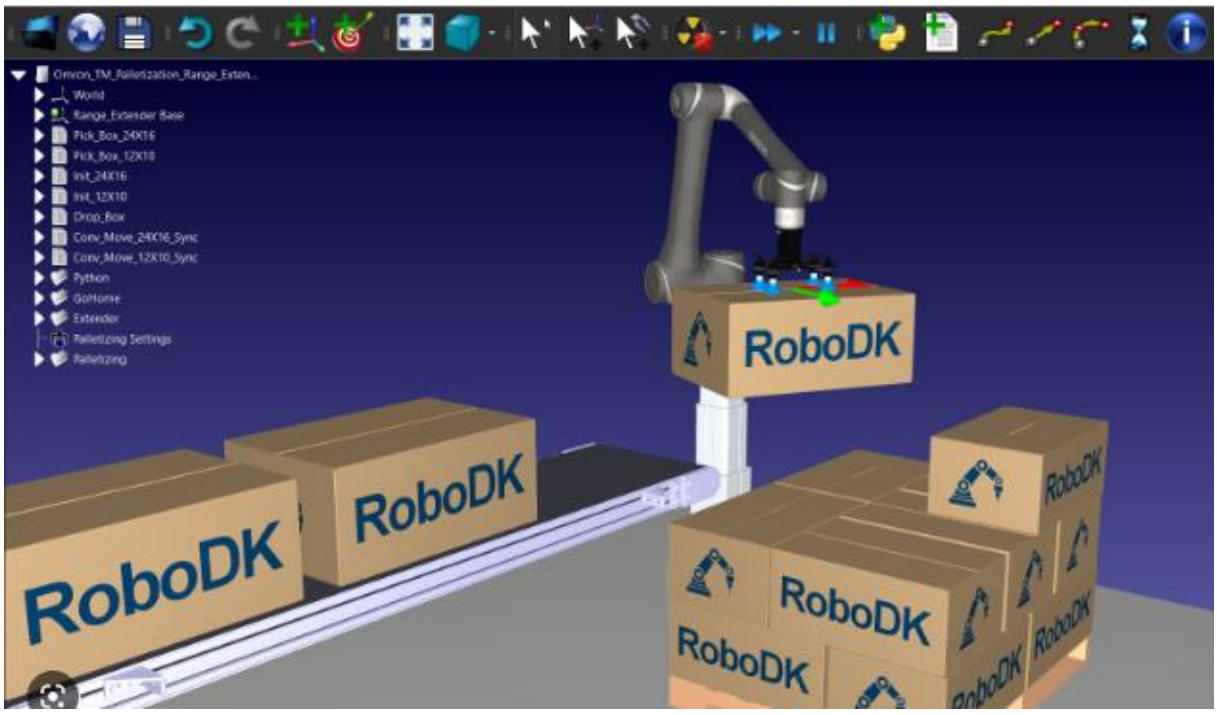
CoppeliaSim from the creators of V-REP
RoboDK
TÜRKİYE ULUSAL AJANSI TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

SİMÜLATÖR – RoboDK

12. PROGRAMIN TANITILMASI

RoboDK, endüstriyel robotlar ve robot programlama için güçlü ve uygun maliyetli bir simülatördür. Ocak 2015'te Albert Nubiola tarafından kurulan RoboDK Robot Geliştirme Kiti, Kanada'nın en prestijli robotik laboratuvarlarından biri olan Kanada'nın Montreal kentindeki ETS Üniversitesi'ndeki CoRo laboratuvarından bir yan şirkettir.

RoboDK ile herhangi bir endüstriyel robotu simüle edebilirsiniz. Doğrudan PC'nizden herhangi bir robot denetleyicisi için robot programları oluşturabilirsiniz. RoboDK'nin sezgisel arayüzü ile programlama becerisi gerekmez. Sadece birkaç tıklama ile herhangi bir robotu çevrimdışı olarak kolayca programlayabilirsiniz. RoboDK, 800'den fazla robot koluyla geniş bir kitaplığa sahiptir.



Resim 12.1. Programın genel görüntüsü

RoboDK'nın temel avantajları:

- RoboDK'nın simülasyon ve çevrimdışı programlama araçlarını kullanmanın avantajı, robotları üretim ortamı dışında programlamanıza izin vermesidir.
- RoboDK ile robotları doğrudan bilgisayarınızdan programlayabilir ve atölye programlamasından kaynaklanan üretim kesintilerini ortadan kaldırabilirsiniz.

RoboDK'nın teknik özellikleri:

- Robot İşleme
  Robot kolunuzu 5 eksenli bir işleme merkezi yada bir 3D yazıcı gibi kullanabilirsiniz. NC kodları(G kodu,APT-CLS) simule edilebilir ve robotik programlara dönüştürülebilir.
  RoboDK, eksen sınırları ve çarpışmalardan kaçınarak robot yolunu otomatik olarak optimize eder.

- Çevrimdışı Programlama
  Endüstriyel robotların simülasyonu ve çevrimdışı programlanması çok kolaydır. Uygulamanızı birkaç dakika içinde simüle etmek için sanal ortamınız oluşturulabilir.
  Herhangi bir robot kontrol cihazı için kolayca robot programları çevrimdışı oluşturulabilir. Artık satıcıya özel programlamayı öğrenmenize gerek yok.

- Robot Kütüphanesi
  30'dan fazla farklı robot üreticisinin endüstriyel robot kolları, dış eksenleri ve araçlarından oluşan geniş bir kütüphaneye sahiptir. RoboDK şimdi 50 robot üreticisinden 600'ün üzerinde robot kolundan, dünya çapında 50'den fazla iş ortağından ve binlerce aktif kullanıcıdan oluşan kapsamlı bir kitaplık sunuyor.
  İşleme, kaynak, kesme, boyama, denetleme, çapak alma ve diğer tüm robotik uygulamalar için herhangi bir robotu kolayca kullanabilirsiniz!

- Robot Doğruluğu
  Hassasiyeti ve üretim sonuçlarını iyileştirmek için robot kolunuz kalibre edilebilir. ISO9283 robot performans testlerini çalıştırıabilirsiniz. Bir ballbar testi ile robotlar sertifikalandırılabilir.

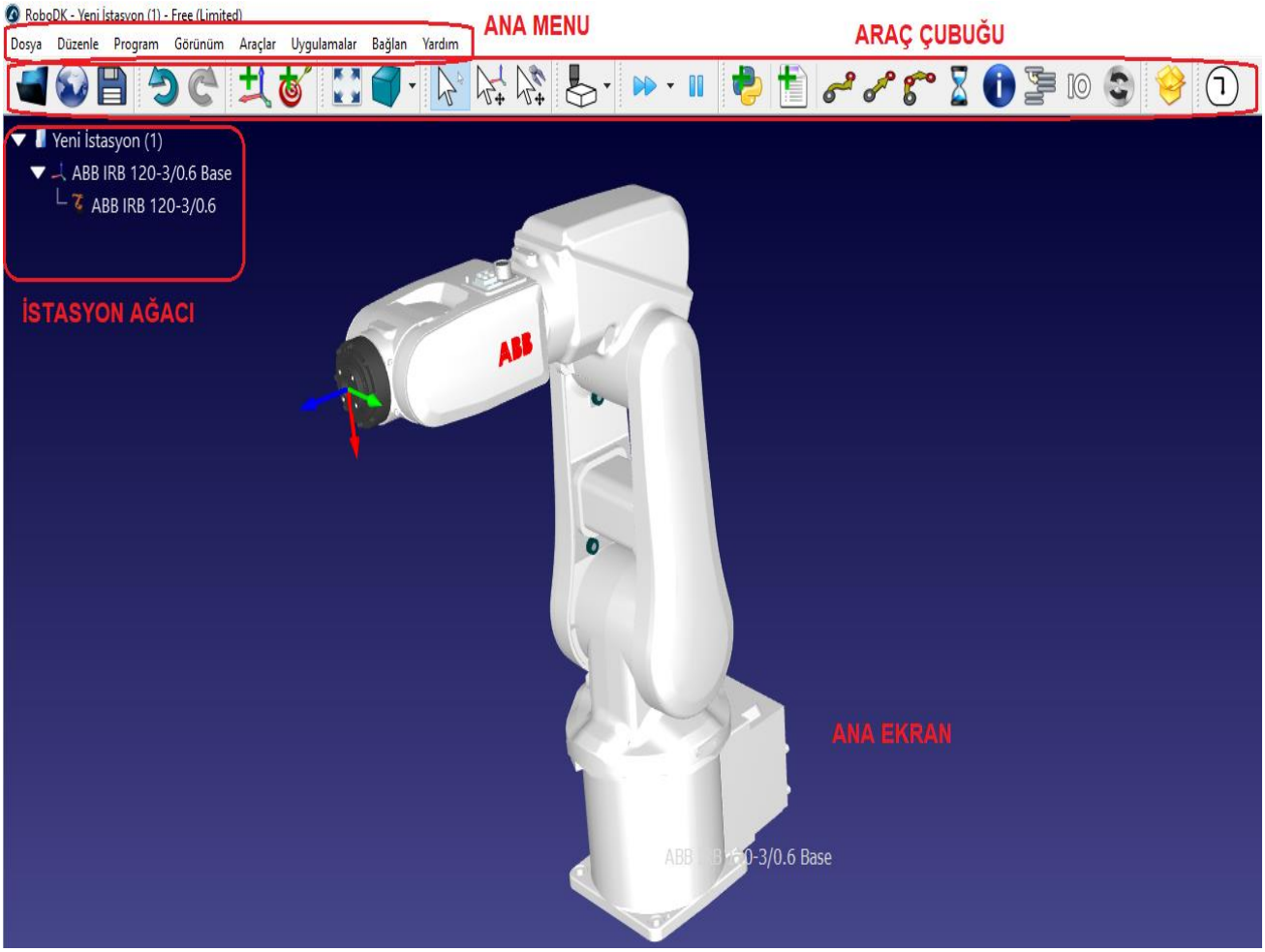
- Programları Robotunuza Yükleyin

  RoboDK Post Prosesörleri aşağıdakiler dahil bir çok robot kontrol cihazını destekler:

  ABB RAPID (mod/prg)
  Fanuc LS (LS/TP)
  KUKA KRC/IIWA (SRC/java)
  Motoman Inform (JBI)
  Universal Robots (URP/script)
  ...ve daha fazlası!

13. TEMEL KULLANIM

RoboDK, https://robodk.com/download web sitesinden yüklendiğinde masaüstünüzde RoboDK kısayolunu oluşacaktır.
RoboDK programını başlatmak için kısayola çift klikleyin.
RoboDK penceresi bir Ana Menü, bir Araç Çubuğu, bir Durum Çubuğu ve Ana Ekran içerir. Ana Ekrandaki İstasyon Ağacı, robotlar, referans çerçeveleri, araçlar, programlar vb. gibi istasyonda bulunan tüm öğeleri içerir.



Resim 13.1. RoboDK çalışma ekranı

Seçme	Pan (Taşıma)	Çevirme	Zoom
Sağ Klik	Orta Buton Basılı	Sağ Buton Basılı	Orta Buton Çevirme

Resim 13.2. Fare hareketleri

14. ANA MENÜLER

14.1. Araç Çubuğu Menüsü

RoboDK Araç Çubuğu, menüde sık kullanılan işlemlere hızlı erişim sağlayan grafik simgeler içerir.

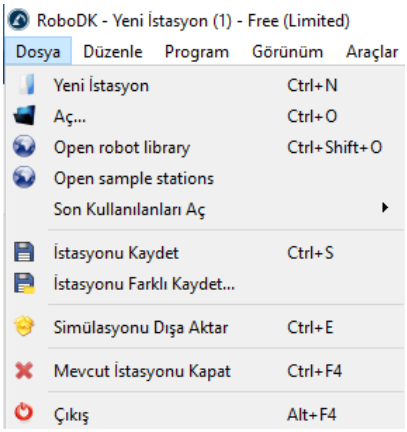
İpucu: Varsayılan araç çubuğunu ayarlamal için **Tools→Toolbar Layout→Set Default Toolbar** seçilmelidir.

Resim 14.1. Yukarıdaki komutlar varsayılan olarak araç çubuğunda mevcuttur.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

14.2. Dosya Menüsü

Dosya menüsünden belgeleri Açmak, Kaydetmek veya Dışa Aktarmak mümkündür.



Resim 14.2. RoboDK Dosya menüsü

Yeni İstasyon ağaçta yeni bir istasyon ekleyecektir. Bir istasyon, bir RDK dosyası olarak yüklenebilir veya kaydedilebilir. RDK dosyası (RDK uzantısı) robotlar ve nesneler hakkındaki tüm bilgileri tutar, böylece içe aktarılan öğrelerin ayrı bir kopyasının saklanması gerekmez.

Aç yeni bir RoboDK dosyası (RDK İstasyonu) yükleyecek veya robot dosyaları için .robot, nesneler için STEP/IGES/STL, araçlar için .tool gibi tanınmış dosya biçimlerini içe aktaracaktır.

Çevrimiçi kitaplığı aç çevrimiçi olarak kullanılabilen kitaplığın bulunduğu yeni bir pencere gösterecektir.

İstasyonu Kaydet RDK dosyasını kaydedecektir. Dosya konumunu belirleme için bu **İstasypnu Farklı Kaydet** seçilmelidir.

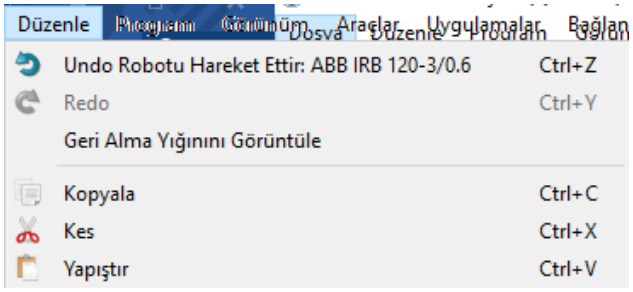
Demo yap çalışma istsyonunu, RoboDK'nın basitleştirilmiş bir sürümüyle birlikte bir EXE dosyası olarak dışa aktaracaktır.

Simülasyonu dışa aktar belirli bir programı veya simülasyonu 3D PDF veya 3D HTML dosyası olarak dışa aktarır.

14.3. Düzenle Menüsü

Geri Al (Ctrl+Z) ve Yinele (Ctrl+Y) eylemlerine Düzenle menüsünden erişilebilir. Geri alma eylemlerinin geçmişi de mevcuttur ve eylemi seçerek değişikliklerin geriye veya ileriye doğru belirli bir duruma geri alınmasına izin verilir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

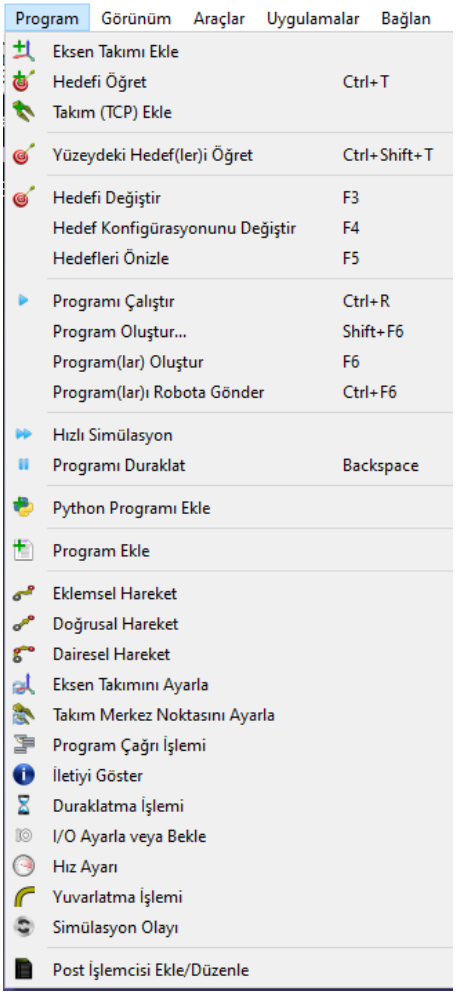


Resim 14.3. RoboDK Düzenle menüsü

Ayrıca Kes (Ctrl+X), Kopyala (Ctrl+C) veya Yapıştır (Ctrl+V) ve Yinele (Ctrl+Y) eylemlerine Düzen menüsünden erişilebilir. Geri alma eylemlerinin geçmişi de mevcuttur ve eylemi seçerek değişikliklerin geriye veya ileriye doğru belirli bir duruma geri alınmasına izin verilir.

14.4. Program Menüsü

Program menüsü, Çevrimdışı Programlama ve program oluşturma ile ilgili tüm bileşenleri içerir. Robotlara yeni programlar, referans çerçeveleri, hedefler veya araçlar eklemek mümkündür. Bu Çevrimdışı Programlama bileşenleri (referans çerçeveleri, araçlar, hedefler vb.), çevrim dışı oluşturulan tüm programlarda görünür.



Resim 14.4. RoboDK Program menüsü

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

 **Eksen Takımı (Referans Ekseni) Ekle** istasyon köküne iliştirilmiş veya başka bir referans ekseni seçilmişse, o referans eksenine eklenmiş yeni bir referans ekseni ekleyecektir.

 **Takım (TCP) Ekle** bir robota yeni bir TCP ekleyecektir. Yeni bir araç (takım) eklemek için geometri gerekmez. Çoklu Araçlar, tek bir araca bağlı aynı geometrinin farklı parçalarına referans verilmesine izin verir.

 **Hedefi Öğret (Ctrl+T)** Aktif robot aracı için Aktif referans eksenine yeni bir hedef ekleyecektir. Aktif referans ekseni ve aktif araç, robot panelinde seçilebilir. Aktif hale getirmek için bir referans eksenine veya bir araca sağ tıklamak da mümkündür.

 **Yüzeydeki Hedef(ler)i Öğret (Ctrl+Shift+T)** kullanıcının kolayca hedef oluşturmak için bir nesnenin noktalarını seçmesine olanak tanır.

 **Program Ekle** RoboDK Grafik Kullanıcı Arayüzü (GUI) kullanılarak oluşturulabilen yeni bir program ekleyecektir. Bu tür robot programlarını oluşturmak veya değiştirmek için herhangi bir programlama deneyimi gerekmez. Robot programı, belirli bir robot için otomatik ve kolay bir şekilde simüle edilebilir ve oluşturulabilir.

Çevrimdışı Programlama belgesinin Program Talimatları bölümü, GUI aracılığıyla mevcut program talimatları hakkında daha fazla bilgi sağlar.

 **Python Programı Ekle** seçeneği, istasyonda RoboDK API'sine bağlanan örnek bir Python programı/makro/komut dosyası/modül içerecektir. RoboDK API'sini kullanan bir Python programı, genel programlama kodundan (Python) robot programlarının oluşturulmasına izin verir. Bir Python programı, istasyona gömülü bir metin dosyası gibidir ve RoboDK'daki belirli görevleri otomatikleştirmek için Python kodu içerir.

 **Post İşlemcisi Ekle/Düzenle**. Post işlemcileri eklemek veya düzenlemek mümkündür. Post işlemciler, belirli bir robot denetleyicisi için programların nasıl oluşturulacağını belirler ve satıcıya özgü sözdizimine uyum sağlamayı sağlar. Post işlemciler, çevrimdışı Programlama Sürecinin son bileşenidir.

14.5. Görünüm Menüsü

3D'de gezinmek için gereken çoğu seçenek Görünüm menüsünde mevcuttur. Bu menüden Döndürmek, Kaydırmak ve Yakınlaştırmak mümkündür (ayrıca 3D görünümü sağ tıklatarak). Ayrıca bir dizüstü bilgisayar dokunmatik yüzeyi (fare yerine) kullanılarak da 3D'de gezinmek gerçekleştirilebilir.

Herhangi bir yönde serbest dönüşe izin vermek için şu seçeneğin işaretini kaldırın: **Görünüm → Dönüşü Hizala.** Aksi takdirde RoboDK, varsayılan olarak XY düzlemini yatay tutmak için istasyon referansını kilitler.

Yıldız tuşunu (*) seçerek robot çalışma alanını göstermek veya gizlemek mümkündür. **F7 tuşu** seçilerek görünen ve görünmeyen öğeler arasında geçiş yapmak da mümkündür.

Referans çerçevelerini **+ veya - tuşuna** birden fazla kez basarak büyütebilir veya küçültebilirsiniz. Eğer çok sayıda öğe görünürse, referans çerçevelerinin boyutunu ayarlamak ve 3D görünümden (örneğin **ALT** tuşunu basılı tutarak) doğru şekilde yakalamak kullanışlı olabilir.

14.6. Araçlar Menüsü

Araçlar menüsünde, 3D görünümün anlık görüntülerini almak, robot izini etkinleştirmek, çarpışma kontrolünü etkinleştirmek veya nokta koordinatlarını ölçmek gibi genel araçlar mevcuttur.

İzi/izlemeyi etkinleştirmek, hareket halindeyken tüm robotların izini gösterecektir.

 **Çarpışmaları Kontrol Et** çarpışma denetimini etkinleştirir veya devre dışı bırakır. Çarpışma kontrolü etkinleştirildiğinde, çarpışma durumunda olan nesneler kırmızı olarak görüntülenecektir. Çarpışma haritası, hangi nesne etkileşimlerinin kontrol edildiğini belirlemeye izin verir.

 **Rengi Değiştir** robotların ve nesnelerin rengini değiştirmeye izin veren küçük bir pencere gösterecektir. Yüzeylerin normal vektörlerini çevirmek de mümkündür.

 **Ölçüm** yerel bir referans çerçevesine veya istasyon referans çerçevesine (mutlak ölçümler) göre noktaların 3B olarak ölçülmesine izin veren bir pencere görüntüler.

Araçlar → Dil'i seçerek RoboDK uygulamasının dilini belirlemek ve tercih edilen dili seçmek mümkündür. RoboDK, seçilen dilde hemen görüntülenecektir.

Araç Çubuğu Düzeni varsayılan araç çubuğunun ayarlanmasına izin verir. Alternatif olarak, daha basit veya daha gelişmiş bir kullanım için bir araç çubuğu belirlemek mümkündür.

 **Seçenekler** ana seçenekler menüsünü açmak için Daha fazla bilgi Seçenekler Menüsü bölümünde mevcuttur.

14.7. Uygulamalar Menüsü

The utilities menu allows performing specific tasks:

 **Takım Merkez Noktasını (TCP) Tanımla** farklı yönler kullanarak bir noktaya ulaşmak için ortak yapılandırmalar gibi gerçek kurulumdan veriler sağlayarak bir robot TCP'nin kalibre edilmesine olanak tanır. Bu prosedür genellikle çoğu robot öğretme pandantifinde bulunur. RoboDK, bir TCP'nin istenildiği kadar çok konfigürasyonla kalibre edilmesini sağlar. Daha fazla konfigürasyon kullanmak, daha doğru bir TCP değeri elde edilmesini sağlar.

 **Eksen Takımını (Referans Ekseni) Tanımla** bir robot temel çerçevesine göre bir referans çerçevesinin tanımlanmasına izin verir. Bu, parçanın gerçek kurulumdan sanal ortama doğru bir şekilde eşleştirilmesini sağlar.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Robotu Kalibre Et robot doğruluğunu iyileştirmek ve robot hata parametrelerini bulmak için bir robot kalibrasyon projesi oluşturmaya olanak tanır. Kalibre edilmiş bir robot, herhangi bir RoboDK Çevrimdışı Programlama projesinde kullanılabilir. Robot kalibrasyonu, robot ölçümlerini almak için ölçüm sistemlerinin kullanılmasını gerektirir. Robot doğruluğu ve tekrarlanabilirliği, kalibrasyondan önce ve/veya sonra ISO9283 ile test edilebilir.

14.8. Bağlan Menüsü

Bir robota bağlanmak ve robot IP'si, FTP kullanıcı adı ve FTP şifresi gibi bağlantı parametrelerini girmek mümkündür. Bir robot bağlantısı kurmak, programların FTP aracılığıyla aktarılmasına veya programların doğrudan PC'den çalıştırılmasına olanak tanır.

14.9. Yardım Menüsü

Yardım (F1) bu belgeleri çevrimiçi olarak açar. Dokümantasyonun PDF versiyonu, her bölümün üst kısmında indirilebilir. F1'e bastığınızda, RoboDK seçili olan öğeyle ilgili Yardım konusunu görüntüler.

14.10. Seçenekler Menüsü

Araçlar **Seçenekler (Shift+O) tuşları** RoboDK ana seçenekler penceresini açar.

Genel Sekme

Ana sekme, temanızı özelleştirme, 3D fare gezinme ayarları, ağacın görünümü, otomatik yedeklemeleri etkinleştirme veya formlarda görüntülenen ondalık basamakları özelleştirme gibi genel seçenekleri içerir.

İstasyon Sekmesi

İstasyon parametreleri, seçenekler menüsünde kullanıcı hesabı ayarları yerine RoboDK projesi (RDK dosyası) ile kaydedilen yegane parametrelerdir.

Görüntüle Sekmesi

Görüntüle sekmesi, 3D ekranın görünümüyle ilgili ayarları özelleştirmenizi sağlar.

Hareket Sekmesi

Hareket sekmesi, robot tekilliklerini ve çarpışmalarını görüntülemek veya önlemek için RoboDK tarafından kullanılan robot simülasyonlarının davranışını ve toleransları özelleştirmenize olanak tanır.

CAD Sekmesi

CAD bölümü (Bilgisayar destekli tasarım), parametrik dosyaların (STEP/STP ve IGES/IGS) içe aktarılması ve bu dosyaların 3D ortamda görüntülenmesi ile ilgili ayarları belirlemenizi sağlar

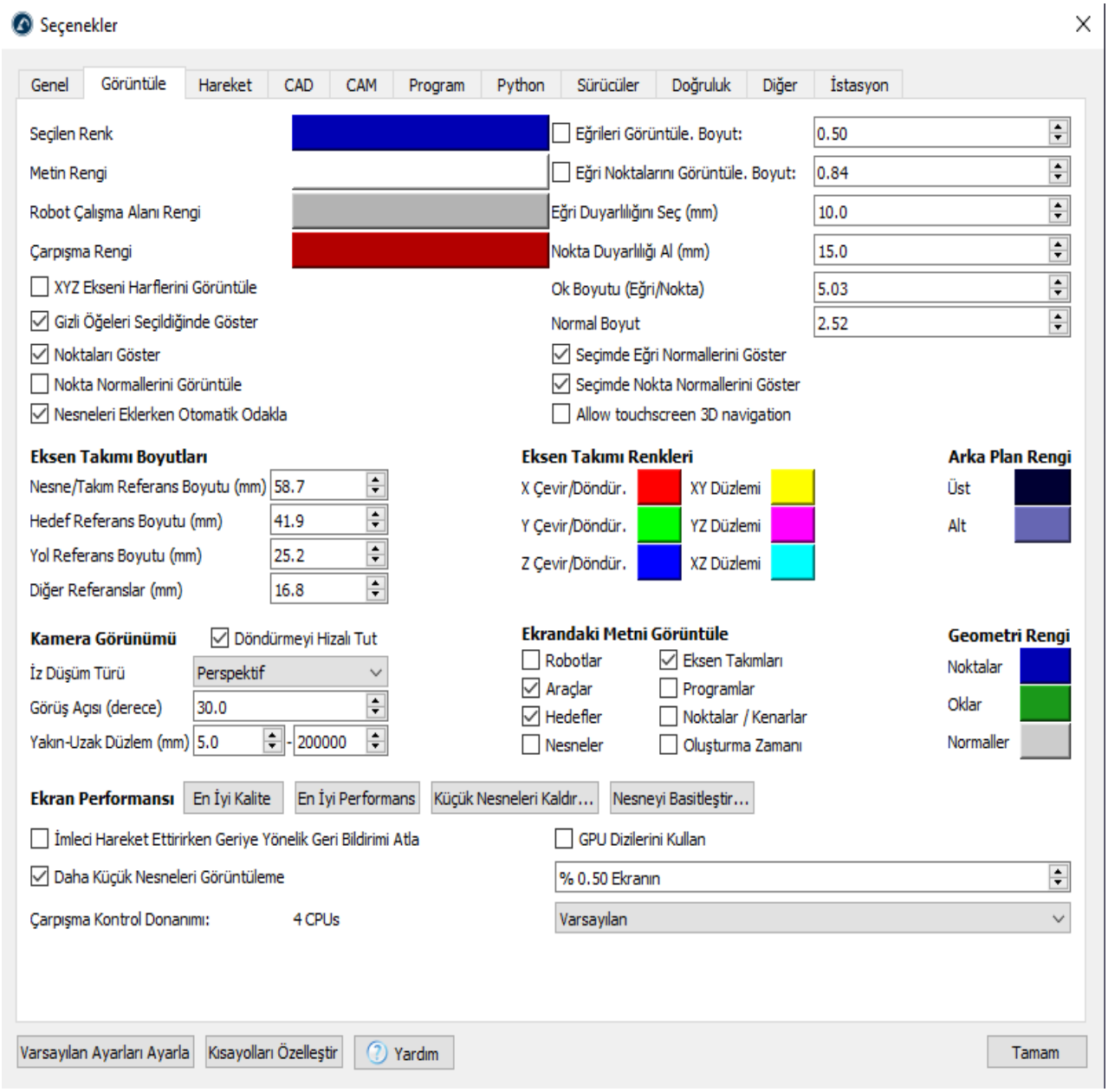
TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

CAM Sekmesi

CAM bölümü (bilgisayar destekli üretim), robot işleme veya 3B yazdırma gibi robot üretim işlemleriyle ilgili tüm ayarları ve CAM yazılımı kullanılarak oluşturulan robot takım yollarının nasıl içe aktarılacağını gösterir.

Program Sekmesi

Program sekmesi, robot programları ve program dosyalarının nasıl oluşturulduğu ile ilgili ayarları özelleştirmenizi sağlar.



Resim 14.5. RoboDK Seçenekler / Görüntüle Sekmesi

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Python Sekmesi

Python sekmesi, RoboDK tarafından kullanılan Python yorumlayıcısının ve Python düzenleyicisinin yolunu ayarlamanıza olanak tanır. Çoğu post işlemci, markaya özel robot programları oluşturmanıza izin vermek için Python'a ihtiyaç duyar. Ayrıca, RoboDK API'sini kullanan bazı örnekler Python'un yüklenmesini gerektirir. RoboDK varsayılan olarak Python 3.7'yi kurar.

15. BAŞLARKEN

Bu başlangıç kılavuzu, robot simülasyonu ve çevrimdışı programlama için RoboDK'da basit bir proje oluşturmanıza yardımcı olacaktır. Bir RoboDK projesinde kullanılan tüm robotlar, nesneler ve araçlar bir RoboDK istasyonu (RDK dosyası) olarak kaydedilir. Bir RoboDK istasyonu, robotlar, araçlar, referans çerçeveleri, hedefler, nesneler ve diğer parametrelerle ilgili tüm ayarları içerir. RoboDK istasyonu tek bir dosyada (RDK uzantısı) saklanır.

15.1. Yeni Proje

Yeni proje başlatmak için **Dosya→Yeni İstasyon (Ctrl+N)** seçilir.

15.2. Robot Seçimi

Projenize PC'nizden veya RoboDK'nın çevrimiçi kitaplığından yeni robotlar eklenebilir. Bunun için

Dosya → Robot Kitaplığını Aç seçilir (**Ctrl+Shift+O**) veya araç çubuğunda ilgili düğmeye basılır. Robot, birkaç saniye içinde otomatik olarak istasyonda görülecektir. Çevrimiçi kitaplık, robot yüklendikten sonra kapatılabilir. Ayrıca RoboDK'nın çevrimiçi kitaplığından Projenize istasyonlar ve eklentilerde ekleyebilirsiniz.

15.3. Bir Araç Oluşturma

RoboDK'da önceden yüklenmiş 3D geometriden yeni robot araçları (TCP'ler) yüklenebilir veya oluşturulabilir.

Bir nesne yüklemek ve onu bir robot aracı olarak ayarlamak için şu adımları izleyin:

Dosya → **Aç** seçilir. Bir nesne olarak eklemek için araç dosyasını seçin (robot temel eksenine eklenecektir). Nesneyi istasyon ağacının içindeki robot öğesine sürükleyip bırakın. Yeni araçlar bir .tool biçimi olarak yüklenebilir veya kaydedilebilir.

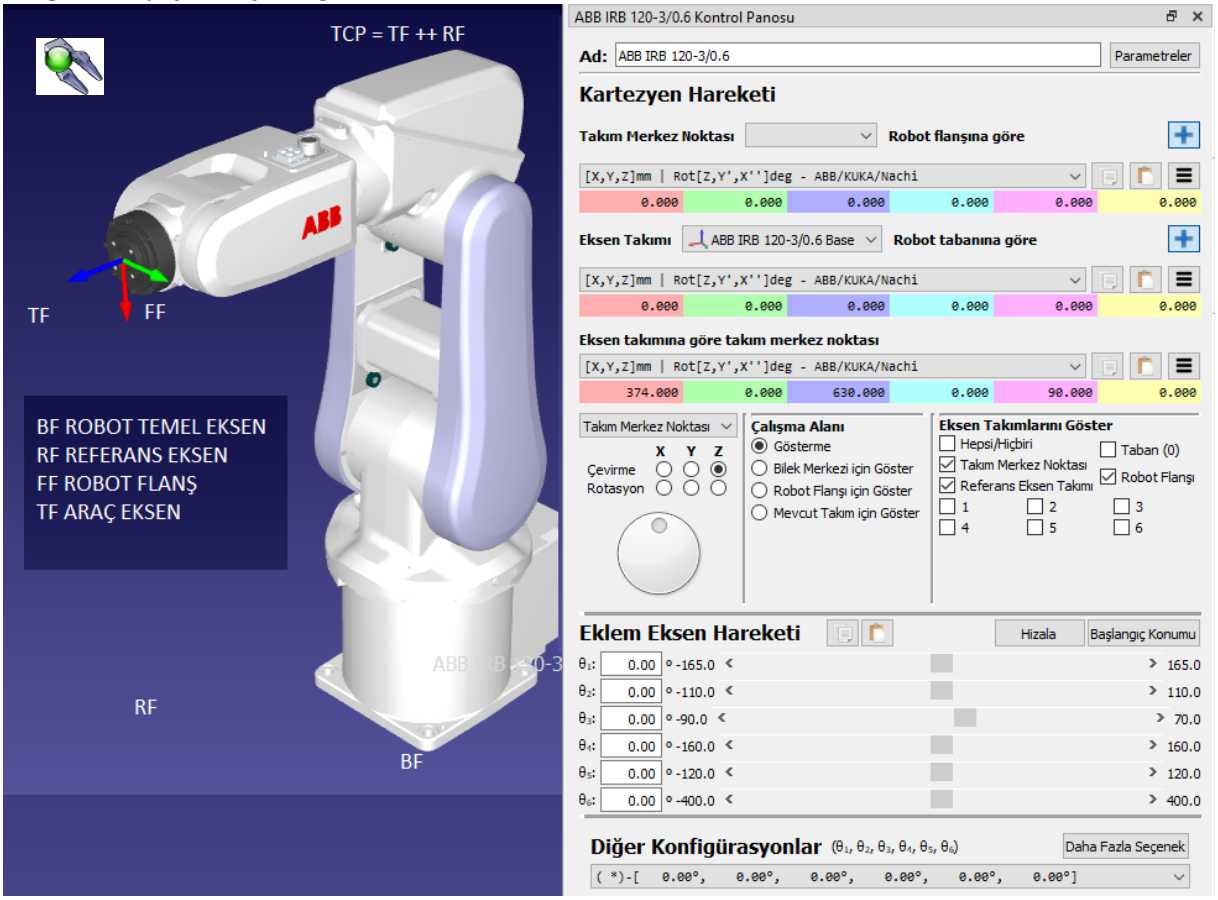
15.4. Robot Paneli

Robot panelini açmak için bir robota çift tıklayın (ağaçta veya 3B görünümde robota çift tıklayabilirsiniz). Eklem Eksen Hareketi elle kumanda bölümünü kullanarak robot eksenlerini elle kumanda etmek ve metin kutularına belirli eklem ekseni değerleri girmek mümkündür. Eklem değerleri ve robot koordinatları, robot kontrolörünüz tarafından görüntülenen değerlerle eşleşmelidir.

Robot eksen sınırlarını değiştirmek için eklem sınırlarını çift tıklayabilirsiniz. Varsayılan olarak RoboDK, robot kontrolörü tarafından kullanılan aynı ortak limitleri (fiziksel donanım limitleri) kullanır.

Kartezyen Hareket bölümü, robot kinematiği ile ilgili tüm bilgileri görüntüler:

Robot Flanşına (FF) göre Takım Ekseni (TF), seçilen Takım Ekseninin Robot Flanşına göre nerede konumlandığını tanımlar. Robot Flanşı her zaman aynıdır, ancak Takım Ekseni, robota monte edilen alete bağlı olarak değişir. Bu ilişki çoğu robot kontrol cihazında UTOOL, ToolData veya sadece Tool olarak da bilinir. Robot Aracı, TCP (Takım Merkez Noktası) olarak da bilinir. Seçilen Takım, "Etkin" araç haline gelir. Etkin takım, yeni hedefler ve programlar oluşturulurken kullanılır. Seçilen takım, simgesinde yeşil bir işaret görüntüler:



Resim 15.1. Rbot paneli ve eksen tanımlamaları

Robot Tabanına (BF) göre Referans Ekseni (RF), Referans ekseninin Robot Ana Eksenine göre nerede konumlandığını tanımlar. Robot Ana Ekseni asla hareket etmez, ancak, herhangi bir nesneyi aynı Robot Ana Eksenine göre konumlandırmak için farklı Referans eksenleri kullanılabilir. Bu ilişki çoğu robot kontrol cihazında UFRAME, WorkObject MFRAME veya Reference olarak da bilinir. Robot panelinde seçilen referans ekseni, "Aktif" referans ekseni olur. Aktif referans ekseni, yeni hedefler ve robot programları için referans olarak kullanılır. Seçilen referans ekseni, simgesinde yeşil bir işaret görüntüler:  .

Referans Eksenine (RF) göre Araç Ekseni (TF), robotun mevcut konumu için aktif referans eksenine göre aktif TCP'nin konumunu gösterir. Robotu hareket ettirmek için bu değeri değiştirin. Eklem eksenleri otomatik olarak yeniden hesaplanır. Bu Kartezyen koordinatlar, robot eksenleriyle birlikte yeni bir hedef oluşturulduğunda (Program→Teach Target) kaydedilir. Hedef, Aktif referans eksenine de iliştirilir.

15.5. İstasyonu Kaydet

Dosya→  **İstasyonu Kaydet**(Ctrl+S) seçilir. Dosyayı helloworld.rdk olarak kaydedin. Pencere başlığı ve İstasyon adı güncellenecektir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 1

ENDÜSTRİYEL ROBOTU ANA EKRANDA HAREKET ETTİRMEK

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, çalışma alanına endüstriyel robot getirebilir.

