

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ELEKTRİK MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**ENDÜSTRİ 4.0 VE ROBOTİK SİSTEMLERDE SANAL
GERÇEKLIK UYGULAMASI**

AKIN ARAS

KOCAELİ 2018

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**ENDÜSTRİ 4.0 VE ROBOTİK SİSTEMLERDE SANAL
GERÇEKLİK UYGULAMASI**

AKIN ARAS

Prof.Dr. Nurettin ABUT
Danışman, Kocaeli Üniv.

Prof.Dr. Engin ÖZDEMİR
Jüri Üyesi, Kocaeli Üniv.

Prof.Dr. Cemil ÖZ
Jüri Üyesi, Sakarya Üniv.

Nurettin Abut

Engin Özdemir

Cemil Öz

Tezin Savunulduğu Tarih: 22.06.2018

ÖNSÖZ VE TEŞEKKÜR

Buhardan bir kinetik enerji kaynağı olarak faydalanılması ilk kez düşünüldüğünde, insanoğlu uzun bir süredir ağırlıklı olarak tarımsal ekonomiyle meşguldü. 18. Yüzyılın başında, buhar makinesinin İngiltere’de tekstil sektöründe ilk kez kullanılması, on yıllar sonra “Sanayi Devrimi” olarak adlandırılmıştır. Ardından, elektriğin keşfi ve sanayide kullanılmasıyla ikinci sanayi devrimi, sonrasında ise elektronik ve otomasyonun yükselişiyle üçüncü sanayi devrimi bu süreci izlemiştir. Bugün ise, nesnelere internetiyle, artırılmış gerçeklikle, makine öğrenimiyle birlikte dördüncü sanayi devrimi ile karşı karşıya bulunmaktayız. Bu süreçte öne çıkmış teknolojilerden biri de sanal gerçekliktir.

Bu çalışmada, otomotiv sektöründe yaygın uygulama alanı olan iki program ABB RobotStudio yazılımında yazılmış ve simülasyonu yapılmış, ardından HTC Vive sanal gerçeklik sistemiyle deneyimlenerek robot programlamada sanal gerçeklik teknolojisi kullanımının avantajlarının gösterilmesi amaçlanmıştır.

Tez çalışması süresince destek ve yönlendirmelerini aldığım değerli danışman hocam Sn. Prof. Dr. Nurettin Abut’a teşekkürlerimi sunarım.

Otomasyon alanındaki çalışmalarında desteklerini aldığım Prof. Dr. Engin Özdemir’e, tez yazım süreci boyunca her adımda desteğini aldığım Yrd. Doç. Dr. Murat Ayaz’a, tez konusunun şekillenmesinde katkısı olan Robo Otomasyon Gn. Md. Haluk Özcan’a, sanal gerçeklik setinin kullanımıyla ilgili bilgi paylaşımlarından dolayı Mekasim Mühendislik Gn. Md. Gökhan Oğuz’a, RobotStudio yazılımıyla çalıştığım süreçte verdikleri bilgi desteği dolayısıyla ABB Robot ekibinden Özgen Öztürk ve Mutlu Kahveci’ye, sanal gerçeklik setinin kullanımında desteğini esirgemeyen Galaktika Game Center ekibine teşekkür ederim.

Tüm eğitim hayatım boyunca en sağlam destekçim olan sevgili aileme ve hayatıma girdiği andan itibaren tez çalışmama yeni bir soluk getirip beni motive eden müstakbel eşim Begüm’e teşekkürlerimi borç bilirim.

Bu çalışma, bugüne kadar hayatıma dokunmuş tüm öğretmenlerime adanmıştır.

Mayıs - 2018

Akın ARAS

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER.....	ii
ŞEKİLLER DİZİNİ	iv
SİMGELER VE KISALTMALAR DİZİNİ.....	vi
ÖZET	vii
ABSTRACT	viii
GİRİŞ	1
1. ENDÜSTRİ 4.0.....	3
1.1. Büyük Veri.....	5
1.2. Değişen İş Kolları	7
1.2.1. Katmanlı imalat	7
1.2.2. Cobot – işbirlikçi robotlar.....	9
1.2.3. Akıllı fabrikalar	10
1.2.3.1. Dizayn süreci.....	10
1.2.3.2. Üretim süreci.....	11
1.2.3.3. Bakım süreci	11
1.2.4. İnsansız hava araçları	12
1.3. Endüstri 4.0'ın Gelir, Maliyet ve Verimlilik Yönünden Faydaları.....	13
2. SANAL DEVREYE ALMA.....	14
2.1. Siemens Process Simulate	15
2.2. ABB RobotStudio	17
2.2.1. Yeni istasyon oluşturulması.....	17
2.2.2. İstasyona robot eklenmesi ve sistemin kurulması.....	19
2.2.3. Robota tool eklenmesi	22
2.2.4. Robotun hareket ettirilmesi ve home pozisyona çekilmesi	24
2.2.5. İş parçası oluşturulması	26
2.2.6. Nokta izleme uygulaması	29
2.2.6.1. İstasyona sehpa ve iş parçası eklenmesi	29
2.2.6.2. Sehpanın üzerine yeni bir “workobject” tanımlanması	32
2.2.6.3. İş parçasının üzerinde robota hedef noktaların tanımlanması.....	34
2.2.6.4. Hedef noktalar kullanılarak robotun izleyeceği yolun tanımlanması.....	40
2.2.6.5. Simulasyon yapılması.....	45
2.2.7. Parça alıp bırakma uygulaması	46
2.2.7.1. Gripper dizayn edilip bir mekanizma olarak tanımlanması.....	47
2.2.7.2. Sehpa dizayn edilmesi	53
2.2.7.3. Gripper ile taşınacak iş parçasının oluşturulması	54
2.2.7.4. Smart Component tanımlanması	55
2.2.7.5. Robot sistemi içinde gerekli I/O sinyallerinin tanımlanması.....	62

2.2.7.6. Station Logic kısmında gerekli konfigürasyonun yapılması	64
2.2.7.7. Robotun izleyeceği yolun tanımlanması.....	65
2.2.7.8. RAPID altyapısında gerekli kodların yazılması.....	65
2.2.7.9. Simulasyon yapılması.....	67
3. SANAL GERÇEKLİK UYGULAMASI	69
3.1. Station Viewer Çıktısının Oluşturulması.....	69
3.2. HTC Vive VR Sistem ile Sanal Gerçeklik Uygulaması.....	70
4. SONUÇLAR VE ÖNERİLER.....	73
KAYNAKLAR.....	75
KİŞİSEL YAYIN VE ESERLER.....	78
ÖZGEÇMİŞ.....	79



ŞEKİLLER DİZİNİ

Şekil 1.1.	Buhar makinesi	3
Şekil 1.2.	Elektrik makinesi	4
Şekil 1.3.	PLC (en öndeki bir I/O modülü)	5
Şekil 1.4.	Endüstri 4.0	6
Şekil 1.5.	Bilgi Değer Döngüsü	6
Şekil 1.6.	Katmanlı imalat tekniğiyle üretilmiş parçalar	7
Şekil 1.7.	Katmanlı imalat tekniğiyle üretilmiş ayakkabı	8
Şekil 1.8.	Katmanlı imalat tekniğiyle üretilmiş kemik desteği	9
Şekil 1.9.	Cobot - İşbirlikçi robot	10
Şekil 1.10.	ABD’li tüketicilerin drone teslimatı algısı	12
Şekil 1.11.	Amazon Prime Air Servisi	13
Şekil 2.1.	Siemens Process Simulate	17
Şekil 2.2.	Yeni istasyon açılış sayfası	18
Şekil 2.3.	Boş istasyon	19
Şekil 2.4.	İstasyona robot eklemek	20
Şekil 2.5.	Robot versiyonunun seçimi	20
Şekil 2.6.	IRB 140 model robot	21
Şekil 2.7.	Robot sisteminin kurulması	21
Şekil 2.8.	Sistem kurulumunda “Options” sayfası	22
Şekil 2.9.	Robota tool eklenmesi	23
Şekil 2.10.	Toolun robota bağlanması	23
Şekil 2.11.	Robotun eksenlerinin bağımsız olarak hareket ettirilmesi	24
Şekil 2.12.	Toolun belli bir noktaya çekilmesi	25
Şekil 2.13.	Toolun belli bir noktada pozisyonunun değiştirilmesi	25
Şekil 2.14.	İş parçası oluşturulması	26
Şekil 2.15.	Robotun görünmez hale getirilmesi	27
Şekil 2.16.	İş parçasının boyutları ve konumunun belirlenmesi	27
Şekil 2.17.	100*200*100 mm boyutlarında iş parçası	28
Şekil 2.18.	İstasyona sehpa ve iş parçası eklenmesi	30
Şekil 2.19.	İş parçasının sehpa üzerine konumlandırılması – 1	30
Şekil 2.20.	İş parçasının sehpa üzerine konumlandırılması – 2	31
Şekil 2.21.	İş parçasının sehpa üzerine konumlandırılması – 3	32
Şekil 2.22.	Sehpanın üzerine yeni bir workobject tanımlanması – 1	33
Şekil 2.23.	Sehpanın üzerine yeni bir workobject tanımlanması – 2	33
Şekil 2.24.	Robotun home pozisyona çekilmesi	35
Şekil 2.25.	Robota hedef nokta tanıtmak – 1	35
Şekil 2.26.	Robota hedef nokta tanıtmak – 2	36
Şekil 2.27.	Robota hedef nokta tanıtmak – 3	37
Şekil 2.28.	Toolun hedef noktadaki pozisyonunun gözlenmesi	37
Şekil 2.29.	Hedef noktayı kendi etrafında döndürmek – 1	38
Şekil 2.30.	Hedef noktayı kendi etrafında döndürmek – 2	39
Şekil 2.31.	Hedef noktayı kendi etrafında döndürmek – 3	39
Şekil 2.32.	Robotun izleyeceği yolun oluşturulması	40

Şekil 2.33. Robotun Zone hassasiyetinin değiştirilmesi	41
Şekil 2.34. Hedef noktalar arasında çizilen yol.....	42
Şekil 2.35. AutoPath oluşturulması	42
Şekil 2.36. Rota için konfigürasyon yapılması.....	44
Şekil 2.37. Yapılan işlemlerin RAPID ile senkronize edilmesi	44
Şekil 2.38. Simulasyon tercihlerinin yapılması.....	45
Şekil 2.39. Gripper için taban parçası oluşturulması	47
Şekil 2.40. Gripper için hareketli parçaların oluşturulması	48
Şekil 2.41. Union özelliği ile parçaların birleştirilmesi	48
Şekil 2.42. Aksamların tanımlanması	49
Şekil 2.43. Hareketlerin tanımlanması ve denenmesi.....	50
Şekil 2.44. Tooldata değerlerinin belirlenmesi.....	51
Şekil 2.45. Toolun durumlarının belirlenmesi – 1	52
Şekil 2.46. Toolun durumlarının belirlenmesi – 2.....	52
Şekil 2.47. Tanımlanmış olan gripperın robota bağlanması.....	53
Şekil 2.48. Sehpa dizayn edilmesi	54
Şekil 2.49. Gripper ile taşınacak iş parçasının oluşturulması	54
Şekil 2.50. Smart Component tanımlanması	55
Şekil 2.51. JointMover ayarlarının yapılması	56
Şekil 2.52. Mechanism Joint Log ekranında aksamların hareket ettirilmesi	57
Şekil 2.53. LineSensorun grippera bağlanması	57
Şekil 2.54. Attacher eklenmesi.....	58
Şekil 2.55. LogicGate eklenmesi ve NOT kapısı olarak seçilmesi	59
Şekil 2.56. SmartComponent blok diyagramı	60
Şekil 2.57. Blok diyagramında Input sinyali tanımlanması	60
Şekil 2.58. SmartComponent blok diyagramının tamamlanması	61
Şekil 2.59. SmartComponentin tut – bırak fonksiyonunun denenmesi	62
Şekil 2.60. PROFIBUS Device tanımlanması.....	63
Şekil 2.61. PROFIBUS Device üzerine I/O sinyalleri tanımlanması	63
Şekil 2.62. Station Logic ekranı	65
Şekil 2.63. RAPID kod editörü.....	66
Şekil 2.64. RAPID kod editöründe programın yazılması	67
Şekil 2.65. I/O Simulator ekranı.....	68
Şekil 3.1. Station Viewer çıktısının oluşturulması	70
Şekil 3.2. Station Viewer ekran görüntüsü.....	71
Şekil 3.3. HTC Vive VR ile robot simulasyonunun izlenmesi – 1.....	71
Şekil 3.4. HTC Vive VR ile robot simulasyonunun izlenmesi – 2.....	72