Öğrenci, çalışma alanındaki endüstriyel robotu kontrol edebilir.

Öğrenci, endüstriyel robotu robot paneli üzerinden çalıştırabilir.

Öğrenci, istasyon çalışmalarını kaydedebilir.

İşlem Basamakları:

1- Masaüstündeki program simgesine çift tıklayarak programı açınız. 

2- **Dosya / Robot Kitaplığını Aç** menüsünden **ABB IRB 120-3/0.6** robotunu seçiniz ve çalışma sayfasına getiriniz.

3- **Dosya / Robot Kitaplığını Aç** menüsünden **Genel Kalem Aracını** seçiniz ve çalışma sayfasına getiriniz.

4- Farenin sağ tuşu, scroll butonu ve basılı tutulan scroll butonunu kullanarak robotunuzu ana ekranda hareket ettiriniz.

5- İstasyon Ağacında robota veya robotun adına çift tıklayarak Robot Panelini açınız.

6- Robotunuzun çalışma alanını görmek için Çalışma Alanı bölgesindeki seçeneklere tıklayınız.

7- Eksenleri Göster alanındaki seçeneklere tıklayarak robotunuzun eksen görünürlüğünü değiştiriniz.

8- Eklem ekseni elle kumanda alanındaki ayar çubuklarını değiştirerek robotunuzun eklemlerini hareket ettiriniz. İlk duruma geri dönmek için Ana Konum düğmesine tekrar basınız.

9- Shift + / tuşları ile ürün isimlerini gizleyiniz.

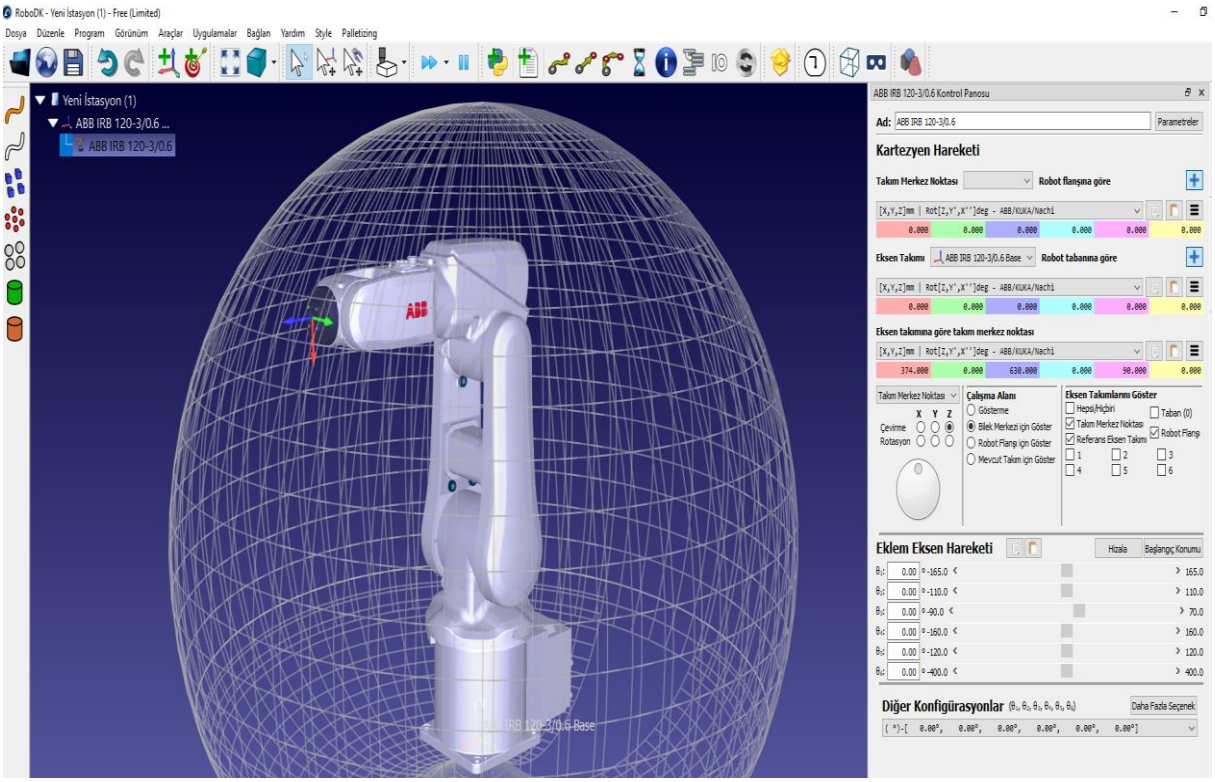
10- Ekrana gelen eksen okları ve  butonuna basarak robotunuzun eklem yerlerini hareket ettiriniz.

11- **Kaydet** butonuna basarak çalışma sayfasını Temrin2 olarak kaydediniz. 

12- Lütfen aynı işlem basamaklarını Universal Robot UR-10 için tekrarlayınız.

13- Cobot ile yaptığınız çalışma sayfasını Temrin 2-1 kaydediniz. .

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 15.2. ABB IRB 120-3/0.6 Endüstriyel Robot çalışma alanı ve Robot Paneli

Çalışma Sorusu:

Lütfen, robotlar ile ilgili payload, work space and reach teknik terimlerinin anlamlarını araştırınız. .

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

15.6. Hedefler Oluşturmak

Robot konumları Hedefler olarak kaydedilir. Kartezyen hedef, aletin konumunu bir koordinat sistemine göre tanımlar. Bir eklem hedefi, robot eklem değerleri verilen robotun konumunu tanımlar.

Not: RoboDK, varsayılan olarak Kartezyen hedefler oluşturur (kırmızı hedefler). Bir hedefi sağ tıklayıp Eklem hedefi olarak ayarlayarak onu Eklem hedefine (yeşil hedeflere) dönüştürebilirsiniz.

Program→  **Hedefi Öğret (Ctrl+T)**, veya araç çubuğundaki ilgili düğme seçilir. Hedef, mevcut robot konumunu (kartezyen ve eklem eksenleri) otomatik olarak hatırlayacaktır.

Çalışma alanına yakın bir ilk yaklaşma konumuna ulaşmak için ortak hedeflerin kullanılması yaygın bir uygulamadır, ardından Kartezyen hedefler, referans ekseni veya takım eksenleri değiştirilirse takım yolunun değişmemesini sağlar.

Bir hedefi sağ tıklayın, ardından kaydedilen pozu ve ortak değerleri görmek için **Seçenekler (F3)** öğesini seçin.

Aşağıdaki renkler varsayılan olarak kullanılır:

X koordinatı → Kırmızı

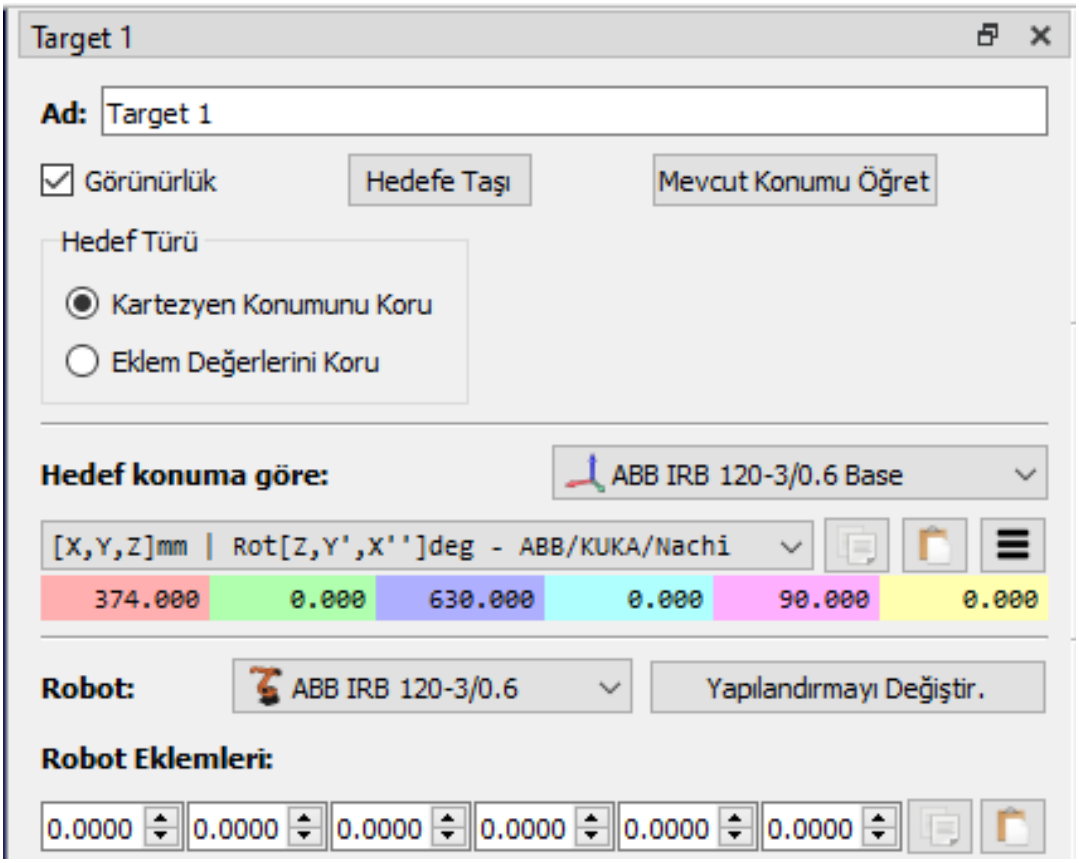
Y koordinatı → Yeşil

Z koordinatı → Mavi

1. Euler dönüşü → Mavi

2. Euler dönüşü → Macenta

3. Euler dönüşü → Sarı



Resim 15.3. Hedef seçenekleri penceresi

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

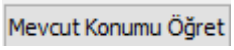
TEMRİN 2

ENDÜSTRİYEL ROBOT İÇİN HEDEFLER OLUŞTURMAK

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, çalışma alanında endüstriyel robot için bazı hedefler oluşturabilir.

Öğrenci, çalışma alanındaki endüstriyel robot için hedeflerin koordinatlarını düzenleyebilir.

Öğrenci,  butonunu veya 

butonunu, endüstriyel robot hedeflerini tasarlamak için kullanabilir.

İşlem Basamakları:

1- Masaüstündeki program simgesine çift tıklayarak programı açınız. 

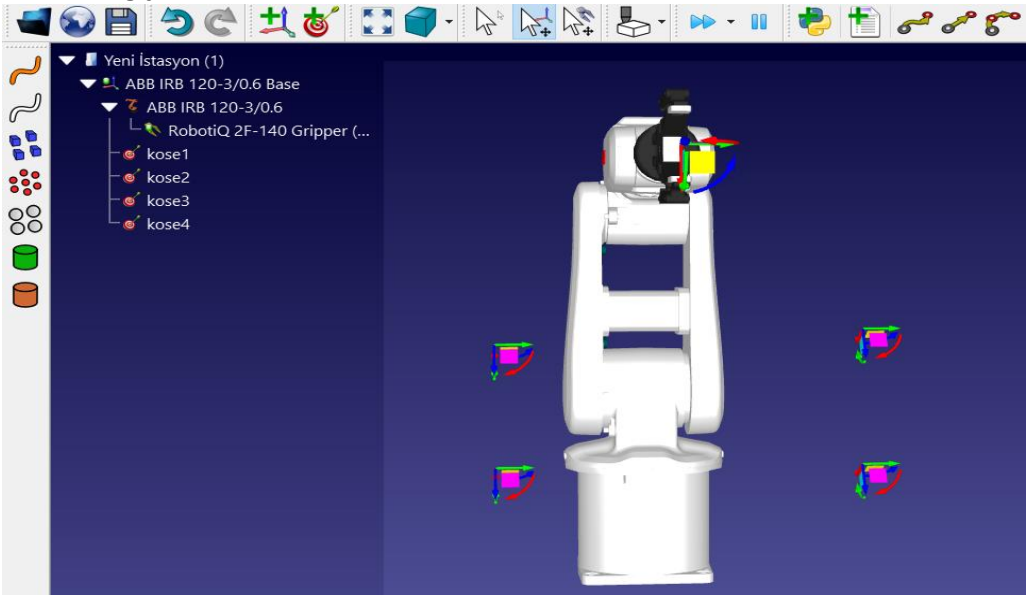
2- **Dosya / Robot Kitaplığını Aç** menüsünden **ABB IRB 120-3/0.6** robotu seçiniz ve çalışma sayfasına getiriniz.

3- **Dosya / Robot Kitaplığınıı Aç** menüsünden **Genel Kalem Aracını** seçin ve çalışma sayfasına getirin.

4-  butonuna basarak robotu hareket ettiriniz. İstediğiniz noktaya ulaştığında  butonuna basarak hedefi kaydediniz.

5- Robotu başka bir yöne hareket ettiriniz. İstasyon Ağacındaki Hedef1 satırına çift tıklayınız ve robotun Hedef1 noktasına geri döndüğünü gözlemleyiniz.

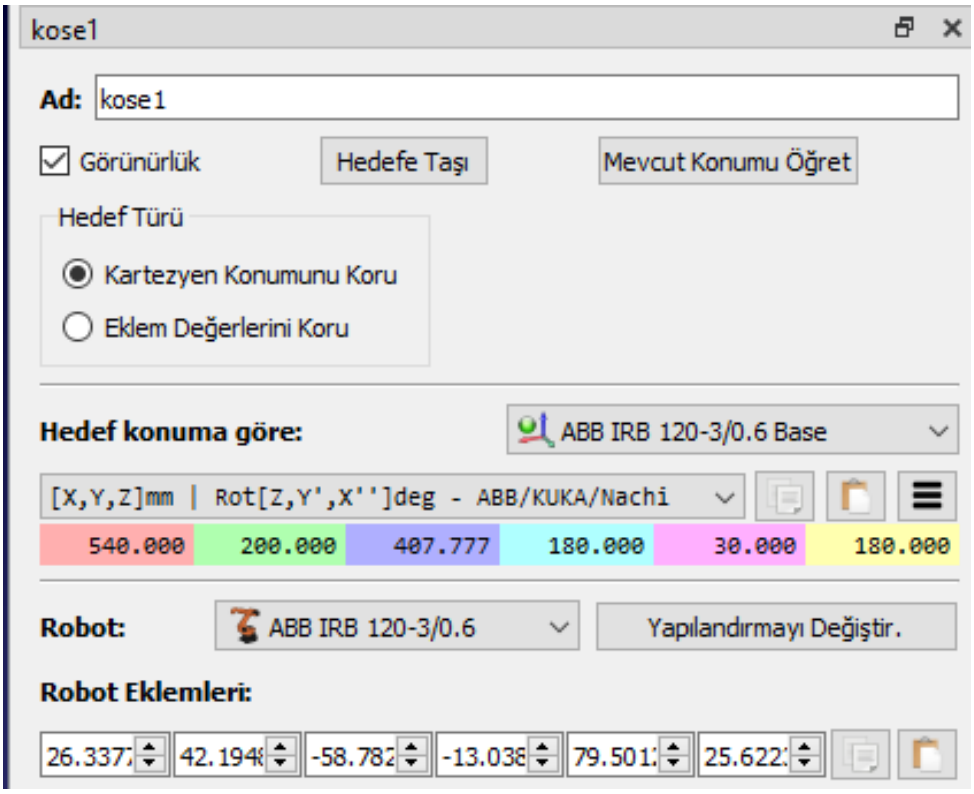
6- Ekranda bir dikdörtgen oluşturmak için 3 hedef koordinat daha oluşturunuz. İstasyon Ağacındaki varış noktalarına gidin ve F2 tuşuna basın. Hedeflerin adlarını Corner1, Corner2, Corner3, Corner4 olarak değiştiriniz.



Resim 15.4. ABB IRB 120-3/0.6 robot için oluşturulan hedefler

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

7- İstasyon Ağacında Corner1'i seçip **F3** tuşuna basarak Seçenekler menüsünü açınız. Menüden açılan Yapılandırmayı Değiştir. butonu ile robotun eklem ayarlarını değiştiriniz.



Resim 15.5. Corner1 için Hedef seçenekleri penceresi

8- Tam bir dikdörtgen oluşturmak için hedef koordinat değerlerini aşağıya giriniz.

Ayarla Kose1 : 540, 300, 390, -180, 30, -180.

Ayarla Kose2 : 540, 200, 390, -180, 30, -180.

Ayarla Kose3 : 540, 200, 240, -180, 30, -180.

Ayarla Kose4 : 540, 300, 240, -180, 30, -180.

9- **Kaydet** butonuna basarak çalışma sayfasını Temrin2 olarak kaydediniz.

10- Lütfen aynı işlem basamaklarını Universal Robot UR-10 için tekrarlayınız.

11- Cobot ile yaptığınız çalışma sayfasını Temrin 2-1 kaydediniz. .

<u>Çalışma Sorusu:</u>

Lütfen ABB robotu için Robot Panelini kullanarak yukarıda verilen hedef koordinatları oluşturunuz.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

16. ROBOT PROGRAMLARI

RoboDK, endüstriyel robot uygulamalarına odaklanan bir simülatördür. Bu, robot programlarının belirli bir robot kolu ve robot kontrolörü için çevrimdışı olarak oluşturulabileceği, simüle edilebileceği ve üretilebileceği anlamına gelir. Başka bir deyişle RoboDK, Çevrimdışı Programlama yazılımıdır.

Kapsamlı kitaplığında endüstriyel robotlar mevcuttur. Endüstriyel robotlar, RoboDK'da, eksen limitleri, hareket hissi ve eksen bağlantısı dahil olmak üzere satıcıya özel kontrolörler kullanılarak davrandıkları şekilde modellenir. RoboDK API, bu programları tamamlamak veya tamamen bir robot programı oluşturmak için kullanılabilir.

16.1. Çevrimdışı Programlama

Çevrimdışı Programlama (veya Çevrim-Dışı Programlama), robotların üretim ortamı dışında programlanması anlamına gelir. Çevrimdışı Programlama, atölye programlamasından (Teach Pendant kullanarak programlama) kaynaklanan üretim kesintilerini ortadan kaldırır.

Simülasyon ve Çevrimdışı Programlama, üretim hücresini kurmadan önce bir robot hücresinin birden fazla senaryosunun çalışılmasına olanak tanır. Bir iş hücresinin tasarımında yaygın olarak yapılan hatalar zamanında tahmin edilebilir.

Çevrimdışı Programlama, robot sistemleri için yatırım getirisini en üst düzeye çıkarmanın en iyi yoludur ve uygun simülasyon araçları gerektirir. Yeni programların benimsenmesi için gereken süre haftalardan tek bir güne indirilebilir ve bu da kısa vadeli üretimin robotlaşmasına olanak tanır.

16.2. Bir Program Oluşturma

Bir programa bir dizi talimat eklenerek bir simülasyon gerçekleştirilebilir. Her talimat, belirli bir denetleyici için belirli bir kodu temsil eder, ancak RoboDK, kod yazmaya gerek kalmadan genel bir şekilde robot programlarını kolayca oluşturmak için bir Grafik Kullanıcı Arayüzü (GUI) sunar.

Bir robot kontrolörüne özgü kod, program oluşturulduğunda otomatik olarak üretilecektir. RoboDK Grafik Kullanıcı Arayüzünü kullanarak yeni bir boş program oluşturmak için:

1. **Program→ Program Ekle** seçiniz veya araç çubunğunda  butonuna basınız.

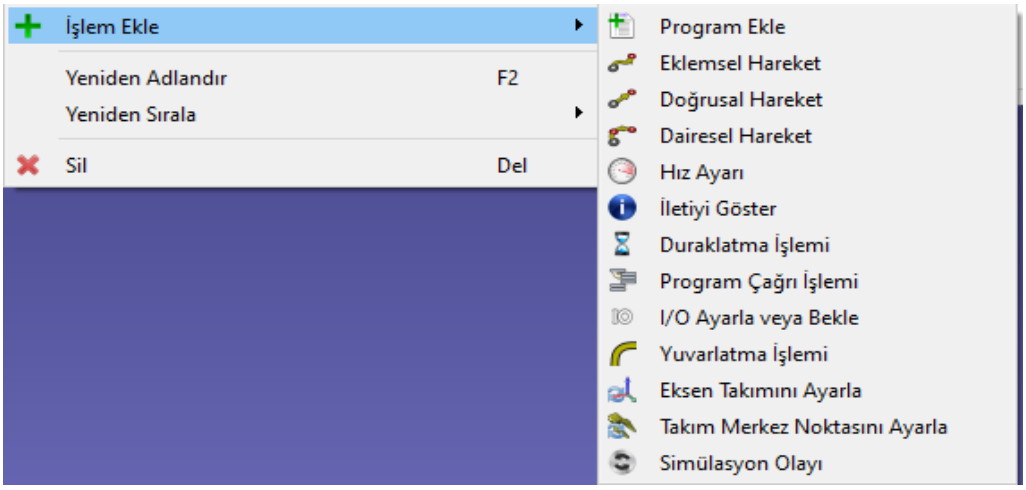
2. Programa isim vermek için **Araçlar→Öğeyi Yeniden Adlandır… (F2)** seçiniz.

Bu hareket boş bir program oluşturur ve yeni komutlar eklenmesine izin verir.

16.3. Program Komutları

Bir programa sağ tıklayarak yeni komutlar eklenebilir veya Program menüsünden bir komut seçilebilir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 16.1. Program komutları

16.3.1. Eklemsel Hareket

Yeni eklemsel hareket komutu eklemek için **Program**→ **Eklemsel Hareket** seçilir. Alternatif olarak, araç çubuğundan ilgili buton da seçilebilir. Komut eklenmeden önce bir hedef seçilmediği takdirde, hareket komutu yeni bir hedef oluşturacak ve bunlar bağlantılı olacaktır. Hedef hareket ettirilirse hareket de değiştirilir.

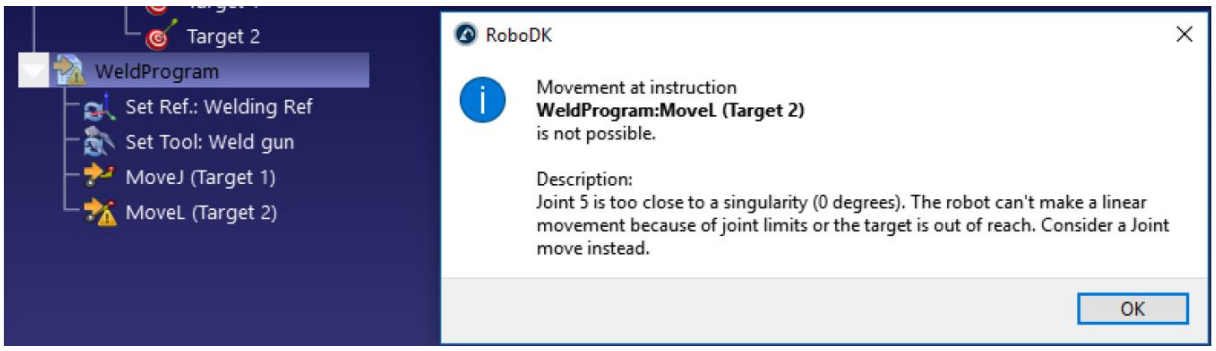
Bu, programa eklenen ilk komut ise, hareket komutundan önce iki komut daha eklenecektir: Referans Ekseni seçimi ve Takım Ekseni seçimi. Bu, program hareket komutuna ulaştığında robotun bu yeni hedefi oluşturmak için kullanılan aynı referansı ve takım eksenlerini kullanmasını sağlayacaktır.

16.3.2. Doğrusal Hareket

Yeni doğrusal hareket komutu eklemek için **Program**→ **Doğrusal Hareket** seçilir. Alternatif olarak, araç çubuğundan ilgili buton da seçilebilir.

Dikkat: Her programın ilk hareketinin Eklemsel hedef kullanılarak Eklemsel Hareket olarak oluşturulması önerilir. Bu, ilk hareketten itibaren istenen konfigürasyonu düzgün bir şekilde oluşturacak ve gerçek robotun simüle edildiği gibi hareket ettiğinden emin olunacaktır.

Eklemsel Hareketlerin aksine, Doğrusal Hareketler robot tekilliklerine ve eksen sınırlarına duyarlıdır. Örneğin 6 eksenli robotlar doğrusal bir hareketin ardından bir tekilliği geçemezler. Aşağıdaki görüntü, Eklem 5'in bir tekilliğe (0 derece) çok yakın olduğunu söyleyen bir örneği göstermektedir. […] Bunun yerine Eklemsel bir hamle düşünülmelidir.



Resim 16.2. Doğrusal hareket uyarı bilgilendirmesi

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

16.3.3. Eksen Takımını Ayarla

Özel referans ekseni kullanmak için **Program→** **Eksen Takımını Ayarla** komutu seçilmelidir. Bu takip eden hareket komutları için kontrolörde verilen referans eksenini güncelleyecek ve simülasyon hedefleri için RoboDK'da robotun aktif referans eksenini değiştirecektir. Bunun anlamı, hareket komutları özel hedeflere göre son referans ekseni ayarlamaları çerçevesinde oluşacaktır.

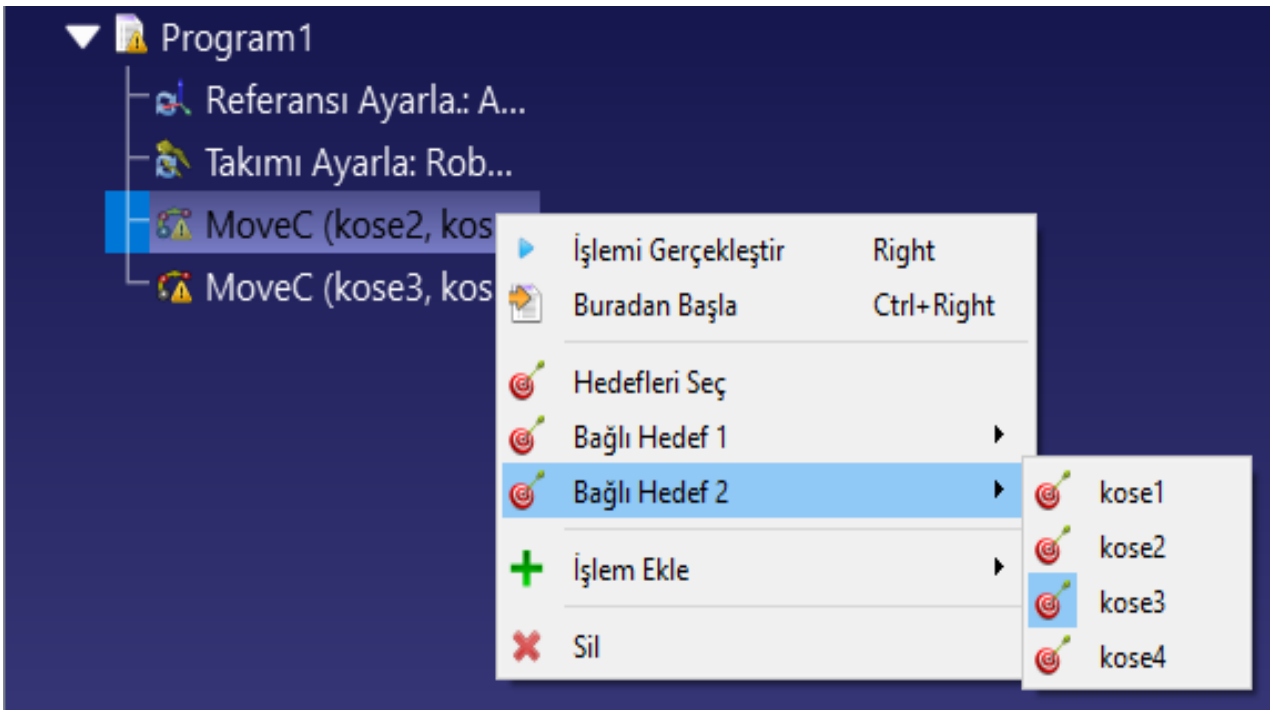
16.3.4. Takım Merkez Noktasını Ayarla

Özel takım ekseni (TCP) kullanmak için **Program→** **Takım Merkez Noktasını Ayarla** komutu seçilmelidir. Bu takip eden hareket komutları için takım eksenini güncelleyecek ve simülasyon hedefleri için RoboDK'da robotun aktif takım eksenini değiştirecektir. Bunun anlamı, hareket komutları özel hedeflere göre son takım ekseni ayarlamaları çerçevesinde oluşacaktır.

16.3.5. Dairesel Hareket

Özel dairesel hareketi kullanmak için **Program→** **Dairesel Hareket Komutu** seçilmelidir. Alternatif olarak araç çubuğundaki butonda kullanılabilir.

Komut eklenmeden önce iki hedef seçilmediği sürece, hareket komutu yeni hedef oluşturmayacaktır. Dairesel hareket komutundan iki taneden fazla hedefi ayrı ayrı eklemek ve aşağıdaki görselde görüldüğü gibi bağlamak gerekiyor.



Resim 16.3. Dairesel hareket komutu

Dairesel yol, robotun bulunduğu noktadan oluşturulan, ilk dairesel noktadan (Target Linked 1) geçen ve bitiş noktasında (Target Linked 2) biten bir yaydır. Tek bir döngüsel komutla tam bir döngü gerçekleştirmek mümkün değildir. Tam bir daire, iki ayrı dairesel harekete bölünmelidir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

16.3.6. Hız Ayarı

Hızı ve/veya ivmeyi değiştiren yeni bir komut eklemek için **Program**→  **Hız Ayarı** komutu seçilir. Eklem uzayında ve kartezyen uzayda değişik hız ve ivme mümkündür. Programda belirli bir hız ve ivme için ilgili şartlar etkinleştirilmelidir. Robotun hızı, bu komutun yürütüldüğü andan itibaren geçerlidir.

Robot hızı, robot parametreleri menüsünden de değiştirilebilir: Robota çift tıklayın, ardından parametreleri seçin.

16.3.7. İletiyi Göster

Teach Pendant'ta bir mesaj görünteleyecek yeni bir komut eklemek için **Program**→  **İletiyi Göster** komutu seçilir.

16.3.8. Duraklat

Program yürütülmesini bir süre veya tamamen durduracak yeni bir komut eklemek için **Program**→  **Duraklat** komutu seçilir.

Operatör programı devam ettirmek isteyene kadar programı duraklatmak için duraklatma gecikmesi değerini -1 olarak ayarlayın. Bu durumda, komut otomatik olarak Durdur olacaktır. Simülasyonda, varsayılan simülasyon oranı 5 için, 5 saniyelik bir duraklamanın simüle edilmesi 1 saniye sürecektir.

16.3.9. Program Çağrı İşlemi

Aktif programadan alt program çağrısı eklemek için **Program**→  **Program Çağrı İşlemi** komutu seçilir.

Varsayılan olarak bu, belirli bir programa yapılan engelleme çağrısıdır. Ancak, bu komutun bulunduğu yere özel kodu girmek için Kod Gir'e geçmek mümkündür. Bu, belirli bir uygulama ve belirli bir denetleyici için yararlı olabilir.

Metin alanını otomatik olarak doldurmak için Program Seç'i seçin. Aksi takdirde, bir metin eşleşmesi de çalışabilir. Komutta kullanılan alt program ile isim eşleşmesi varsa, bu alt program RoboDK'da simüle edilecektir.

Arka arkaya birden çok program çağrısı talimatını otomatik olarak ayarlamak için birden çok satır girin.

16.3.10. IO Ayarla/Bekle

Dijital Çıkışların (DO) durumunu değiştirmek için **Program**→  **IO Ayarla/Bekle** komutu kullanılır. Varsayılan olarak, bu komut **Dijital Çıkışı Ayarla** olarak ayarlanmıştır. Bu komut aynı zamanda belirli bir Dijital Girişin (DI) belirli bir duruma geçmesini beklemeye izin verir.

IO bir değişkense, bir sayı veya metin değeri olabilir. IO bir sabitse, bir sayı (Yanlış için 0 ve Doğru için 1) veya metin değeri olabilir.

16.3.11. Yuvarlatma İşlemi

Yuvarlatma doğruluğunda değişiklik yapmak için **Program**→ **Yuvarlatma İşlemi** komutu seçilir. Yuvarlatma doğruluğu ardışık hareketler arasındaki kenarları yumuşatmak için kullanılır. Bu değişiklik bir program içinde yürütüldüğü andan itibaren etkili olur. Bu nedenle bu değerin bir programın başında ayarlanması normaldir.

Yuvarlatma komutu olmadan, robot her hareketin sonunda 0 hızına ulaşacaktır (bir sonraki hareket bir önceki hareketle teğet değilse). Bu, her hareket için en iyi doğruluğu sağlamak için yüksek hızlanmalara ve hızlı hız değişikliklerine neden olacaktır.

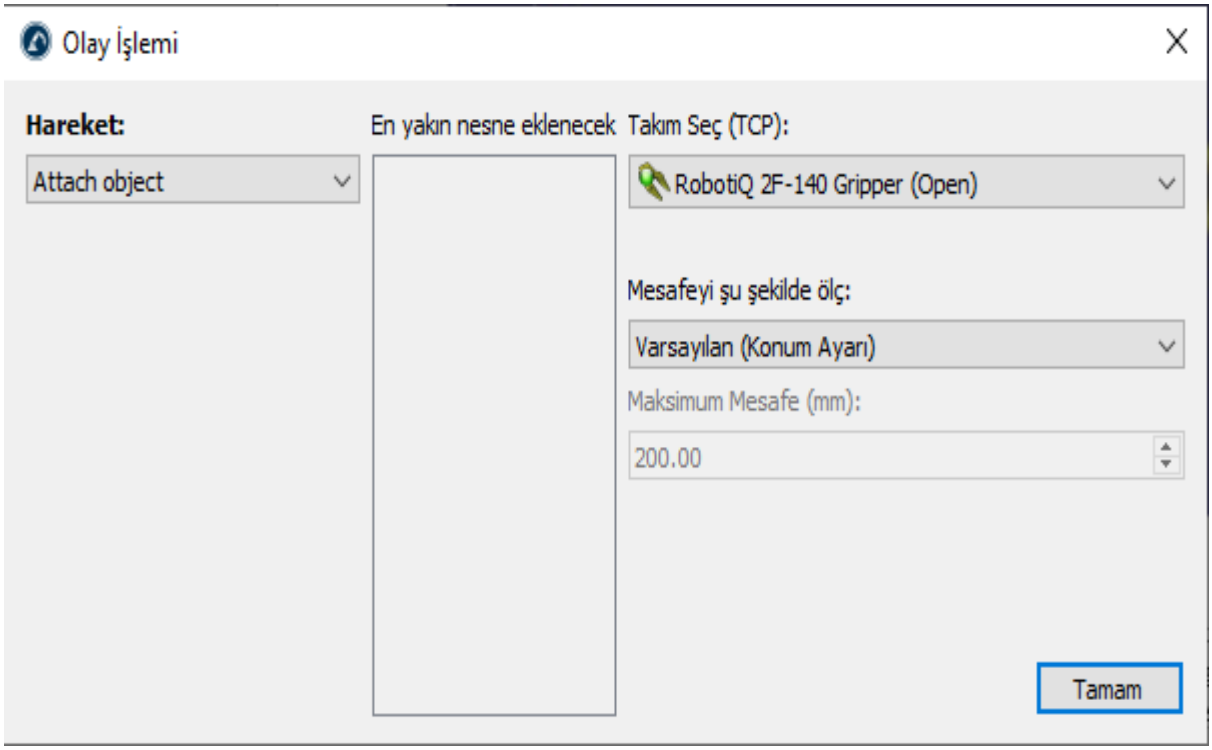
Yüksek bir yuvarlatma değeri, yol kenarlarında doğruluk kaybı karşılığında robot yolu boyunca sabit bir hız sağlayacaktır. Her uygulamaya bağlı olarak, doğruluk ve yumuşak hız arasında iyi bir uzlaşma bulmak yaygındır.

16.3.12. Simülasyon Olayı

Belirli bir simülasyon olayını tetiklemek için Komutu seçilir. Simülasyon olaylarının oluşturulan kod üzerinde hiçbir etkisi yoktur ve yalnızca simülasyon amacıyla belirli bir olayı tetiklemek için kullanılır.

Grafik kullanıcı arabirimini kullanan simülasyon etkinlikleri şunları yapmanıza olanak tanır:

- nesneleri robot araçlarına iliştirme veya ayırma
- nesneleri veya araçları gösterme veya gizleme
- nesnelerin ve referans eksenlerinin konumunu değiştirme



Resim 16.4. Simülasyon etkinlik komut penceresi

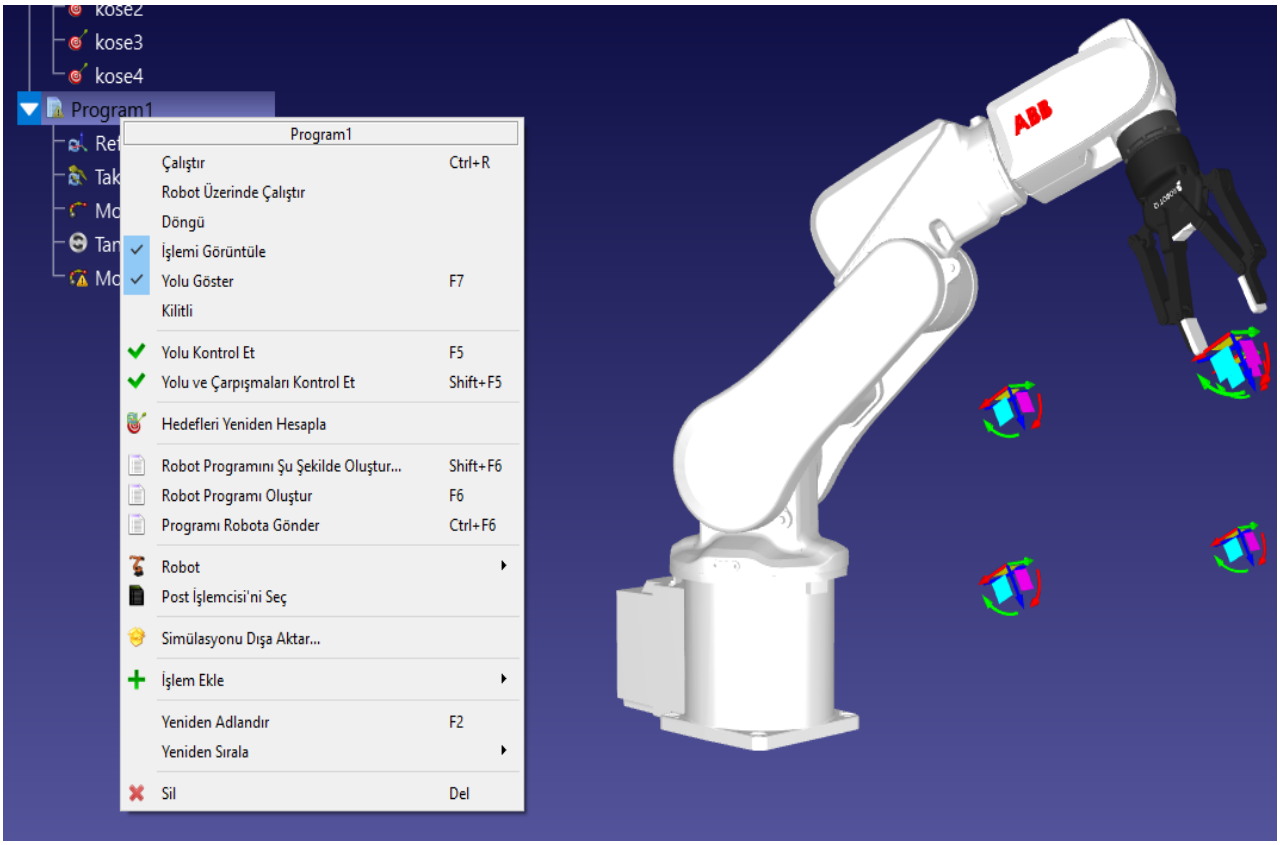
17. SIMÜLASYON PROGRAMI

Program simülasyonunu başlatmak için program ikonun çift klikleyin. 