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

CAD	: Computer Aided Design (Bilgisayar Destekli Tasarım)
I/O	: Input/Output (Giriş/Çıkış)
IoT	: Internet of Things (Nesnelerin İnterneti)
PLC	: Programmable Logic Controller (Programlanabilir Lojik Kontrolör)
VC	: Virtual Commissioning (Sanal Devreye Alma)
VR	: Virtual Reality (Sanal Gerçeklik)

ENDÜSTRİ 4.0 VE ROBOTİK SİSTEMLERDE SANAL GERÇEKLIK UYGULAMASI

ÖZET

Toplumların tarihsel gelişimi boyunca üretimin sürekli daha az maliyet ile daha çok çıktı alacak şekilde geliştiği görülmektedir. Bu süreçte önce makineleşen üretim, sonradan seri üretim hatlarına ve gelişmiş otomasyon sistemlerine doğru evrilmiştir. Seri üretim hatlarına robotların dahil olmasıyla insanların ürettiğinden çok daha fazla üretim adetlerine ve yüksek kalite değerlerine ulaşılmıştır. Bu gelişmeler sanayide daha çok robotun tercih edilmesine yol açmıştır. Robotlu üretim hatlarının dizaynı ve devreye alınması sürecindeki iş güvenliği riskleri ve kurulum süresi maliyetleri değerlendirildiğinde, çevrimdışı robot programlamanın ve simulasyonun önemi daha iyi anlaşılmaktadır. Son yıllarda tüm dünyanın gündeminde olan Endüstri 4.0 konsepti ise, bilgi teknolojilerini üretim sahasında daha aktif kılarak her alanda dijitalleşmenin yolunu açmıştır. Bu bağlamda ülke kalkınmanın en önemli anahtarlarından birisi, bu teknolojilere hakim olmak ve onları en etkin şekilde kullanabilmektir. Endüstri 4.0 ve robot programlamayla ilgili sanayi uygulamaları ve akademik çalışmalar incelendiğinde, sanal gerçeklik teknolojisinin robotlu üretim hattı dizayn süreçlerine uygulanmasının yeterince işlenmediği görülmüştür. Bu tez çalışması ile, robotlu üretim hattı dizayn ve geliştirme süreçlerinde sanal gerçeklik teknolojisi kullanımının önemini vurgulanması amaçlanmıştır. Bu bağlamda ABB RobotStudio yazılımında, otomotiv sektöründe yaygın uygulama alanı olan iki örnek program yazılıp simulasyonu yapılmış, ardından HTC Vive sanal gerçeklik setiyle sanal gerçeklik ortamında deneyimlenmiştir.

Anahtar Kelimeler: Endüstri 4.0, Otomasyon, Robotik, RobotStudio, Sanal Gerçeklik.

INDUSTRY 4.0 AND VIRTUAL REALITY APPLICATION ON ROBOTIC SYSTEMS

ABSTRACT

It is seen that the manufacturing has been improved in its way to bring more outcome with less cost along the historical progress of the communities. The manufacturing that became mechanized at first in this process, transformed into serial production lines and into advanced automation systems afterwards. Together with the robots that were integrated into the serial production lines, much higher production quantities and higher quality levels were reached comparing with the levels that humans reached before. These news resulted with more robots to be preferred in industry. The benefits of offline robot programming and simulation are better understood when the health and safety risks and the cost of installation time in the design and commissioning of robotic production lines are assessed. The Industry 4.0 concept, which has been on the agenda of the whole world in recent years, has opened the way to digitalization in every field by making information technologies more active in production. In this context, one of the most important keys of country development is to dominate these technologies and to be able to use them most effectively. When examining industrial applications and academic studies related to Industry 4.0 and robot programming, it has been seen that the application of virtual reality technology to robotic production line design processes has not been sufficiently processed. With this thesis, it is aimed to emphasize the importance of using virtual reality technology in designing and developing of robotic production lines. In this context, two different applications that have wide usage area in automotive sector were programmed and simulated in ABB RobotStudio software, then they were experienced in virtual reality environment with the HTC Vive virtual reality system.

Keywords: Industry 4.0, Automation, Robotics, RobotStudio, Virtual Reality.

GİRİŞ

Teknolojinin gelişimi ve üretime yansımaları tarihi boyunca, üretimin sürekli olarak daha az maliyetle daha çok çıktı vermesi beklenmiştir. Sürekli değişen piyasa koşullarında rekabet edebilmek için, üretimdeki verimliliği artırıp maliyetleri azaltmak kaçınılmaz bir gerekliliktir. Bu ihtiyaca cevap verebilmek adına, otomasyon ve kontrol sistemlerinin gelişim sürecinde endüstriyel robotlar ortaya çıkmıştır.

Endüstriyel robotlar temel olarak, insanların yaptığı rutin işlerde insanların yerlerini alıp verimliliği arttırmak üzere tasarlanmıştır. İş güvenliği riski olmadan ağır işleri yapabilmeleri, hızlı iş çıkarabilme kapasiteleri, yüksek kalite çıktıları, periyodik bakımları vaktinde ve doğru olarak yapıldığı sürece uzun zaman çalışabilmeleri gibi başlıca avantajları, sanayideki kullanım alanlarını hızla arttırmalarını sağlamıştır.

Robotlar ülkemizde ve dünyada çok çeşitli sektörlerde kullanım alanı bulmuşlardır. Otomotiv sektöründe kaynak, boya, montaj, kalite kontrol gibi alanlar, gıda sektöründe üretim, paketleme, kalite kontrol ve ilaç sektöründe de yine benzer alanlar buna örnek gösterilebilmektedir. Sanayide hızla daha çok yer edinen robotların kurulumu, programlanması ve devreye alınması önemli bir teknik beceri olarak öne çıkmaktadır.

Önceleri çevrimiçi yapılan programlamada robotun ve dolayısıyla üretim uzun süreler durdurulması gerekmektedir. Çevrimdışı robot programlamada ise robotlu üretim hatları ofis ortamında bilgisayarda dizayn edilip, programlanıp, simule edilebilmektedir. Günümüzde robotlu üretim hatları önce bilgisayar ortamında dizayn edilmekte ve programlanmakta, ardından hat fiziksel olarak kurulup robot programlarının ince ayarları yapılmaktadır. Çalışan üretim hatlarında ise çevrim sürelerini iyileştirme çalışmaları, yine uzun süreler hattı durdurmadan ofis ortamında yapılmakta, ardından planlı duruş dönemlerinde devreye alınabilmektedir.

Son yıllarda hızla tüm sektörlerde bir dönüşüm dalgasına yol açan Endüstri 4.0 konseptiyle gelişen teknolojilerden biri olan sanal gerçeklik teknolojisinin robot programlama sürecine entegre edilmesiyle, robotlu üretim hatlarının kurulumu ve çevrim süresi iyileştirme çalışmaları çok daha farklı bir boyut kazanacaktır. Bu tez çalışması, bu alandaki akademik ve sanayi çalışmalarına ışık tutabilmek amacıyla yapılmıştır.

Çevrimdışı robot programlama alanındaki literatür taramasında çevrimdışı robot programlama için ABB RobotStudio yazılımının kullanımına [1-5], RobotStudio yazılımında ark kaynağı uygulamasına [6], RobotStudio yazılımında paletleme uygulamasına [7], Robot Studio yazılımına dışarıdan bir CAD modeli eklenerek yapılan uygulamaya [8] rastlanmıştır.

Sanal gerçeklik alanındaki literatür taramasında sanal gerçekliğin tarihsel gelişimi çalışmasına [9], sanal gerçekliğin uygulama alanlarına [10], eğitim sektöründe sanal gerçeklik uygulamalarına [11-13], turizm sektöründe sanal gerçeklik uygulamalarına [14], klinik terapide sanal gerçeklik uygulamalarına [15], sanal gerçeklik ile artırılmış gerçekliğin teknolojik açıdan karşılaştırmalarına [16-17] rastlanmıştır.

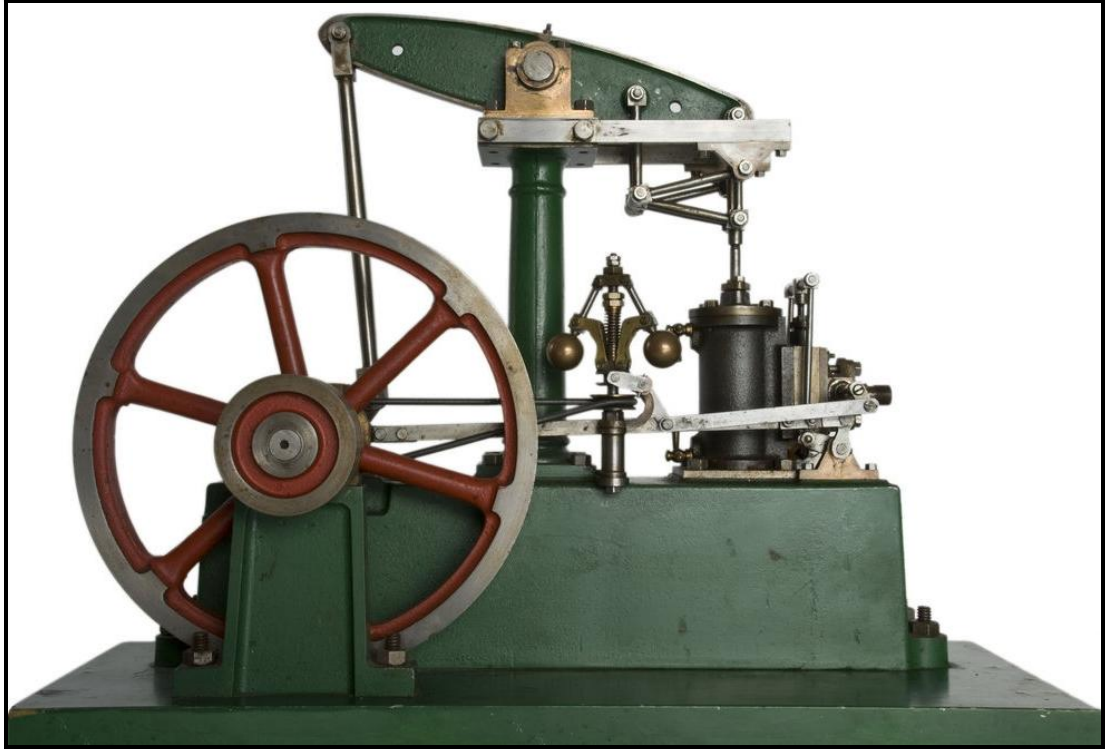
Bu çalışmada, robotlu üretim hatlarının dizaynı ve programlanması süreçlerini zaman, maliyet ve iş güvenliği açısından iyileştirebilmek amacıyla sanal gerçeklik teknolojisinin kullanımı incelenmektedir. Bu bağlamda, bu çalışmanın ilk kısmında sanayinin günümüze kadar geçirdiği devrimler tarihsel açıdan incelenip Endüstri 4.0 konseptinin sanayide yol açtığı dönüşümler ortaya konulmuştur.

Çalışmanın ikinci kısmında, ABB RobotStudio yazılımının kullanımı incelenmiş, otomotiv sektöründe yoğun kullanım alanı olan nokta izleme ve parça al-bırak uygulamaları programlanmış ve simülasyonları yapılmıştır.

Çalışmanın son bölümünde, RobotStudio yazılımında hazırlanan simülasyonlar, HTC Vive sanal gerçeklik sistemiyle deneyimlenmiş ve sonuçlar paylaşılmıştır.

1. ENDÜSTRİ 4.0

18. yüzyılın başlarında buhar gücünün keşfiyle, Şekil 1.1.'de bir örneği görülen ilk buhar makineleri üretilmiş ve dokuma tezgahlarında kullanılmaya başlanmıştır. İnsan kas gücünün yerini buhar makinelerinin almasıyla, önceleri tarımda çalışan iş gücü zaman içinde makineleşen üretime kaymıştır. Bu durum kırsaldan kentlere göçe yol açmış, büyük devletleri daha fazla üretebilmek amacıyla hammadde arayışına yönlendirmiştir. Bu süreç dünya tarihinde sanayi devrimi olarak adlandırılmaktadır.



Şekil 1.1. Buhar makinesi [18]

Bundan yaklaşık 100 yıl sonra, elektriğin keşfi ve Şekil 1.2.'de bir örneği görülen elektrik motorlarının sanayiye entegre olmasıyla, seri üretim hatları ortaya çıkmıştır. İnsanların seri üretim hatlarında çalışmaları sonucunda üretim adetleri hızla yükselmiştir. Bu sayede daha az zamanda daha çok sanayi ürünü piyasaya sürülebilmiştir. Ürünlerin maliyetinin düşmesi ve ulaşılabilirliğinin artması gibi süreçler toplumsal hayatı da etkilemiştir.

Böylece kentlerin kalabalıklaşması, daha fazla işgücünün ortaya çıkmasını sağlamış ve beraberinde daha fazla sanayi üretimini getirmiştir. Sanayi üretiminin elektrik enerjisi ile dönüşüme uğradığı bu dönem, ikinci sanayi devrimi olarak değerlendirilmektedir.



Şekil 1.2. Elektrik makinesi [19]

20. yüzyılda elektroniğin gelişmesi ve günümüzden 40-50 yıl kadar önce, Şekil 1.3.'te örnekleri görülen PLC'lerin geliştirilip sanayiye entegre olmasıyla üretimde otomasyona geçilmiştir. PLC'lerin sanayide kullanımıyla seri üretim hatlarının kontrol edilebilirliği artmıştır. Bu durum üretimde çeşitliliğin ve üretim çıktılarının artmasını sağlamıştır. İnsan kas gücüne gereksinimin azaldığı bu dönemde, teknik bilgi ve beceriye olan ihtiyaç artmıştır. Otomasyonun sanayi üretimini dönüştürdüğü bu dönem, üçüncü sanayi devrimi olarak değerlendirilmektedir.

Otomasyonla üretim sürecinde sensörler, valfler, motorlar, robotlar gibi çok çeşitli girdi ve çıktı araçları tarafından bilgi üretilmektedir. Üretilen bu büyük miktarda bilginin işlenilip yeniden kullanılması ve bunun da ötesinde tüm nesnelerin veri üretebilir hale getirilmesi yaklaşımı, dördüncü sanayi devrimi kavramını doğurmuştur. Bu bağlamda "Endüstri 4.0" ifadesi ilk kez 2011 yılında Hannover Fuarı'nda kullanılmıştır.



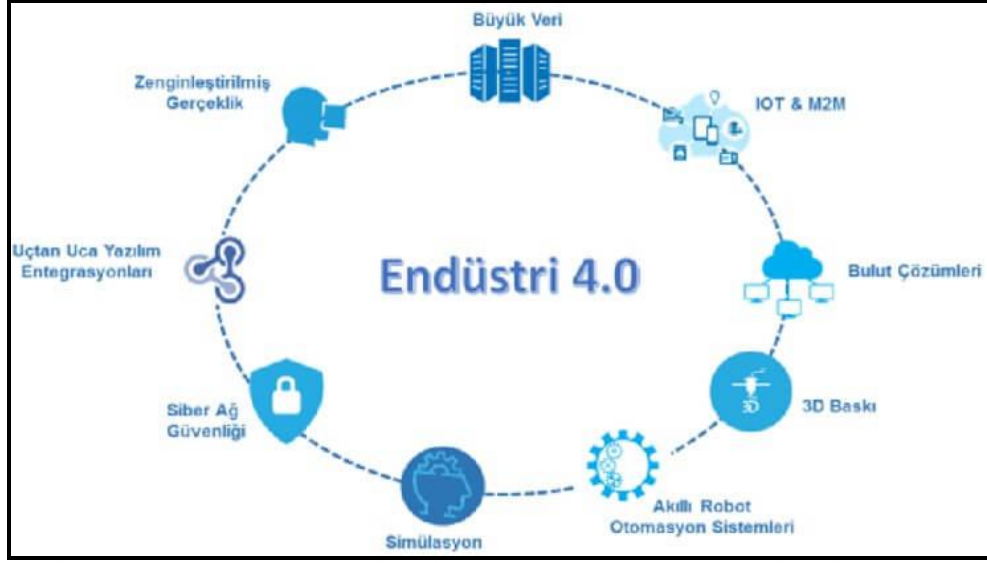
Şekil 1.3. PLC (en öndeki bir I/O modülü) [20]

Bir çalışmaya göre [21], IoT teknolojisinin üretim alanlarına entegrasyonu, dördüncü sanayi devriminin ana tetikleyici sebeplerindendir. Siber fiziksel sistemler, şirketlerin tüm makinelerini, tedarik zinciri sistemlerini, üretim sistemlerini tek bir network ağı altında toplayabilmelerine fırsat vermektedir. Bu durum akıllı makinelerin ve üretim alanlarının, birbirleri arasında otonom olarak bilgi alışverişi yapmalarını ve kendi iş sistemlerini bağımsız olarak denetleyebilmelerini sağlamaktadır. Endüstri 4.0 konseptinde akıllı fabrikalar, akıllı ürünlerin yalnızca piyasa taleplerine göre değil, aynı zamanda bireysel tüketici taleplerini de göz önüne alarak daha esnek koşullarda üretilebileceği yeteneklerle donatılmaktadır. Bu dinamik iş modelinin arkasındaki kilit etken ise üretim alanlarından elde edilen büyük veridir.

1.1. Büyük Veri

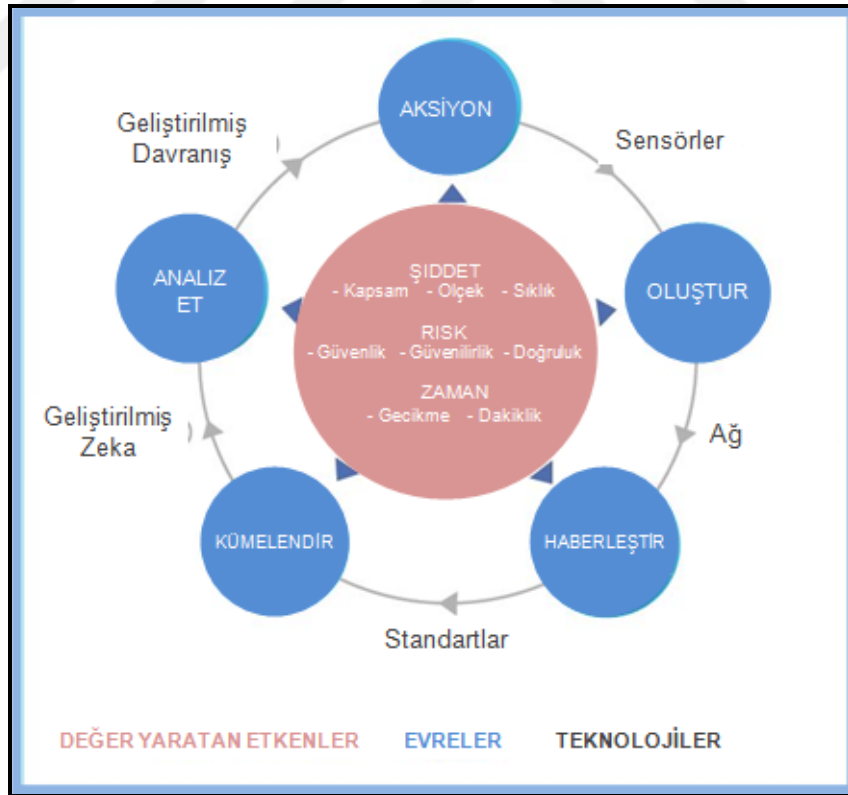
Dördüncü sanayi devriminin kilit noktası, çeşitli nesnelere tarafından üretilen büyük miktarda verinin işlenmesi sonucu üretilen değerli bilgidir.

Nesnelerden sağlanan büyük miktarda verinin analiz edilip tekrar kullanılması ile son yıllarda “akıllı fabrikalar”, “siber fiziksel sistemler”, “arttırılmış gerçeklik”, “otonom robotlar”, “3d yazıcılar”, “bulut bilişim” ve “nesnelerin interneti” olarak adlandırılan ve Şekil 1.4.’te gösterilen yeni araştırma ve uygulama alanları ortaya çıkmıştır.



Şekil 1.4. Endüstri 4.0 [22]

Yapılan bir çalışmaya göre [23], IoT teknolojisi nerdeyse tüm nesnelere bir veri kaynağına dönüştürmeyi amaçlamaktadır. Bu çalışmada, bilgiyi değere dönüştürmek için izlenen yollar sırasıyla Şekil 1.5.' te gösterilmiştir.



Şekil 1.5. Bilgi Değer Döngüsü [23]

Bu döngüye göre, bir hareket sensör tarafından yakalanıp tanımlanmakta ve bu yolla bilgi üretilmektedir. Üretilen bilgiler network üzerinden haberleşmekte ve belirli standartlar çerçevesinde biraraya gelmektedir. Ardından bu bilgiler arttırılmış zeka ile analiz edilmektedir. Analiz sonuçlarının başlangıçtaki davranış üzerine uygulanmasıyla devrenin sonunda geliştirilmiş davranış elde edilmektedir.

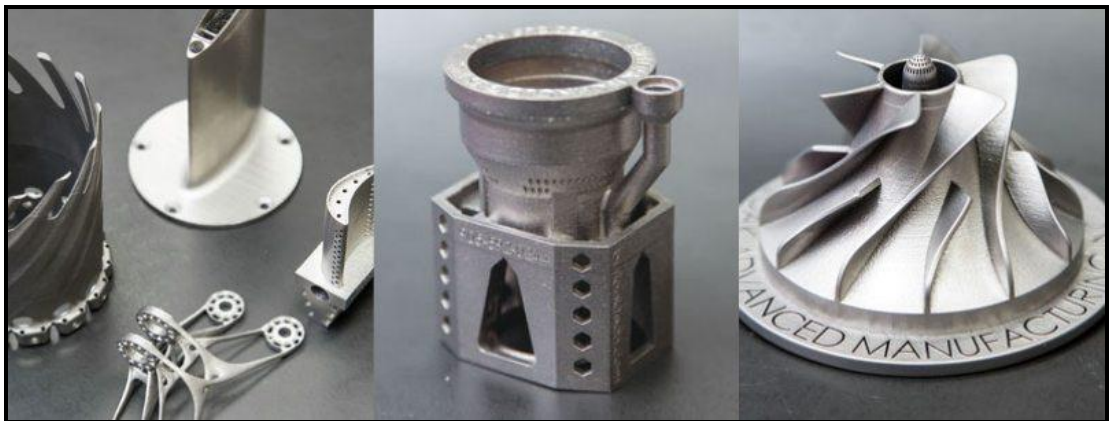
1.2. Değişen İş Kolları

Değer yaratan bu bilgi sanayiye ve toplumu da birçok farklı alanda dönüştürmektedir. Sanayinin dönüşüme uğradığı bu dönemde birçok iş kolu kaybolmakta ve yerine birçok yeni iş kolu ortaya çıkmaktadır.

Otomasyon çağında robotik sistemler, imalat alanlarındaki tekrar eden işleri yaparak vasıfsız işgücünün yerini hızla işgal etmiştir. Bununla birlikte otomasyon, robotik, mekatronik gibi bu dönemde ortaya çıkan yeni uzmanlık alanları da yeni iş alanları doğurmuştur. Endüstri 4.0 çağında akıllı sistemler, imalat ve ofis ortamlarında, kalite kontrol veya planlama gibi tekrar eden işleri gerçekleştiren vasıflı işgücünün yerini alması öngörülmektedir. Otomasyon çağındaki gibi, tekrar eden işleri insanlardan alan bu süreç, aşağıda tarif edilen yeni teknolojilerin ortaya çıkmasına ve yayılmasına önderlik etmiştir:

1.2.1. Katmanlı imalat

3d yazıcıların geliştirilmesiyle birlikte katmanlı imalat teknolojisi yaygınlaşmaya başlamıştır. Bu teknolojiyle, Şekil 1.6.'da örnekleri görülen ve geleneksel üretim teknikleriyle üretilmesi zor olan parçalar hızla ve düşük maliyetle üretilmektedir.