Veya:

1. Program üzerinde sağ tıklayın,
2. **Çalıştır** seçin.

Program ikonu çift tıklanırsa altta bir simülasyon çubuğu görünecektir. Simülasyon çubuğu kullanarak simülasyon hareketini ileri geri kaydırmak mümkün olacaktır.



Resim 17.1 RoboDK içerisinde program simülasyonu

Simülasyonu hızlandırmak için **Program→**  **Hızlı Simülasyon** seçilir. (veya **Space** tuşuna basılır). Bu komut araç çubuğunda da yer almaktadır.

RoboDK varsayılan olarak gerçek zamandan 5 kat daha hızlı simülasyon yapar. Bu, bir programın yürütülmesi 30 saniye sürüryorsa, 30/5=6 saniyede simüle edileceği anlamına gelir. Simülasyonun hızlandırılması **Araçlar→Seçenekler→Hareket** menüsünden ayarlanabilir.

Simülasyonu duraklatmak için **Program→**  **Duraklat** seçilir. (veya **backspace** tuşuna basılır). Simülasyonu durdurmak için **Esc** tuşuna basılır veya program ikonu tekrar çift tıklanır. Adım adım yürütmek için her komuta ayrı ayrı çift tıklanır. Bir hareket komutunu sağ tıklanır ve buradan **Başlat** seçilirse, program yürütmeye o komuttan devam eder.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 3

ROBOT PROGRAMLAMA

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, kaydedilmiş istasyon çalışmalarını açabilir.

Öğrenci, robot kolunu hedefler arasında hareket ettirebilir.

Öğrenci, hedefler arasındaki hareket türlerini öğrenebilir.

Öğrenci, çalışma alanının önemini kavrayabilir.

İşlem Basamakları:

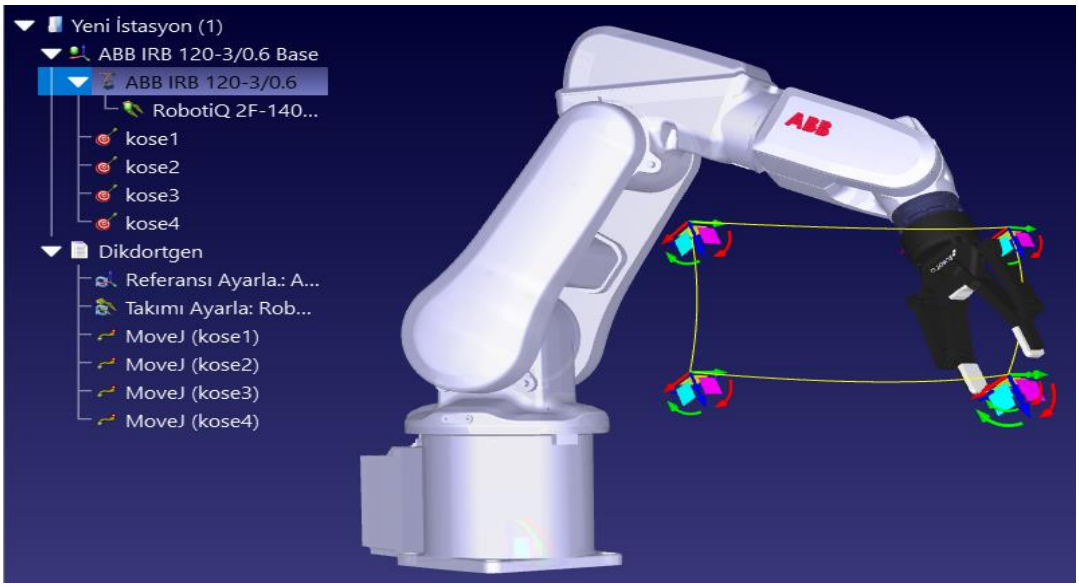
1- Masaüstündeki program simgesine çift tıklayarak programı açınız. 

2- Daha önce oluşturulan çalışma sayfasını açmak için  butonuna basınız ve dosyayı seçiniz.

3-  butonuna basarak bir program bloğu ekleyiniz. Eklediğiniz program İstasyon Ağacı'nda Prog1 olarak görülecektir. Prog1 satırını seçiniz**. F2** tuşuna basarak programın adını Rectangle yapınız.

4-  hareket butonlarını kullanarak robotumuzu hareketlendireceğiz.  Eklemsel Hareket butonuna basınız. Target5, Set Ref, Set Tool and MoveJ satırları İstasyon Ağacı'nda görülecektir.

5- Target5 istenmiyor. Çünkü hareket listemizde 4 hedef var. Bu yüzden Target5'i siliniz. Daha sonra: CTRL tuşuna basılı tutarak kose1 hedefini ve Dikdortgen programını seçiniz. Eklemsel hareket butonuna basınız. Ve MoveJ (Kose11) komutu ağaca gelecektir. Tüm köşeler için aynı işlemi tekrarlayınız.

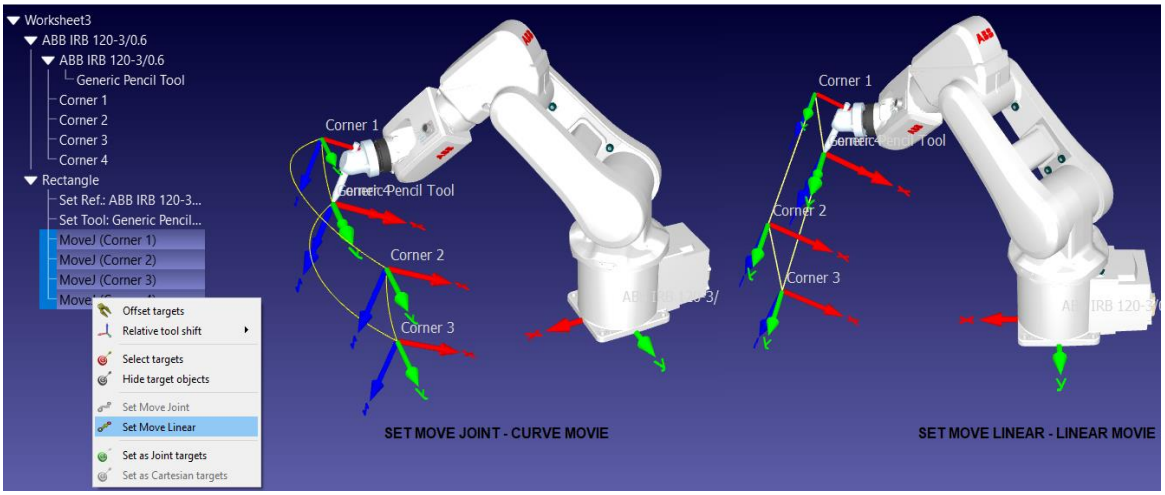


Resim 17.2. MoveJ komutları ile Dikdortgen programı

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

6- Dikdortgen programına çift tıkladığımızda robotumuzun ekranda bir dikdörtgen yaptığını göreceksiniz. Robot kolunun hedefler arasındaki hareketi kavislidir.

7- Eklediğimiz dört MoveJ komutunu seçerek, sağ tuşa basnız. Açılan menüden Doğrusal Hareket Ayarla komutunu seçiniz. Böylece komutların tamamı MoveL oldu. Böylece hedefler arasındaki hareket doğrusallaştı.



Resim 17.3. MoveJ ve MoveL komutlarının karşılaştırılması

8- İstasyon Ağacında sırasıyla Köşe Hedefleri, Kalem Aracı ve ABB robotu seçiniz ve sağ tuşa basınız. Görünen menüden **Görünürlük** onayını kaldıranız. Böylece ekranda çok temiz bir dikdörtgen çizen bir robot belirecektir

9- Dikdörtgen üzerine sağ tıklandığında açılan menüden **Döngü** komutunu seçiniz. Böylece işlem sonsuz bir döngüye girecektir.

10-  butonuna basarak çalışma sayfasını worksheet 3 olarak kaydediniz.

11- Lütfen Universal Robot UR-10 için aynı işlem basamaklarını tekrarlayınız.

12- Cobot ile yaptığınız bu istasyonu worksheet 3-1 olarak kaydediniz.

Çalışma Sorusu:

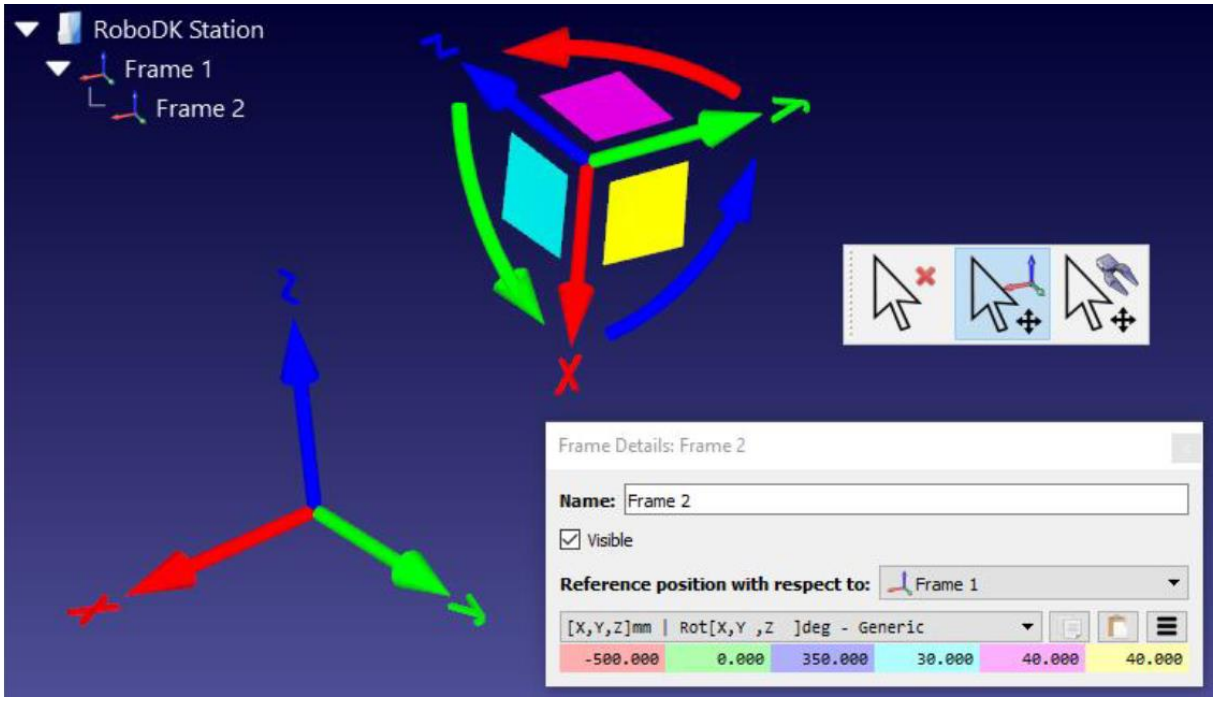
Lütfen ABB robot için AB kelimesinin hedef koordinatlarını oluşturun.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

18. REFERANS EKSENLERİ

Bir Referans Ekseni, bir öğenin konumunu, belirli bir konumu ve yönü verilen başka bir öğeye göre tanımlar. Bu öğe bir nesne, bir robot veya başka bir referans ekseni olabilir. Tüm Çevrimdışı Programlama uygulamaları, simülasyonu buna göre güncellemek için robota göre nesneyi bulmak üzere bir referans ekseni tanımlamayı gerektirir. Belirli bir ilişkiyi tanımlamak için İstasyon Ağacı içindeki herhangi bir referans eksenini veya nesneyi sürükleyip bırakmak yeterlidir.

Referans eksenlerini birbirine göre hareket ettirmek için **Alt** tuşunu basılı tutun. Alternatif olarak, araç çubuğunda ilgili düğmeyi seçin:  . Ardından, referansı ekranda fare ile sürükleyin. Referans taşınırken, ilgili koordinat değerleri güncellenecektir.



Resim 18.1 Referans Eksenler ve RoboDK programında eksen detayları

Bir referans ekseninin başka bir referans eksenine göre ilişkisi aynı zamanda **poz** (konum ve yön) olarak da bilinir. Bir **poz**, oryantasyon için XYZ konumu ve Euler açıları, XYZ konumu ve Kuaterniyon değerleri veya bir 4x4 matrisi ile temsil edilebilir.

Aşağıdaki renkler varsayılan olarak kullanılır:

X koordinatı → Kırmızı

Y koordinatı → Yeşil

Z koordinatı → Mavi

1. Euler dönüşü → Mavi

2. Euler dönüşü → Macenta

3. Euler dönüşü → Sarı

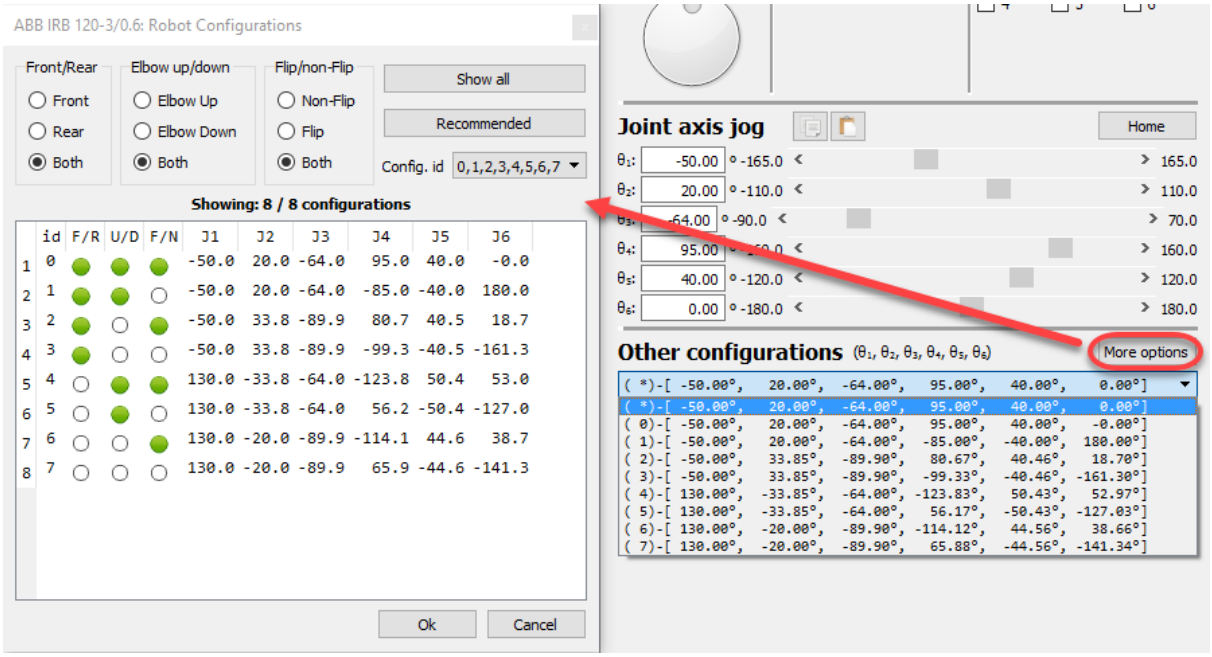
TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

19. ROBOT KONFIGÜRASYONU

Bir robot konfigürasyonu, robotun belirli bir durumunu tanımlar. Yapılandırmayı değiştirmek, bir tekilliği aşmayı gerektirir. Robot kontrolörleri doğrusal bir hareket yapılırken bir tekilliği geçemezler (bunun için eklemsel bir hareket gerekir).

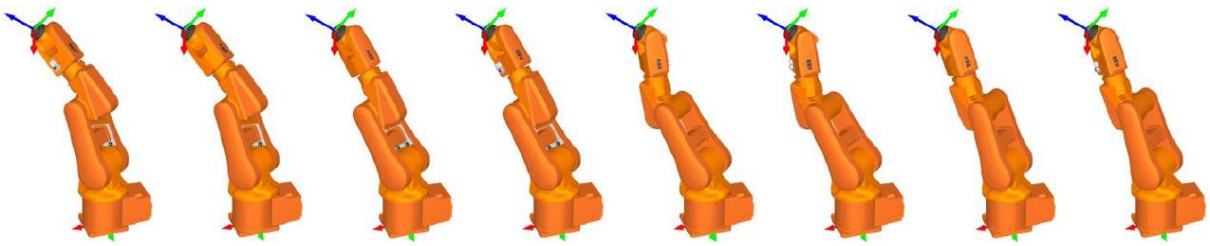
Başka bir deyişle, iki hedef arasında doğrusal bir hareket gerçekleştirmek için robot konfigürasyonu, ilk ve son noktalar da dahil olmak üzere tüm hareket için aynı olmalıdır.

Bir robota sağ tıklayın ve robot konfigürasyonları penceresini açmak için **Konfigürasyonu Değiştir** öğesini seçin. Robot panelinde Diğer seçenekler seçilerek de bu pencerenin açılması mümkündür.



Resim 19.1Robot kongigürasyonları penceresi ile robot paneli

Standart bir 6 eksenli robot için, her robot ekseninin bir tam tur hareket edebileceğini varsayarsak, robotun herhangi bir konumu için tipik olarak 8 farklı konfigürasyon vardır. Uygulamada, eklem sınırları robota bağlı olarak az ya da çok kısıtlanabilir. Bu nedenle, robota bağlı olarak belirli bir konum için 1 ila 100'den fazla farklı robot konfigürasyonuna sahip olmak mümkün olabilir.



Resim 19.2 Aynı robot poziryonu için farklı konfigürasyonlar.

Bir robot konfigürasyonu, robotla bir konuma ulaşmanın belirli bir yolunu (montaj modu) tanımlar. Örneğin, robotun dirseği yukarı veya aşağı olabilir, aynı zamanda hedefe dönük olabilir, olmayabilir ve hedefe geri ulaşmak için 180 derece dönebilir. Toplamda bu 2*2*2=8 konfigürasyon sağlar.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 4

ROBOT İÇİN EKSEN TANIMLAMA

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, robotlar için eksen oluşturabilir.

Öğrenci, çalışma alanı içerisinde eksen tanımlayabilir.

Öğrenci, eksenin önemini kavrayabilir.

İşlem Basamakları:

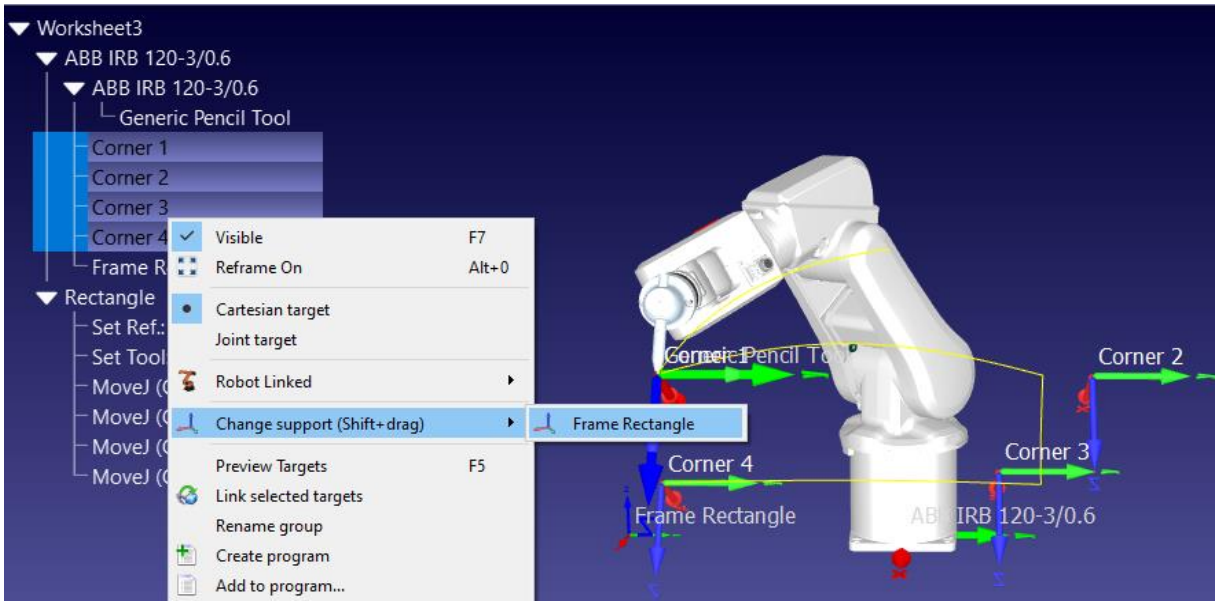
1- Masaüstündeki program simgesine çift tıklayarak programı açınız. 

2- Daha önce oluşturulan çalışma sayfasını açmak için  butonuna basınız ve dosyayı seçiniz.

3- Tüm hedeflerimizi bir eksen yapalım. İlk önce  tuşuna basarak ekrana bir Eksen getiriniz. İstasyon Ağacında Frame2 olarak görünür.

4- Çalışma sayfasında oluşan Frame2'yi fare ile tutarak istasyon ağacında robotun alt kısmına getiriniz. Ardından F2 tuşunu kullanarak Frame2'nin adını Frame Dikdortgen olarak değiştiriniz.

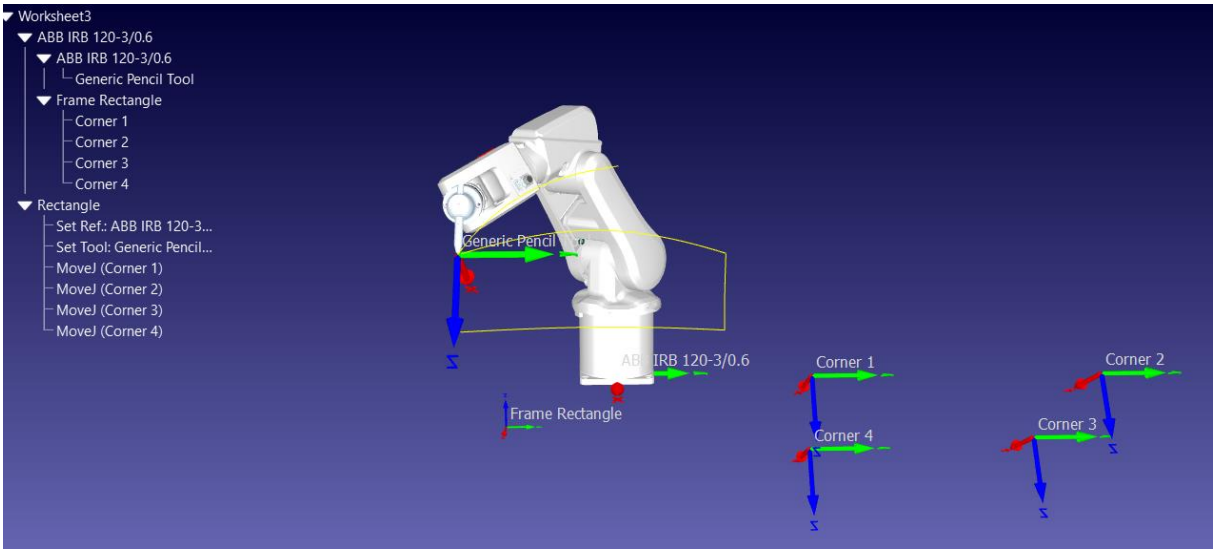
5- Ekseni oluşturduktan sonra şimdi tanımlayalım. Kose1, Kose2, Kose3 ve Kose4'ü seçiniz. Sağ tuş ile açılan menüden **Üst Öğe Değiştir** komutunu seçiniz. Frame Dikdortgen satırına tıklayınız.



Resim 19.3 Hedeflerin Rectangle eksenine taşınması

- Tüm Köşe Hedefleri Dikdortgen ekseni altında oluşturulur. Şimdi ekseni hareket ettirerek bu dört köşeyi hareket ettiriyorsunuz.  butonunu kullanarak bunu deneyebilirsiniz.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 19.4 Hedefler Rectangle eksenine taşındı

6- ABB IRB 120-3/0.6 panel Robot Panelinden **Reference Frame** Frame Rectangle seçiniz. Ardından İstasyon Ağacı'ndaki robot adına sağ tıklayarak menüyü açınız. Buradan Active Reference Frame seçiniz. Aynı işlemi Dikdortgen ekseni için tekrarlayınız.

7- Çalıştırdığınızda program çalışmayacaktır. Çünkü oluşturulan Eksen istasyona tanıtılmadı. Set Ref'e gidiniz ve sağ tıklama ile menüyü açınız. Set Reference link 'te Frame Rectangle seçiniz. Böylece oluşturduğunuz Frame2 istasyondaki tüm elemanlara tanıtılmış oldu. Frame2'yi hareket ettirerek robotu istenilen 4 koordinatta çalıştırmış olursunuz.

8- Robota sağ tıklayarak robot konfigürasyonları penceresini açınız. Konfigürasyonu değiştir öğesini seçiniz. Robot panelinde Diğer seçenekler seçilerek de bu pencerenin açılması mümkündür. Ve diğer yapılandırmaları seçerek robottaki değişimleri gözlemleyiniz.

9- butonuna basarak çalışma sayfasını worksheet 4 olarak kaydediniz.

10- Lütfen Universal Robot UR-10 için aynı işlem basamaklarını tekrarlayınız.

11- Cobot ile yaptığınız bu istasyonu worksheet 4-1 olarak kaydediniz.

Çalışma Sorusu:

Lütfen programın Yardım menüsündeki Eksen başlıklarını okuyunuz. Örnek uygulamalar getiriniz.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

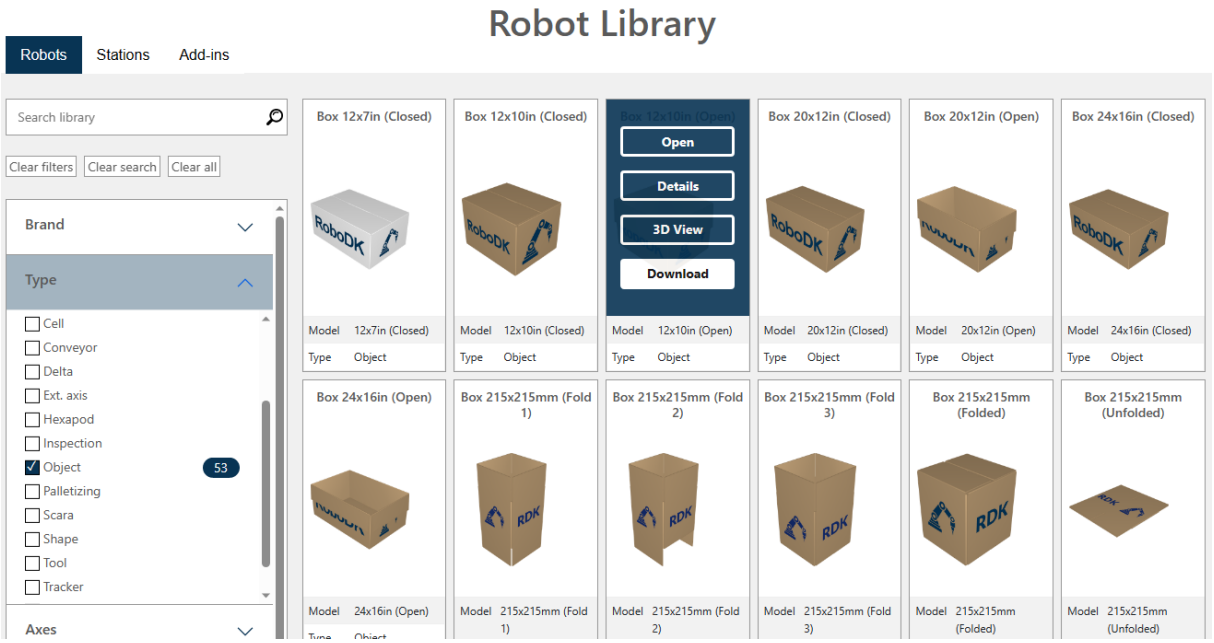
20. 3D NESNELERİ DAHİL ETME

RoboDK, STL, STEP (veya STP) ve IGES (veya IGS) formatları gibi çoğu standart 3D formatı destekler, böylece örneğin ücretsiz Tinkercad yazılımını kullanarak kendi 3D nesnelerinizi oluşturabilir ve bunları RoboDK'nın projesine aktarabilirsiniz. WRML, 3DS veya OBJ gibi diğer formatlar da desteklenir (STEP ve IGES, Mac ve Linux sürümlerinde desteklenmez)
Yeni bir 3D dosyası yüklemek için şu adımlar izlenir:

1. **Dosya➔ Aç** seçin.
2. Bilgisayardan 3D nesne seçin.
3. Alternatif olarak nesneyi RoboDKçalışma sayfasına sürükleyip, bırakın.
4. İstasyon Ağacı'ndaki nesneleri yeniden sıralamak için sağ tıklamayı basılı tutarak nesneyi sürükleyip bırakın.

RoboDK'nın PC'nizdeki varsayılan kitaplığından (C:/RoboDK/Library) veya RoboDK'nın çevrimiçi kitaplığından projenize tablo, kutu, şişe, zemin gibi yeni nesneler de ekleyebilirsiniz.

Dosya→Robot kitaplığını açlibrary (Ctrl+Shift+O) seçilir veya araç çubuğundaki butonu seçilir. RoboDK's online kitaplığından **Type** menüsünde **Object** seçilir.

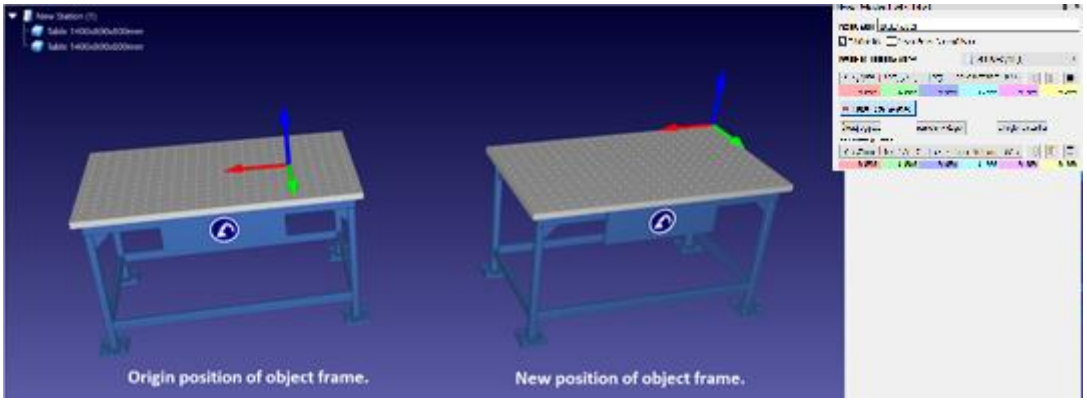


Resim 20.1. RoboDK's online kitaplığı

Nesneyi seçin ve **Download** 'a tıklayın. Seçilen nesne, projenizde birkaç saniye içinde otomatik olarak görünmelidir. Nesne yüklendikten sonra çevrimiçi kitaplık kapatılabilir.

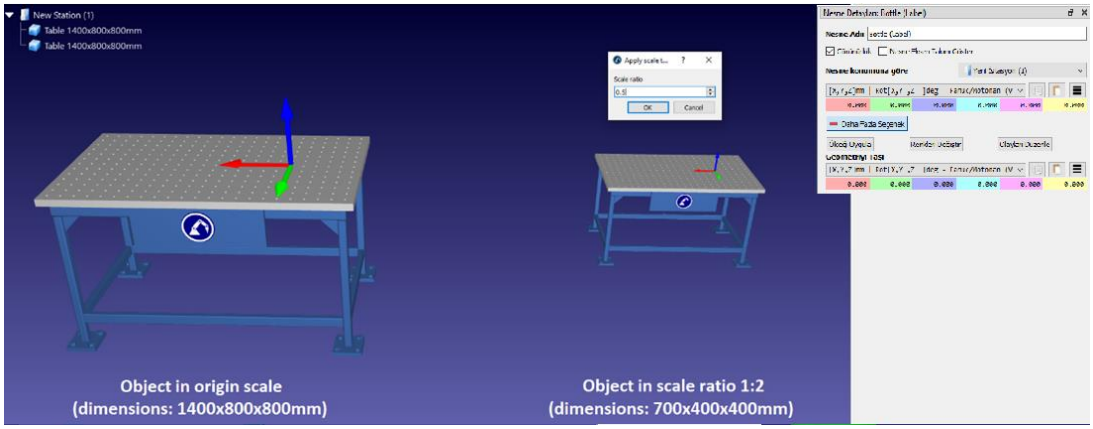
Her nesnenin, nesneyi başka bir nesneye, robota veya başka bir referans çerçevesine göre yerleştirmeye yardımcı olan bir nesne ekseni vardır. İsterseniz içe aktarılan nesnede nesne ekseninin konumunu değiştirebilirsiniz. **Nesne Detayları** penceresini açmak için nesneye çift tıklayın ve **Daha Fazla Seçenek**'e tıklayın. Ardından, nesne çerçevesinin konumunu değiştirmek için **Geometriyi Taşı** seçeneğini kullanarak koordinat değerlerini girebilirsiniz. Ardından, değişiklikleri onaylamak ve nesne çerçevesinin orijin konumunu yeniden tanımlamak için **Taşımayı uygula**'yı tıklayın.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



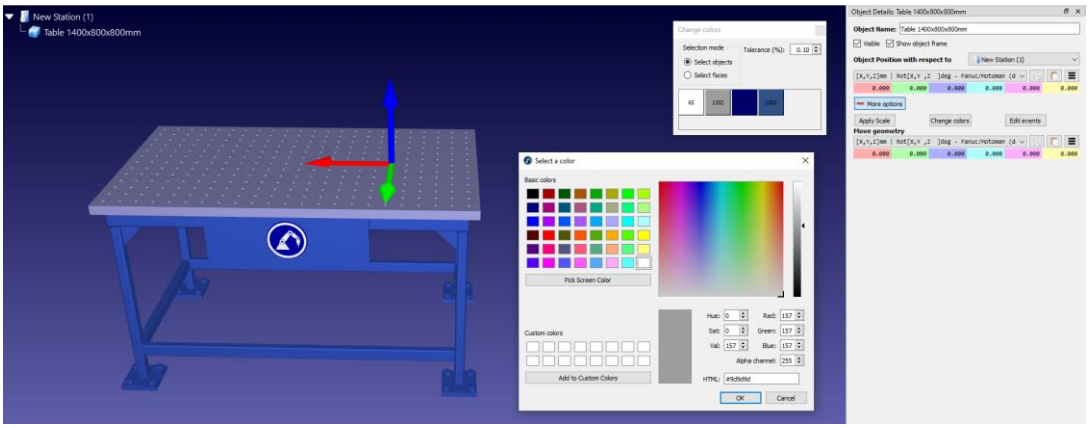
Resim 20.2. Aynı masa için farklı nesne ekseni pozisyonları

Aynı **Nesne Detayları** penceresinde içe aktarılan nesnenin ölçeğini veya rengini değiştirmek de mümkündür. İçe aktarılan nesnenin boyutlarını (ölçeği) değiştirmek için **Ölçeği Uygula**'ya tıklayın. Ölçeği uygula penceresi görünecektir. Ardından pencerede ölçek oranını girin ve **Tamam**'a tıklayarak değişiklikleri onaylayın.



Resim 20.3. Aynı masa için farklı ölçekler

İçe aktarılan bir nesnenin rengini değiştirmek için **Renkleri Değiştir**'e tıklayın. Renkleri değiştir penceresi görünecektir. Ardından, **Renkleri değiştir** penceresinde nesneye tıklayın ve değiştirmek istediğiniz rengi seçin, çünkü her rengi ayrı ayrı değiştirmek mümkündür. Bir renk seçin penceresi görünecektir. Ardından yeni rengi seçin ve **Tamam**'a tıklayarak değişiklikleri onaylayın.



Resim 20.4. Renk değiştirme adımları

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 5

ROBOT İÇİN ÇALIŞMA ALANI OLUŞTURMA

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, çalışma alana 3D nesneler ekleyebilir.

Öğrenci, 3D nesne pozisyonunu refrerans ekseni çerçevesinde değiştirebilir.

Öğrenci, 3D nesnelerin renk ve büyüklüğünü değiştirebilir.

İşlem Basamakları:

1- Masaüstündeki program simgesine çift tıklayarak programı açınız.

2- **Dosya / Robot Kitaplığını Aç** menüsünden **UR10** robotu seçiniz ve çalışma sayfasına getiriniz.

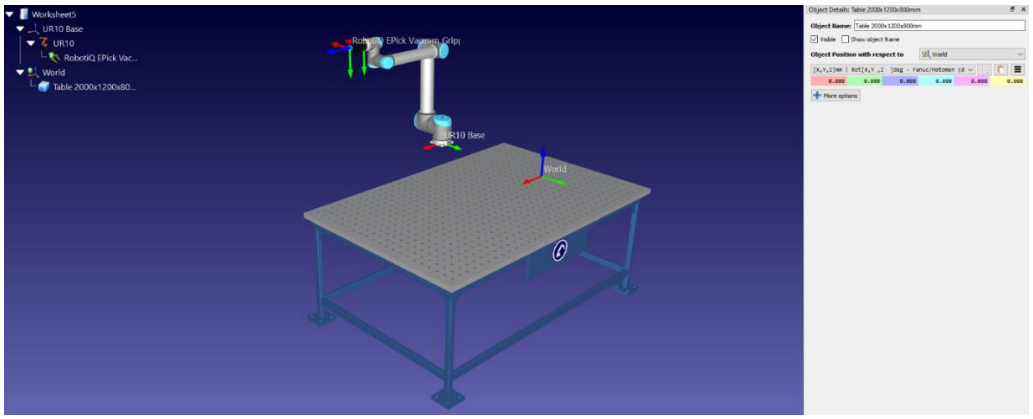
3- **Dosya / Robot Kitaplığınıı Aç** menüsünden **RobotiQ EPick Vacuum Gripper (1 Cup)** seçin ve çalışma sayfasına getirin. Robot takımı, UR10 Base çerçevesine otomatik olarak eklenecektir.