Şekil 1.6. Katmanlı imalat tekniğiyle üretilmiş parçalar [24]

Parçaların katman katman üretilmesi, değişen pazar koşullarına veya bireysel müşteri taleplerine hızla cevap verebilme yeteneğini de birlikte getirmiştir. Bu şekilde dizayn değişimi hızla üretime yansıtılabilmekte, prototip maliyeti düşürülmekte ve tek bir ürün bile kar edilebilir maliyetlerle üretilebilmektedir.

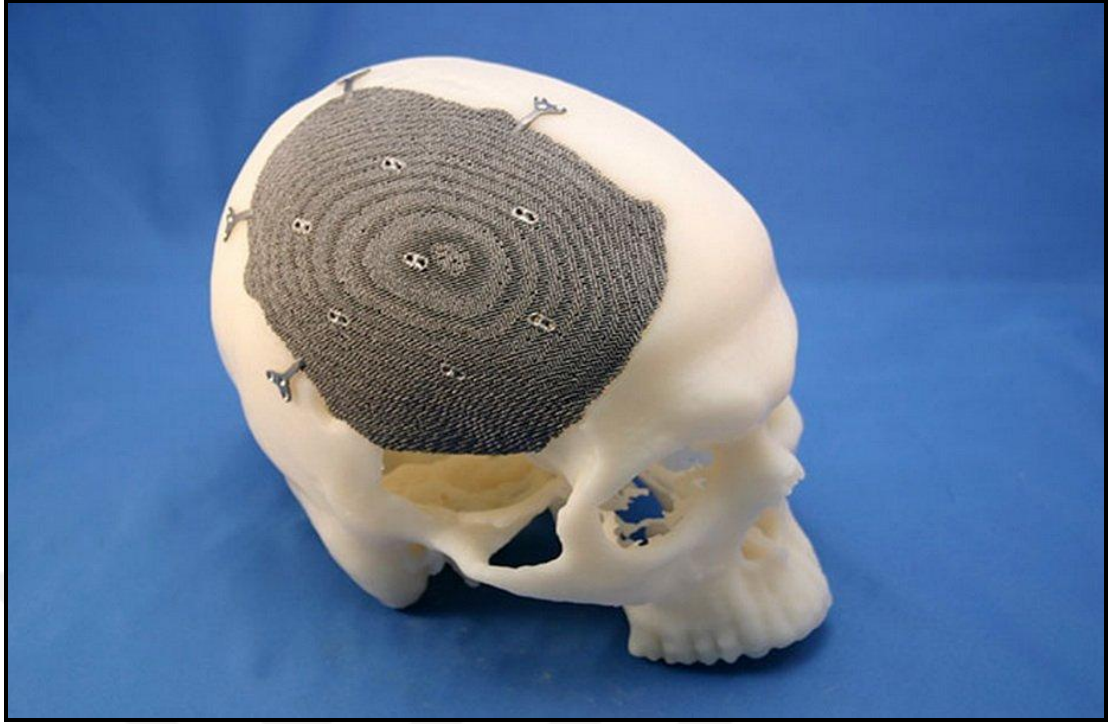
Katmanlı imalat tekniği yalnızca sanayi üretimi değil, birçok sektörde yeni ufuklar açmıştır. İnsanların yalnızca dizayn kodlarını satın alıp istediği ürünü ürettirebilmesi veya kendi evlerine alacakları bir 3d yazıcı ile üretebilmeleri fırsatıyla moda sektörü yeni bir yola girmektedir. Şekil 1.7.'de, katmanlı imalat tekniğiyle üretilmiş bir çift spor ayakkabı görülmektedir.



Şekil 1.7. Katmanlı imalat tekniğiyle üretilmiş ayakkabı [25]

Katmanlı imalat tekniği medikal alanda da yeniliklere yol açmıştır. Yapay uzuv üretilebilmesi, vücudun deforme olmuş bölümlerine destek parçalar üretilebilmesi, bu tekniğin dizayn esnekliğinin bir sonucu olarak öne çıkmaktadır.

Şekil 1.8.'de, katmanlı imalat tekniğiyle üretilmiş ve kafatası modeli üzerine montajı yapılmış bir parça görülmektedir.



Şekil 1.8. Katmanlı imalat tekniğiyle üretilmiş kemik desteği [26]

1.2.2. Cobot – işbirlikçi robotlar

Geleneksel endüstriyel robotlar programlanmış işleri yapmaktadır. Bu işleri yaparken çevrelerindeki insanlara tehlike yaratabilecekleri için de çalışma alanlarının çevresi çitlerle çevrilmekte ve hat güvenlik kapılarıyla programlarının sürekliliği sağlanmaktadır. Cobotlar ise güvenlik çitlerine ihtiyaç duymadan insanlarla birlikte güvenle çalışabilmektedir.

Şekil 1.9.'da bir örneği görülen Cobotlar, donatıldıkları sensör sistemleri sayesinde, en ufak dokunuşlara dahi hassasiyet gösterip anında hareketsiz kalmaktadırlar. Bu özellikleri, insanlarla yüksek güvenlik içinde çalışabilmelerini sağlamaktadır.

Üretim ortamlarında keskin kenarlı veya sıcak malzemeleri taşıyabilmeleri, erişilmesi zor yerlere erişebilmeleri gibi özellikleriyle insanlar için riskli olan işleri devralabilmeleri de, cobotların önemli avantajlarından. Böylece iş kazaları azalmakta ve insanlar üretim ortamlarında katma değeri daha yüksek işlere odaklanabilmektedirler.



Şekil 1.9. Cobot - İşbirlikçi robot [27]

Geleneksel endüstriyel robotlara göre daha hafif olan cobotlar üretim hattında çok daha farklı yerlerde konumlanabilirken aynı zamanda bu hafiflikleri dolayısıyla istendiği zaman başka bir bölgeye taşınıp başka işlerde kullanılabilirler.

Cobotlar geleneksel endüstriyel robotlar gibi özel programlama dillerine bağımlı olmayıp, öğrenebilme yeteneğine de sahiptirler. Böylece örneğin birlikte çalıştığı insan tarafından bir konuma hareket ettirilen bir cobot, bu hareketi öğrenip bir dahaki sefer otomatik olarak kendisi yapabilmektedir.

1.2.3. Akıllı fabrikalar

Akıllı fabrikalar dizayn, üretim ve bakım süreçleri olarak üç safhada ele alınabilir.

1.2.3.1. Dizayn süreci

Geleneksel fabrikalarda dizayn süreci bilgisayar ortamında tamamlandıktan sonra prototip üretimine geçilmektedir. Bunun sonrasında prototipten edinilen bilgiler, ardından gelen seri imalat fazında edinilen bilgiler, en sonunda ise müşterilerden gelen geribildirimler ile dizaynda iyileştirmeler yapılmaktadır.

Akıllı fabrikalarda ise dizayn süreci, prototip, seri imalat ve müşteri bir bütün olarak ele alınmaktadır. Bir müşterinin talebiyle tekil bir parçanın dahi o anda dizayn edilip kar edebilecek şekilde üretilebilmesi bu konseptin temelini oluşturmaktadır. Pazar dinamiklerine hızla yanıt verebilmesi, dizaynı hızla değiştirip yeni hat kurmadan

mevcut üretim hattında yeni ürünü üretebilmesi, akıllı bir fabrikadan beklenen özelliklerdendir.

1.2.3.2. Üretim süreci

Geleneksel fabrikalarda üretim süreçleri otomasyon çağında kontrol edilebilirlik yönünden gelişmiş olmasıyla birlikte, birçok sistem hala manuel olarak işlemektedir. Ekipmanların etkinlikleri operatörlerin tuttıkları arıza ve çıktı kayıtlarıyla değerlendirilmekte, bu ise verilerin güvenilirliği ve kayıt tutulması için harcanan zaman açısından dezavantaj yaratmaktadır.

Akıllı fabrikalarda ise üretim sistemine dahil olan tüm ekipmanların arıza ve çıktı kayıtlarının gerçek zamanlı olarak izlenebilir olması amaçlanmaktadır. Bulut bilişim teknolojisi, tüm bu verinin gerçek zamanlı izlenebilmesi için tercih edilmektedir. Bu şekilde ekipmanların etkinliği daha az insan eforuyla ve güvenilir bilgilerle takip edilebilmektedir.

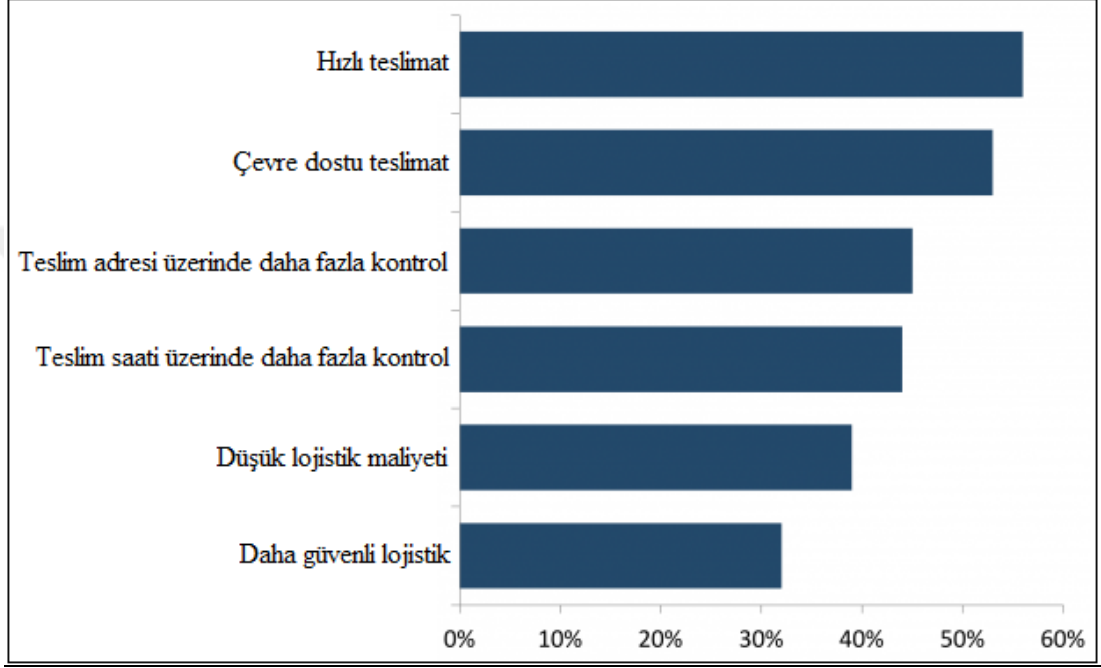
1.2.3.3. Bakım süreci

Geleneksel fabrikalardaki kestirimci bakım süreçleri, ekipmanların belirli parametrelerinin operatör tarafından izlenip, ardından özel cihazlarla analiz edilerek arızanın ortaya çıkmadan önce tespit edilip düzeltilmesini amaçlamaktadır. Kestirimci bakımda ekipmana genellikle yağ analizi, sıcaklık analizi, ultrasonik test, titreşim analizi gibi testler yapılmaktadır.

Akıllı fabrikalarda ise kestirimci bakım için ekipmanın yanına gidip gözlem yapmaya gerek yoktur. Bunun yerine ekipmanın üzerine yerleştirilmiş gelişmiş sensörler ile yağı, sıcaklığı, titreşim seviyesi gibi kestirimci bakımda analiz edilen tüm değerler sürekli olarak gözlem altında tutulmaktadır. Elde edilen bu büyük veri özel bir yazılım ile analiz edilip sonuçların normal değerler ile karşılaştırılmasıyla ekipmanın arızaya geçebilme durumu önceden kestirilmekte ve müdahale edilebilmektedir.

1.2.4. İnsansız hava araçları

Üzerinde mürettebatı olmayıp, yerden veya uydu üzerinden kontrol edilebilen hava araçları insansız hava aracı olarak adlandırılmaktadır. Askeri alanda bunların silahlı modelleri de olduğu gibi, bireysel kullanım için tasarlananlar daha çok drone olarak bilinmektedir.



Şekil 1.10. ABD’li tüketicilerin drone teslimatı algısı [28]

Dronelar ağırlıklı olarak bireysel hobi amaçlı pazarlanmakla birlikte, bazı sektörlerde yeni iş modellerine öncülük etmektedir. E-ticaret sektöründe faaliyet gösteren büyük firmalar, droneli teslimat yöntemlerini denemeye başlamışlardır. Amazon firmasının şimdilik kısıtlı mesafelere kısıtlı ürün gamıyla pilot denemelerini yaptığı “Prime Air” servisi bunlardan biridir. Bu projeye tüketicinin evinde internet üzerinden satın aldığı ürünün bir drone ile çok kısa bir süre içerisinde tüketicinin evine varışı hedeflenmektedir. ABD’li tüketiciler üzerinde yapılan bir araştırmanın Şekil 1.10. da görülen sonuçları da tüketicilerin drone teslimatına genel olarak olumlu yaklaşımlarını göstermektedir.



Şekil 1.11. Amazon Prime Air Servisi [29]

1.3. Endüstri 4.0'ın Gelir, Maliyet ve Verimlilik Yönünden Faydaları

Büyük verinin akıllı bilgiye dönüştürülmesiyle birlikte, Endüstri 4.0'ın çığır açan gelir, maliyet ve verimlilik avantajları ortaya çıkmıştır.

Bu alanda bir araştırma [30], Endüstri 4.0'ın ürün ve servisleri dijitalleştirme, yeni dijital ürünler ortaya çıkarma, büyük veri analizinden servis alanında faydalanma, ürünleri kişiselleştirme yoluyla işletmelerin gelirlerini arttırdığını ortaya koymuştur. Bununla birlikte büyük veri analizinden gerçek zamanlı kalite kontrol alanında faydalanma, üretimin esnekliğini ve adaptasyonunu arttırma, kestirimci algoritmalar kullanarak ekipman etkinliğini arttırma yoluyla sistemlerin verimliliği arttırdığını da ortaya koymuştur.

Endüstri 4.0'ın bu gibi faydalarının yanısıra, “Sanal Devreye Alma” bu süreçte gelişen yeni teknolojilerden birisi olarak öne çıkmaktadır.

2. SANAL DEVREYE ALMA

Üretim hatları için sanal devreye alma teknolojisi, dördüncü sanayi devriminin getirdiği yeniliklerden biri olarak ön plana çıkmaktadır. Geleneksel yaklaşımda, üretim hattının otomasyon programı yazıldıktan sonra hat gerçek ortamda mekanik olarak kurulmaktadır. Sonrasında otomasyon programı PLC üzerinden sahadaki cihazlarla test edilip, üretim hattının devreye alma işlemi yapılmaktadır. Bu süreçte programdaki ve mekanik kurulumdaki hatalar gözlenmekte ve giderilmekte, aynı zamanda hatta üretilecek ürünün seri üretim için istenen kalite seviyesine ulaşması amaçlanmaktadır.

Yapılan bir çalışmada, devreye almanın fabrika kurulum sürecinin %25 ini ve kontrol yazılımındaki hataları gidermenin ise bu sürenin %15 ini oluşturduğu belirtilmiştir. Bu çalışmada, bu probleme çözüm olarak sanal devreye alma önerilmiştir. [31]

Sanal devreye alma teknolojisinde, devreye alma süreci bilgisayar ortamında ve PLC (gerçek veya sanal) yardımıyla gerçekleştirilmektedir. Bunun amacı planlama ve dizayn süreçlerindeki hataların önceden ortaya çıkarılması ve giderilmesidir. Bunun için üretim hattının eksiksiz mekanik dizaynı yapılmakta ve hatta kullanılacak tüm elemanlar simulasyon yazılımında modellenmektedir. Otomasyon programının bu simulasyon ile haberleştirilmesi sayesinde sistemin gerçek zamanlı çalışması gözlenmekte ve tüm devreye alma çalışmaları bu sistem üzerinde yapılabilmektedir.

Sanal devreye alma uygulamalarını sınıflandırma ve standartlaştırma üzerine yapılan bir çalışmada [32], sanal devreye alma sistemleri detaylarına göre 4 seviyeye ayrılmıştır. Bunlar sırasıyla aşağıdaki gibi tariflenmiştir:

- Makroskopik: Bu seviyede modüller kabaca gösterilir ve içinde hat bileşenleri yoktur. Sadece zaman temelli davranış ele alınmıştır, işlenen veya hareket ettirilen ürünler ele alınmamıştır.
- Mezoskopik (dinamik olmayan): Modüller daha detaylı gösterilir ve içinde hat bileşenleri olan elektriksel (sensor, aktuator) veya mekanik (hidrolik, prömatik)

cihazlar da gösterilir. Zaman temelli davranış ele alınmıştır, ilave olarak işlenen veya hareket ettirilen ürünler de ele alınır. Ancak hareket, önceden tanımlı yollar üzerinde mümkündür.

- Mezoskopik (dinamik): Modüller daha detaylı gösterilir ve hat bileşenleri fiziksel veya dinamik davranış gösterebilir. Ürünler için hareket, önceden tanımlı yollar üzerinde mümkündür.
- Mikroskopik: Modüller daha detaylı gösterilir ve hem hat bileşenleri, hem de ürünler fiziksel ve dinamik davranışlar gösterirler. Ürünler serbestçe hareket edebilir.

Sanal devreye alma üzerine yapılan deneysel bir çalışmada [33] sanal devreye almanın, geleneksel devreye almaya göre zaman yönünden avantajı gösterilmiştir. Bu çalışma 2 grup kontrol programcısıyla yürütülmüştür. Her bir grup 30 kişiden oluşmaktadır. Bir grup, bir makinanın devreye alınması için sanal devreye alma yöntemi uygulamıştır. Sonuçlar, sanal devreye alma uygulamayan diğer grup ile karşılaştırılmıştır. 10 adet aktuator çıkışı ve 17 adet sensor girişi kullanılan 1 adet Siemens S7-300 PLC ile bir konserve kutusu presi, test ünitesi olarak kullanılmıştır. Bir grup PLC'yi programlamış ve ardından programı gerçek makina üzerinde, gerçek devreye alma sürecinde test etmiştir. Diğer grup ise sanal bir makina modeli kullanarak programlama yapmıştır. Bu grup programlarını sanal model üzerinde doğruladıktan sonra gerçek makine üzerinde devreye alma sürecine geçmiştir. İkinci grup bu yolla %75 daha kısa devreye alma süresi elde etmiştir. Bu sonuç temel olarak programın sanal devreye alma sürecinde artırılmış olan performansından kaynaklanmaktadır ve bu tekniğin avantajını göstermektedir. Bununla birlikte, sanal makine modeli ikinci grubun çalışmasının başında hazır bulunuyordu ve bu modelin tasarım süreci, bu çalışmada göz önünde bulundurulmamıştır. Çalışmanın sonucunda yazarlar, basitleştirilmiş ve hızlandırılmış bir model oluşturma sistemine olan gereksinimi işaret etmişlerdir. [34]

2.1. Siemens Process Simulate

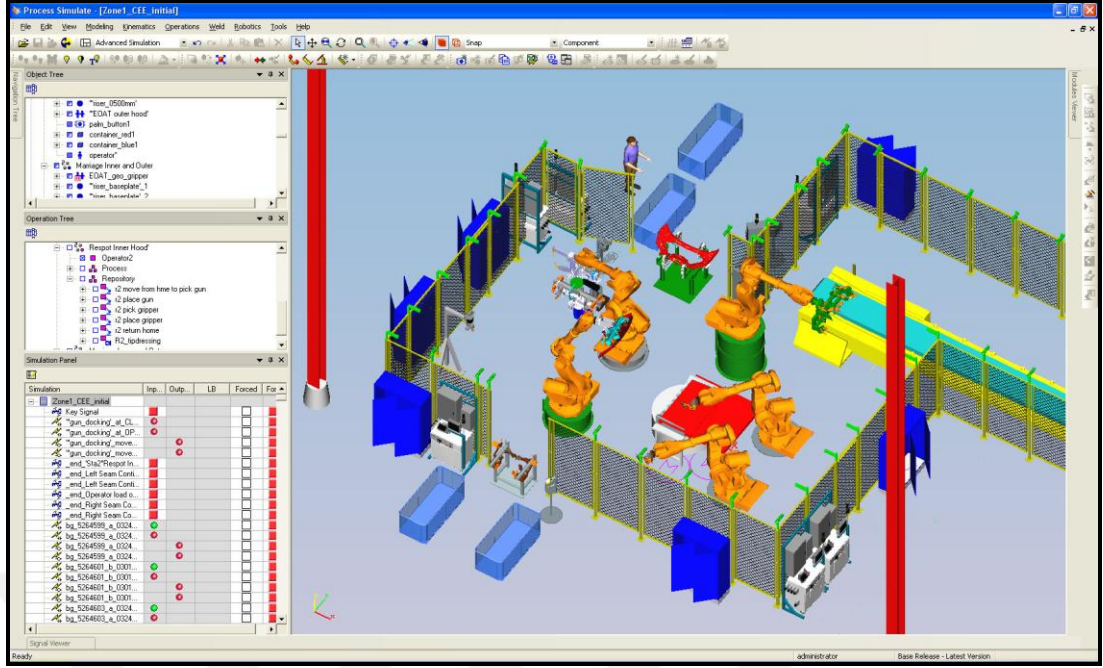
Process Simulate, Siemens tarafından geliştirilmiş ve imalat proseslerinin 3 boyutlu ortam üzerinde dijital olarak tasarlanıp denenmesini sağlayan bir yazılımdır. [35]

Yapılan bir çalışmada [36], Siemens Process Simulate yazılımı üzerinde sanal devreye alma sürecinin basitleştirilebilmesi ve hızlandırılabilmesi için sırasıyla takip edilmesi gereken adımlar şu şekilde belirtilmiştir:

- Sistemin karakterize edilmesi.
- Bilgisayar destekli tasarım yapılması.
- Sanal ortamın oluşturulması.
- Sanal ortamın test edilmesi.
- Sanal ortamın bir dijital ikiz olarak kullanılması.

Bu adımlar şu şekilde detaylandırılabilir:

- Sistemi karakterize etmek için sistemin çalışma prensipleri tanımlanmalıdır.
- Sistem NX (Siemens PLM Yazılımı) veya CATIA gibi bir CAD yazılımında dizayn edilmelidir. Dizayn için herhangi bir CAD yazılımı kullanılabilir, bununla birlikte hazırlanan tasarımın .JT dosya uzantısına dönüştürülebilmesi için sonunda tasarımın yine NX yazılımına alınması gerekmektedir.
- .JT dosya uzantılı tasarım Process Simulate yazılımına alınmalı ve burada .COJT dosya uzantısına çevrilmelidir.
- Process Simulate yazılımı üzerinde hareketler tanımlanarak sanal ortam oluşturulmalıdır. Bu süreçte iç lojik blokları da tanımlanmalıdır.
- PLC programı hazırlanmalıdır.
- PLC programı, PLCSIM üzerinden Process Simulate yazılımıyla entegre olmalıdır. PLCSIM, Siemens tarafından geliştirilmiş ve PLC programlarının simüle edilmesini ve test edilmesini sağlayan bir yazılımdır.
- PLC programı simulasyon üzerinde doğru bir şekilde çalıştıktan sonra PLC'ye aktarılmalı ve gerçek sistem üzerinde test edilmelidir.
- PLC programı gerçek sistem üzerinde de doğru bir şekilde çalıştıktan sonra, simulasyon bir dijital ikiz olarak kullanılabilir. Bu aşamada PLC ve Process Simulate arasındaki bağlantı Siemens OPC yazılımı üzerinden sağlanmaktadır.



Şekil 2.1. Siemens Process Simulate [37]

2.2. ABB RobotStudio

RobotStudio, ABB firmasının geliştirdiği ve robot istasyonlarının modellenildiği, çevrimdışı programlandığı ve simule edilebildiği bir yazılımdır.