4- **Eksen Takımı Ekle** butonuna basarak ekrana yeni bir eksen getirelim. İstasyon Ağacı'nda Frame2 olarak görünür. O zaman **F2** tuşunu kullanarak Frame2'nin adını World olarak yeniden adlandıralım.

5- **Dosya / Robot Kitaplığınıı Aç** menüsünden **Table (Masa) 2000x1200x800mm** seçin ve çalışma sayfasına getirin.

6- İstasyon Ağacı'nda Table 2000x1200x800mm'yi seçip sağ tuş ile açılan menüden **Hedefi Değiştir** komutunu seçelim. World eksen çizgisine tıklayalım. Artık Table 2000x1200x800mm World ekseninin altındadır.

7- İstasyon Ağacı'ndaki Table 2000x1200x800mm üzerine çift tıklayarak **Nesne Detayları** penceresini açın. Ardından, aşağıdaki şekilde gösterildiği gibi Table eksen konumunu World eksenine göre ortalamak için XYZ koordinat değerlerini 0 olarak değiştirin.



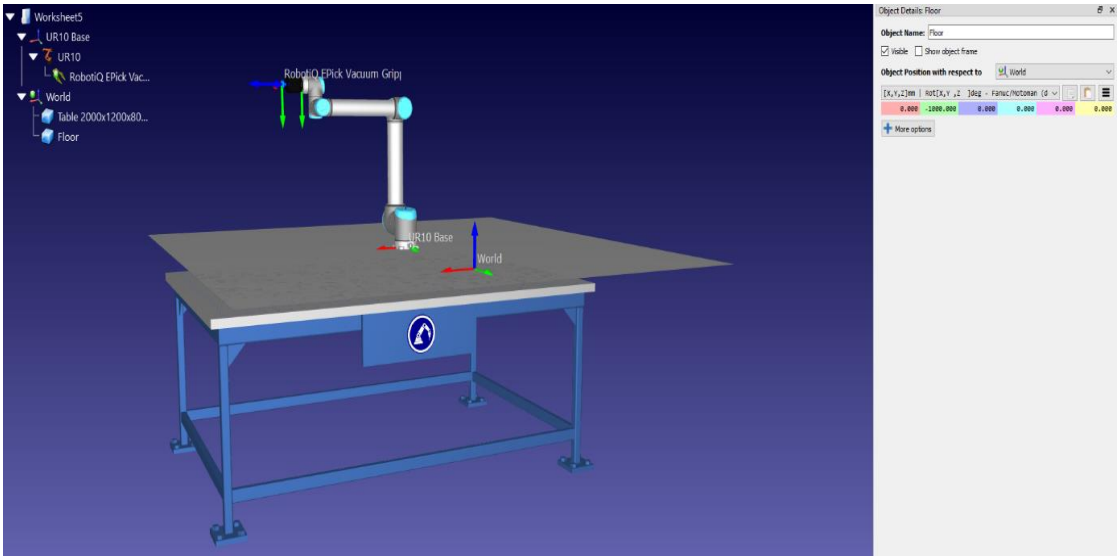
Resim 20.5. Table World eksenine taşındı.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

8- **Dosya / Robot Kitaplığınıı Aç** menüsünden **Floor** (Zemin) seçin ve çalışma sayfasına taşıyın.

9- İstasyon Ağacı'nda Floor'u seçip sağ tuş ile açılan menüden **Hedefi Değiştir** komutunu seçelim. World eksen çizgisine tıklayalım. Artık Flour, World ekseninin altındadır.

10- Flour, aşağıdaki şekilde gösterildiği gibi, örneğin Table'da yanlış konumda görünüyorsa, World eksenine göre Flour'un konumunun değiştirilmesi gerekecektir.



Resim 20.6. Table üzerindeki zeminin yanlış pozisyonuna bir örnek.

Nesne Detayları penceresini açmak için İstasyon Ağacı'ndaki Flour'a çift tıklayın. Ardından, Zemini doğru konuma taşımak için XYZ koordinat değerlerini değiştirin.

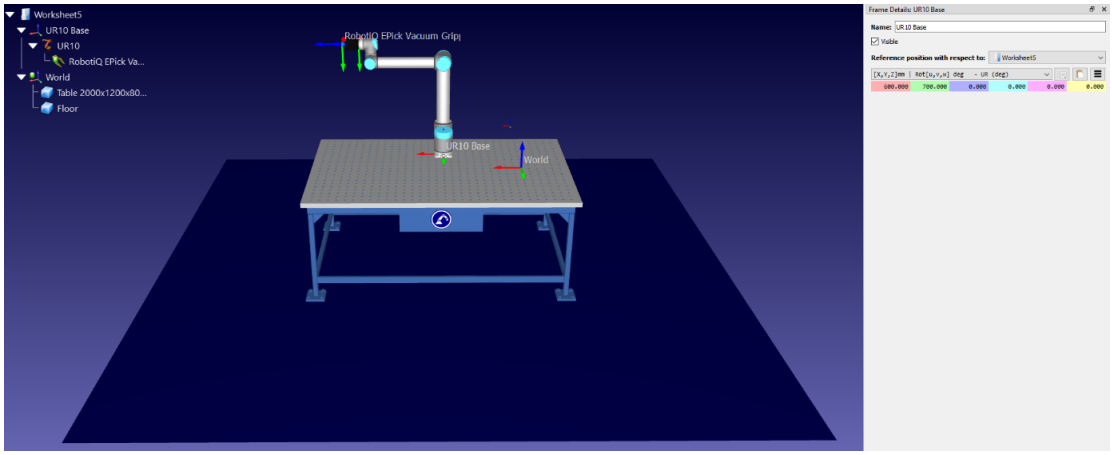
İpucu: Table'ın yüksekliği 800 mm'dir.

11- Flour yüzeyini arttıralım. Nesne Detayları penceresinde **Diğer seçenekler**'e tıklayın ve Flour'un ölçeğini 1,0'dan 1,5'e değiştirmek için **Ölçeği Uygula** seçeneğini belirleyin.

12- Flour rengini değiştirelim. Nesne Ayrıntıları penceresinde **Diğer seçenekler**'e tıklayın ve Flour'un rengini değiştirmek için **Renkleri değiştir** seçeneğini belirleyin.

13- UR10 robotumuzu masanın üzerine yerleştirelim. Çerçeve Ayrıntıları penceresini açmak için İstasyon Ağacı'ndaki UR10 Taban çerçevesine çift tıklayın. Ardından, UR10 robotunu aşağıdaki şekilde gösterilen konuma taşımak için XYZ koordinat değerlerini değiştirin.

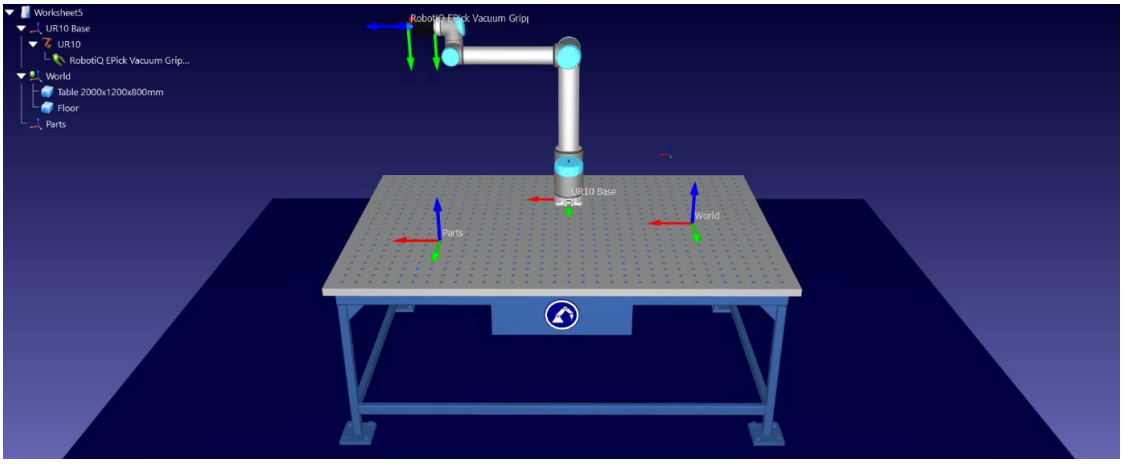
TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 20.7. UR 10 cobot'un masadaki yerleşimi

14- **Eksen Takımı Ekle** butonuna basarak ekrana yeni bir eksen getirelim. İstasyon Ağacı'nda Frame3 olarak görünür. Ardından, **F2** tuşunu kullanarak Frame3'ün adını **Parts (**Parçalar) olarak değiştirelim.

15- Parts eksenini masaya yerleştirelim. **Eksen Takımı Detayları** penceresini açmak için İstasyon Ağacı'ndaki Parts eksenine çift tıklayın. Ardından, Parts eksenini aşağıdaki şekilde gösterilen konuma taşımak için XYZ koordinat değerlerini değiştirin.



Şekil 20.8. The Parts eksenin masaya yerleştirildi.

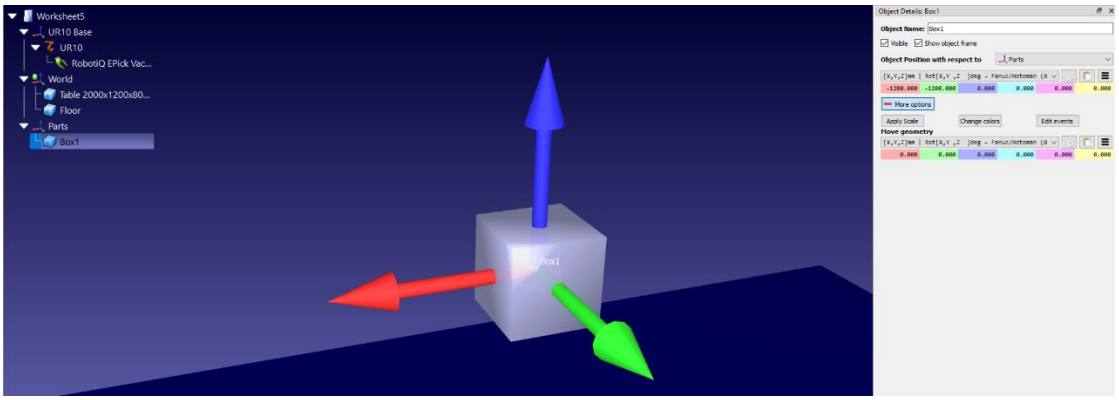
16- Bilgisayarınızdaki RoboDK'nın varsayılan kitaplığından (C:/RoboDK/Library) yeni nesne eklemek için **Dosya➔ Aç**'ı seçin. Ardından **kutu**yu seçin ve çalışma sayfasına getirin. **F2** tuşunu kullanarak kutunun adını **Box1** olarak yeniden adlandırın.

17- İstasyon Ağacı'nda Box1'i seçip sağ tuş ile açılan menüden **Üst öğe değiştir** komutunu seçelim. Frame Parts satırına tıklayalım. Artık Box1, Parts ekseninin altındadır.

18- Kutu1'i Masanın üzerine yerleştirelim. **Nesne Detayları** penceresini açmak için İstasyon Ağacı'ndaki Parts1'e çift tıklayın. Ardından Box1'i Masadaki Parts ekseni konumuna yerleştirmek için koordinat değerlerini [X, Y, Z] = [0, 0, 50] mm olarak değiştirin.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Box1 eksenini görünür kılmak için **Nesne Detayları** penceresindeki **Nesne Eksen Takımını Göster** seçeneğini işaretleyin. Aşağıdaki şekilde görebileceğiniz gibi, Box1 ekseni nesnenin merkez noktasında bulunur, bu nedenle XYZ koordinat değerlerini girerken bunu hesaba katmalısınız. Box1 ekseninin konumunu nesnenin tabanına taşımak için **Geometriyi Taşı** seçeneğini de kullanabilirsiniz (Box1 boyutları 100x100x100mm'dir).



Resim 20.9. Kutu1 eksen yerleşimi

19-  butonuna basarak çalışma sayfasını worksheet 4 olarak kaydediniz.

20- Lütfen Universal Robot UR-10 için aynı işlem basamaklarını tekrarlayınız.

21- Cobot ile yaptığınız bu istasyonu worksheet 4-1 olarak kaydediniz.

22- Lütfen çalışma sayfasına Kutu2 ismi ile bir kutu daha ekleyiniz. Rengini değiştirip, Kutu1'in yanına yerleştiriniz.

23- Bu eklemeden sonra çalışma sayfanızı Temrin 5-1 ile kaydediniz. 

Çalışma Sorusu:

Lütfen RoboDK Kütüphanesinde başka hangi nesnelerin mevcut olduğunu kontrol edin.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 6

AL VE YERLEŞTİR UYGULAMA PROGRAMLAMASI

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, robot kola Takım ekleyebilir.

Öğrenci, robot koldan Takım çıkarabilir.

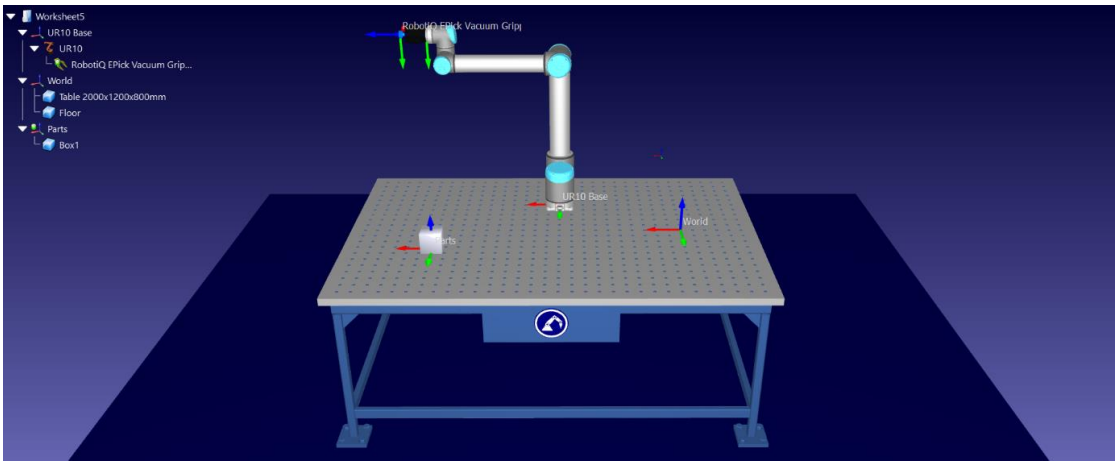
Öğrenci, ana programdan alt programlar çağırabilir.

İşlem Basamakları:

1- Masaüstündeki program simgesine çift tıklayarak programı açınız. 

2- Daha önce oluşturulan çalışma sayfasını açmak için  butonuna basınız ve Temrin 5 dosyasını seçiniz.

Çalışma istasyonunuz aşağıdaki gibi gözükecektir;

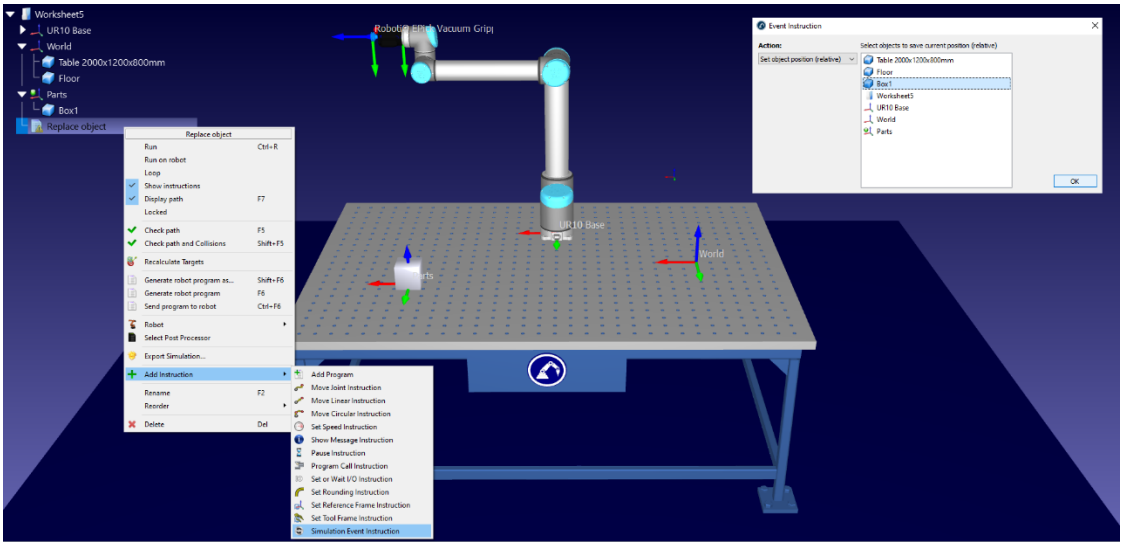


Resim 20.10. Temrin 5 çalışma istasyonu

3- Lütfen başlamak için **Program Ekle**  butonuna basınız. Yeni programınız İstasyon Ağacı'na **Prog1** olarak gelecektir. Prog1 satırını seçiniz ve F2 tuşunu kullanarak ismini **Replace object** olarak değiştiriniz. Replace object Box1'i ilk konumuna getirmek için kullanılacaktır.

Çalışma Ağacı'nda Replace Object satırında iken sağ tuşu tıklayınız. Replace object program menüsü görülecektir. Menüdeki **İşlem Ekle** satırından **Simülasyon Olayı Komutunu** seçiniz. Simülasyon Olayı pencerei görülecektir. **Set object position (relative)** komutunu **Hareket** açılır menüsünden seçiniz. Sonra çıkan pencereden Box1'i seçiniz ve **Tamam** butonuna tıklayınız. Şimdi İstasyon Ağacı'nda Replace object satırına çift tıklandığında Box1 ilk pozisyonuna gidecektir.

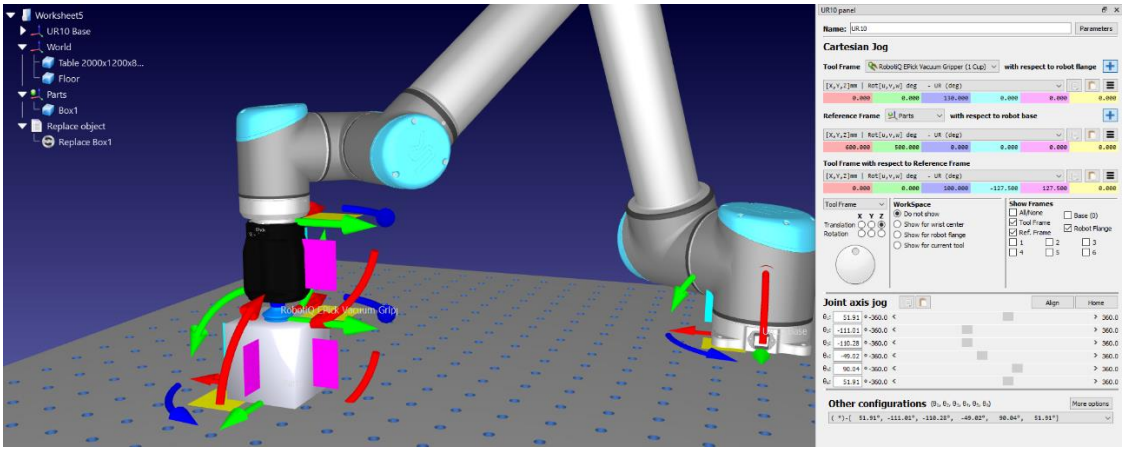
TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 20.11. Replace object programına Simülasyon Olayı komutunun eklenmesi

4- Şimdi Al ve Yerleştir uygulaması için bazı hedefler oluşturalım.  UR10 robotun eklemlerini eksen oklarını kullanarak hareket ettiriniz.

Önce robotu, RobotiQ EPick Vakum Tutucu aşağıdaki şekilde gösterildiği gibi Kutu1'in üst kısmının ortasına gelecek şekilde hareket ettirmeye çalışınız.



Resim 20.12. Vakum tutucu pozisyonlaması

UR10 robota çift tıklayınız ve açılan **Robot Panelini** kullanarak RobotiQ EPick Vacuum tutucuyu düzgün yerleştiriniz. Sonra **Takım Merkez Noktası** RobotiQ EPick Vacuum, **Eksen Takımı** Parts R**obot tabanına göre** olacak şekilde seçiniz ve sonra koordinat **değerlerini Eksen takımına göre takım merkez noktası**'nda değiştiriniz.

İstediğiniz noktay ulaştığınızda  butonuna basarak hedefi kaydediniz. Hedef1 Target 1 olarak İstasyon Ağacında gözükecektir. Daha sonra ismini PICK olarak değiştiriniz.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

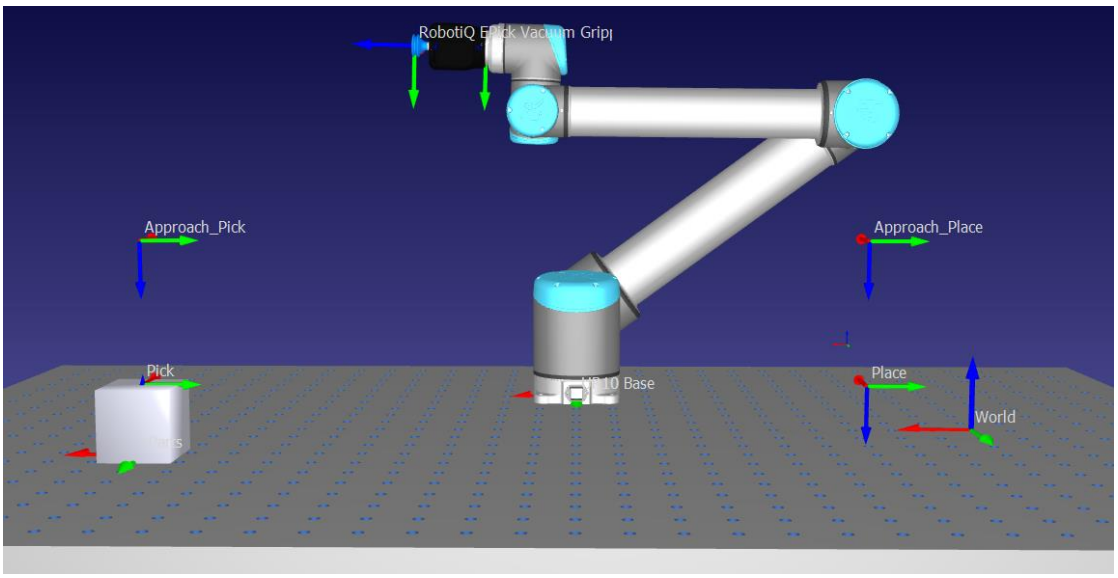
5- UR10 paneli ve butonu kullanrak 3 hedef daha oluşturunuz:

Approach_Pick – Pick hedefinden Z ekseninde 200mm taşındı.

Place – Pick hedefinden X ekseninde -1000mm taşındı.

Approach_Place – Place hedefinden Z ekseninde 200mm taşındı.

4 hedefte aşağıdaki şekilde yerleştirilmiş olacaktır.



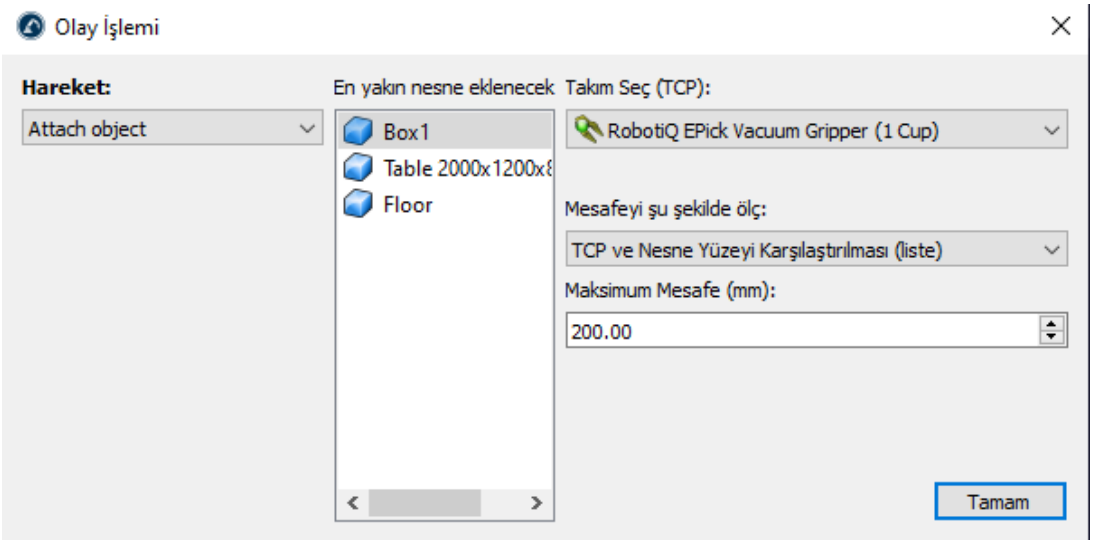
Resim 20.13. Hedeflerin yerleştirilmesi

6- İstasyon Ağacı'ndaki Approach_Pick hedefine çift tıklayınız, ardından butonuna basarak yeni bir program ekleyiniz**. F2** tuşuna basarak eklenen programın adını **Pick_Program** olarak değiştiriniz. Set Ref, Set Tool ve MoveJ satırları, **Pick_Program**'da otomatik olarak görünecektir.

7- CTRL tuşu ile İstasyon Ağacı'nda Pick hedefini ve Pick_Program'ı seçiniz ve Doğrusal Hareket butonuna basınız. Bundan sonra Pick_Program'a sadece MoveL (Pick) komutu gelir.

8- Pick_Program'a çift tıkladığımızda robotumuzun Approach_Pick'e ve ardından Pick hedeflerine ilerlediğini görüyoruz. RobotiQ EPick Vakumlu Tutucu daha sonra Box1'i tutmalıdır. Bunu **Simulasyon Olayı** komutu kullanarak yapacağız, bu yüzden **Olay İşlemi** penceresini açmak için butonuna basınız. Pencerede **Hareket** olarak **Attach object**'i seçiniz ve ardından **Mesafeyi şu şekilde ölç** olarak **TCP ve Nesne Yüzeyi (liste)** öğesini seçiniz ve nesneler listesinden Kutu1'i seçerek ve değişiklikleri Tamam diyerek onaylayınız. Şu andan itibaren Box1, RobotiQ EPick Vakum Tutucuya takılıdır.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 20.14. Attach object hareketi için Olay İşlemi penceresi.

9- Şimdi robotumuz Box1'i Approach_Pick pozisypnuna almalıdır. Approach_Pick hedefini ve Pick_Program seçiniz İstasyon Ağacı'nda **CTRL** tuşunu kullanarak. **Doğrusal Hareket** butonuna basınız. Bundan sonra Pick_Program'a sadece MoveL (Approach_Pick) komutu gelir. Pick_Program tamamlandı ve aşağıdaki şekildeki gibi görünmelidir.

Resim 20.15. Pick_Program

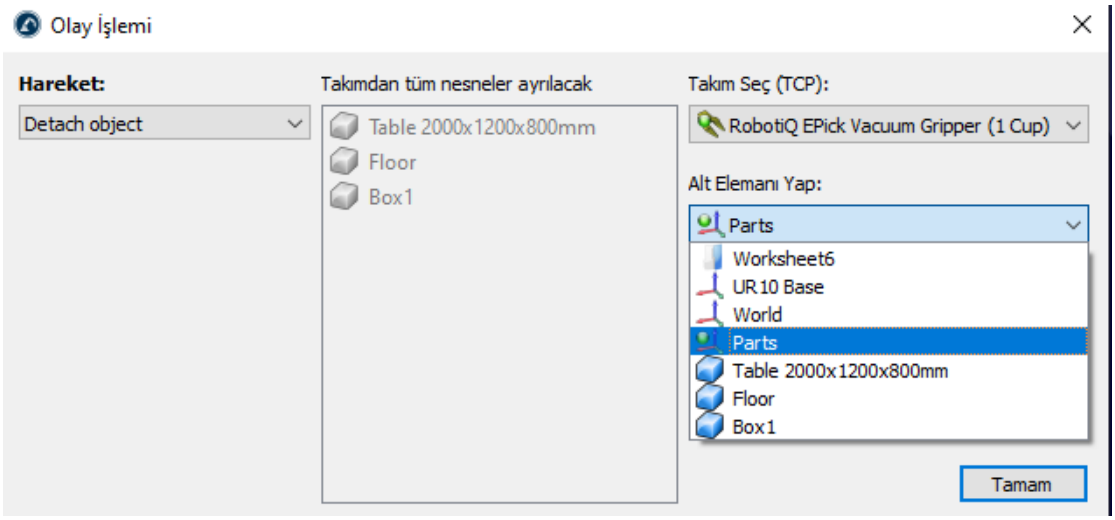
10- İstasyon Ağacındaki Approach_Place hedefine çift tıklayınız, ardından tuşuna basarak yeni bir program ekleyiniz. F2 tuşuna basarak eklenen programı **Place_Program** olarak yeniden adlandırınız. Set Ref, Set Tool ve MoveJ satırları, Place_Program'da otomatik olarak görünecektir.

11- **CTRL** tuşu ile İstasyon Ağacı'nda Place hedefini ve Place_Program'ı seçiniz. **Doğrusal Hareket** butonuna basınız. Bundan sonra Place_Program'a sadece MoveL (Yerleştir) komutu gelir.

12- Pick_Program'a çift tıkladığımızda ve ardından Place_Program'a çift tıkladığımızda, robotumuzun Box1'i orijin konumundan Approach_Place'e ve ardından Place hedeflerine taşıdığını görüyoruz. RobotiQ EPick Vakumlu Tutucu daha sonra Box1'i Place konumunda bırakmalıdır. Bunu, **Simulasyon Olayı** komutu kullanarak tekrar yapacağız. Bu yüzden **Olay İşlemi** penceresini açmak için butonuna tıklayınız. Bu kez pencerede **Hareket** olarak **Detach object** seçiniz ve ardından **Ana**

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Eleman Yap olarak Parts'ı seçiniz. Değişiklikleri Tamam diyerek onaylayınız. Artık Box1, RobotiQ EPick Vakum Tutucudan ayrılmış ve Parts eksenine takılmıştır.



Resim 20.16. Detach object hareketi için Olay İşlemi penceresi.

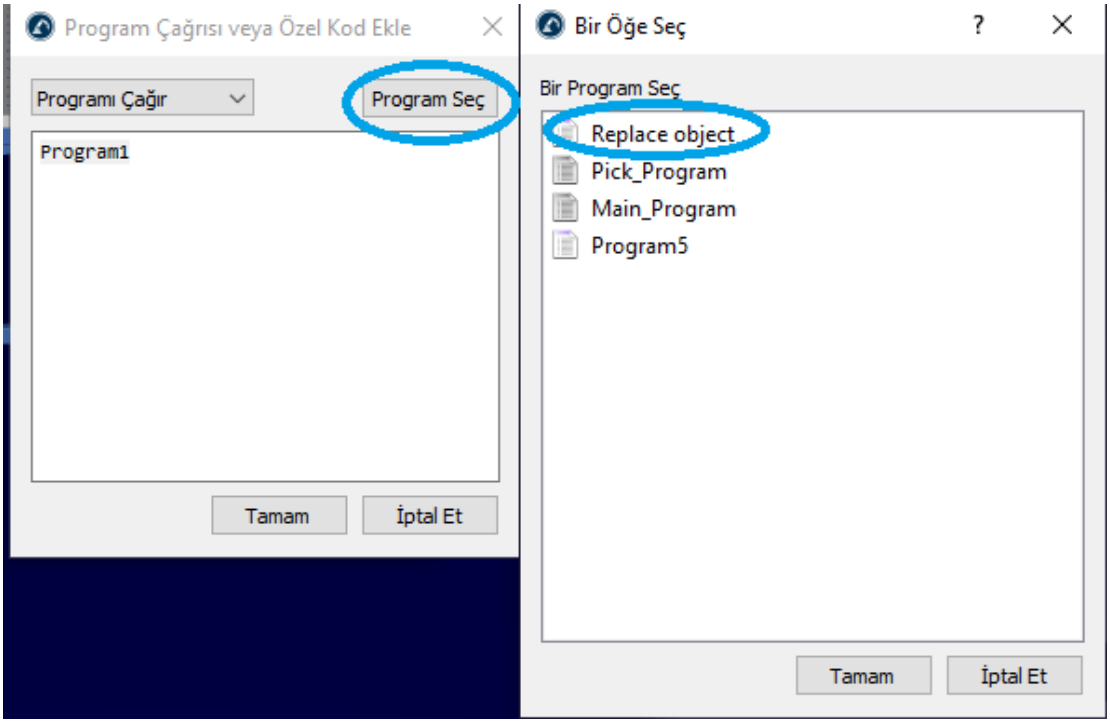
13- Al ve Yerleştir uygulamamızın sonunda UR10 robotu Approach_Place konumuna geri çekilmelidir. **CTRL** tuşu ile İstasyon Ağacında Approach_Place hedefini ve Place_Program'ı seçiniz. **Doğrusal Hareket** butonuna tıklayınız. Bundan sonra Place_Program'a sadece **MoveL** (Approach_Place) komutu gelir. Place_Program tamamlandı ve aşağıdaki şekildeki gibi görünmelidir.

Figure 20.17. Place_Program

14- Tüm programları sıra ile yürüten bir ana program oluşturalım. yeni bir program eklemek için butouna tıklayınız ve **F2** tuşunu kullanarak eklenen program adını **Main_Program** yapınız.

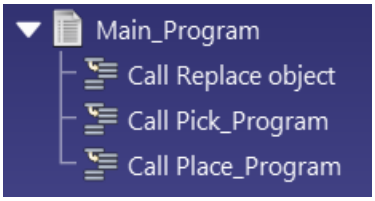
15- Program Çağrısı penceresini açmak için butonuna tıklayınız ve pencerede Program Seç'e tıklayınız. Ardından, Main_Program'a ilk olarak eklemek istediğimiz program olarak Replace object seçiniz. Tamam'a tıklayarak tüm değişiklikleri onaylayınız.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 20.18. Program Çağrısı Komut penceresi

Pick_Program and Place_Program için önceki adımı tekrarlayınız. Main_Program aşağıdaki şekilde görülecektir.



Resim 20.19. Main_Program

16- Main_Program satırına çifttıkladığınızda tüm simülasyon çalışacaktır. Simülasyonun sürekli çalışması için Main_Program'a sağ tıklayın ve **Loop** komutunu seçiniz.

İstasyon Ağacı'na sağ tıklayıp açılan menüden **Display path** onayını kaldırırsanız, robotunuzun hareket izleri ekranda gözükmez.

17- **Dosya→İstasyonu farklı kaydet seçilerek** temrin 6 olarak çalışmanızı kaydediniz.

18- Lütfen çalışma sayfası 5-1'i açın ve Al ve Yerleştir uygulamasını iki kutu ile programlamayı deneyiniz. Temrin 6-1 olarak kaydediniz.

Çalışma Sorusu:

Programdan seçilen hareket için robotun hızı nasıl değiştirilir?

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

21. ÇARPIŞMA ÖNLEYİCİ

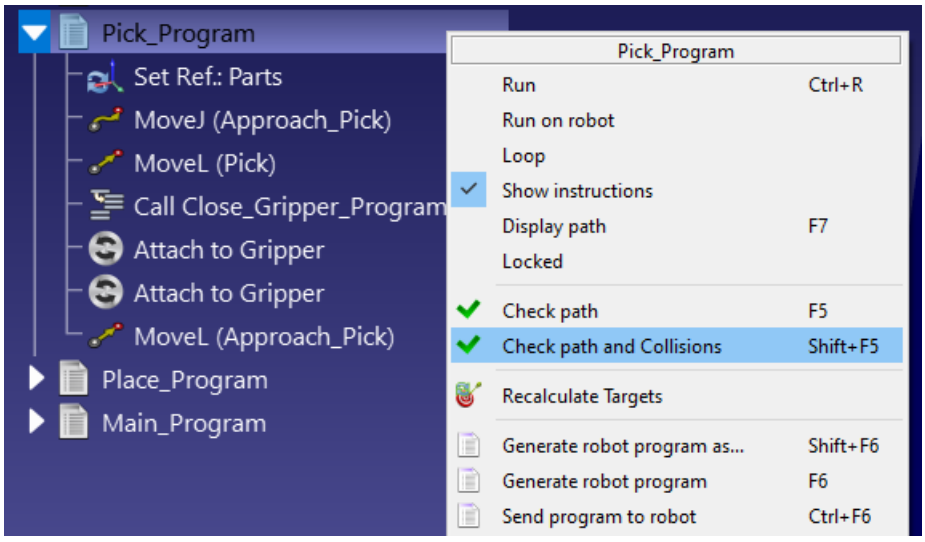
RoboDK ile çarpışma kontrolü, gerçek kurulumumuzda çarpışmaları önlememize yardımcı olabilir. Çarpışma kontrolü, çarpışmaları görsel olarak kontrol etme, robot işleme projeleri için çarpışmaları otomatik olarak önleme veya otomatik olarak çarpışmasız programlar oluşturmak için çarpışmasız bir harita oluşturma gibi farklı şekillerde kullanılabilir. RoboDK'daki sanal ortamın gerçek kurulumu tam olarak temsil etmeyebileceğini unutmayın. Bu nedenle, çarpışmaları güvenli bir şekilde önlemek için bir toleransın hesaba katılması önerilir. Kurulumumuzun daha büyük ve daha basitleştirilmiş 3B modellerini yükleyerek bunu yapabiliriz. Örneğin, iş milini basit bir küp olarak modelleyebiliriz (yalnızca çarpışma kontrolü için kullanılır).

Araçlar➔  **Çarpışmaları Kontrol Et** seçilerek çarpışma önleyici açılıp, kapatılır. Eğer çarpışma önleyici aktif ise, çarpışma algılandığında tüm robot hareketleri ve programlar duracaktır. Çarpışma durumu devam ettiği sürece bütün nesneler, takımlar and robot bağlantıları kırmızı renk olacaktır.

Bir çarpışma algılansa bile bir programı simüle etmeye devam etmeyi tercih ederseniz **Araçlar➔Seçenekler➔Hareket** menüsü seçilmeli ve **Bir Çarpışma Algılandığında Robot Hareketlerini Durdur** seçeneğindeki onay kaldırılmalıdır.

Çarpışmalara karşı bir programı güvenli bir şekilde kontrol etmek için şu adımları izleyin:

1. İstasyon Ağacı'nda program üzerine sağ tıklayın.
2. **Yolu ve Çarpışmaları Kontrol Et seçiniz. (Shift+F5)**. Bu seçenek, yolun uygun olup olmadığını hızlı bir şekilde kontrol eder (**Yolu kontrol et - F5** ile aynı) ve ardından herhangi bir çarpışma olmadığını doğrular.

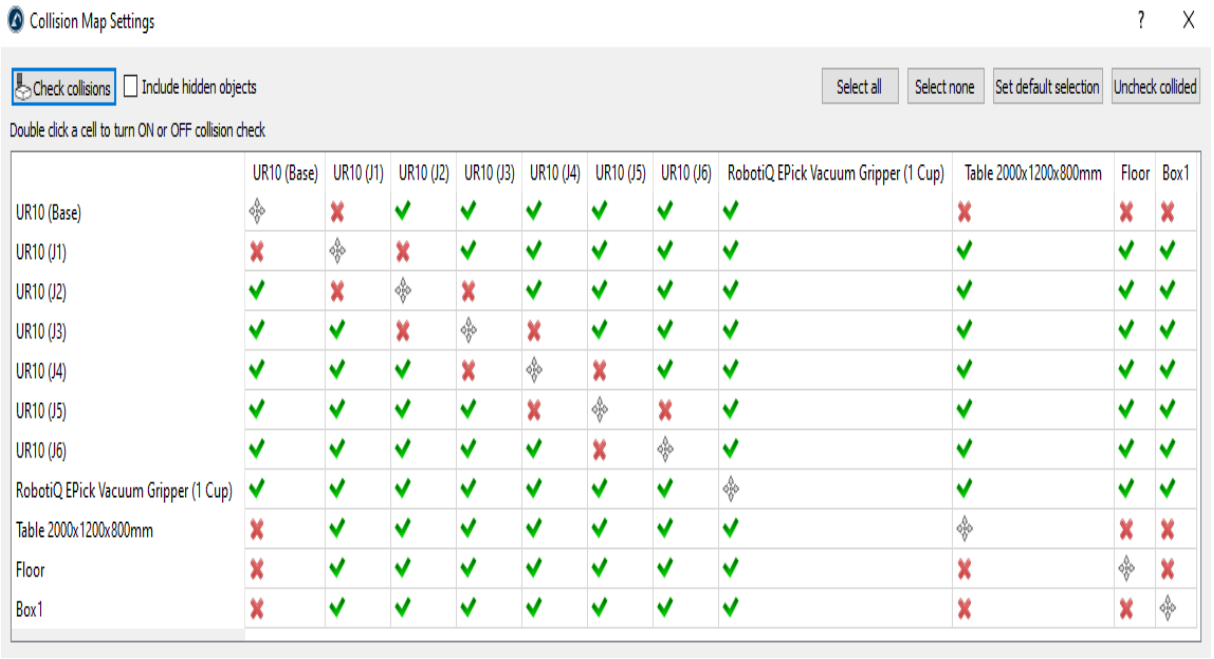


Resim 21.1. Yolu ve Çarpışmaları Kontrol Et seçeneğinin aktif edilmesi.

Herhangi bir nesne çifti arasındaki etkileşimin çarpışma açısından kontrol edilmesi gerekip gerekmediğini belirtebiliriz. Hücremizdeki tüm hareketli nesneler ile çarpışma kontrol durumu arasındaki ilişkiyi görüntülemek için **Araçlar➔**  **Çarpışma Haritası** seçilmelidir. Bu ilişki için çakışma denetimini etkinleştirmek veya devre dışı bırakmak için bir hücreye çift tıklayın. Emniyetli bir seçimi otomatik olarak ayarlamak için **Varsayılan Seçimi Ayarla** seçilmelidir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Varsayılan olarak RoboDK, tüm robot bağlantıları, nesneler ve takımları dahil olmak üzere istasyondaki tüm hareketli nesneler arasındaki çarpışmaları kontrol eder. Bir istisna olarak, ardışık robot eklemleri her zaman temas halinde olabileceğinden, çarpışmalara karşı kontrol edilmez.
Belirli bir nesne için diğer tüm nesnelerle çarpışma denetimini On veya OFF konuma getirmek için köşegendeki bir hücreye çift tıklanmalıdır.



Resim 21.2. Çarpışma Haritası Ayarlar penceresi

22. ROBOT PROGRAMI YARATMAK

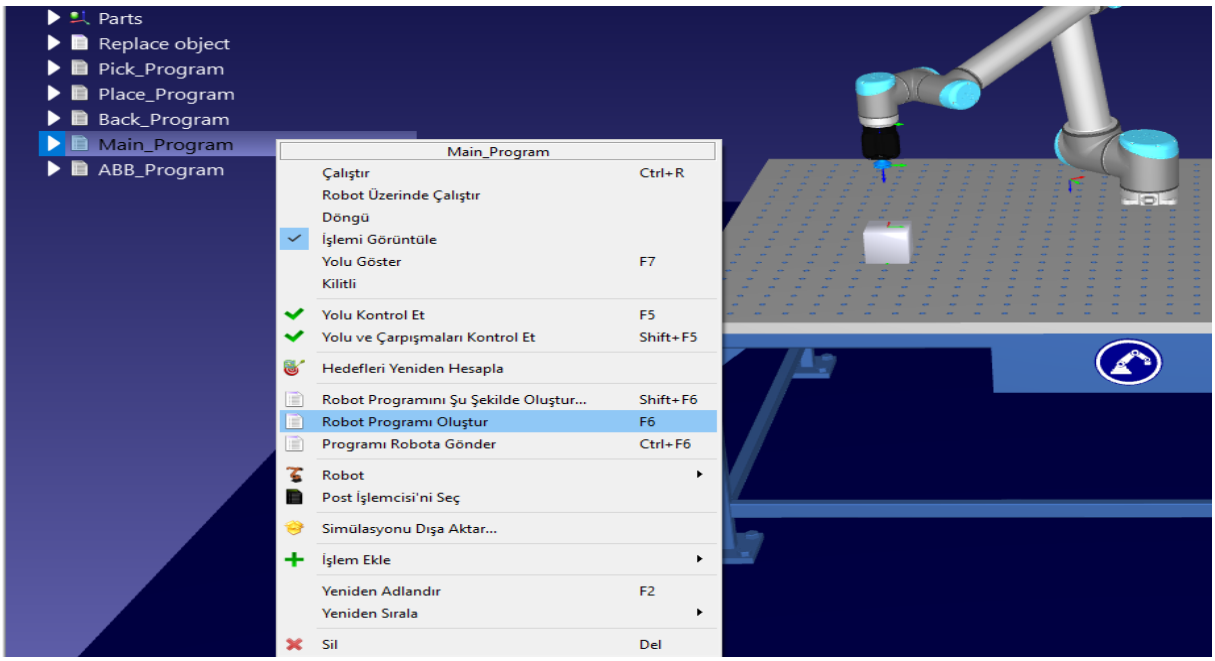
Simülasyonu RoboDK'da hazır hale getirdikten sonra, robot programını kolayca oluşturabilir, böylece tek bir kod satırı yazmak zorunda kalmadan programı robot kontrol cihazında çalıştırabilirsiniz. RoboDK simülasyonundan belirli bir robot programına dönüştürme, bir Son İşlemci tarafından yapılır. Son İşlemci, belirli bir robot için robot programlarının nasıl oluşturulması gerektiğini tanımlar. RoboDK'de varsayılan olarak her robotun belirli/varsayılan bir post işlemcisi vardır.

Herhangi bir programı tek tek veya alt programlar dahil olmak üzere ana programı dışa aktarabilirsiniz.

Robot işlemcisinin gerektirdiği robot programı oluşturmak için şu adımları izleyin:

1. **Program** menüsünü açınız.
2. **Program(lar) Oluştur** (F6) seçiniz.

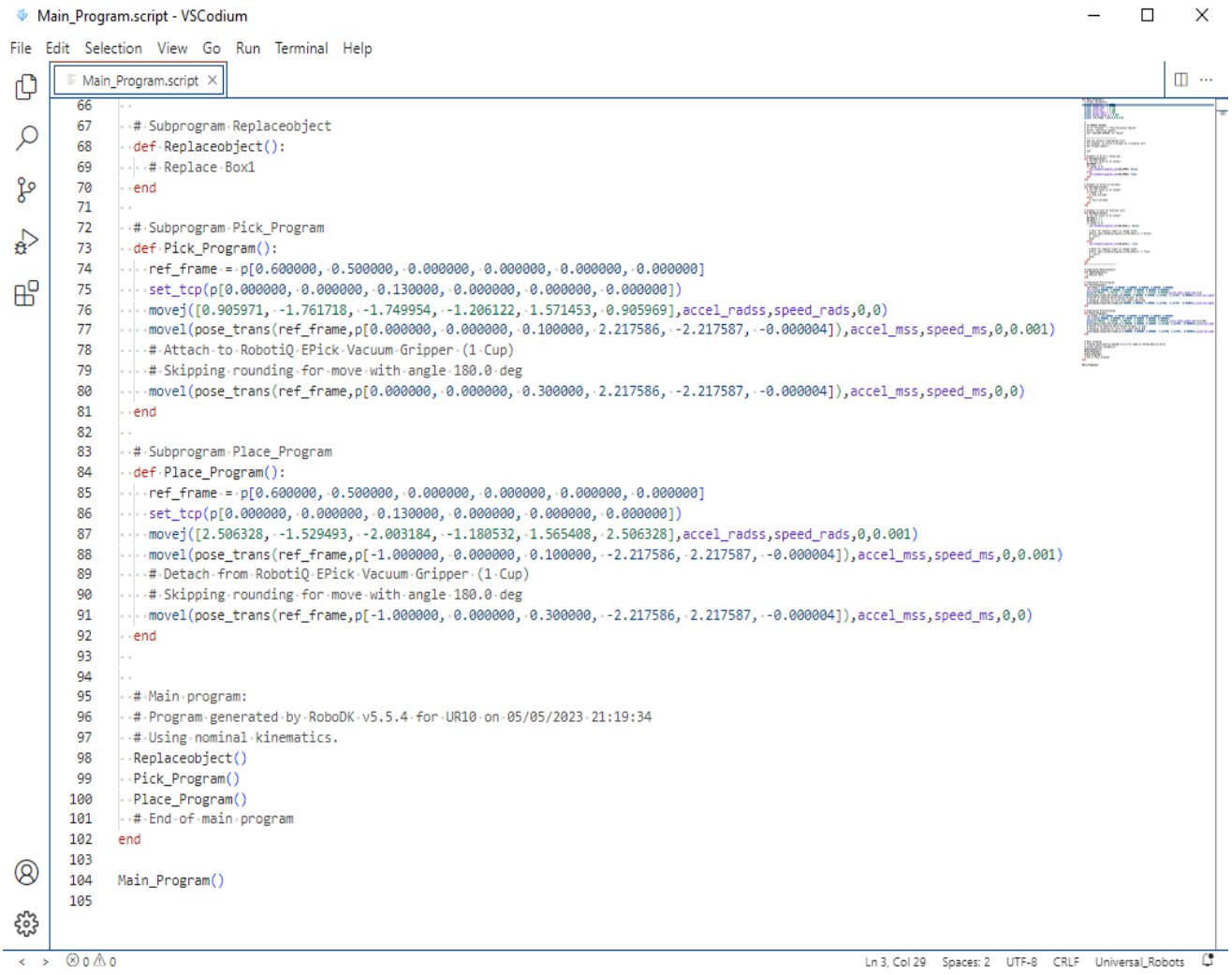
Diğer bir yol olarak ana program üzerinde iken farenin sağ tuşu tıklanır ve açılan menüden R**obot Program (F6) Oluştur** satırı seçilebilir.



Resim 22.1. Robot programı oluşturmak

Aynı anda birden fazla program oluşturmak için birden fazla program seçilebilir. Ctrl tuşunu basılı tutularak birden fazla program seçilir. Robot Programını Şu Şekilde Oluştur (Shift+F6) seçildiğinde, kullanıcıda programı kaydetmek için bir konum sağlamasını isteyen bir pencere açılır.

Elde ettiğiniz dosya, programı çevrimdışı oluşturmanın sonucudur. Dosya, RoboDK'da simüle edilen hareketlerin aynısını üretim ortamında çalıştırmak için kullanılan robot kontrol cihazına gönderilebilir.



Resim 22.2. Oluşturulmuş robot program örneği

Robota düzgün bir şekilde bağlanırsak, program menüsünde aşağıdaki seçeneklerden birini de seçebiliriz:

1. 1. Programı FTP (Çevrimdışı Programlama) aracılığıyla göndermek için **Programı robota gönder (Ctrl+F6)** .
2. Programı her çalıştırdığımızda (Çevrimiçi Programlama) programı adım adım çalıştırmak için **Robotta AÇılş** seçeneğini işaretleyin. Bu, programın simüle edildiği gibi aynı anda robot üzerinde yürütülmesine izin verir. Çevrimiçi Programlama için robot sürücüleri gereklidir.

Robotta Çalış seçeneği, robot sürücülerinin düzgün çalışmasını gerektirir. Bu sürücüler, robot denetleyicide ek yazılım seçenekleri ve/veya robot denetleyicide belirli bir kurulum gerektirebilir (UR robotları için durum böyle değildir).

TEMRİN 7

ROBOTIQ MEKANİK TUTUCUNUN KULLANILMASI

Bu çalışma yaprağının sonundaki amaçlar şunlardır:

Öğrenci, robot kola Takım ekleyebilir.

Öğrenci, RoboDK projelerinde mekanik tutucu kullanabilir.

Öğrenci tutucunun açılma seviyesini ayarlayabilir.

Öğrenci tutucuyu açmak ve kapatmak için alt programlar oluşturabilir.

İşlem Basamakları:

1- Masaüstündeki program simgesine çift tıklayarak programı açınız. 

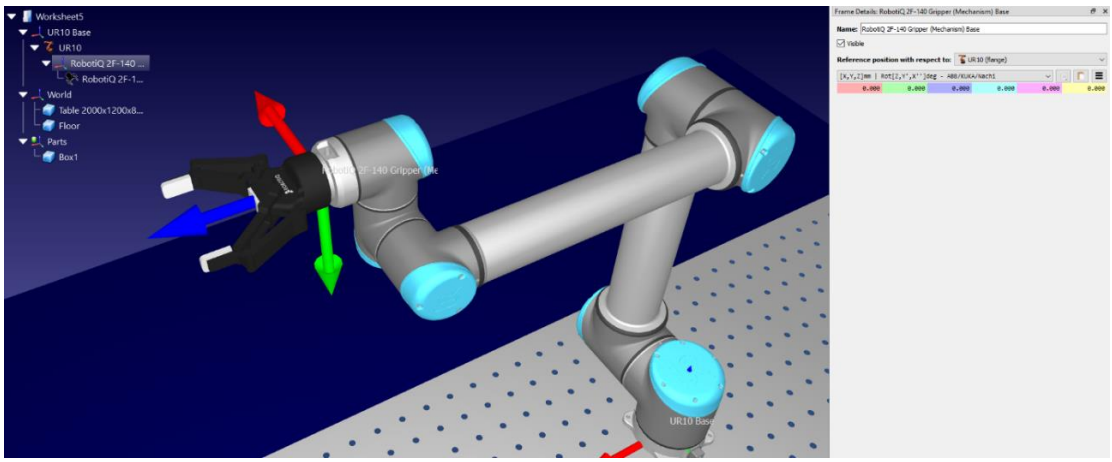
2- Daha önce oluşturulan çalışma sayfasını açmak için  butonuna basınız ve Temrin 5 dosyasını seçiniz.

3- Bu projede yer alan takımı kaldıralım. İstasyon Ağacı'ndaki RobotiQ EPick Vacuum Gripper tıklayınız ve Del tuşuna basarak siliniz.

4- **Dosya / Open Robot Library** menüsünde **RobotiQ-2F-140-Gripper-Mechanism**'i seçiniz ve çalışma sayfasına getiriniz.

5- RobotiQ 2F-140 Gripper (Mechanism) Base İstasyon Ağacı'nda görünecektir. Tutucuyu tutarak İstasyon Ağacı'ndaki UR10'un altına sürükleyerek, bırakınız.

6- İstasyon Ağacı'nda RobotiQ 2F-140 Gripper (Mechanism) Base tutucuya çift tıklayarak, **Takım Detayları** penceresini açınız. Ardından XYZ koordinat değerlerini 0 olacak şekilde değiştiriniz. RobotiQ 2F-140 Gripper UR10 tabanına eklenir.



Resim 22.3. The RobotiQ 2F-140 Gripper-Mechanism UR10 robota eklendi.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

7- Şimdi RobotiQ 2F-140 Gripper (Mechanism) için takım tanımlayalım. İstasyon Ağacı'nda RobotiQ 2F-140 Gripper (Mechanism) satırı üzerinde sağ tuşa tıklayalım. Menü açılacaktır. Menüden **Takım (TCP) Ekle** seçiniz. **Tool 1** İstasyon Ağacı'nda görünür ve **Tool 1 TCP** Ana Ekranda görünür. Aşağıda gösterilmektedir.

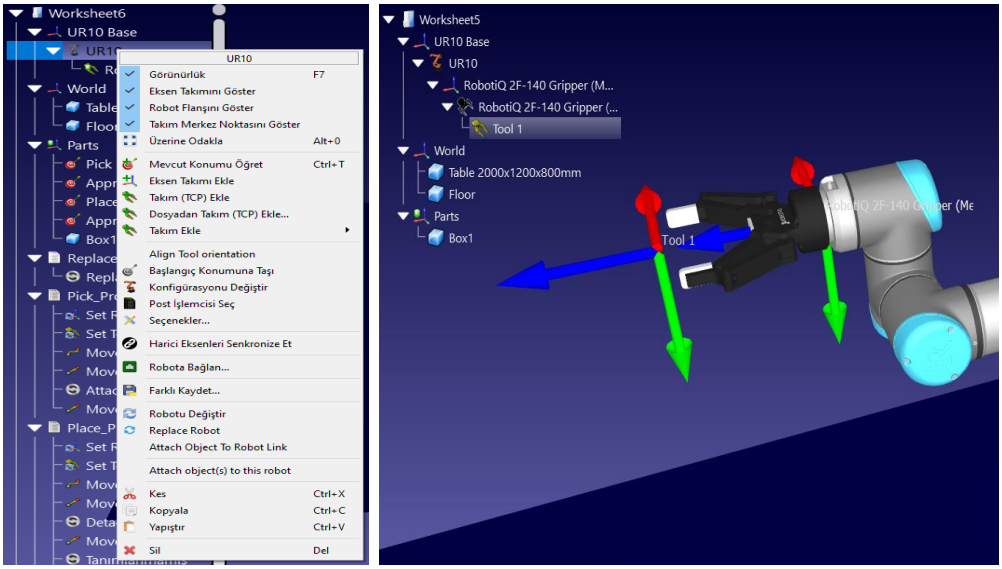


Figure 22.4. RobotiQ 2F-140 Gripper (Mechanism) için Takım tanımı

8- Varsayılan olarak RoboDK TCP'yi [X, Y, Z] = [0, 0, 200] mm. Konumunda tanımlayacaktır. Bu koordinatlar manuel olarak değiştirirlebilir. İstasyon Ağacı'nda Tool1'e çift tıklayarak Takım Detayları penceresini açınız. Koordinat değerlerini [X, Y, Z] = [0, 0, 155] mm olarak değiştirerek **Tool 1 TCP**'yi RobotiQ 2F-140 Gripper (Mechanism) flanşına biraz daha yaklaştırınız.**Takım Adı: Gripper**, **TCP'yi** göster onayı kaldırılmalıdır.

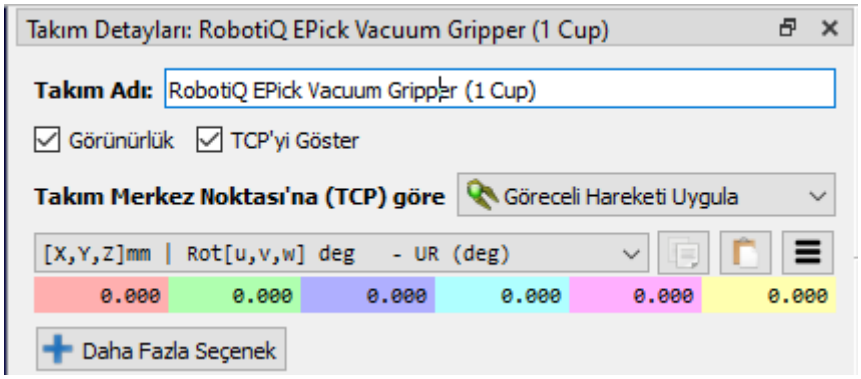
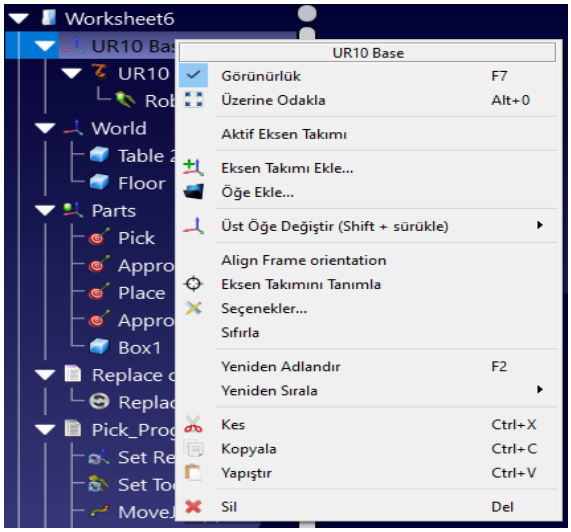


Figure 22.5. Tool 1 için Takım Detayları penceresi.

9- İstasyon Ağacı'nda RobotiQ 2F-140 Gripper (Mechanism) Base satırı üzerinde sağ tuş tıklanır. The RobotiQ 2F-140 Gripper (Mechanism) Base menenüsü açılır. **Aktif Eksen Takımı** menüden seçilir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 22.6. The RobotiQ 2F-140 Gripper (Mechanism) Base menüsü

10- İstasyon Ağacı'nda RobotiQ 2F-140 Gripper (Mechanism) çift tıklanır ve RobotiQ 2F-140 Gripper (Mechanism) panel penceresi açılır. Pencerede tutucunun açılış seviyesini 0-140mm değerleri arasında değitirebilirsiniz.

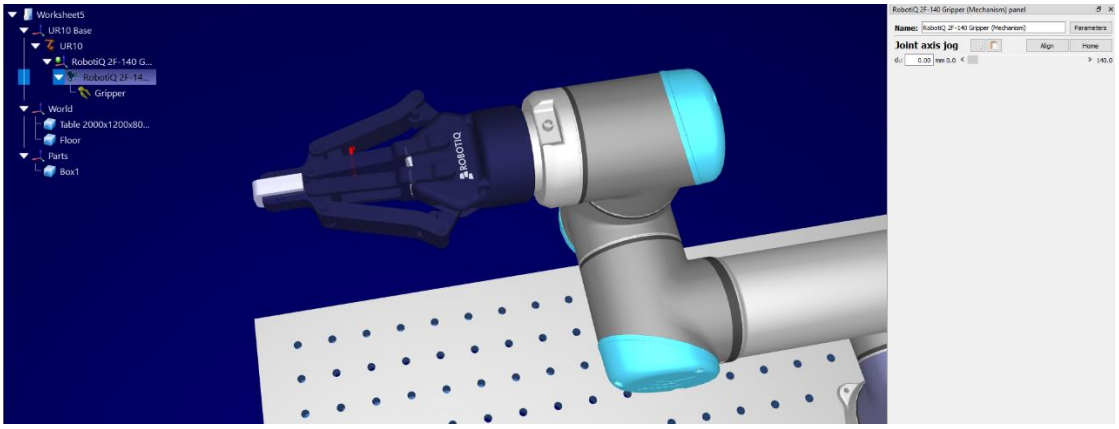


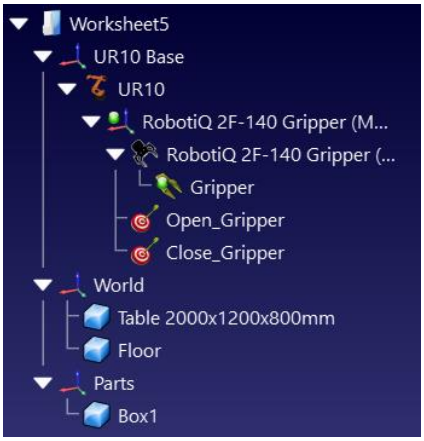
Figure 22.7. The RobotiQ 2F-140 Gripper (Mechanism) panel penceresi

11- Tutucu açılış seviyesini **140 mm** olarak ayarlayınız RobotiQ 2F-140 Gripper (Mechanism) panel penceresinde ve sonra  butonuna tıklayarak tutucu açıklığı için hedef oluşturunuz. Target 1 İstasyon Ağacı'nda Gripper altında görünecektir. Target 1 adını **Open_Gripper** olarak değiştiriniz.

ÖNEMLİ: RobotiQ 2F-140 Gripper (Mechanism) panel penceresini kapatınız! Eğer pencereyi kapatmadan başka bir hedef oluşturulursa, sadece birinci hedef çalışacaktır.

12- Tekrar RobotiQ 2F-140 Gripper (Mechanism) panel penceresini açınız. Bundan sonraki adımlarda Box1 kutusunu Tut ve Yerleştir yapacağız, dolayısıyla tutucu açılış seviyesini **100 mm** (Box1 ölçüleri 100x100x100mm) olarak ayarlayınız ve sonra  butonuna basarak tutucu kapanışı için hedef oluşturunuz. gripper. Target 2 İstasyon Ağacı'nda Gripper altında görünecektir. Target 2 adını **Close_Gripper** olarak değiştiriniz.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Resim 22.8. Open_Gripper ve Close_Gripper hedeflerinin İstasyon Ağacı'nda görünümü.

13- İstasyon Ağacı'nda Open_Gripper hedefi çift tıklayınız, sonra butonuna tıklayarak yeni bir program ekleyiniz ve eklediğinz programın adını F2 tuşunu kullanarak **Open_Gripper_Program** olarak değiştiriniz. Set Ref, Set Tool and MoveJ satırları otomatik olarak Open_Gripper _Program içerisinde oluşacaktır.

14- İstasyon Ağacı'nda Open_Gripper hedefi çift tıklayınız, sonra butonuna tıklayarak yeni bir program ekleyiniz ve eklediğinz programın adını F2 tuşunu kullanarak **Close_Gripper_Program** olarak değiştiriniz. Set Ref, Set Tool and MoveJ satırları otomatik olarak Close_Gripper _Program içerisinde oluşacaktır.

Resim 22.9. Open_Gripper_Program ve Close_Gripper_Program

15- Close_Gripper_Program çift tıklandığında tutucu kapanış simülasyonu and Open_Gripper _Program çift tıklandığında tutucu açılış simülasyonları çalışacaktır. Eğer doğru çalışırsa, projemizdeki herhangi başka bir programda Close_Gripper _Program ve Open_Gripper _Program çağrılarını ekleyebilirsiniz.

16- Artık oluşturulan tutucu için Tut ve Yerleştir uygulaması programını yapılabilir.

Başlangıçta, herhangi bir zamanda Box1'i orijinal konumuna geri getirmek için lütfen **Replace object** programı oluşturun. (Temrin 6, 3. İşlem Basamağı).

17- Robot paneli açmak için robotu çift tıklayınız. **Eksen Takımı "Parts" Robot tabanına göre** seçilir, sonra **Eksen takımına göre takım merkez noktası** koordinat değerlerini aşağıdaki gibi girerek 4 hedef oluşturunuz:

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Pick	0,	0,	100,	180,	0,	0.
Approach_Pick	0,	0,	300,	180,	0,	0.
Place	-1000,	0,	100,	180,	0,	0.
Approach_Place	-1000,	0,	300,	180,	0,	0.

İstediğiniz noktaya ulaştığınızda, hedefi oluşturmak için  butonuna tıklayınınız ve ardından oluşturulan hedefi yeniden adlandırınız.

18- İstasyon Ağacı'nda Approach_Pick hedefi çift tıklayınız, sonra  butonuna tıklayarak yeni bir program ekleyiniz ve eklediğiniz programın adını F2 tuşunu kullanarak **Pick_Program** olarak değiştiriniz. Set Ref, Set Tool and MoveJ satırları otomatik olarak Pick_Program içerisinde oluşacaktır.

19- CTRL tuşunu kullanarak İstasyon Ağacı'nda Pick hedefini ve Pick_Program'ı seçiniz.  Doğrusal Hareket butonuna tıklayınız. MoveL (Pick) komutu Pick_Program'da oluşacaktır.

20- Program Çağrısı penceresini açmak için  butonuna tıklayınız ve **Program Seç** tuşunu seçiniz. Sonra Pick_Program'a eklemek istediğiniz program olarak Close_Gripper_Program seçiniz ve Tamam tuşu ile onaylayınız.

21- Olay İşlemi penceresini açmak için  butonuna tıklayınız. **Hareket** başlığında **Attach object** seçiniz, sonra **Mesafeyi şu şekilde ölç: TCP vs. Nesne Konumu (liste)** seçilir ve gelen nesne listesinden Box1 seçilir ve Tamam tuşu tıklanır.

22- CTRL tuşunu kullanarak İstasyon Ağacı'nda Approach_Pick hedefini ve Pick_Program'ı seçiniz.  Doğrusal Hareket butonuna tıklayınız. MoveL (Approach_Pick) komutu Pick_Program'da oluşacaktır. Böylece Pick_Program aşağıdaki gibi tamamlanacaktır.

Resim 22.10. Pick_Program

23- İstasyon Ağacı'nda Approach_Place hedefi çift tıklayınız, sonra  butonuna tıklayarak yeni bir program ekleyiniz ve eklediğiniz programın adını F2 tuşunu kullanarak **Place_Program** olarak değiştiriniz. Set Ref ve MoveJ satırları otomatik olarak Place_Program içerisinde oluşacaktır.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

24- **CTRL** tuşunu kullanarak İstasyon Ağacı'nda Place hedefini ve Place_Program'ı seçiniz. **Doğrusal Hareket** butonuna tıklayınız. MoveL (Place) komutu Place_Program'da oluşacaktır.

25- **Program Çağrısı** penceresini açmak için butonuna tıklayınız ve **Program Seç** tuşunu seçiniz. Sonra Place_Program'a eklemek istediğiniz program olarak Open_Gripper_Program seçiniz ve Tamam tuşu ile onaylayınız.

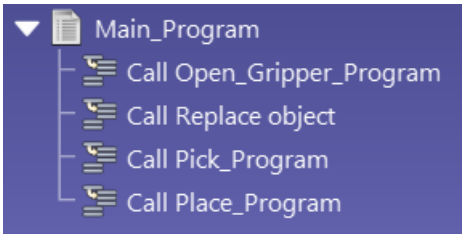
26- Olay İşlemi penceresini açmak için butonuna tıklayınız. **Hareket** başlığında **Detach object** seçiniz, sonra **Ana Parçaya ekle, Parst** Tamam tuşu tıklanarak değişiklikleri onaylayınız.

27- CTRL tuşunu kullanarak İstasyon Ağacı'nda Approach_Place hedefini ve Place_Program'ı seçiniz. Doğrusal Hareket butonuna tıklayınız. MoveL (Approach_Place) komutu Place_Program'da oluşacaktır. Böylece Place_Program aşağıdaki gibi tamamlanacaktır.

Resim 20.11 Place_Program

28- Tüm programlarımızı sırası ile yürüten Ana Programımızı yapalım. Let's create a main robot program that executes all our programs sequentially. Yeni bir program eklemek için butonuna bainız ve F2 tuşunu kullanrak ismini **Main_Program** yapınız.

29- **Program Çağrısı** penceresini açmak için butonuna tıklayınız ve **Program Seç** tuşunu seçiniz. Main_Program başlangıcında tutucunun açık olduğundan emin olmak için, Open_Gripper_Program'ı seçerek , Main_Program'a ilk onu ekleyiniz**. Tamam**'a tıklayarak onaylayınız. Replace object program, Pick_Program ve Place_Program için önceki adımları tekrarlayınız.



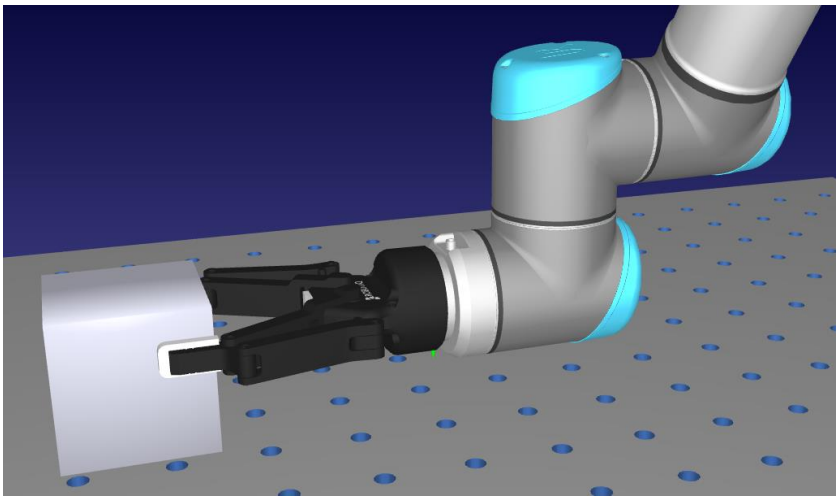
Resim 20.12. Main_Program

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

30- Main_Program simüle etmek için çift tıklayınız. Main_Program üzerinde sağ tuşa basarak açılan menüden **Loop** seçerek simülasyonu sonsuz yapınız. to make it simulate in a loop.

31- **Dosya→İstasyonu farklı kaydet'i** seçerek, çalışmayı Temrin 7 olarak kaydediniz.

32- Al ve Yerleştir uygulamasını, aşağıdaki şekilde gösterildiği gibi RobotiQ-2F-140-Gripper-Mechanism tutucusunu yatay konumlandırarak programlamayı deneyiniz ve Temrin 7-1 olarak kaydediniz.



Resim 20.13. The RobotiQ-2F-140-Gripper-Mechanism tutucusu yatay pozisyonlama

Çalışma Sorusu:

RoboDK projelerinde mekanik tutucuları açıp-kapamak için başka yöntem var mı? Lütfen bilgisayarnızdaki (C:/RoboDK/Library) kütüphanede yer alan **Example-06.a-Pick and place – Mecademic** örneğine bakınız.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

PYTHON PROGRAMLAMA DİLİ

PYTHON ÇALIŞMA SAYFALARININ YAPISI

ÇALIŞMA SAYFASI 8	
1.Çalışma Sayfasının Amacı:	• RoboDK API Programını Tanıma (Uygulama Arayüzü) • Robotik kolu bir konumdan diğerine hareket ettirmek için ilk Pyhton komutlarının öğtenilmesi
İşlem Adımları:	Köşeleri takip eden dikdörtgen oluşturmak Dört köşeli uzayda bir dikdörtgenin oluşturulması: Çalışma Sayfası 3'ün iyileştirilmesine yönelik talimatlar (Robot Programlama)
Çalışma Sorusu	

Python İpuçları:	• RoboDK API: https://robodk.com/doc/en/RoboDK-API.html#RoboDKAPI
	• Python API: https://robodk.com/doc/en/RoboDK-API.html#PythonAPI
	• İçe Aktarma Kütüphanesi: https://robodk.com/doc/en/PythonAPI/robodk.html
	• Talimatların incelenmesi:
	  ◦ Item()
	  ◦ MoveJ()
	  ◦ MoveL()

Matematik İpuçları:	Matris matematiğine giriş
Referans Dosyası (.RDK):	**Worksheet_PY_1**
ÇALIŞMA SAYFASI 9	
2.Çalışma Sayfasının Amacı:	• **Poz** verilerini anlama: aktif aracın aktif referans sistemine(referans çerçevesi) göre konumu ve yönelimi • Çeviri matrisinin nasıl uygulanacağını bilmek
İşlem Adımları:	Hedef noktadan uzayda bir kare oluşturma Başlangıç noktasından itibaren uzayda bir kare oluşturma talimatları
Çalışma Sorusu	

Python İpuçları:	Çalışma Talimatları:
	  ◦ Pose()
	  ◦ transl()

Referans Dosyası (.RDK):	**Worksheet_PY_2**
ÇALIŞMA SAYFASI 10	
3.Çalışma Sayfasının Amacı:	• Uzayda hareket etmek için çevirme ve döndürme matrisinin nasıl uygulanacağını bilmek • “for” döngüsünün nasıl kullanılacağını bilmek
İşlem adımları:	İki hedef noktası kullanarak uzayda bir beşgen oluşturmak

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

	Uzayda, merkezden ve son konumlandırma noktasından bir beşgen oluşturma talimatları
Çalışma Sorusu	
Python İpuçları:	”for” döngüsü çalışması Çalışma talimatları: • rotz()
Robotik İpuçları:	• Döndürme matrisi • Roto/çeviri dönüşümleri
Referrans Dosyası (.RDK):	**Worksheet_PY_3**
ÇALIŞMA SAYFASI 11	
4.Çalışma Sayfasının Amacı:	• Uzayda hareket etmek için çevirme ve döndürme matrisinin nasıl uygulanacağını bilmek • Genelleştirilmiş yinelemeler yapmak için “for” döngüsünün nasıl kullanılacağını bilmek • Kullanıcı tarafından veri girmek için etkileşimli iletişim kutularının nasıl oluşturulacağını bilmek
İşlem Adımları:	İki hedef noktası verilen uzayda düzgün bir çokgen oluşturma Merkezinden itibaren uzayda bir çokgen oluşturma, kenar sayısını ve yarıçapını seçme olanağına sahip olma talimatları
Çalışma Sorusu	
Python İpuçları:	Çalışma talimatları • InputDialog()
Robotik İpuçları:	• Döndürme matrisi • Roto/çeviri dönüşümleri
Referans Dosyası (.RDK):	**Worksheet_PY_4** **Worksheet_PY_4.1**
ÇALIŞMA SAYFASI 12	
5.Çalışma Sayfasının Amacı:	• Kutupsal koordinatların nasıl kullanılacağını bilmek • Yinelemeler yapmak için ”for” döngüsünün nasıl kullanıldığını bilmek • Kullanıcı tarafından veri girmek için etkileşimli iletişim kutularının nasıl oluşturulacağını bilmek
İşlem Adımları:	Merkezinden ve yarıçapından uzayda düzgün bir altıgen oluşturma Uzaydaki noktaların bir referans sisteminden diğerine yerleştirilmesini kullanarak merkezinden ve yarıçapından bir altıgen yapma talimatları
Çalışma Sorusu	
Python İpuçları:	Çalışma talimatları • Pos() • setPos()
Matematik İpuçları:	Referans sistemi çevirisi Kutupsal koordinatlar
Referans Dosyası(.RDK):	**Worksheet_PY_5** **Worksheet_PY_5.1**

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

23. RoboDK YAZILIMINDA PYTHON

Python, nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir. seviyeli bir programlama dilidir.