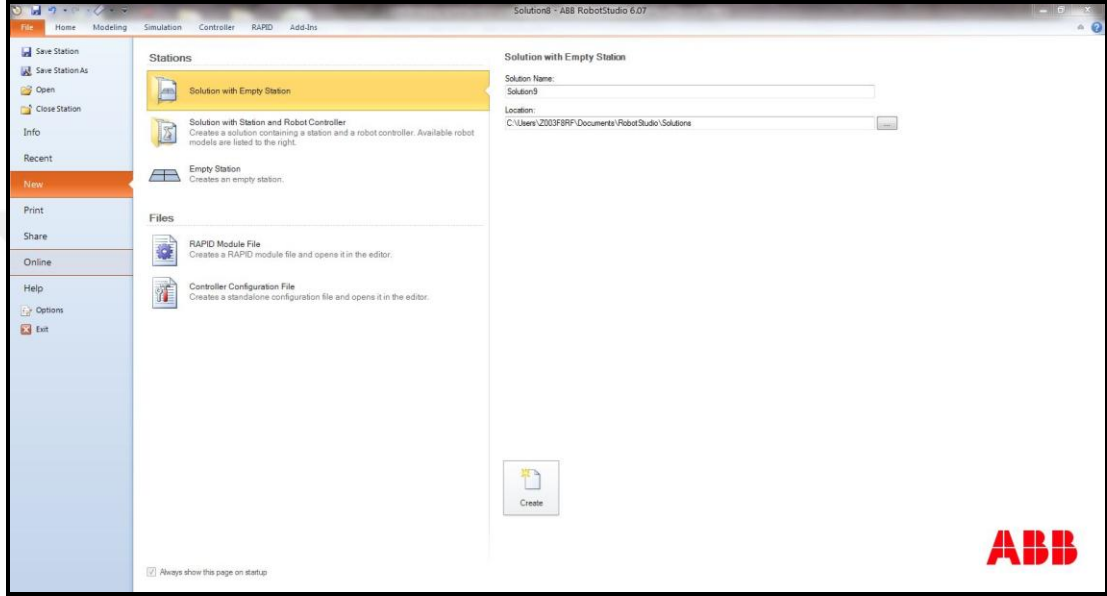
Sanal devreye alma teknolojisinde çevrimdışı modelleme ve programlama ile simulasyonun güçlü avantajları ortaya çıkmaktadır. Bu avantajları aşağıdaki gibi sıralayabiliriz:

- Üretim hattını fiziksel olarak kurmadan önce sanal ortamda yapılan dizayn ile sistemin davranışı gözlenebilir ve bu aşamada sistem optimize edilebilir.
- Üretim hatları için çevrim zamanı iyileştirme projelerinde, üretim sekteye uğratılmadan program iyileştirmeleri ofis ortamında çevrimdışı olarak yapılabilir.
- Robot istasyonunun dizaynı, robotun programlanması gibi teknik beceriler proje ve bakım ekiplerine iş güvenliği riski yaratmadan ve istenildiği kadar deneme yoluyla kazandırılabilir.

2.2.1. Yeni istasyon oluşturulması

ABB RobotStudio programını açtığımızda ilk olarak Şekil 2.2.'de görülen sayfaya karşılaşmaktayız. Burada “New” sekmesinde, istasyon oluşturmak için çeşitli seçenekler kullanıcının seçimine sunulmaktadır.

Boş bir istasyon açıp robotları sonradan ekleyebilmek için burada “Solution with Empty Station” seçeneği seçilebilir. Bu seçeneğin hemen sağ yanında bulunan “Solution Name” kutucuğunda istasyonumuza isim verebilmekte ve onun altında bulunan “Location” kutucuğunda da istasyonumuzun kaydedileceği dosya yolunu belirleyebilmekteyiz. Bu seçimlerin ardından ekranın orta alt kısmında görülen “Create” butonuna tıklanmalıdır.



Şekil 2.2. Yeni istasyon açılış sayfası

“Create” butonuna tıklanmasının ardından Şekil 2.3.’te görülen boş istasyonumuz açılmaktadır. Boş istasyonun ilk açılışında standart olarak dört bölümlü bir düzlem gelmektedir. Düzlemin hemen sol alt köşesinde görülen x-y-z eksen göstergesi, düzlemde konumlanacak cisimlerin koordinat düzlemindeki konumları için bir kılavuz görevi görmektedir.

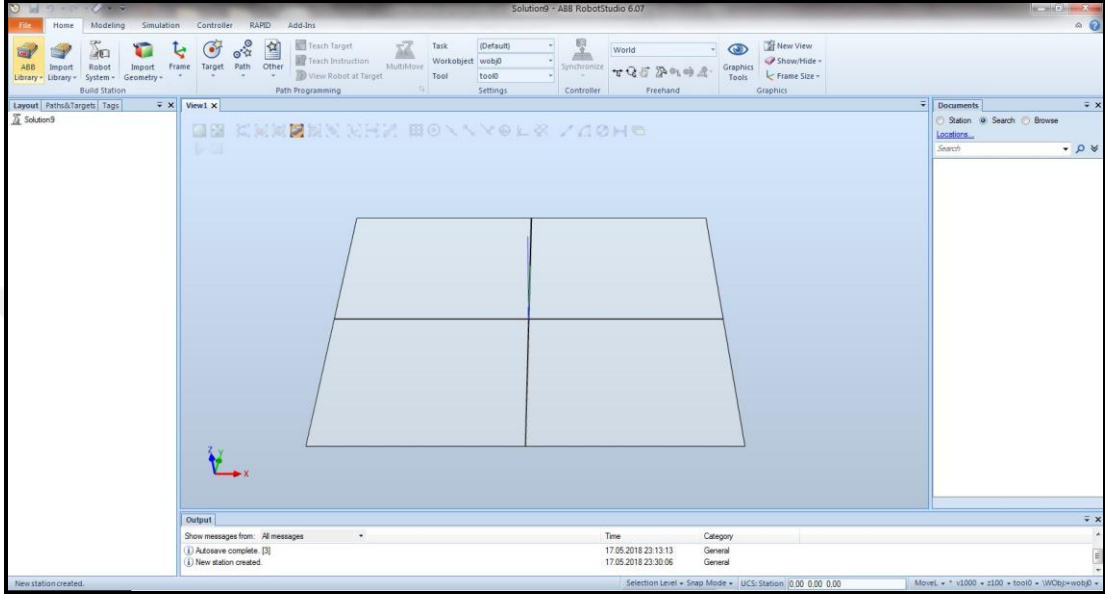
Bu aşamada istasyonda henüz bir robot tanımlı olmadığı için sayfanın sol tarafındaki “Layout” sekmesi altında yalnızca istasyonumuzun ismi olan “Solution9” ifadesini görmekteyiz.

Sayfanın sağ alt köşesinde görülen ve başlangıçta standart olarak seçili gelen ifadeleri aşağıdaki gibi tarifleyebiliriz:

- MoveL: Doğrusal hareket.
- v1000: Robotun hızı.
- z100: Robotun hedef noktalara yaklaşma hassasiyeti.

- tool0: Robotun ucuna bağlanıp işi yapan ekipman.
- wobj0: Ana koordinat sistemi.

Robotun izleyeceği güzergahın tanımlanmasının tariflendiği ileriki kısımlarda, bu ifadeler daha detaylı açıklanmıştır.

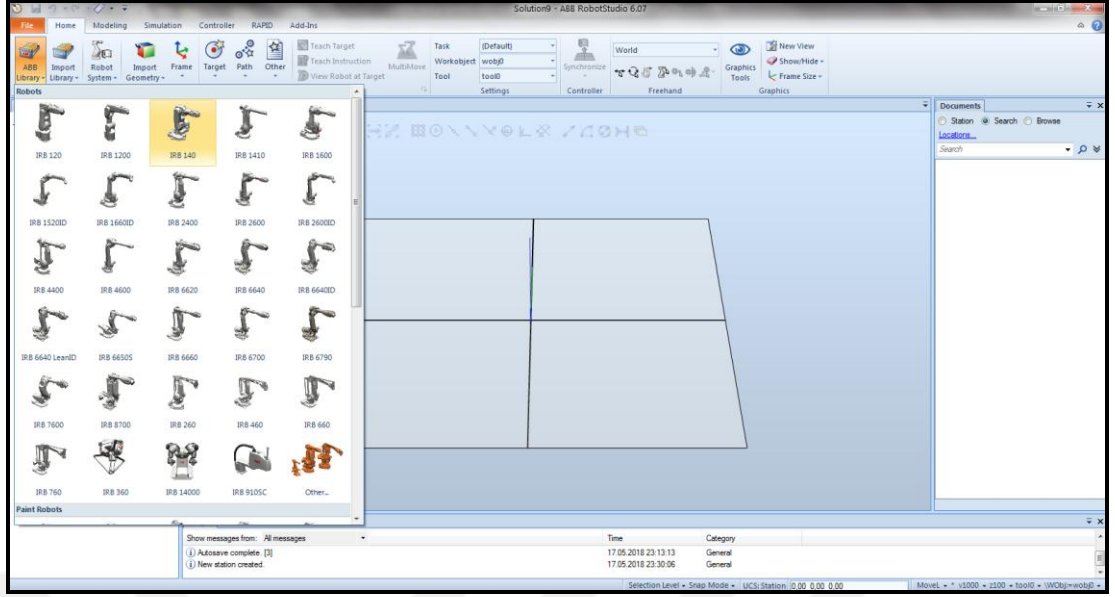


Şekil 2.3. Boş istasyon

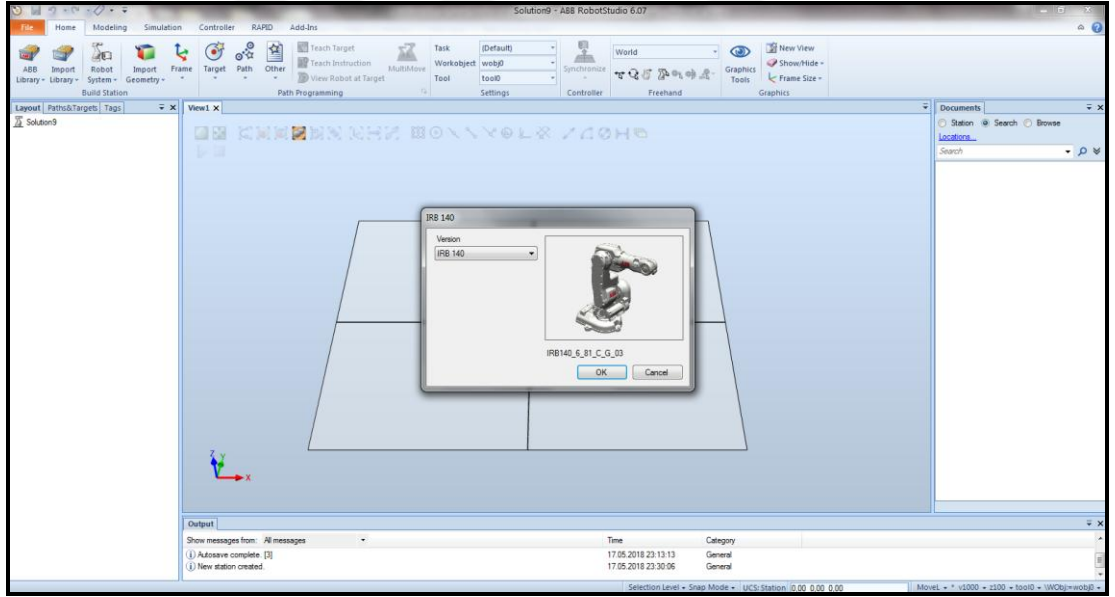
2.2.2. İstasyona robot eklenmesi ve sistemin kurulması

Boş istasyona ilk olarak bir robot ekleyebilmek için “ABB Library” sekmesi açılmalıdır. Bu sekmede ABB robotları, Şekil 2.4.’te görüldüğü gibi ürün kodlarıyla birlikte listelenmektedir. Uygulamanın gerektirdiği boyutlar, hareket kabiliyeti ve ağırlık taşıyabilme kapasitesi gibi seçim kriterleri göz önünde bulundurularak bu kısımdan uygun bir robot seçilmelidir. Bu çalışmada IRB 140 modeli robot seçilmiştir.

Seçilen robot üzerine tıkladıktan sonra robotun boyu ve uzanabilme kapasitesine göre farklı versiyonları varsa eğer, bu seçimi de yapabilmek adına Şekil 2.5.’teki ekran gelmektedir. Nispeten basit bir model olan IRB 140 için IRB 140 ve IRB 140T olarak 2 farklı versiyon gelmektedir. Burada IRB 140 versiyonu seçilip devam edilmiştir.