Girintilere dayalı basit söz dizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Söz dizimi ayrıntısına dikkat etmeye gerek kalmadığı için özellikle programlamaya yeni başlayanlar tarafından da tercih edilmektedir.

Modüler yapısı, sınıf dizgesini (sistem) ve her türlü veri alanı girişini destekler. Hemen hemen her türlü platformda çalışabilir (Unix, Linux, Mac, Windows, Amiga, Symbian). Python ile sistem programlama, kullanıcı arabirimi programlama, ağ programlama, web programlama, uygulama ve veri tabanı yazılımı programlama gibi birçok alanda yazılım geliştirebilirsiniz. (wikipedia.org)

RoboDK platformunda da phyton dilini kullanarak robot kolların istediğimiz şekilde hareket etmesini sağlayacağız.

Robolink alt modülü, RoboDK ve Python arasındaki köprüdür. RoboDK öğe ağacındaki her nesne alınabilir ve bu nesne öğesi tarafından temsil edilir. Bir öğe bir robot, bir referans çerçevesi, bir araç, bir nesne veya istasyon ağacında görünen herhangi bir öğe olabilir.

https://robodk.com/doc/en/RoboDK-API.html

classrobodk.robolink.Item(link, ptr_item=0, itemtype=- 1)

Öğe sınıfı, RoboDK istasyonundaki bir öğeyi temsil eder. Bir öğe bir robot, bir çerçeve, bir araç, bir nesne, bir hedef, istasyon ağacında görünen herhangi bir öğe olabilir. Bir öğe aynı zamanda diğer öğelerin eklenebileceği bir düğüm (alt öğeler) olarak da görülebilir. Her öğenin bir üst öğesi/düğümü vardır ve bir veya daha fazla alt öğeye/düğüme sahip olabilir.

Item(*name*, *itemtype=None*)

Bir öğeyi adına göre döndürür. Tam eşleşme yoksa son en yakın eşleşmeyi döndürür. itemtype ile ne tür bir öğe aradığınızı belirtin. Bu, 2 öğenin aynı ada ancak farklı türde olması durumunda kullanışlıdır. (değişkenleri kontrol et ITEM_TYPE_*)

Parametreler

- **name** (*str*) – öğenin adı (RoboDK istasyon ağacında gösterilen öğenin adı)
- **itemtype** (*int*)) – Alınacak öğenin türü (benzer ad eşleşmeleri varsa karışıklığı önler). ITEM_TYPE_* kullan

Mevcut öğe türleri

```
ITEM_TYPE_STATION=1          # station item (.rdk files)
ITEM_TYPE_ROBOT=2            # robot item (.robot files)
ITEM_TYPE_FRAME=3            # reference frame item
ITEM_TYPE_TOOL=4             # tool item (.tool files or tools without geometry)
ITEM_TYPE_OBJECT=5           # object item (.stl, .step, .iges, ...)
ITEM_TYPE_TARGET=6           # target item
ITEM_TYPE_PROGRAM=8          # program item (made using the GUI)
ITEM_TYPE_PROGRAM_PYTHON=10  # Python program or macro
```

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

23.1. MoveJ Komutu

MoveJ(*target*, *blocking=True*)

Bir robotu belirli bir hedefe hareket ettirir (“Move Joınt” modu). Bu fonksiyon robotun hareketlerini tamamlamasını bekler (bloklar). Bu fonksiyon Programlarda da çağrılabilecek programın sonuna kadar yeni bir hareket talimatı ekleyecektir. Bunun bir program öğesiyle kullanılması durumunda programa yeni bir doğrusal hareket talimatı eklenecektir. Programlara yeni hareket talimatları eklerken önemli not: yalnızca hedef öğeler desteklenir, pozlar desteklenmez.

Parametreler

- **target** (**Mat** , list of joints or **Item**) – Taşınacak hedef. Robot eklemleri (Nx1 veya 1xN), poz (4x4) veya hedef (öğe işaretçisi) olabilir
- **blocking** (*bool*) – Robotun hareketi tamamlamasını beklemek için True'ya ayarlayın (varsayılan=True). Engellenmeyen bir çağrı yapmak için false olarak ayarlayın. İpucu: Yanlış olarak ayarlanırsa robotun hala hareket edip etmediğini kontrol etmek için robot.Busy() işlevini kullanın.

23.2. MoveL Komutu

MoveL(*target*, *blocking=True*)

Bir robotu belirli bir hedefe hareket ettirir (“Move Linear” modu). Bu fonksiyon robotun hareketlerini tamamlamasını bekler (bloklar). Bu fonksiyon Programlarda da çağrılabilecek programın sonuna kadar yeni bir hareket talimatı ekleyecektir. Bunun bir program öğesiyle kullanılması durumunda programa yeni bir doğrusal hareket talimatı eklenecektir. Programlara yeni hareket talimatları eklerken önemli not: yalnızca hedef öğeler desteklenir, pozlar desteklenmez.

Parametreler

- **target** (**Mat** , list of joints or **Item**) – Taşınacak hedef. Robot eklemleri (Nx1 veya 1xN), poz (4x4) veya hedef (öğe işaretçisi) olabilir
- **blocking** (*bool*) – Robotun hareketi tamamlamasını beklemek için True'ya ayarlayın (varsayılan=True). Engellenmeyen bir çağrı yapmak için false olarak ayarlayın. İpucu: Yanlış olarak ayarlanırsa robotun hala hareket edip etmediğini kontrol etmek için robot.Busy() işlevini kullanın.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 8

KÖŞELERİ TAKİP EDEN DİKDÖRTGEN OLUŞTURMAK

Bu çalışma sayfasının amaçları şunlardır:

- RoboDK API Programını Tanıma (Uygulama Arayüzü)
- Robotik kolu bir konumdan diğerine hareket ettirmek için ilk Pyhton komutlarının öğtenilmesi.

İşlem Basamakları:

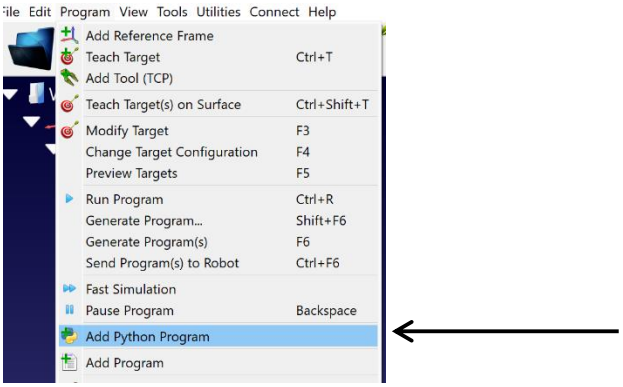
1- Masaüstündeki program kısayoluna çift tıklayın. 

2- Oluşturulan **Worksheet 3** dosyasını açmak için  butonuna basın.

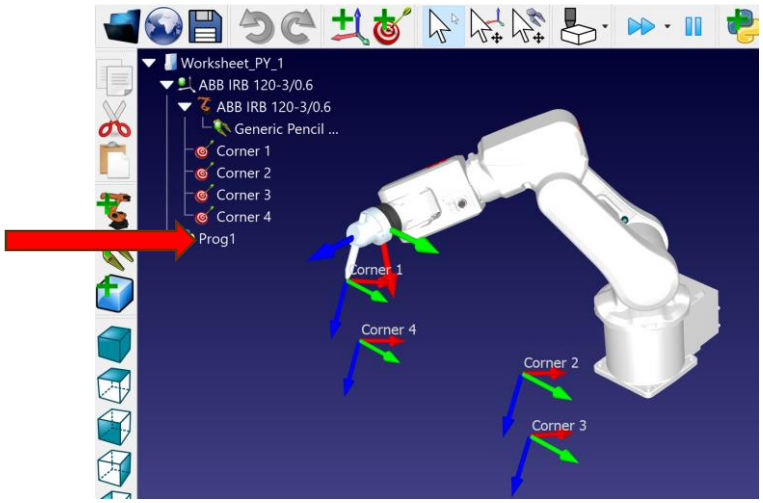
3- İstasyonu **Worksheet_PY_1** olarak kaydedin

4- Dikdörtgene sağ tıklayın ve silin.

5- Program menüsünden→Add Python Program:



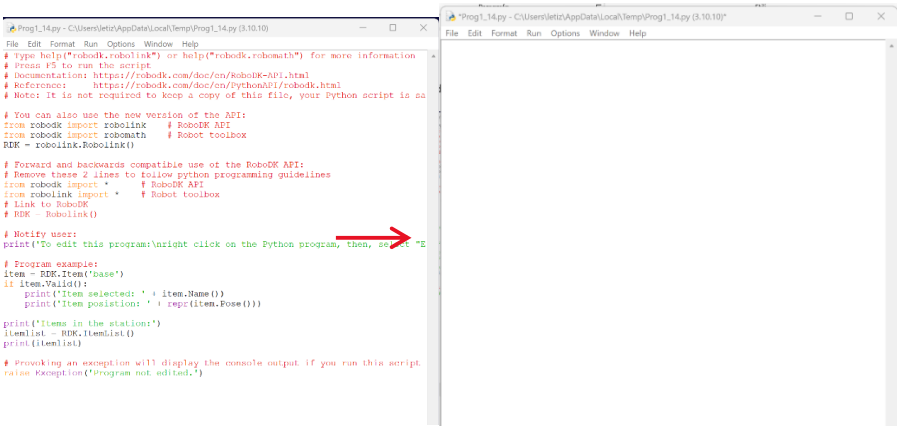
Diğer bir yol olarak Araç Çubuğu menüsündeki simgeyi kullanabilirsiniz.



TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

6- Prog1' e **sağ** tıklayıp **Python** kodlarını düzenle diyoruz.

7- Yeni bir pencere açılacaktır, metni seçin ve tümünü silin:

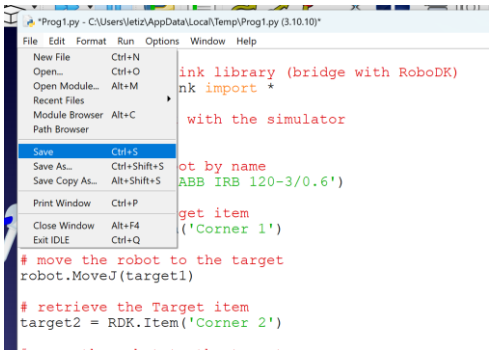


8- Python kodlarını yazmaya başlıyoruz.

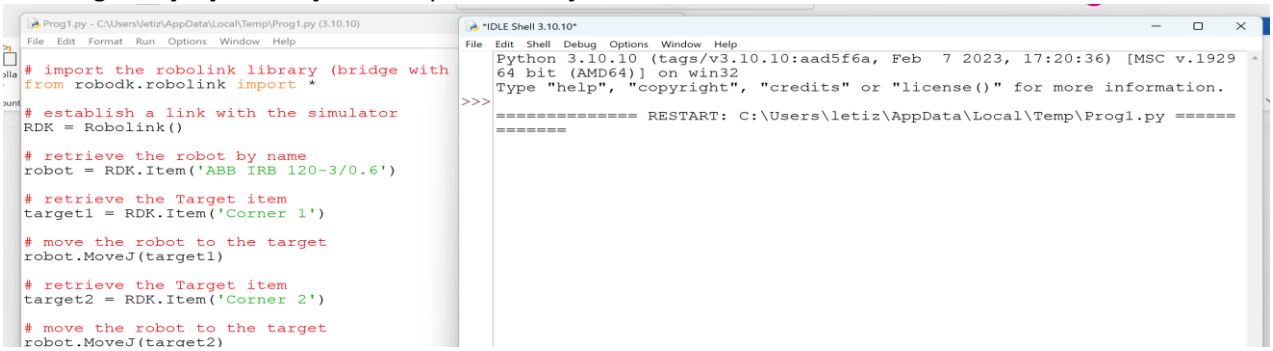
```
# import the robolink library (bridge with RoboDK)
from robodk.robolink import *
# establish a link with the simulator
RDK = Robolink()
# retrieve the robot by name
robot = RDK.Item('ABB IRB 120-3/0.6')
# retrieve the Target item
target1 = RDK.Item('Corner 1')
# move the robot to the target 1
robot.MoveJ(target1)
# retrieve the Target item
target2 = RDK.Item('Corner 2')
# move the robot to the target 2
robot.MoveJ(target2)
# retrieve the Target item
target3 = RDK.Item('Corner 3')
# move the robot to the target 3
robot.MoveJ(target3)
# retrieve the Target item
target4 = RDK.Item('Corner 4')
# move the robot to the target 4
robot.MoveJ(target4)
# retrieve the Target item
target1 = RDK.Item('Corner 1')
# move the robot to the target 1
robot.MoveJ(target1)
```

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

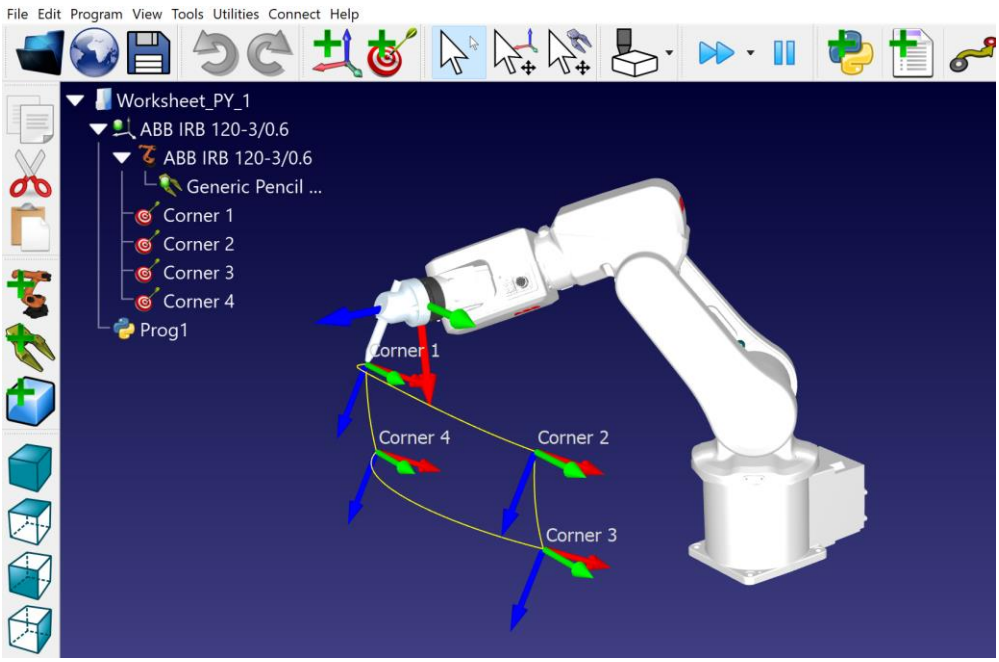
9- Programı kaydedin:



10- **Run-Run Module' ü** seçiniz,
Programı çalıştırmak için kabuk penceresi açılır:



11- Ana ekrana geri dönün, Prog1'e çift tıklayarak robot, **Worksheet-3** de açıklanan yolu takip ederek bir dikdörtgenin ana hatlarını tanımlar:



12- Lütfen aynı adımları Universal Robot UR10 için tekrarlayınız ve bu istasyonu **Whorksheet_PY1.1** adı altında kaydediniz.

13- Lütfen programı MoveJ() yerine MoveL() komutunu kullanarak değiştirin.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

24. PYTHON DEĞİŞKENLERİ VE DEĞİŞKEN KURALLARI

Python'da değişken tanımlamak için bir komutu yoktur. Değişken adını bildirip, içerisine değer atamanız yeterlidir.

Fakat değişken adı verirken dikkat etmemiz gereken bazı kurallar var. Bunlar:

a) Değişken isimleri büyük küçük harf duyarlıdır. Örneğin; değişken isminin **adres** ya da **Adres** olması bu değişkenlerin farklı iki değişken olduğunu gösterir.

b) Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir. Ancak sayılar başa gelmez. Örneğin sayi1 doğru bir isimlendirmeyken **1sayi** doğru bir isimlendirme değildir.

c) Değişken isimlendirilirken alt tire (_) kullanılabilir. Ancak boşluk ve diğer özel karakterler (?,%,!, ., + vb.) kullanılmaz. Örneğin **ev adresi** ya da **kimlik%no** gibi değişken isimleri kurallara aykırı olduğundan hataya neden olacaktır.

d) Değişken isimlendirilirken phyton programlama dilindeki komutları örneğin if, for, true vb. Ifadeleri kullanılmamalıdır.

Bu kurallar çerçevesinde aşağıda doğru tanımlanmış bazı değişken örnekleri görülmektedir:

```
live_city=“Izmir”

exam_grade=80

rate3=5.7

RDK = Robolink()

robot = RDK.Item('ABB IRB 120-3/0.6')

target1 = RDK.Item('Corner 1')
```

Yukarıda tanımlana; yasadigi_sehir, sinav_notu, oran3, RDK, robot ve target1 birer değişkendirler ve kendilerine birer değer atanmıştır.

24.1. Python Operatörleri

Operatörler, değişkenler ve veriler üzerinde işlem yaparak yeni değerler üretilmesini sağlayan sembollerdir. Python programlama diline yeni başlayanlar için aritmetiksel, atama, karşılaştırma, mantıksal ve kimlik operatörleri öğrenmek son derece önemlidir.

24.1.1.Arithmetic Operatörler

Değişkenler içerisinde saklanılan değerler üzerinde matematiksel işlem yapmak için kullanılan operatörlerdir.

Operatör	Tanımı	Örnek
+	Toplama	a+b
-	Çıkarma	a-b
*	Çarpma	a*b
/	Bölme	a/b
%	Mod alma (Bir sayının diğer sayıya bölümünden kalan)	a%b

**	Kuvvet alma (ab)	a**b
//	Tam sayı bölme (Bölme işleminde sadece tam kısım alınır.)	a//

24.1.2. Atama Operatörleri

Bir değişkendeki değerin başka bir değişkene aktarılabilmesi için ya da bir işlem sonucunun başka bir değişkene aktarılabilmesi için kullanılan operatörlerdir.

Operatör	Örnek	Açıklama
=	a=2	a değişkenine 2 değeri atanmıştır.
+=	a+=2	a değişkenine 2 değerini ekleyerek yine a değişkenine atanmıştır. a=a+2 anlamına gelmektedir.
-=	a-=2	a değişkeninden 2 değeri çıkarılarak yine a değişkenine atanmıştır. a=a-2 anlamına gelmektedir.
*=	a*=2	a değişkeni 2 ile çarpılarak yine a değişkenine atanmıştır. a=a*2 anlamına gelmektedir.
/=	a/=2	a değişkeni 2 değerine bölünerek yine a değişkenine atanmıştır. a=a/2 anlamına gelmektedir.
%=	a%=2	a değişkenin 2 değeri ile modu alınarak yine a değişkenine atanmıştır. a=a%2 anlamına gelmektedir.

24.1.3. Karşılaştırma Operatörleri

İki değişkenin değerini karşılaştırıp ona göre işlem yaptırmak istendiğinde kullanılan operatörlerdir.

Operatör	Tanımı	Örnek
==	Eşittir	a==b
!=	Eşit değildir	a!=b
<	Küçüktür	a<b
>	Büyüktür	a>b
<=	Küçük eşittir	a<=b
>=	Büyük eşittir	a>=

24.1.4. Mantıksal Operatörler

İki ya da daha fazla değişkenin değerini kontrol ettirerek program akışına karar vermek için kullanılan operatörlerdir.

Operatör	Örnek	Açıklama
and	a<3 and b>=5	İki veya daha fazla şartın tamamının doğru olması durumunda True değerini döndürür. Buradaki örnekte a değişkeni 3'ten küçük ve b değişkeni 5'e eşit ya da 5'ten büyük olursa True değeri döndürülür.
or	a<3 or b>4	İki veya daha fazla şartın en az birinin doğru olması durumunda True değerini döndürür. Buradaki örnekte a değişkeninin 3'ten küçük olması ya da b değişkenin 4'ten büyük olması True değeri döndürmek için yeterlidir.
not	not(a<3)	Durumu tersine çevirmek (True ise False; False ise True) için kullanılır. Buradaki örnekte parantez içindeki mantıksal sınamanın sonucu tersine çevrilir. İfadenin not komutu olmadan yazıldığında true döndüreceği varsayıldığında bu haliyle false döndürecektir.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

24.2. Python Veri Türleri

Python'da genel olarak string (metinsel), numbers (sayısal),boolean, list (liste), tuple (demet), dictionary (sözlük) ve set (küme) gibi veri tipleri bulunmaktadır. Başlangıç aşamasında bunlardan en fazla kullanacağımız değişken türleri aşağıda yer almaktadır.

24.2.1. String (Metinsel) Veri Tipi
Tek ya da çift tırnak içlerine yazılan karakter dizileridir. Burada karakter harf (t,c), rakam (1,9,2,3) ya da özel semboller (&,/) olabilir. String veri tipleri tek ya da çift tırnak içinde yazılır

```
name="Mert"
surname="Yılmaz"
```

24.2.2. Numbers (Sayısal) Veri Tipi
Sayısal verileri tutan veri tiplerine verilen addır. Python'da sayısal veri tipleri genel olarak int, float ve complex veri tipleridir.

```
sayi=1919
pi_degeri=3.14
```

24.2.3. Python Listeleri
Farklı verilerin bir dizi hâlinde tutulduğu koleksiyonlara liste adı verilir. int, float, string gibi farklı veri tiplerini tek bir listede tutabilirsiniz. Birden fazla veriyi sıralı ve değiştirilebilen bir yapıda tutmak için listeler kullanılır. Python programlama dilinde listeler iki köşeli parantez ile tanımlanmaktadır.

```
ilk_liste=["Ankara", 312, 0.6]
```

24.2.4 Sözlük (dict)
Anahtar-değer çiftlerini saklar. Anahtarlar benzersiz olmalıdır. {} içinde anahtar:değer şeklinde tanımlanır.

```
thisdict = {
  "brand": "BMW
  "model": "X3",
  "year": 2020
}
```

➔ **Python in RoboDK:**

Pose()

Bir nesnenin, hedefin veya referans çerçevesinin göreli pozunu döndürür. Örneğin, bir nesnenin, hedefin veya referans çerçevesinin esasına (ağaçta iliştirildiği öğeye) göre konumu. Robot öğeleri için bu işlev, etkin aracın etkin referans çerçevesine göre pozunu döndürür.

Pozu **Mat** olarak döndürür.

İpucu: Pozu bir robot markasına özel olarak XYZABC'ye (mm olarak XYZ konumu ve derece olarak ABC yönelimi) dönüştürmek için robodk modülündeki bir Pose_2_* işlevini (**robomath.Pose_2_KUKA** gibi) kullanın.

robodk.robomath.transl(tx, ty=None, tz=None)

Bir çeviri matrisi (mm) döndürür.

$$T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Parameters

- **tx** (*float*) – X ekseni boyunca Çevirme
- **ty** (*float*) – Y ekseni boyunca Çevirme
- **tz** (*float*) – Z ekseni boyunca Çevirme

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 9

HEDEF NOKTADAN UZAYDA KARE OLUŞTURMA

Bu çalışma sayfasının amaçları şunlardır:

- **Poz** verilerini anlama: aktif aracın aktif referans sistemine(referans çerçevesi) göre konumu ve yönelimi
- Çeviri matrisinin nasıl uygulanacağını bilmek

İşlem Adımları:

1- Masaüstündeki program simgesine çift tıklayarak programı açın. 

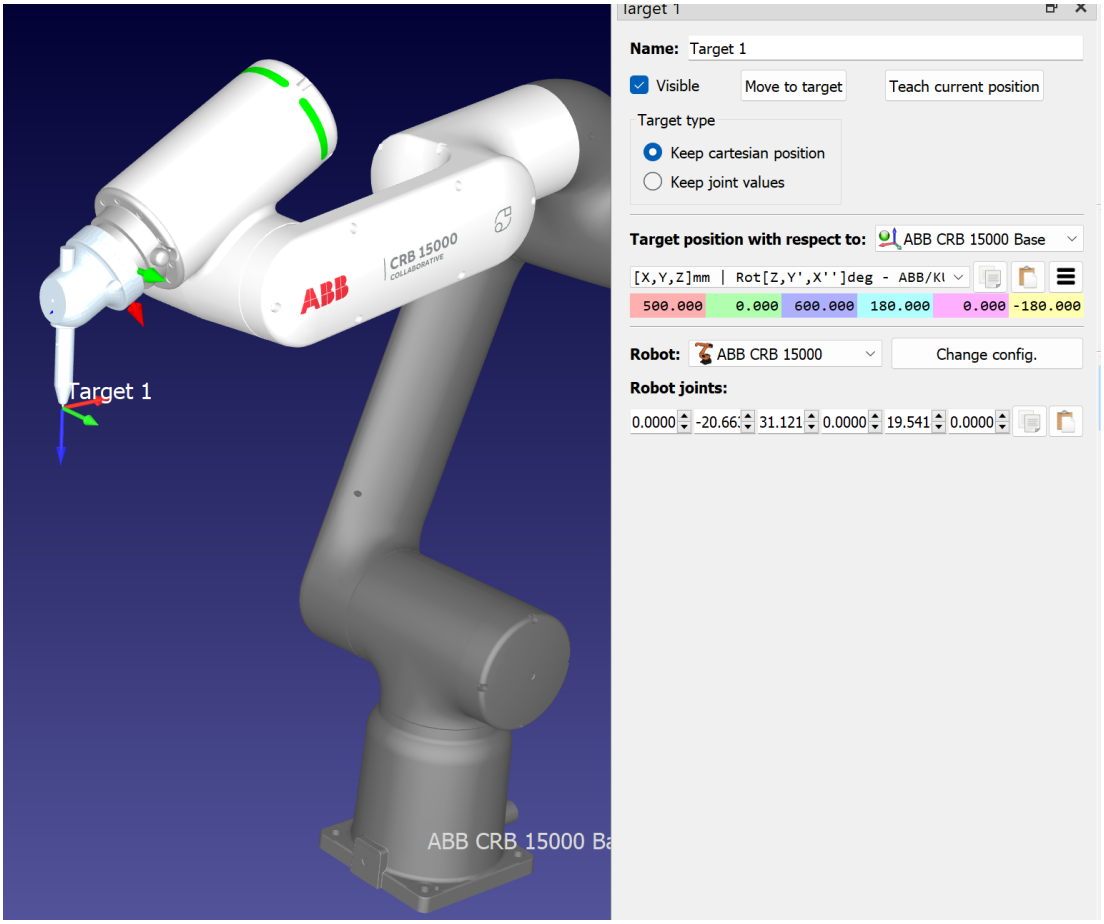
2- Yeni bir istasyon oluşturun ve Robot kütüphanesi **(CTRL+SHIFT+O**) menüsünden **ABB CRB 1500'i** seçin.

3- Kütüphaneden bir "**Generic pencil tool"** aracı seçin.

4- İstasyonu **Worksheet_PY_2** olarak kaydedin.

5- Aşağıdaki konfigürasyonla bir hedef noktası oluşturun:

Set **Target 1** to: 500, 0, 600, 180, 0, -180



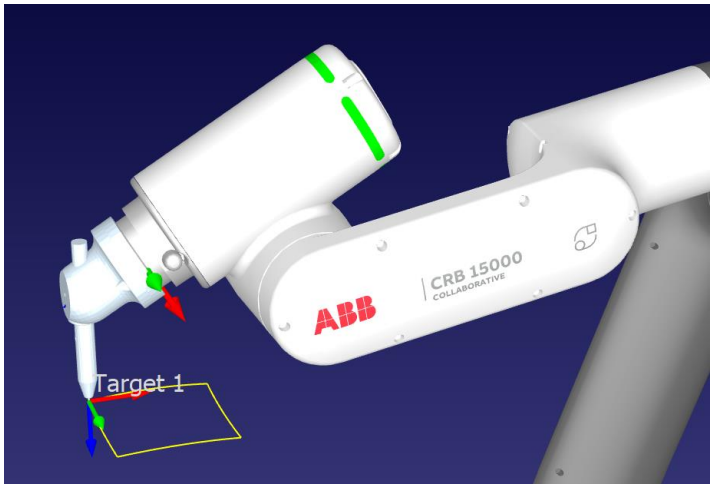
6- Python programını ikonla veya Program menüsüyle ekleyin.

7- Aşağıdaki programı yazın:

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

```
# import the robolink library (bridge with RoboDK)
from robodk.robolink import *
# establish a link with the simulator
RDK = Robolink()
# retrieve the robot by name
robot = RDK.Item(' ABB CRB 15000')
#create home position in the Target 1
home=RDK.Item('Target 1')
# import the robotics toolbox
from robodk.robomath import *
# create the vertex1 in a position 100 mm along the X axis of the tool
#with respect to the home position
vertex_1 = home.Pose()*transl(100,0,0)
# move to the vertex_1
robot.MoveJ(vertex_1)
# create the vertex2 in a position 100 mm along the Y axis of the tool
#with respect to the vertex1 position
vertex_2 = vertex_1*transl(0,100,0)
# move to the vertex_2
robot.MoveJ(vertex_2)
# create the vertex3 in a position 100 mm along the -X axis of the tool
#with respect to the vertex2 position
vertex_3 = vertex_2*transl(-100,0,0)
# move to the vertex_3
robot.MoveJ(vertex_3)
# create the vertex4 in a position 100 mm along the -Y axis of the tool
#with respect to the vertex3 position
vertex_4 = vertex_3*transl(0,-100,0)
# move to the vertex_4
robot.MoveJ(vertex_4)
```

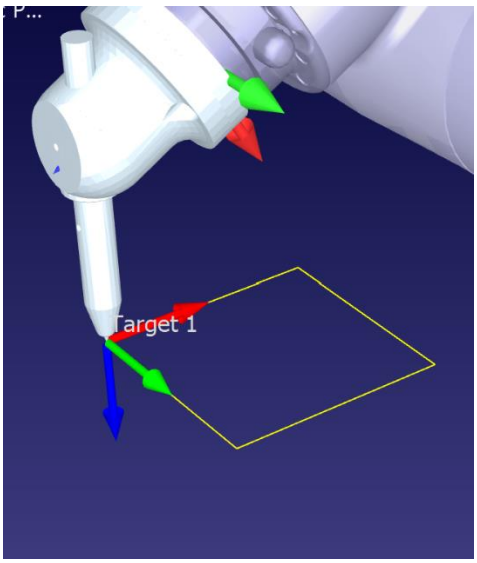
8- Robot, aktif araçla kenarı 100 olan bir kareyi tarif ediyor:



TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

9- Lütfen kodu, kenarları 100 ve 50 olan bir dikdörtgen olacak şekilde değiştirin

10- Lütfen programı MoveJ() yerine MoveL() komutunu kullanarak değiştirin:



TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

25. KARAR VE DÖNGÜ YAPILARI

Bir program akışında programın çalışma yönünü belirlemek için gerektiği yerlerde Döngülerden ve karar yapılarından faydalanılmaktadır.

25.1. Karar Yapısı – If..elif

İki ya da daha fazla değişkenin değerlerini control ettirerek kodların istenilen şartlara göre dallanıp şarta uygun kodların çalışmasını istiyorsak karar yapılarını kullanırız.

Örnek 1:

```
a = 33
b = 200
if b > a:
  print("b a'dan büyüktür")
```

Örnek 2:

```
a = 33
b = 33
if b > a:
  print("b a'dan büyüktür")
elif a == b:
  print("a and b eşittir")
```

25.2. While Döngüsü

Bir koşul doğru olduğu sürece bir dizi ifadeyi çalıştırabiliriz.

Örnek 1: i 6'dan küçük olduğu sürece ekrana yazdıran programın kodları

```
i = 1
while i < 6:
  print(i)
  i += 1
```

Program Çıktısı

```
1
2
3
4
5
```

25.3. For Döngüsü

Bir for döngüsü, bir dizi (yani bir liste, bir tanımlama grubu, bir sözlük, bir küme veya bir dize) üzerinde yineleme yapmak için kullanılır.

Örnek Her meyveyi bir meyve listesinde yazdırın:

```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

Program Çıktısı

```
apple
banana
cherry
```

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

➔ **Python in RoboDK:**

robodk.robomath.rotz(rz)

Returns a rotation matrix around the Z axis (radians)

$$R_x(\theta) = \begin{bmatrix} c_\theta & -s_\theta & 0 & 0 \\ s_\theta & c_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Parameters

ry (*float*) – rotation around Y axis in radians

TEMRİN 10

İKİ HEDEF NOKTASI KULLANARAK UZAYDA BİR BEŞGEN OLUŞTURMAK

Bu çalışma sayfasının amaçları şunlardır:

- Uzayda hareket etmek için çevirme ve döndürme matrisinin nasıl uygulanacağını bilmek
- "for" döngüsünün nasıl kullanılacağını bilmek

İşlem Adımları:

1- Masaüstündeki RoboDK programının kısayol simgesine çift tıklayın. 

2- Yeni bir istasyon oluşturun, Robot kütüphanesi (CTRL+SHIFT+O) menüsünden **KUKA KR 6 R900 sixx Base** robotunu seçin.

3- Kütüphaneden **weld_gun'** u seçin**.**

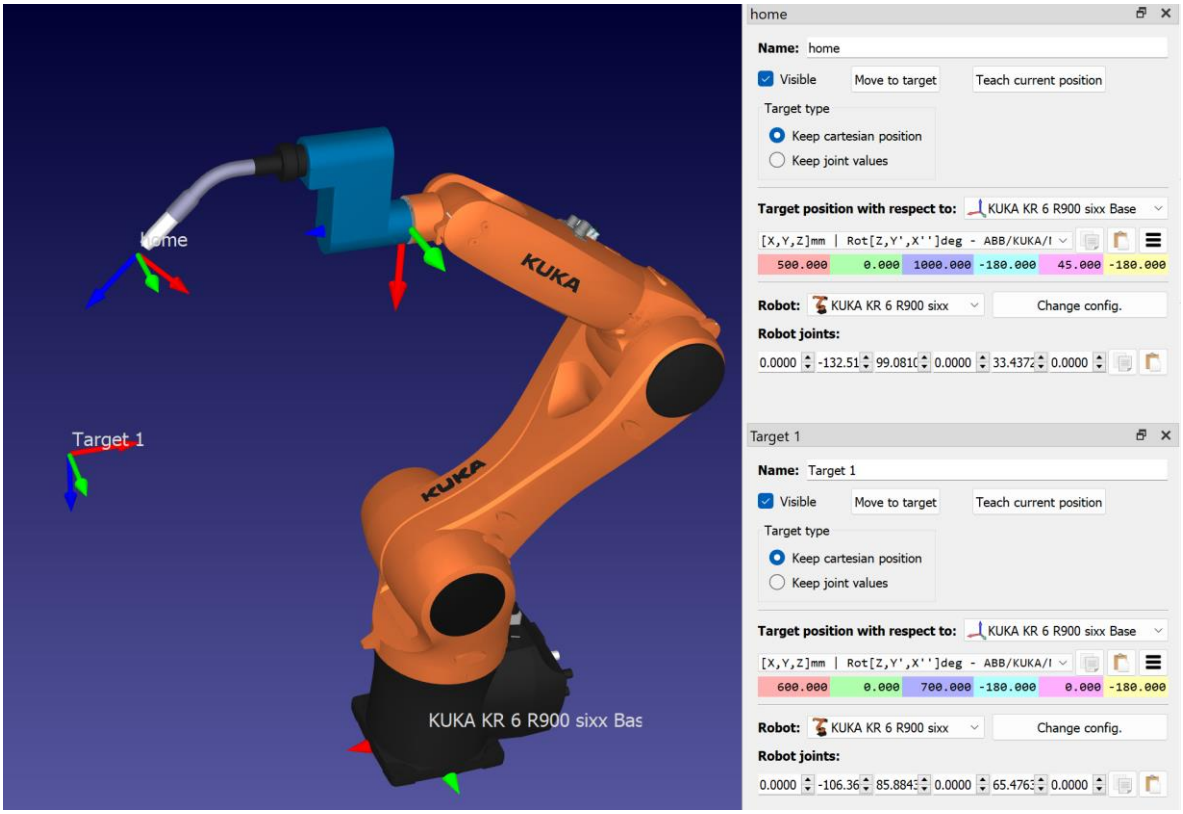
4- İstasyonu **Whorksheet_PY_3** olarak kaydedin.

5- Aşağıdaki konfigürasyonla bir hedef noktası oluşturun:

Set **Target 1** to: 600, 0, 700, -180, 0, -180

6- Aşağıdaki konfigürasyonla başka bir hedef noktası oluşturun:

Set **home** to: 500, 0, 1000, -180, 45, -180



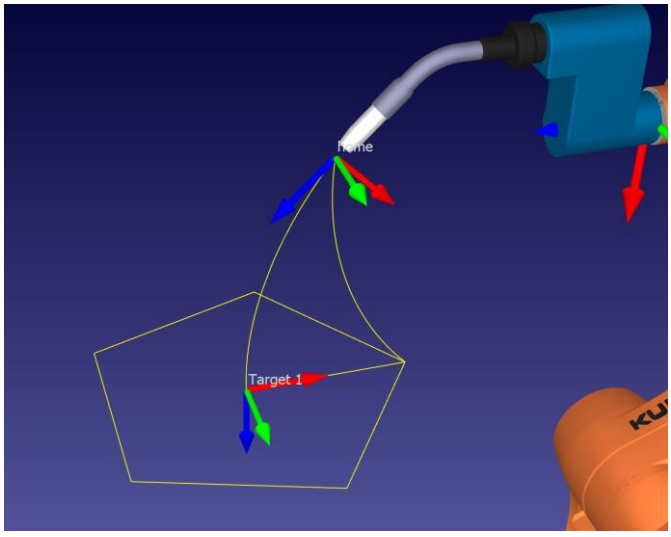
7- Python programını ikonla veya Program menüsüyle ekleyin.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

8- Aşağıdaki kodları yazın:

```
#API to communicate with RoboDK
#Import the robolink library (bridge with RoboDK)
from robodk.robolink import*
# basic matrix operations
from robodk import *
# Any interaction with RoboDK must be done through Robolink()
# estabilish a link with the simulator
RDK = Robolink()
# get the robot item, (retrieve the robot by name):
robot = RDK.Item('KUKA KR 6 R900 sixx')
# get the home target:
home = RDK.Item('home')
# get the pentagon center:
center = RDK.Item('Target 1')
# get the pose of the ref (4x4 matrix):
poseref= center.Pose()
# move robot to home than to center
robot.MoveJ(home)
robot.MoveJ(center)
# make a pentagon :
for i in range(6):
    ang = i*2*pi/5 #angle: 0,60,120...
    posei = poseref*rotz(ang)*transl(200,0,0)*rotz(-ang)
    robot.MoveL(posei)
# move back to home:
robot.MoveJ(home)
```

9- Robot beşgen şeklinde bir yol çizecek ve başlangıç pozisyonuna dönecektir.



TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

26.InputDialog KULLANIMI

'Ok' ve 'Cancel' düğmeleriyle beraber bir giriş iletişim kutusu gösterilecektir.

```
robodk.robodialogs.InputDialog(msg, value, title=None, default_button=False, default_value=None, embed=False, actions=None, *args, **kwargs)
```

Giriş alanı otomatik olarak aşağıdaki very türleri için oluşacaktır :

- Base types: bool, int, float, str
- list or tuple of base types
- dropdown formatted as [int, [str, str, …]]. e.g. [1, [‘Option #1’, ‘Option #2’]] where 1 means the default selected option is Option #2.
- dictionary of supported types, where the key is the field’s label. e.g. {‘This is a bool!’ : True}.
- Temel türler: bool, int, float, str
- Temel türlerin listesi veya dizisi
- [int, [str, str, …]] olarak biçimlendirilmiş açılır menü. Örneğin. [1, ['option #1', 'option #2']] burada 1, varsayılan olarak seçilen seçeneğin Option #2 olduğu anlamına gelir.
- Desteklenen türlerin sözlüğü; burada anahtar alanın etiketidir. Örneğin. {'Bu çok harika!' : True}.

Parametreler

- **msg** (*str*) – Kullanıcıya mesaj (ne girileceğini açıklar)
- **value** – I girişin başlangıç değeri (desteklenen türlere bakın)
- **title** (*embed*) – Pencere başlığı, isteğe bağlı
- **default_button** – Girişi varsayılana, varsayılanı false değerine yeniden başlatmak için bir düğme göster
- **default_value** – Geri yüklenecek varsayılan değerler. Sağlanmadığı takdirde orijinal değerler kullanılacaktır
- **title** – Pencereyi RoboDK'nin içine yerleştirin, varsayılan değer false'tur
- **actions** (*list of tuples of str, callable*) – Düğme olarak eklenecek isteğe bağlı eylem callback listesi, [(str, callable), …] olarak biçimlendirilmiştir. e.g. [(“Button #1”, action_1), (“Button #2”, action_2)]

Geri Dönüşler

Kullanıcı “OK” ‘ i tıkladığında kullanıcı girişi None olur.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Geri Dönüş Türü

Desteklenen türlere bakın

Örnek:

```
print(InputDialog('This is as input dialog.\n\nEnter an integer:', 0))
print(InputDialog('This is as input dialog.\n\nEnter a float:', 0.0))
print(InputDialog('This is as input dialog.\n\nEnter text:', ''))
print(InputDialog('This is as input dialog.\n\nSet a boolean:', False))
print(InputDialog('This is as input dialog.\n\nSelect from a dropdown:', [0, ['RoboDK is the best', 'I love RoboDK!', "Can't hate it, can I?"]]))
print(InputDialog('This is as input dialog.\n\nSet multiple entries:', {
   'Enter an integer:': 0,
   'Enter a float:': 0.0,
   'Set a boolean:': False,
   'Enter text:': '',
   'Select from a dropdown:': [0, ['RoboDK is the best!', 'I love RoboDK!', "Can't hate it, can I?"]],
   'Edit int list:': [0, 0, 0],
   'Edit float list:': [0., 0.],
}))
```

TEMRİN 11

İKİ HEDEF NOKTASI VERİLEN UZAYDA DÜZGÜN BİR ÇOKGEN OLUŞTURMA

Bu çalışma sayfasının amaçları şunlardır:

- Uzayda hareket etmek için çevirme ve döndürme matrisinin nasıl uygulanacağını bilmek.
- Genelleştirilmiş yinelemeler yapmak için "for" döngüsünün nasıl kullanılacağını bilmek.
- Kullanıcı tarafından veri girmek için etkileşimli iletişim kutularının nasıl oluşturulacağını bilmek.

İşlem Basamakları:

1- Masaüstündeki RoboDK programının simgesine çift tıklayın. 

2- **Whorksheet_PY_3'** yi aç.

3- **Worksheet_PY_4** olarak kaydet.

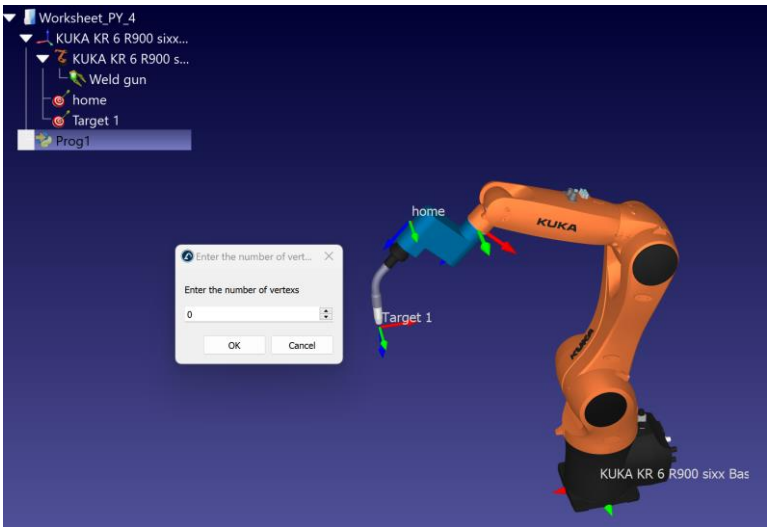
4- Python programını açın ve kodu bu şekilde değiştirin:

```
#API to communicate with RoboDK
#Import the robolink library (bridge with RoboDK)
from robodk.robolink import*
# basic matrix operations
from robodk import *
# Any interaction with RoboDK must be done through Robolink()
# estabilish a link with the simulator
RDK = Robolink()
# get the robot item, (retrieve the robot by name):
robot = RDK.Item('KUKA KR 6 R900 sixx')
# get the home target:
home = RDK.Item('home')
# get the pentagon center:
center = RDK.Item('Target 1')
# get the pose of the ref (4x4 matrix):
poseref= center.Pose()
# move robot to home than to center
robot.MoveJ(home)
robot.MoveJ(center)
#Through a dialog window, define the number of vertexs of the polygon
nvertexs= InputDialog('Enter the number of vertexs', 0 , )
nvertexs= int(nvertexs)
# make a poligon around center :
for i in range(nvertexs+1):
    ang = i*2*pi/nvertexs #angle: 0,60,120...
    posei = poseref*rotz(ang)*transl(200,0,0)*rotz(-ang)
    robot.MoveL(posei)
# move back to home:
```

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

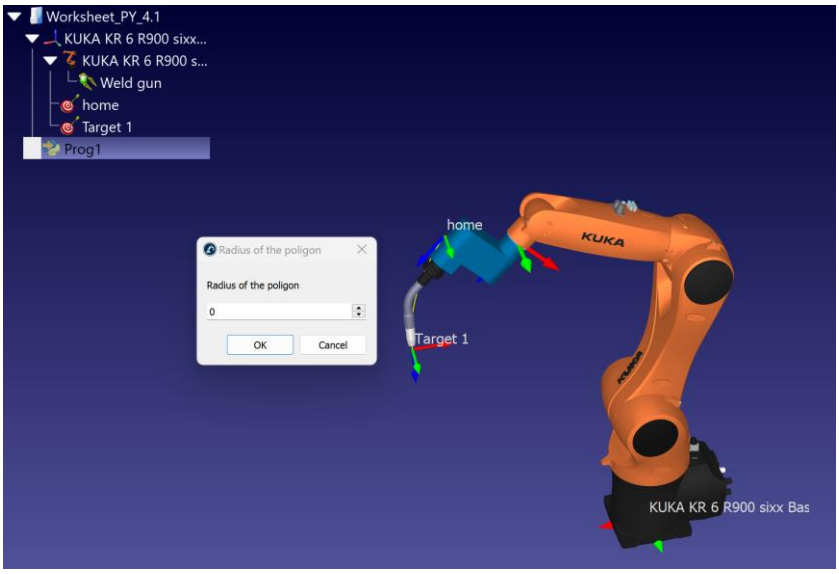
```
robot.MoveJ(home)
```

5- Program, bir iletişim kutusu aracılığıyla çokgenin kenar sayısını seçmenize izin verecektir:



6- Lütfen poligonun yarıçapını soran pencereyi ekleyerek programı değiştirin ve **Worksheet_PY_4.1** olarak kaydedin:

```
#Through a dialog window, define the radius of the poligon
radius= InputDialog('Radius of the poligon', 0 , )
radius= int(radius)
```



Sorular:

Çokgenin kenar sayısı arttıkça ne olur?

Çokgenin yarıçapı için herhangi bir değer girmek mümkün mü?

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

27.POSITION COMMANDS

Pos()

Returns the position of a pose (assumes that a 4x4 homogeneous matrix is being used)

setPos(*newpos*)

Sets the XYZ position of a pose (assumes that a 4x4 homogeneous matrix is being used)

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

TEMRİN 12

MERKEZİNDEN VE YARIÇAPINDAN UZAYDA DÜZGÜN BİR ALTIGEN OLUŞTURMA

Bu çalışma safasının amacı:

- Kutupsal koordinatların nasıl kullanılacağını bilmek
- Yinelemeler yapmak için ”for” döngüsünün nasıl kullanıldığını bilmek
- Kullanıcı tarafından veri girmek için etkileşimli iletişim kutularının nasıl oluşturulacağını bilmek

İşlem Adımları:

1- Masaüstündeki RoboDK programının simgesine çift tıklayın. 

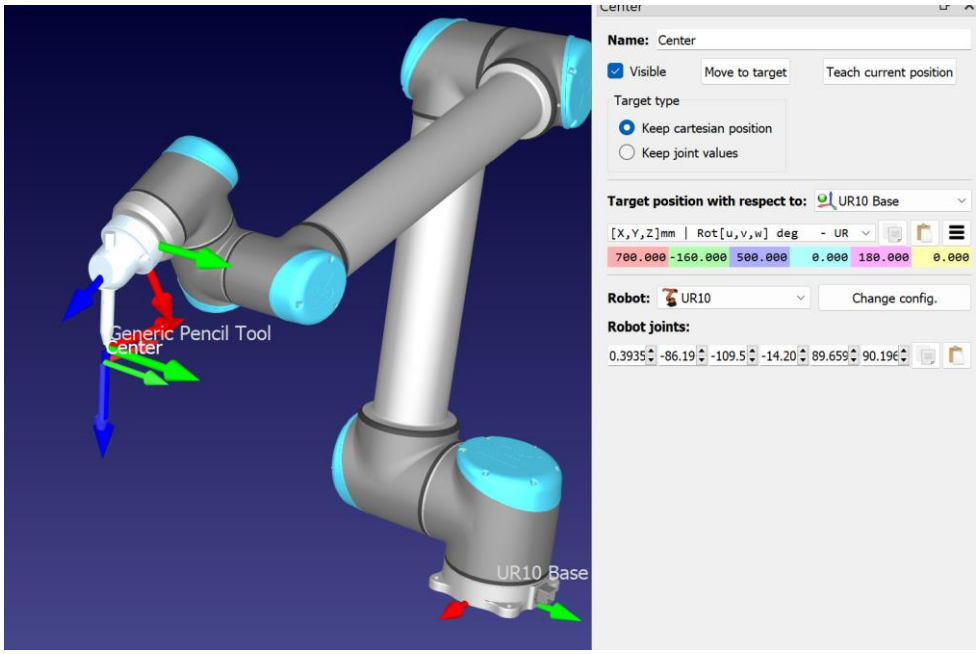
2- Yeni bir istasyon oluşturun, Robot kütüphanesinden **UR10** robotunu seçiniz.

3- Kütüphaneden **Generic pencil tool** aracını seçiniz.

4- İstasyonu **Worksheet_PY_5** olarak kaydediniz.

5- Sonraki yapılandırma ve merkeze göre bir hedef noktası oluşturun. Aşağıdaki konfigürasyona sahip bir hedef noktası oluşturun ve adını “center” olarak değiştirin:

Set **Center** to: 700, -160, 500, 0, 180, 0



6- Python programını ikonla veya Program menüsüyle ekleyin.

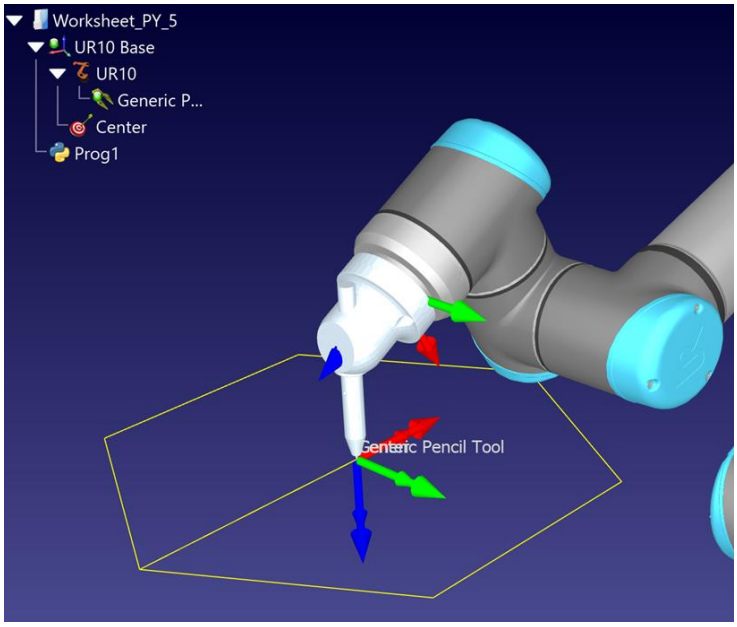
7- Aşağıdaki kodları yazın:

```
# Draw a hexagon around a point called center
```

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

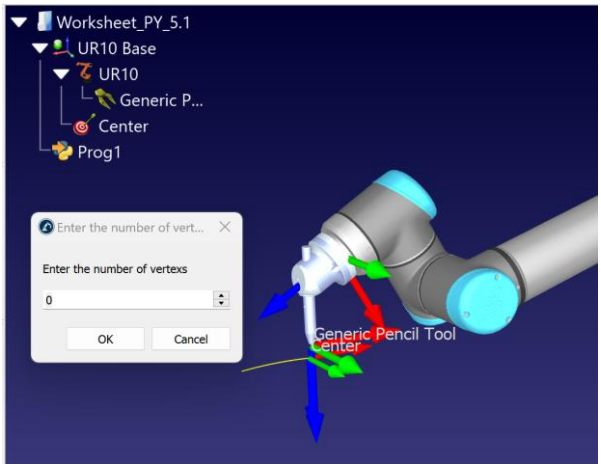
```
#API to communicate with RoboDK
#Import the robolink library (bridge with RoboDK)
from robolink import *
# basic matrix operations
# Math toolbox for robots
from robodk import *
# Start the RoboDK API:
RDK = Robolink()
 # Get the robot (first robot found):
robot = RDK.Item('', ITEM_TYPE_ROBOT)
# Get the reference target by name:
center = RDK.Item('Center')
center_pose = center.Pose()
xyz_ref = center_pose.Pos()
 # Move the robot to the reference point:
robot.MoveJ(center)
 # Draw a hexagon around the reference target:
for i in range(7):
   # Angle = 0,60,120,...,360
   ang = i*2*pi/6
   # Polygon radius
   R = 200
   # Calculate the new position:
   x = xyz_ref[0] + R*cos(ang) # new X coordinate
   y = xyz_ref[1] + R*sin(ang) # new Y coordinate
   z = xyz_ref[2]             # new Z coordinate
   center_pose.setPos([x,y,z])
      # Move to the new target:
   robot.MoveL(center_pose)
 # Move back to the reference target:
robot.MoveL(center)
```

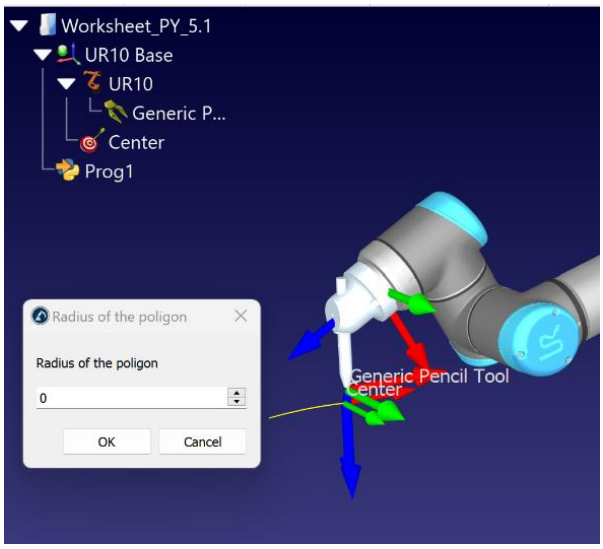
Program, flanşa bağlı aracın düzenli bir çokgen tanımlamasına olanak tanır:

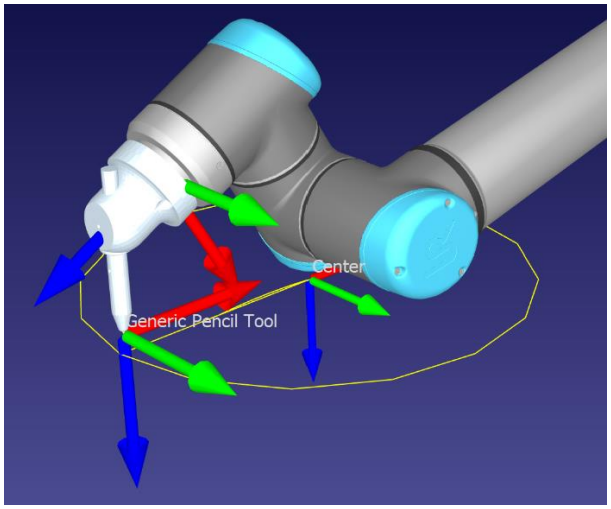


8- Çokgenin kenar sayısını ve yarıçapını soran etkileşimli iletişim kutusunu ekleyerek programı değiştirin ve **Worksheet_PY_5.1** olarak kaydedin:

```
#Through a dialog window, define the number of vertexs of the polygon
nvertexs= InputDialog('Enter the number of vertexs', 0 , )
nvertexs= int(nvertexs)
#Through a dialog window, define the radius of the poligon
radius= InputDialog('Radius of the poligon', 0 , )
radius= int(radius)
 # Draw a hexagon around the reference target:
for i in range(nvertexs+1):
   # Angle = 0,60,120,...,360
   ang = i*2*pi/nvertexs
   # Calculate the new position:
   x = xyz_ref[0] + radius*cos(ang) # new X coordinate
   y = xyz_ref[1] + radius*sin(ang) # new Y coordinate
   z = xyz_ref[2]             # new Z coordinate
   center_pose.setPos([x,y,z])
    # Move to the new target:
   robot.MoveL(center_pose)
```







Çalışma Sorusu:

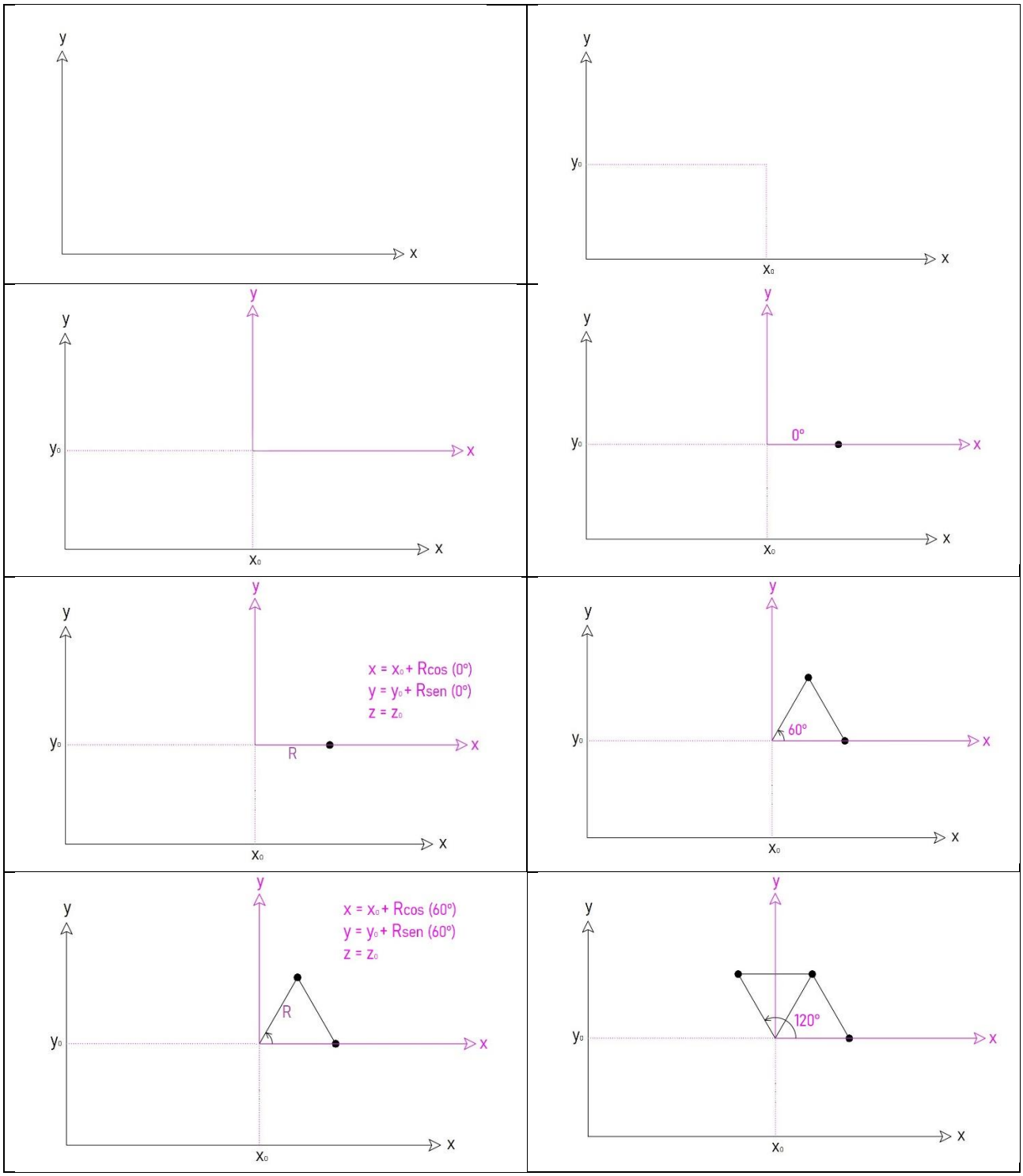
Uzaydaki referans sisteminin değişimini inceleyin

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

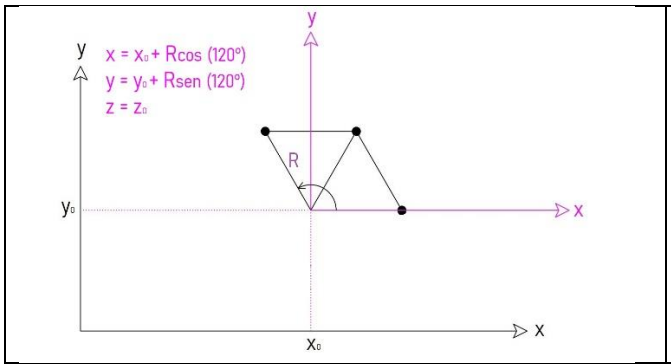
Matematik İpuçları:

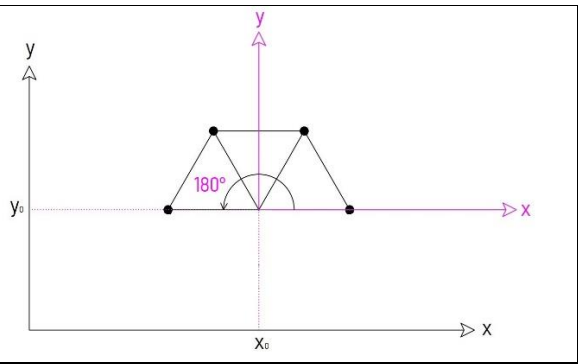
Kutupsal koordinatlar

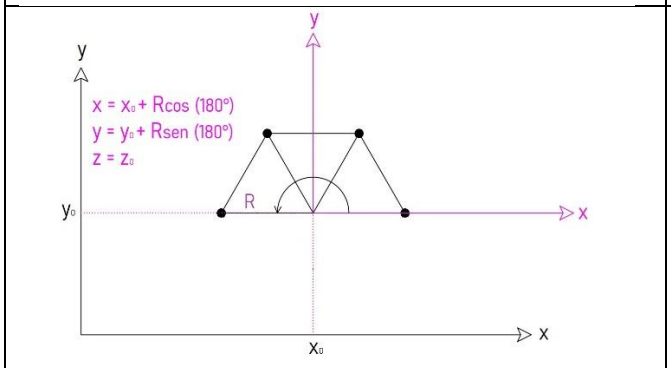
For döngüsü yoluyla altıgenin yapısı:

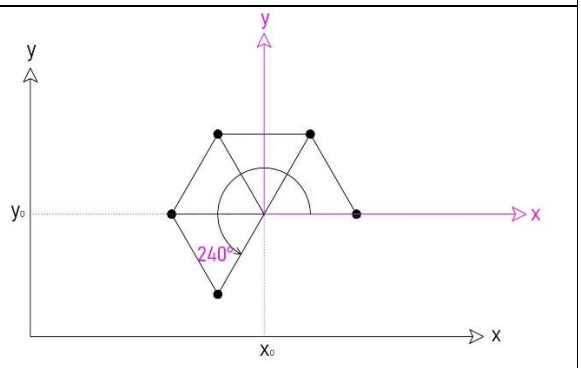


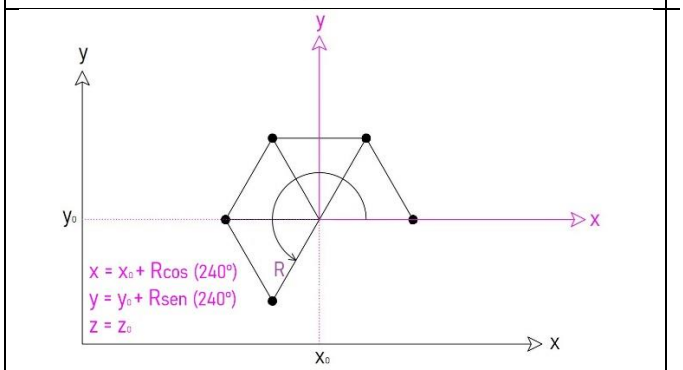
TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

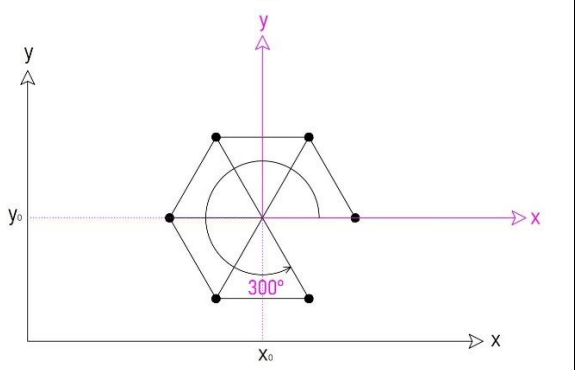


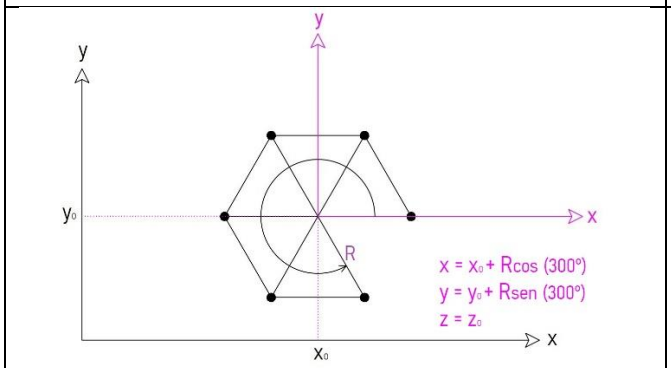


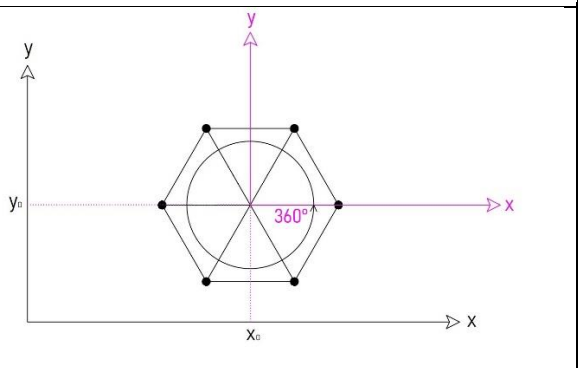


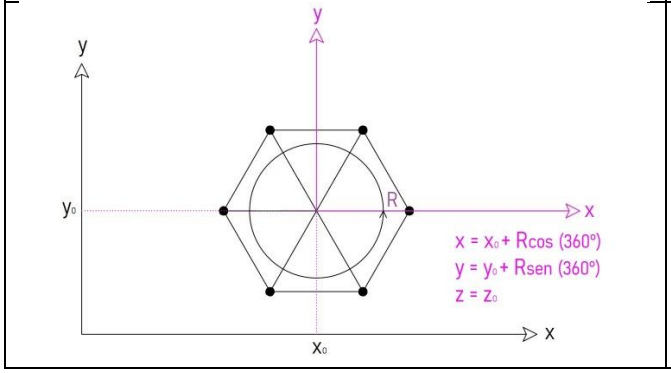












TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

BAKIM – ONARIM İLKELERİ

Bu bölümde robotların bakım-onarım ve güvenlik kullanımına ilişkin ilkeler açıklanmaktadır. Robot, uç efektör veya çevre ekipmanı olmadan çalışamaz. Robot, uç efektör ve çevre ekipmanlarıyla birleştirilip sistemin montajı yapıldığında düzgün çalışabilir. Yani robot sistemin bir parçasıdır.

28. ROBOT SİSTEM BİLEŞENLERİ

Günümüzde işbirlikçi robotlar dendiğinde, işçiler ile birlikte çalışan robotlar akıla gelmektedir. Dolayısıyla aşağıdaki bileşenlerin güvenli çalışmasını sağlamak gerekmektedir.

Şekil 1. İşbirlikçi Robot Sistem Örneği

- Robot
- Robot Kontrolcüsü
- Robot Teach Pendant
- Uç Efektör
- Diğer Çevre Birimleri
- Çalışma Parçası

Teknisyenler robot sisteminin risk değerlendirmesini yapar ve ihtiyaç ortaya çıktıkça güvenlik için aşağıdaki unsurların sisteme ilave edilmelidir.

- Güvenlik Kapısı
- Kilitli Kapı
- Kilitleme Devreleri

Aşağıdaki robot bileşenlerinin güvenliği ise üretici firmalar tarafından zaten sağlanmıştır.

- Robot
- Robot kontrolcücü ve teach pendant

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

29. KULLANICI VE KULLANICI GÜVENLİĞİ TANIMLAMA

29.1. Kullanıcıları Tanımlama

Kullanıcılar bir işbirliköçi robot sisteminde aşağıdaki gibi isimlendirilmişlerdir.

İşbirlikçi çalışan:

- İşbirliğine dayalı çalışma alanına girer, robotla çalışır.
- Robotu doğrudan zorlayarak robotun tutumunu değiştirir; örneğin robotu iterek durdurma işlevi.
- İşbirliğine dayalı çalışan için ayarlanan operatör düğmesiyle programı yeniden başlatır.

Operator:

- Robot kontrolcüsü gücünü AÇ/KAPA yapar.
- Operatör panelden robot programını çalıştırır.

Programcı:

- Robotu çalıştırır ve öğretme işlemini el kontrol paneli kullanarak gerçekleştirir.
- Robotu çalıştırır ve öğretme işlemini doğrudan öğretmeyi kullanarak gerçekleştirir.

Bekım-Onarım teknisyeni:

- Robot kullanır.
- Roboru güvenlik çiti içerisinde tutar.
- Bakım-Onarım gerçekleştirir. (onarım, ayarlar, yerleştirme)

Aşağıdaki tablo işbirlikçi robotun çalışmalarını göstermektedir. Bu tabloda “O” sembolü personelin yapmasına izin verilen çalışmayı ifade etmektedir.

	İşbirlikçi Çalışan	Operator	Programcı Öğretme Operatörü	Bakım-Onarım Teknisyeni
Robot kontrolcüsünü Açma/Kapama		O	O	O
Çalışma modunu seçme			O	O
Erişim modunu seçme			O	O
El Paneli ile program seçimi			O	O
Harici devre ile program seçimi			O	O
Operatör paneli ile program çalıştırma		O	O	O
El Paneli ile program çalıştırma			O	O
Operatör panelinden alarm durdurma			O	O
El panelinden alarm durdurma			O	O
Operatör panelinde veri ayarı			O	O
El paneli ile öğretme			O	O
Direkt öğretme ile öğretme			O	O
Operatör panelinden acil durdurma	O	O	O	O
El panelinden acil durdurma	O	O	O	O
Operatör panelinden bakım-onarım				O
El panelinden bakım-onarım				O
Çalışma sahasına erişim	O	O	O	O
İşbirlikçi işçi için ayarlanmış operatör butonu ile restart yapma	O	O	O	O

Tablo 1. İşbirlikçi robotta çalışma listesi

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

29.2. Kullanıcı Güvenliği

Kullanıcı güvenliği birincil güvenlik hususudur. Otomatik çalışma sırasında robotun çalışma alanına girmek çok tehlikeli olduğundan yeterli güvenlik önlemleri alınmalıdır.
Aşağıda genel güvenlik önlemleri listelenmektedir. Kullanıcı güvenliğini sağlamak için dikkatli bir değerlendirme yapılmalıdır.

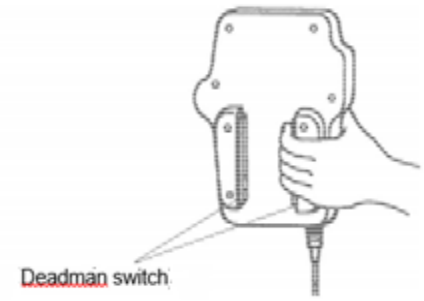
29.2.1. Programlama Güvenliği

Robota öğretirken verirken operatörün robotun çalışma alanına girmesi gerekir. Özellikle el kumandası paneli operatörü kendi güvenliğini sağlamalıdır.

1- Robotun çalışma alanına girilmesi özellikle gerekmedikçe tüm görevleri alan dışında gerçekleştirin.

2- Robota öğretmeden önce robotun ve çevre birimlerinin normal durumda olduğunu kontrol edin.

3- Robota öğretmek için robot çalışma alanına girmek kaçınılmazsa, güvenlik cihazlarının (ACİL DURDURMA düğmesi, DEADMAN anahtarı gibi) konumları, ayarları ve diğer koşullar kontrol edilmelidir.



Şekil 2. Aleti Aktifleştirme (Deadman switch) (iPendant) FANUC

4- Programcı robotun çalışma alanına başka kimsenin girmemesine son derece dikkat etmelidir.

5- Programlama mümkün olduğunca güvenlik çitinin dışında yapılmalıdır. Güvenlik çiti alanında programlama yapılması gerekiyorsa programcı aşağıdaki önlemleri almalıdır:

* Güvenlik çiti alanına girmeden önce bölgede herhangi bir tehlikeli durum riski bulunmadığından emin olun.

* Gerektiğinde acil stop butonuna basmaya hazır olun.

* Robotu düşük hızda çalıştırın.

* Programlamaya başlamadan önce, çevresel ekipmanlara veya hareketlere yönelik hiçbir uzaktan talimatın çalışan kişiye zarar vermeyeceğinden emin olmak için tüm sistem durumunu kontrol edin.

6- Operatör, Kontak Durdurma fonksiyonu aktif olduğu durumda çalışmalıdır.

7- Kontak Durdurmayı geçici olarak devre dışı bırakmak için gerekli, yaygınlaştırmak için önlem almak Kontak Durdurma fonksiyonunu devre dışı bırakır.

8- Operatör panelini kullanarak sistemi başlatmak için robotun çalışma alanında kimsenin bulunmadığından ve robotun çalışma alanında herhangi bir anormal durumun bulunmadığından emin olunmalıdır.

9- Bir program tamamlandığında test işletiminin aşağıdaki prosedüre göre yapıldığından emin olunmalıdır.

(a) Programı tek adım modunda düşük hızda en az bir çalışma döngüsü boyunca çalıştırın.

(b) Programı sürekli çalışma modunda düşük hızda en az bir çalışma döngüsü boyunca çalıştırın.

(c) Programı, sürekli çalışma modunda ara hızda bir çalışma döngüsü boyunca çalıştırın ve zamanlamadaki gecikme nedeniyle herhangi bir anormallik meydana gelmediğini kontrol edin.

(d) Programı, normal çalışma hızında sürekli çalışma modunda bir çalışma döngüsü boyunca çalıştırın ve sistemin sorunsuz bir şekilde otomatik olarak çalıştığını kontrol edin.

(e) Yukarıdaki test işlemiyle programın eksiksizliğini kontrol ettikten sonra, programı otomatik çalışma modunda yürütün.

10- Sistemi otomatik çalışma modunda çalıştırırken, el kumanda paneli operatörü robotun çalışma alanını terk etmelidir.

29.2.2. Bakım – Onarım Güvenliği

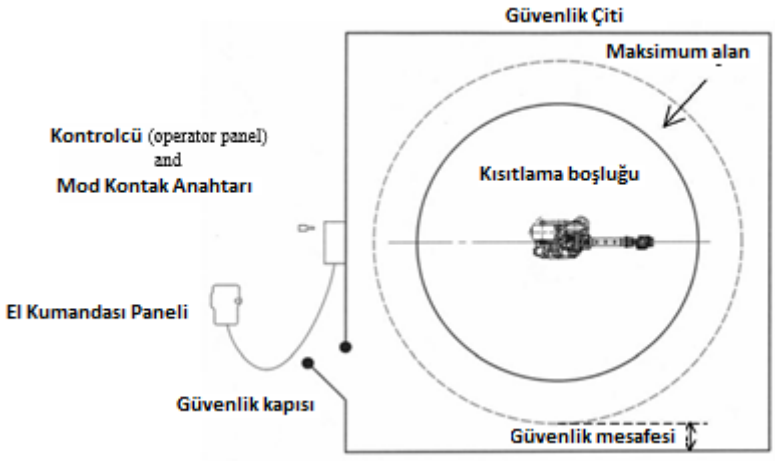
Bakım-Onarım teknisyeni personelinin güvenliği için aşağıdaki hususlara azami dikkat gösterilmedir.

1- Çalışma sırasında kesinlikle bölgede bulunmamalıdır.

2- Bakım işlemleri sırasında robotun veya sistemin açık tutulması halinde tehlikeli bir durum ortaya çıkabilir. Bu nedenle herhangi bir bakım işlemi için robotun ve sistemin kapalı duruma getirilmesi gerekir. Gerekirse başka bir kişinin robotu ve/veya sistemi açmasını önlemek için kilit bulunmalıdır. Güç açık durumdayken bakımın yapılması gerekiyorsa acil durdurma düğmesine basılmalıdır.

3- Robotun çalışma alanına güç açıkken girmek gerekirse, alana girmeden önce operatör panelindeki acil durdurma düğmesine veya öğretme kumandasına basılmalıdır. Bakım personeli, bakım çalışmalarının sürdüğünü belirtmeli ve başkalarının robotu dikkatsizce çalıştırmasına izin vermemeye dikkat etmelidir.

4- Bakım görevlisi güvenlik çiti ile çevrelenmiş alana girerken etrafta tehlikeli bir durum olmadığından emin olmak için tüm sistemi kontrol etmelidir. Tehlikeli bir durum mevcutken işçinin güvenlik alanına girmesi gerektiğinde son derece dikkatli olunmalı ve tüm sistem durumu dikkatle izlenmelidir.



Şekil 3. Güvenlik Çiti ve Güvenlik Kapısı örneği

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

5- Pnömatik sistemin bakımına başlamadan önce besleme basıncı kesilmeli ve borulardaki basınç sıfırlanmalıdır.

6- Öğretmeden önce robotun ve çevre birimlerinin normal durumda olup olmadığı kontrol edilmelidir.

7- Robotun çalışma alanında birisi varken robot otomatik modda çalıştırılmamalıdır.

8- Güvenlik çiti veya robotun çalışma alanı içerisinde kaçış yollarının açık olduğundan emin olunmalıdır.

9- Robotun üzerine bir alet takıldığında veya robot dışında bantlı konveyör gibi herhangi bir hareketli cihaz takıldığında, bu hareketlere dikkat edilmesi gerekmektedir.

10- Operatör panelinin yakınına, olası tehlikeyi gördüğünde ACİL DURDURMA butonuna basabilecek bir uzman görevlendirilmelidir.

11- Bir parçanın değiştirilmesi durumunda robot firması ile iletişime geçilmelidir. Yanlış prosedür, robotun ve çalışanın ciddi şekilde zarar görmesine neden olabilir.

12- Bileşenler değiştirilirken veya yeniden takılırken sistemde yabancı madde olmadığından emin olunmalıdır.

13- Muayene sırasında kontrolördeki her bir ünite veya baskılı devre kartını tutarken tekrar elektrik çarpmasından korunmak için devre kesici kapatılmalıdır.

14- Üretici firma tarafından tavsiye edilen sigorta kullanılmamalıdır.

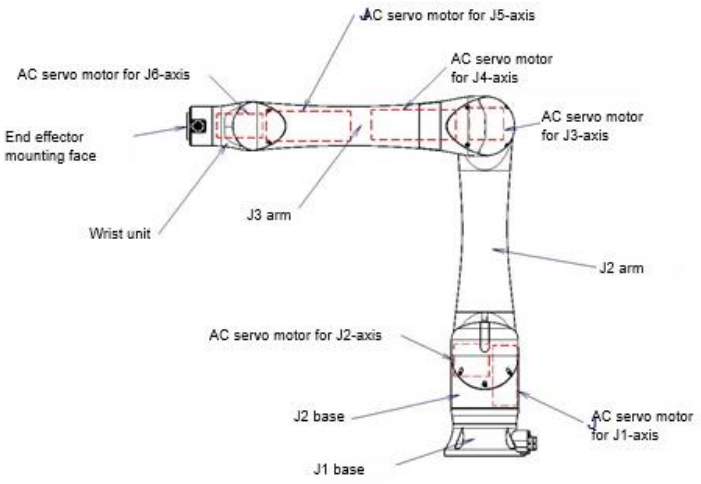
15- Bakım işi tamamlandıktan sonra robot sistemini yeniden başlatırken, çalışma alanında kimsenin bulunmadığından, robotun ve çevre birimlerinin anormal olmadığından önceden emin olunmalıdır.

16- Motorun veya frenin çıkarılması durumunda, düşmeyi önlemek için kol önceden vinç veya başka bir ekipmanla askıya alınmalıdır.

17- Zemine yağ döküldüğünde, düşmeyi önlemek için mümkün olan en kısa sürede temizlenmelidir.

18- Aşağıdaki kısımlar ısınabilir. Bakım görevlisinin sıcak durumdayken böyle bir parçaya dokunması gerekiyorsa, ısıya dayanıklı eldiven giymesi veya başka koruyucu aletler kullanması gerekir.

Servo motor, Kumanda İçi, Redüktör, Şanzıman ve Bilek ünitesi



Şekil 4. Mekanik Ünite

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

19- Bakımlar uygun aydınlatma ile yapılmalıdır.

20- Motor, redüktör ya da başka bir ağır yük taşınırken bakım çalışanlarını aşırı yükten korumak için vinç ya da başka bir ekipman kullanılmalıdır. Aksi takdirde bakım çalışanları ciddi şekilde yaralanabilir.

21- Bakım sırasında dahi robotun üzerine kesinlikle çıkılmamalı ve üzerine basılmamalıdır. Böyle bir girişimde bulunulması halinde robot olumsuz etkilenecektir. Ayrıca yanlış bir adım işçinin yaralanmasına neden olabilir.

22- Yüksek yerde bakım yaparken basamaklar emniyete alınmalı ve emniyet kemerini takılmalıdır.

23- Bakım tamamladıktan sonra güvenlik çitindeki robotun etrafına dökülen tüm yağ, su ve metal talaşlar temizlenmelidir.

24- Parça değişiminde ilgili tüm cıvata ve aksamlar orijinal yerlerine tekrar takılmalıdır. Bazı parçalar eksik veya dışarıda kaldıysa, hepsi kullanılana kadar değiştirme işlemi tekrarlanmalıdır.

25- Bakım sırasında robotun hareket etmesi gerekiyorsa aşağıdaki önlemler alınmalıdır:

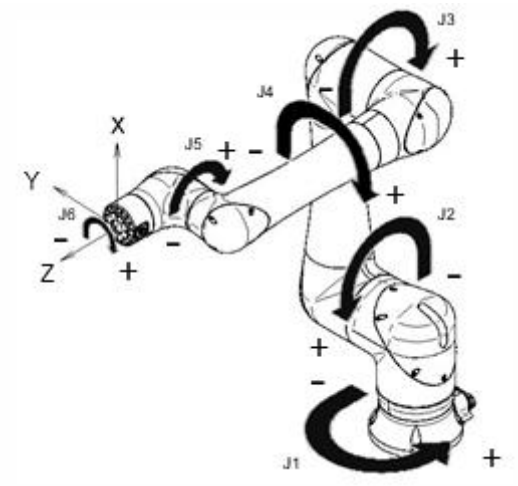
* Bir kaçış yolu emniyete alınmalıdır. Bakım hareketi sırasında tüm sistem sürekli olarak izlenmelidir.

* Gerektiğinde acil stop butonuna basmak için gerekli tedbir alınmalıdır.

26- Periyodik muayene zamanında yapılmalıdır. Periyodik muayenenin yapılmaması, robotun performansını veya hizmet ömrünü olumsuz etkileyebilir ve bir kazaya neden olabilir.

27- Bazı parçalar değiştirildikten sonra önceden belirlenen yönteme göre bir deneme çalıştırması yapılmalıdır. Test çalıştırması sırasında bakım personeli, ihtiyaç duyulduğunda güvenlik çitinin dışında çalışmalıdır.

28- Güvenlik çiti veya robotun çalışma alanı içerisinde kaçış yollarının engellenmediğinden emin olunmalıdır.



Şekil 5. Herbir eksen koordinatları

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

30. GÜVENLİK ARAÇLARI, ÇEVRE BİRİMLERİ VE MEKANİZMA

30.1. Pogramlama Araçlarında Önlemler

1- Tehlikeli bir durumu tespit etmek için bir limit anahtar veya başka bir sensör kullanılmalı ve gerekirse program sensör sinyali alındığında robotu durduracak şekilde tasarlanmalıdır.

2- Robotun kendisi normal olmasına rağmen diğer robotlarda veya çevresel cihazlarda anormal bir durum oluştuğunda robotu durduracak program tasarlanmalıdır.

3- Robotun ve çevre birimlerinin senkronize hareket ettiği bir sistem için programlamada birbirlerine müdahale etmemelerine dikkat edilmelidir.

4- Robotun sistemdeki tüm cihazların durumlarını algılayabilmesi ve durumlara göre durdurulabilmesi için robot ile çevre birimleri arasında uygun bir arayüz sağlanmalıdır.

30.2. Mekanizma Araçlarında Önlemler

1- Robot sisteminin bileşen hücreleri temiz tutulmalı, robot yağ, su ve toz etkisinden izole edilmiş bir yerde çalıştırılmalıdır.

2- Kesme sıvısı ve temizleme sıvısı için onaylanmamış sıvı kullanılmamalıdır.

3- Robotun hareketini sınırlamak için limit anahtarları veya mekanik durdurucuları kullanılmalıdır ve böylece robotun çevresel cihazlara veya aletlere çarpması önlenmelidir.

4- Mekanik ünite kabloları ile ilgili aşağıdaki uyarılara dikkat edilmelidir. Önlemlere uyulmaması mekanik sorunlara neden olabilir.

* Gerekli kullanıcı arayüzüne sahip mekanik ünite kablosu kullanılmalıdır.

* Mekanik ünitenin içine kullanıcı kablosu veya hortum eklenmemelidir.

* Kablolar mekanik ünite dışına eklendiğinde, mekanik ünitenin hareketini engellememelidir.

* Kullanıcı çevre ekipmanı robot mekanik ünitesine takılırken, ekipmanın robotun kendisine müdahale etmemesine dikkat edilmelidir.

5- Robotun çalışma sırasında sık sık kapanması robotun sorun yaşamasına neden olur. Durdurmak gerektiğinde robotun hızı azaltılıp, acil olmadığında bekleterek, durdurarak veya döngü yaparak durdurduktan sonra kapatma işlemi gerçekleştirilmelidir. Olası durdurmalar;

* Kötü ürün oluştuğunda, acil durdurma ile hat durdurulur ve robotun gücü kesilir.

* Değişiklik gerektiğinde emniyet çiti açılarak emniyet anahtarı çalıştırılır ve çalışma sırasında robotun kapanması sağlanır.

* Operatör acil stop butonuna sık sık basar ve hat durur.

* Güvenlik sinyaline bağlı bir alan sensörü veya paspas anahtarı rutin olarak çalışır ve robotun kapanması gerçekleşir.

* İkili Kontrol Güvenliği (DCS) için uygun olmayan bir ayar nedeniyle düzenli olarak güç kesintisi yaşanır.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

30.3. Programlama Mekanizmasında Önlemler

1-Robotların çalışma alanları çakıştığında robotların hareketlerinin birbirini engellememesine dikkat edilmelidir.

2- Robot için bir hareket programında önceden belirlenmiş iş orijininin belirttiğildiğinden emin olunmalıdır ve hareket orijinden başlayıp orijinde bitecek şekilde programlanmalıdır.

3- Operatörün robot hareketinin sona erdiğini bir bakışta kolayca ayırt etmesi mümkün kılınmalıdır.

a) Pnömatik, hidrolik ve elektrikli aktüatörleri kontrol etmek için, her kontrol komutunun gerçekleşmesinden sonra gerekli zaman gecikmesi dikkatlice düşünülmeli ve güvenli kontrol sağlanmalıdır.

b) Uç efektöre bir limit anahtarı bağlanmalı ve uç efektörün durumu izlenerek robot sistemi kontrol edilmelidir.

30.4. Acil ve Anormal Durumlarda Kolu Tahrik Gücü Olmadan Hareket Ettirme

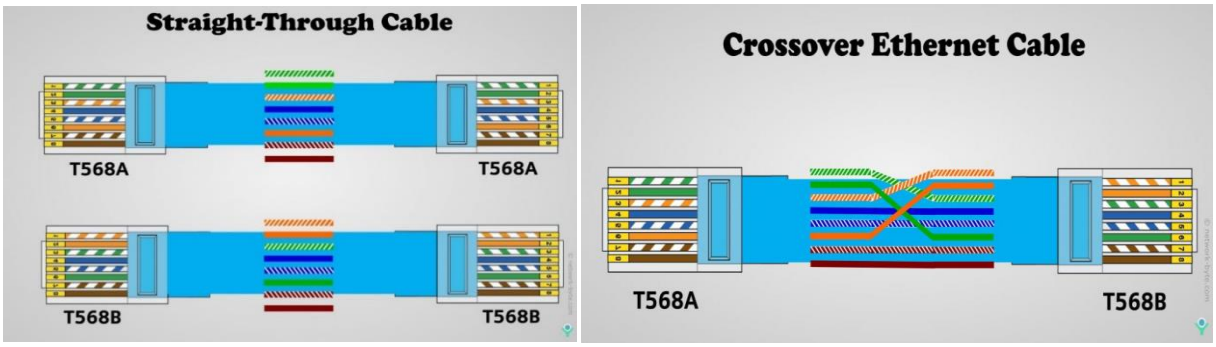
Acil veya anormal durumlarda (örneğin robotun içinde sıkışıp kalan veya robot tarafından sıkıştırılan kişiler) fren serbest bırakma ünitesi, robot eksenlerini tahrik gücü olmadan hareket ettirmek için kullanılabilir.

Fren serbest bırakma ünitesi yöntemi ve robotu destekleme yöntemi içinrobot kullanılmaya başlanmadan önce robot kılavuz okunmalıdır.

30.5. Ağ Sisteminde Önlemler

Ağ cihazlarını birbirine bağlamak için standart kablolar kullanılır. Ağ cihazlarını UTP kablo ile bağlamak için kullanılan Ethernet kablosunun ucu RJ45 konnektör ile sonlandırılmaktadır. UTP kablo, RJ45 veya 8P8C konnektör üzerinde sonlandırmak için kullanılan 4 çift veya 8 farklı renkte telden oluşur. EIA (Elektronik Endüstrileri Birliği) ve TIA (Telekomünikasyon Endüstrisi Birliği) tarafından standartlaştırılan Ethernet kablosunun renk kodlaması, iki standart EIA/TIA-568-A ve EIA/TIA-568-B'dir.

Gerektiğinde ağ kabloları teknisyenler tarafından yapılabilmelidir.



Şekil 6. Bağlantı Kabloları

TEMRİN 13

ROBOTLAR İÇİN AĞ KABLOSU YAPIMI VE TESTİ

Bu çalışma sayfasının amacı:

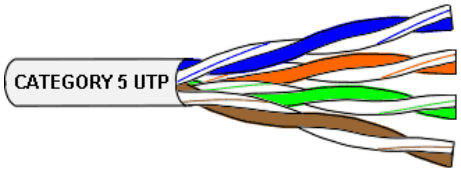
- Ağ kablosu yapımını öğrenme
- Robot ve bilgisayar arasında kullanılacak ağ kablosu yapma.
- Yapılan kabloyu test etme.

Giriş :

UTP kablo içerisinde 4 çift bakır kablo bulunur. Kabloların birbirleri üzerindeki elektromagnetik etkisini azaltmak için, bakır kablolar ikişer ikişer sarılı durumdadırlar. Çevresinin küçük olmasından dolayı kablo kanallarında daha az yer kaplamakta ve büyük ağ kurulumlarında çok avantaj sağlamaktadır.

Çift bükümlü kabloları sonlandırmak için RJ(Registered Jack) serisi konnektörler kullanılır. RJ serisinde onlarca konnektör çeşidi vardır. Bunların içinde en yaygın olanları telefon sistemlerinde kullanılan Kategori 2 (Cat2) kabloları sonlandıran RJ-12 ve UTP ile STP kabloların sonlandırılmasında kullanılan RJ-45 konnektörleridir.

Bu konnektörler kabloya takılırken bazı aletler gerekmektedir. Bu aletler kabloyu soymak, bükümlü çiftleri ayırmak, kabloyu kesmek ve kabloyu konnektöre takmak için gerekli olan aletlerdir.



Şekil 7. UTP Cat5 Kablo

Malzeme Listesi:

- 2 adet RJ-45 konnektörü (Cat5)
- 2 adet RJ-45 yalıtkan kapağı
- 0,5 metre Cat 5e Kablo
- 1 adet RJ-45, RJ-12 konnektörleri için kullanılan sıkıştırma pensesi
- 1 adet Bükümlü çiftlerin temizlenmeleri ve kesilmeleri için alet

İşlem Adımları:

1- Önce hazırlanacak kablo kesilir ve ucuna yalıtkan kapak takılır. Kapaklar, kablonun eğilip bükülmesi esnasında zarar görmesini engeller.

2- İzolasyonun en dış katını çıkarmak için gerekli olan aletle kablonun üst katı halka olarak kesilir ve çıkarılır.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

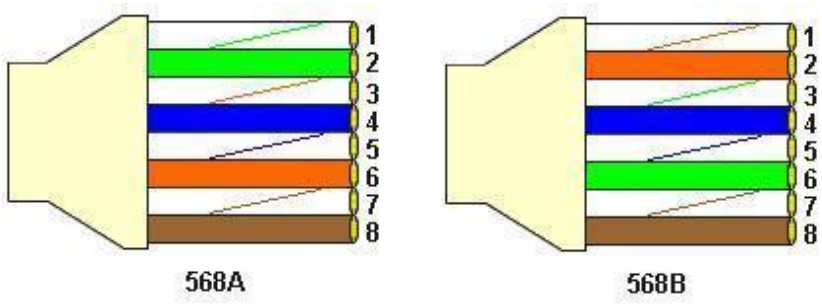
3- Kablonun konnektöre sokulabilmesi için bükümlü çiftler çözülmelidir. Çiftler, kablonun kılıfının kenarına kadar çözülürler. Çiftlerin bir sıra olarak yerleştirilmesi gerekmektedir. Bunun için kablo, yassı biçimli yapılır.

4- Çiftler, paralel olarak yerleştirilen iletkenlerden yassı bir katın oluşturulacağı şekilde koyulmalıdır. Kablo sıkma pensesi ile kablonun kılıfının kenarından iletkenlerinin aşağı yukarı 14 mm'lik parçası kesilir.

5- İletkenler, seçilen standarda (T568A veya T568B) uygun olarak renk sırasına koyulur. Bu sıralamada yaygın olan standart EIA/TIA-T568B (soldan sağa: turuncu- beyaz, turuncu, yeşil- beyaz, mavi, mavi- beyaz, yeşil, kahverengi- beyaz, kahverengi) standardıdır.

Eğer kablo bir PC'den bir ağ cihazına takılacaksa kablonun her iki ucundaki konnektör de aynı standarda göre hazırlanmalıdır. (Düz Bağlantı)

Eğer kablo bir ağ cihazından diğer bir ağ cihazına ya da bir PC'den diğer bir PC'ye takılacaksa o zaman kablonun uçlarındaki konnektörlerden birbirinden farklı standartlara göre hazırlanmalıdır. (Çapraz Bağlantı)



Şekil 8. EIA/TIA Kablo bağlantı standartları

6- İletkenler konnektörde, ayrı ayrı kanallarda bulunacakları ve kablonun kılıfının konnektöre en az 6 mm gireceği şekilde ayarlanmalıdır. Konnektörün sabitleyici anahtarı aşağıya yönlendirilmelidir.

7- İletkenler konnektöre sonuna kadar sokulmalıdır. Konnektörün uç kısmında bulunan bıçakların iletkenlerle temas sağlayabilmesi için kablo konnektöre iyice oturtulmalıdır. Konnektör saydam plastikten üretildiğinden dolayı iletkenlerin durumu görsel olarak kontrol edilebilir. Bu aşamadan sonra yapılan hatanın geri dönüşü olmayacağı için iletkenler iyi kontrol edilmelidir.

8- Bükümlü çiftleri konnektörün bıçaklarıyla bağlamak için RJ-45 pensesi kullanılır. Bu işlem ile konnektörün bıçakları konnektörün içine girer, iletkenlerin kılıflarını keserler ve kablonun telleri arasına girerek elektrik kontağını sağlarlar. RJ-45 pensesi sayesinde konnektör kabloya çıkmayacak şekilde monte edilmiş olur.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

9- Klasik konstrüksiyon olarak yapılan konnektörde kablo yassı lata (uzunluğu, aşağı yukarı 7 mm) biçiminde baskı ile sabitleştirilir. Konnektörün entegre bir parçası olan bu baskı sayesinde kablo iyice sıkıştırılmış olur. Bu sayede kablonun yükü öndeki bıçaklara binmez.

10- Konnektörün yalıtkan kapağı takılır.

11- Kablonun konnektöre takılma işlemi tamamlanmış olur. Burada emin olmak için kablo ve konnektör küçük bir kuvvetle zıt yönlere doğru çekilerek montajın sağlamlığı kontrol edilir.

12- Son olarak üretilen kablonun her iki ucu da kablo test cihazları aracılığıyla test edilir.

Şekil 9. Kablo test ekipmanları

Rapor:

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

4. KONTROLLER VE BAKIM - ONARIM

Robot sisteminin sürekli güvenli çalışmasını sağlamak için robot ve robot sistemi, bir inceleme ve bakım programına sahip olmalıdır.

Denetim ve bakım programında robot ve robot sistemi üreticisinin tavsiyeleri dikkate alınmalıdır.

Robotlar veya robot sistemi üzerinde bakım veya onarım yapan personel, gerekli görevleri güvenli bir şekilde gerçekleştirmek için gerekli prosedürler konusunda eğitilmelidir.

Robot sistemlerinin bakımını ve onarımını yapan personel tehlikelerden korunmalıdır.

Mümkün olduğunda bakım, robot kolunun önceden belirlenmiş bir konuma yerleştirilmesiyle korunan alan veya robot çalışma alanı veya üretim alanının dışında gerçekleştirilmelidir.

Risk değerlendirmesinin sonuçları göz önüne alınarak, işbirlikçi robot konusunda eğitilmiş kişilerin bakım sırasında robotun çalışma alanına ve üretim mahalline kolayca erişmesi kabul edilebilir. Bu durumda kontak durdurma fonksiyonunun etkinleştirildiği acil durumlar için doğrulanmalıdır.

4.1. PERİYODİK BAKIM - ONARIM

Bu bölümde açıklanan periyodik bakım prosedürlerinde, örnek olarak alınan FANUC robotunun yılda 3840 saate kadar kullanıldığı varsayılmaktadır. Robot kullanımının 3840 saati/yılı aştığı durumlarda verilen bakım sıklıklarını buna göre ayarlanmalıdır.

Yeni (daha yüksek) frekansları hesaplamak için fiili çalışma süresi/yıl ile 3840 saat/yıl oranı kullanılmalıdır. Örneğin, robotu 3 yıllık veya 11520 saatlik önerilen bakım aralığıyla yılda 7680 saat kullanırken, bakım sıklığını belirlemek için aşağıdaki hesaplamayı kullanın: 3 yıl / 2 = her 1,5 yılda bir bakım yapın.

Günlük bakım ve periyodik bakım/muayene, uzun süreler boyunca güvenilir robot performansı sağlar.

4.1.1. Günlük Kontroller ve Bakım – Onarım

(1) Günlük Bakım

Sistemi her gün çalıştırmadan önce sistemin her parçası temizlenmeli ve sistem parçalarında hasar veya çatlak olup olmadığını kontrol edilmelidir:

Çalıştırmadan önce,

El kumandası paneline bağlı kabloda aşırı bükülme olup olmadığını kontrol edin.

Denetleyiciyi ve çevresel aygıtları anormallik açısından kontrol edin.

Güvenlik fonksiyonunu kontrol edin.

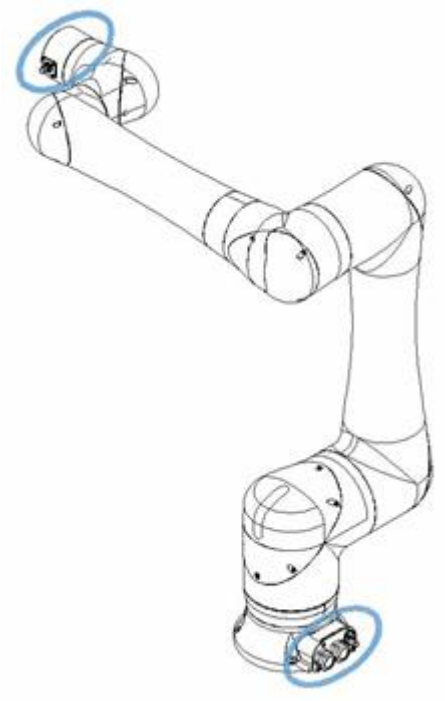
Çalıştırmadan sonra,

Servis işleminin sonunda robotu uygun konuma getirin ve ardından kontrol cihazını kapatın.

Her parçayı temizleyin ve herhangi bir hasar veya çatlak olup olmadığını kontrol edin.

Kontrol ünitesinin havalandırma portu ve fan motoru tozluysa tozu silin.

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



Şekil 6. Konektör denetim noktaları

(2) Bir ayın sonunda

Fanın normal şekilde dönüp dönmediğini kontrol edin. Fanda kir ve toz birikmişse fanı kullanım kılavuzunda belirtildiği gibi temizleyin.

(3) Altı ayda bir yapılan periyodik kontrol,

Fanda ve transformatörde kir ve toz birikmişse fanı ve transformatörü kullanım kılavuzunda belirtildiği gibi temizleyin.

(4) Battery daily check

Ana kartın ön panelindeki pili her 4 yılda bir değiştirin.

(5) Bakım-onarım araçları

AC/DC voltmetre

Oscilloscope with a frequency range of 5 MHz or higher, two channels

Yıldız tornavidalar: Büyük, orta ve küçük

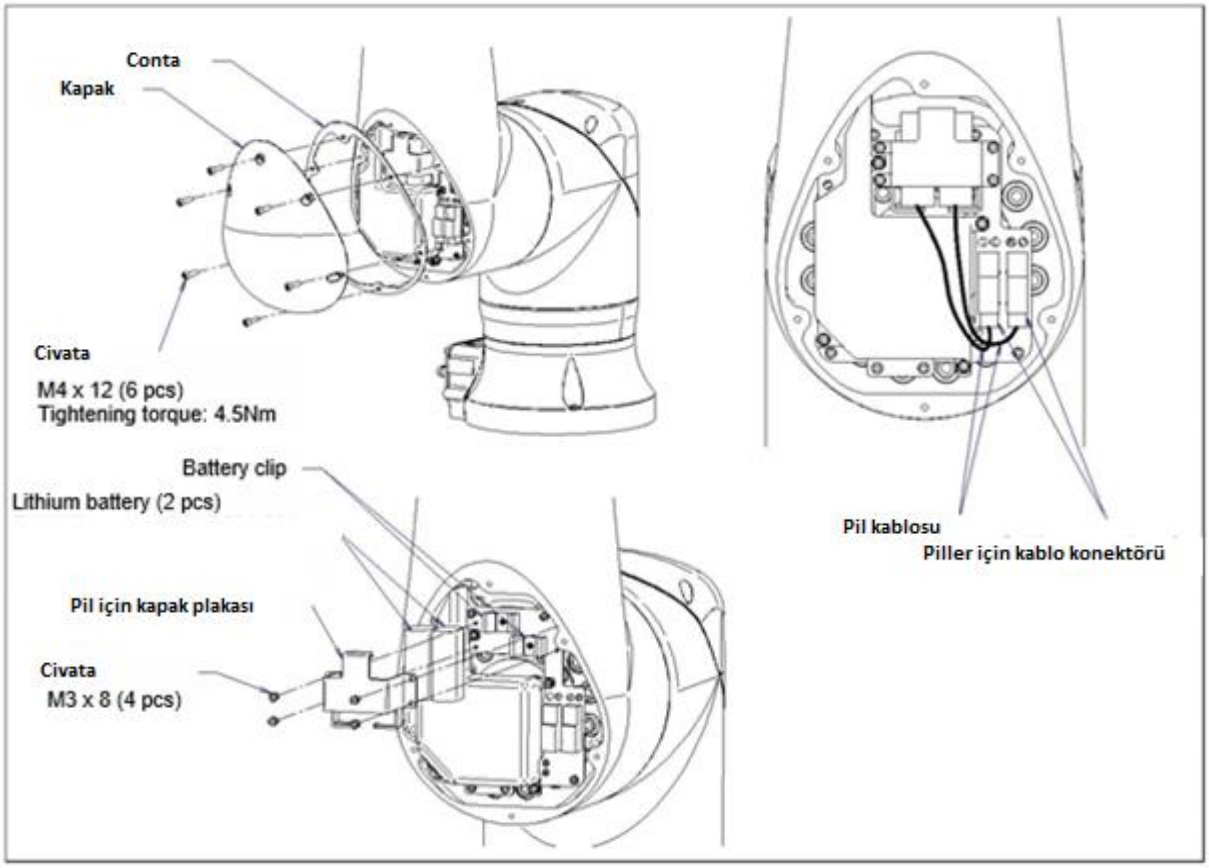
Düz başlı tornavidalar: Büyük, orta ve küçük

Somun tornavida seti (Metrik)

Pense

Kargaburun

Yankeski



Şekil 7. Pil değiştirme

Öğeleri Kontrol Et	Kontrol Noktaları ve Yönetim
Yağ sızıntısı	Her bir bağlantının contalı kısmında yağ olup olmadığını kontrol edin. Yağ sızıntısı var ise temizleyin
Titreşim, anormal sesler	Titreşim ve anormal sesler olup olmadığını kontrol edin
Konumlandırma doğruluğu	Robotun öğretilen konumlarının daha önce öğretilen konumlardan sapmadığını kontrol edin.
Düzgün çalışma için çevresel aygıtlar	Çevresel aygıtların, robottan ve çevresel diğer aygıtlardan gelen komutlara göre düzgün çalışıp çalışmadığını kontrol edin.
Her eksen için frenleme	Servo gücü kapatıldığında uç efektörünün aşağı boşluğunun 5mm dahilinde olup olmadığını kontrol edin.
Uyarılar	El kumandası panelindeki alarm ekranında beklenmeyen uyarıların oluşup oluşmadığını kontrol ediniz.

Tablo 2. Kontrol Noktaları Listesi

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

4.1.2. Periyodik Kontroller ve Bakım – Onarım

Toplam çalışma süresine veya bakım zamanına göre aşağıda önerilen aralıklarla, aşağıdaki öğeleri kontrol edin, hangisi önce gelirse. (○ : Öğenin gerçekleştirilmesi gerekiyor.)

PERİYODİK BAKIM TABLOSU

		Öğeler \ Toplam İşletim Zamanı H	Zamanı kontrol et	Gres yağı miktarı	First check 320	3 months 960	6 months 1920	9 months 2880	1 year 3840	4800	5760	6720	2 years 7680	8640	9600	10560
Mechanical unit	1	Dış hasar veya soyulmuş boya olup olmadığını kontrol edin	0.1H	-		○	○	○	○	○	○	○	○	○	○	○
	2	Suyu kontrol edin	0.1H	-		○	○	○	○	○	○	○	○	○	○	○
	3	Uç efektör kablosunu kontrol edin	0.1H	-		○			○				○			
	4	Açıktaki konnektörü kontrol edin	0.1H	-		○			○				○			
	5	Uç efektör civatasını sıkın	0.1H	-		○			○				○			
	6	Kapağı ve ana civatayı sıkın	1.0H	-		○			○				○			
	7	Sıçramayı ve tozu giderin	1.0H			○			○				○			
Controller	8	Robot kablosunu, el kumandası paneli kablosunu ve bağlantı kablolarını kontrol edin	0.2H	-		○			○				○			
	9	Kontrolcü havalandırma sistemini temizleyin	0.2H	-	○	○	○	○	○	○	○	○	○	○	○	○

Tablo 3. Cobot Periyodik Bakım-Onarım Tablo Örneği (FANUC)

Yukarıdaki bakım tablosuna ek olarak aşağıdaki işlemler de yapılabilmektedir;

Robot sistemini güvende tutmak için lütfen operatör kılavuzunda veya bakım kılavuzunda belirtilen periyodik bakımları yapın.

Ayrıca lütfen sistemin her parçasını temizleyin ve herhangi bir hasar veya çatlak olup olmadığını görsel olarak kontrol edin.

Günlük kontrol öğeleri aşağıdaki gibidir (ancak bunlarla sınırlı değildir).

Giriş gücü voltajı

Pnömatik basınç

Bağlantı kablolarının hasar görmesi

Konektörlerin gevşekliği

Yağlama

Acil durdurma fonksiyonları

Öğretme kumandasındaki deadman anahtarının etkinliği

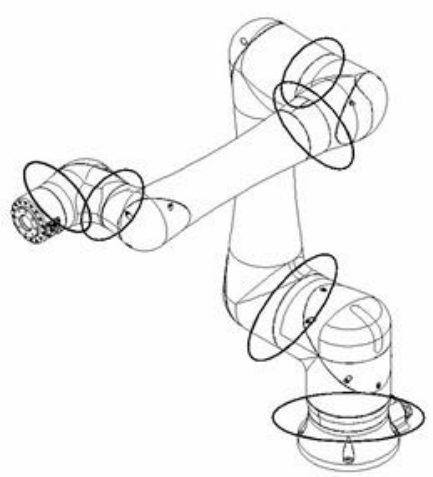
Emniyet kapısı kilitleri (robot sisteminin emniyet kapısı kilitleri olması durumunda)

Robot hareketinden kaynaklanan titreşim ve gürültü

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

Çevresel cihazların işlevleri

Robot ve çevresel cihazların armatürleri



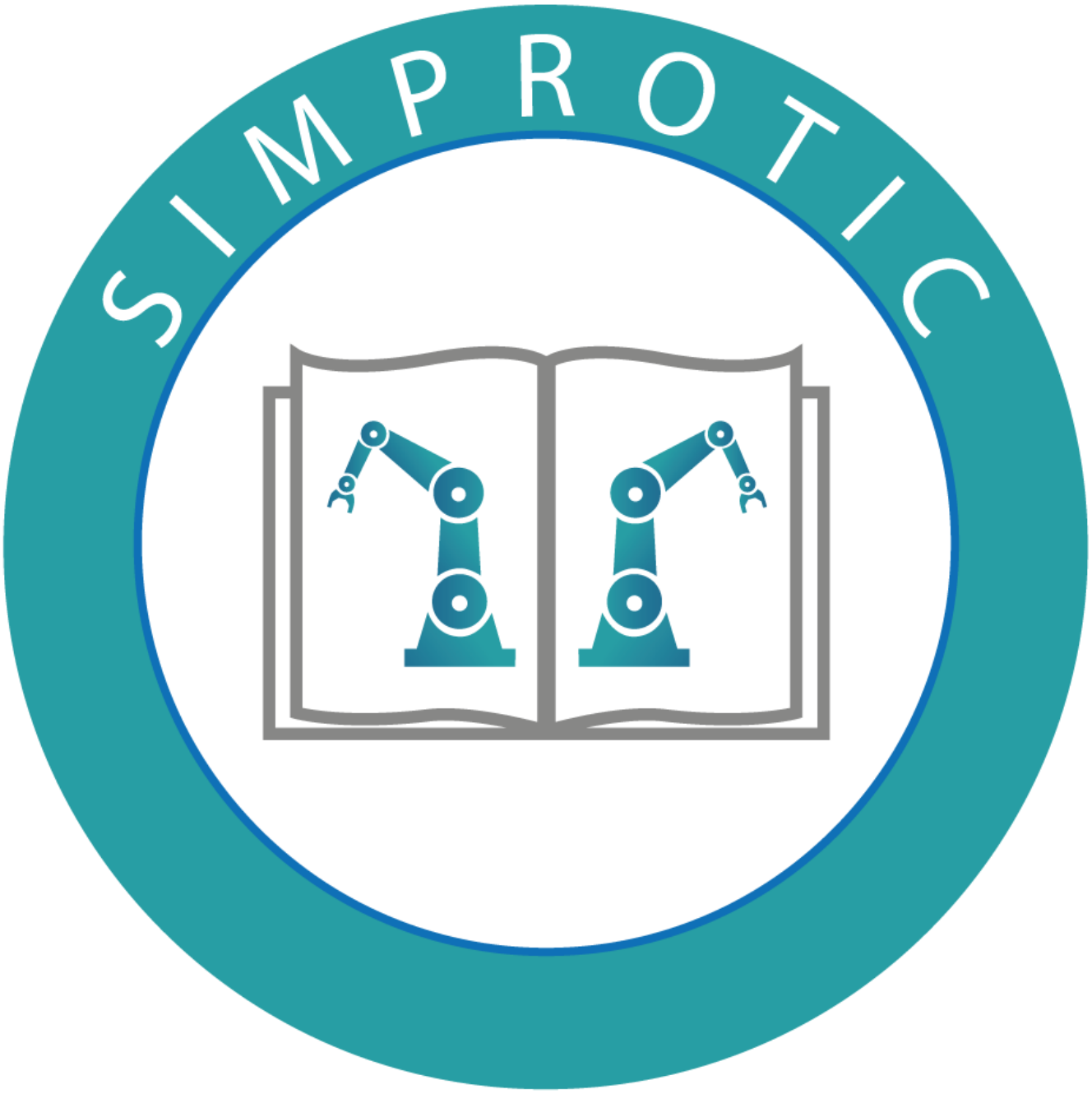
Şekil 6. Parçalardaki yağ sızıntısı kontrolü

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir

KAYNAKÇA

1- Dr. Serkan DİŞLİTAŞ. Endüstriyel Robot Programlama, Çorum 2015
2- RoboDK Temel Kullanım Kılavuzu, https://robodk.com/
3- Universal Robot eBookları, https://www.universal-robots.com/tr/akademi/
4- International Federation of Robotics Raporları, Frankfurt 2018
5- Milli Eğitim Bakanlığı - Programlama Temelleri 9, Ankara 2023
6- FANUC Bakım-Onarım Kılavuzu, https://fanuc.eu/

TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY
Avrupa Birliği tarafından finanse edilmektedir



TÜRKİYE ULUSAL AJANSI
TURKISH NATIONAL AGENCY

Avrupa Birliği tarafından
finanse edilmektedir