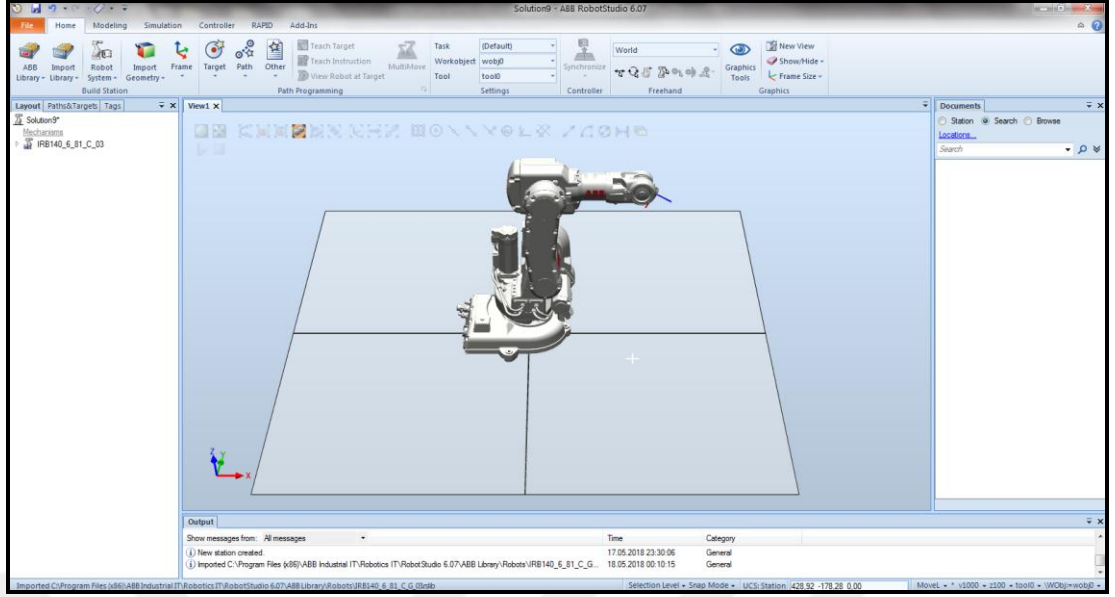


Şekil 2.4. İstasyona robot eklemek



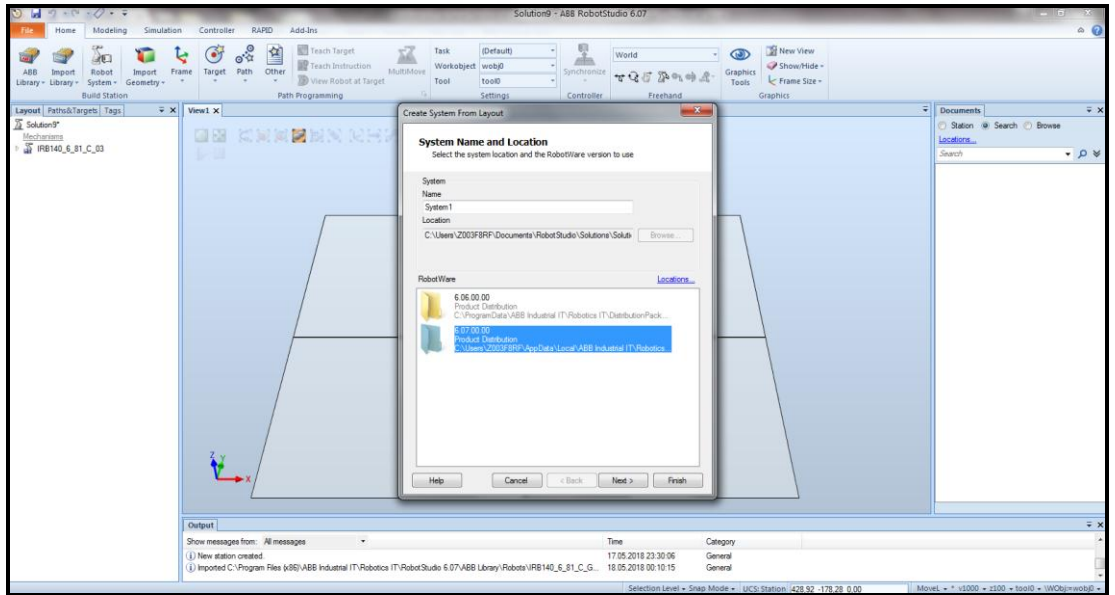
Şekil 2.5. Robot versiyonunun seçimi

Burada “OK” tıklandıktan sonra robot seçimi yapılmış olur ve seçilen robot, Şekil 2.6.’da görüldüğü gibi koordinat düzleminin merkezi üzerine konumlanmış olarak ekrana gelir. Bu aşamada ekranın sol tarafındaki Layout sekmesinde, “Solution9” ifadesinin altında robotun bir mekanizma olarak tanımlandığı görülmektedir.



Şekil 2.6. IRB 140 model robot

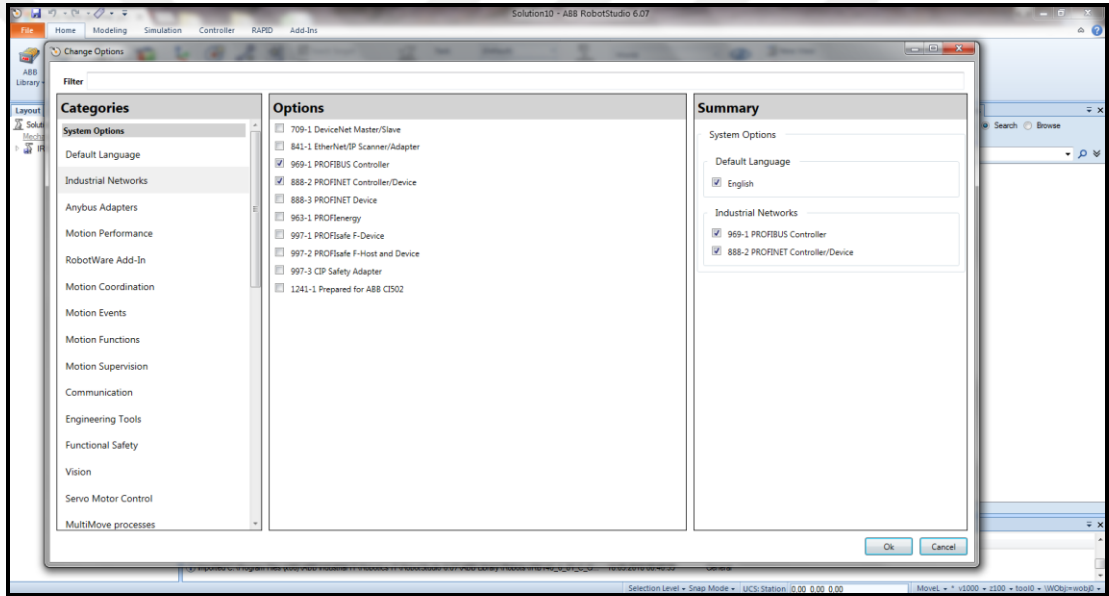
Bu aşamada istasyona robot eklenmiş ancak henüz robotun sistemi kurulmadığı için herhangi bir çalışma yaptırılmamaktadır. Sistemi kurmak için ise önce “Robot System” ve ardından bu seçeneğin altından gelen “From Layout” seçeneği tıklanmalıdır.



Şekil 2.7. Robot sisteminin kurulması

“From Layout” seçiminin ardından Şekil 2.7.’de görülen ekran gelmektedir. Çalışmanın yapıldığı bilgisayarda programın 6.06 ve 6.07 sürümlerinin kurulu olması dolayısıyla burada iki seçenek sunulmaktadır. Burada ilk kurulan sisteme standart

olarak “System1” ismi verilebilmekte veya istendiği gibi değiştirilebilmektedir. Sistemin kurulacağı lokasyon için ise değiştirme fırsatı verilmemekte ve sistem zorunlu olarak istasyonun içine kurulmaktadır. Bu aşamada herhangi başka seçim yapılmadan sırasıyla “Next” butonları ve en sonunda “Finish” butonu tıklanarak sistem kurulabilmektedir. Sistemin uygulamaya özel seçimlerle özelleştirilmesi isteniyorsa, bu seçimin iki adım sonrasında “Options” kutucuğu tıklanıp, Şekil 2.8.’de görülen sayfada istenilen seçimler yapılabilmektedir. Bu çalışmada, uygulamada kullanılabilmesi amacıyla “Options” sayfasındaki “Industrial Networks” sekmesi altından PROFIBUS ve PROFINET seçimleri de eklenerek sistem kurulumu tamamlanmıştır.



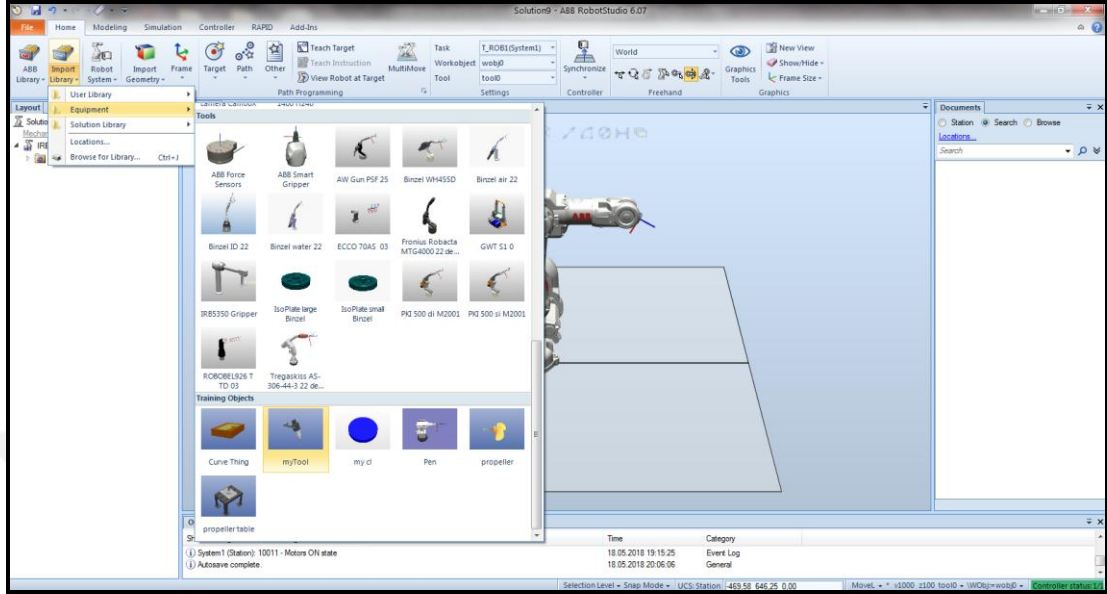
Şekil 2.8. Sistem kurulumunda “Options” sayfası

2.2.3. Robota tool eklenmesi

Sistem kurulduktan sonra robot programlanmaya hazırdır. Ancak robota iş yaptırabilmek için en uçtaki eksene bir tool eklenmelidir. Tool, basit tanımla, robota bağlanıp işi yapan ekipmandır. Robotun bir parçası değildir ancak robota ilave bir eksen gibi eklenerek robot tarafından kontrol edilebilmektedir.

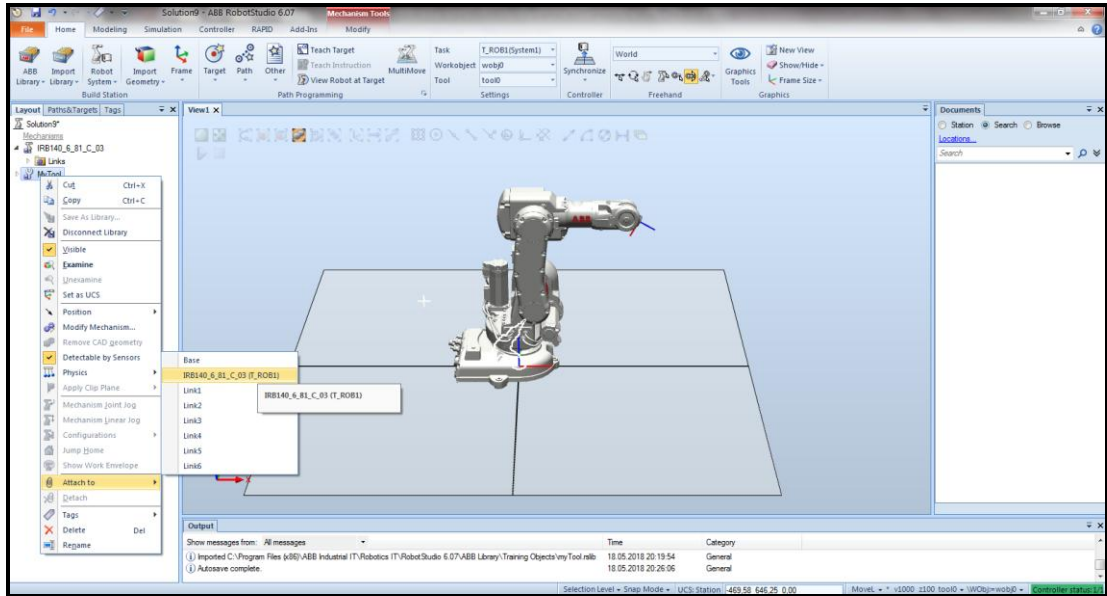
Tool eklemek için “Import Library” sekmesinde “Equipment” başlığı altında, “Tools” kısmındaki hazır toollardan seçim yapılabilir. Tool eklemeyi göstermek için Şekil 2.9.’da görüldüğü gibi, uygulamada “Equipment” başlığı altında, “Training

Objects” kısmında “myTool” seçilmiştir. Seçilen toolun üstüne tıklandığında, tool koordinat düzleminin merkezinde ortaya çıkmaktadır.



Şekil 2.9. Robota tool eklenmesi

Bunun ardından toolu robota bağlamak için Şekil 2.10.’da görüldüğü gibi toolun üstüne sağ tıklanıp “Attach to” kısmında robotun adı tıklanmalıdır. Ardından gelen onay sorusunda evet seçildiğinde, tool robota bağlanmaktadır.

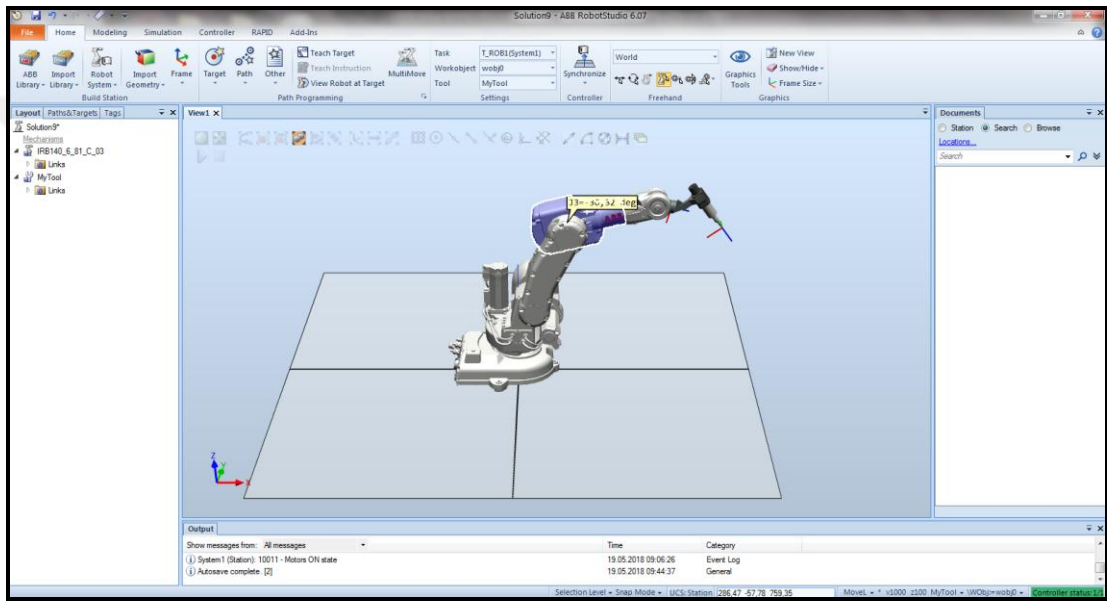


Şekil 2.10. Toolun robota bağlanması

2.2.4. Robotun hareket ettirilmesi ve home pozisyona çekilmesi

Robotun veya istasyondaki herhangi bir parçanın hareket ettirilmesi için “Home” sekmesi altında, “Freehand” başlığı altındaki komutlar kullanılabilir.

Robotun eksenleri birbirinden bağımsız olarak hareket ettirilebilmektedir. Bunun “Jog Joint” komutu tıkladıktan sonra, Şekil 2.11.’de görüldüğü gibi, hareket ettirilmek istenen eksen üzerinde farelin sol tuşuna basılı tutularak istendiği yöne doğru çekilmelidir.



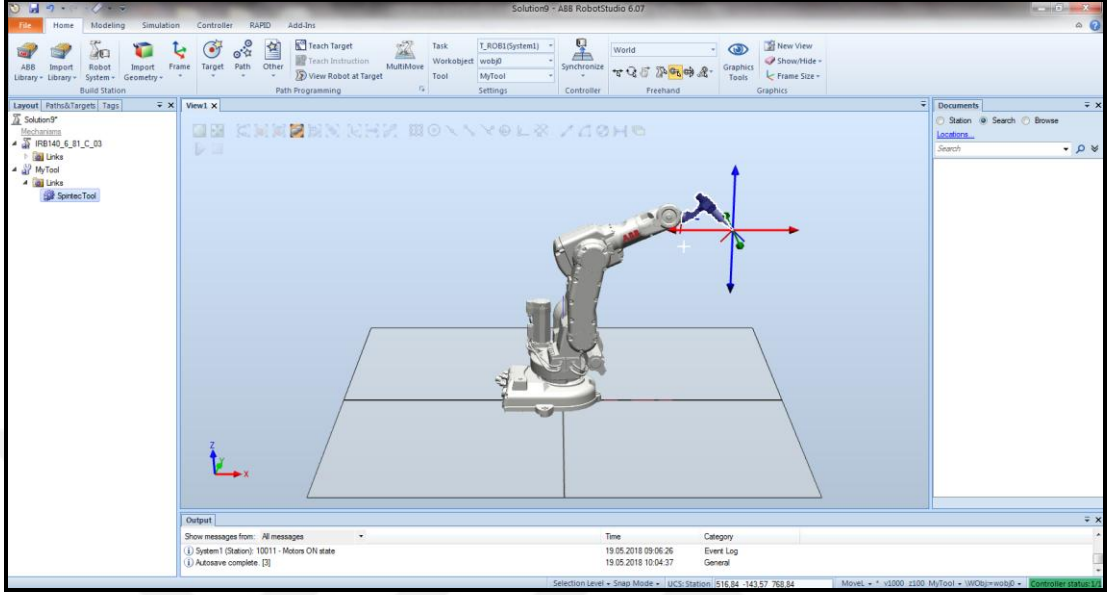
Şekil 2.11. Robotun eksenlerinin bağımsız olarak hareket ettirilmesi

Bu şekilde robotun her bir eksenini birbirinden bağımsız olarak, robotun mekanik olarak ulaşabileceği sınırlar dahilinde, hareket ettirilebilmektedir.

Belirli bir nesne üzerinde çalışılacağı zaman, toolun bir noktaya gitmesi istendiğinde, robotun her bir eksenleriyle ayrı ayrı çalışmaya gerek yoktur. Bunun için yine “Freehand” başlığı altında “Jog Linear” komutu kullanılmaktadır. Bu komut tıklandıktan sonra robotun üzerine tıklanınca, toolun ucunda Şekil 2.12.’de görüldüğü gibi bir hareket aracı ortaya çıkmaktadır.

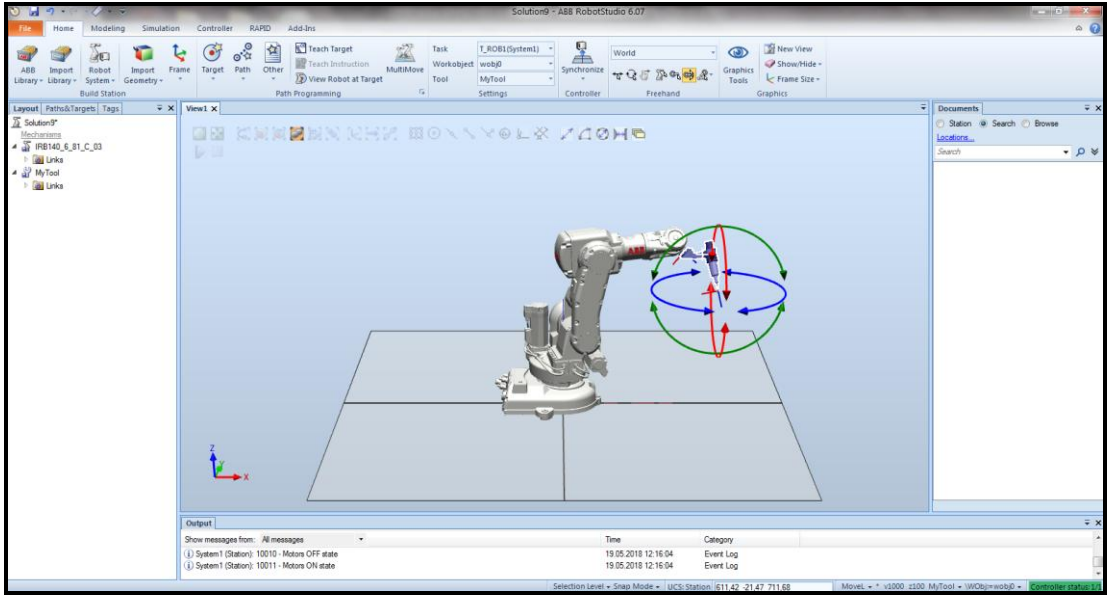
Bu hareket aracını x, y veya z eksenlerinde ileri veya geri yönde hareket ettirerek tool istenen konuma götürülebilmektedir. Bu sırada robotun eksenleri üzerinde ayrıca çalışmaya gerek yoktur. Lakin hareket çubuğu istenilen noktaya çekilirken, robot

tüm eksenlerini, toolu o noktaya ulařtırmak için en uygun şekilde koordinasyon halinde hareket ettirmektedir.



Şekil 2.12. Toolun belli bir noktaya çekilmesi

Toolun ucu istenilen noktaya getirildiđi halde, toolun bu noktada farklı bir şekilde pozisyonlandırılması istendiđinde, yine “Freehand” başlığı altında “Jog Reorient” komutu kullanılmaktadır. Bu komut tıklandıktan sonra robotun üzerine tıklanınca, toolun ucunda Şekil 2.13’te görüldüğü gibi bir döndürme aracı ortaya çıkmaktadır.

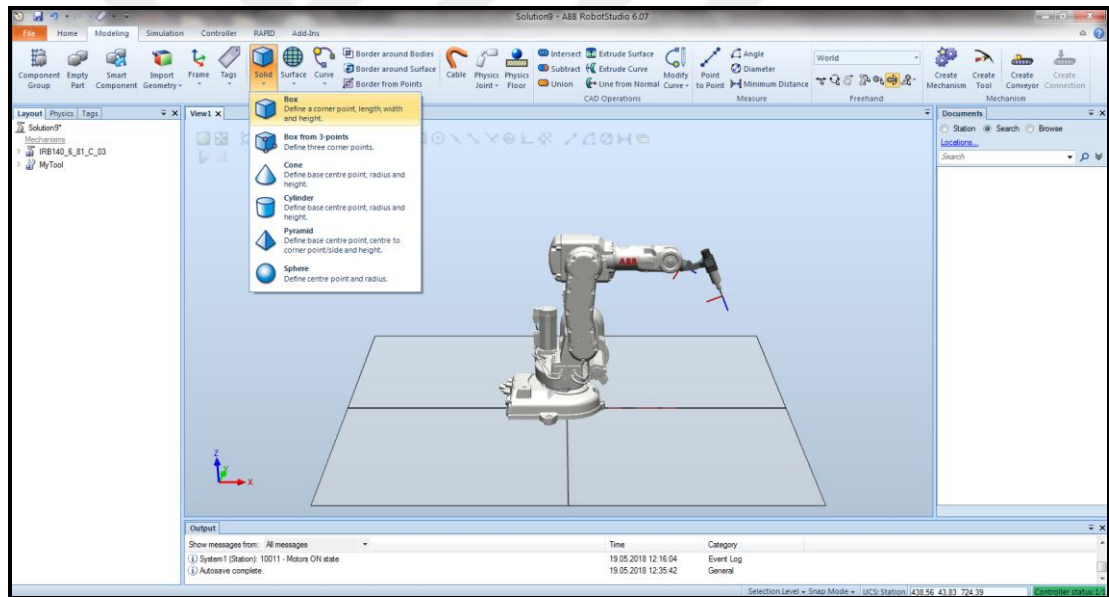


Şekil 2.13. Toolun belli bir noktada pozisyonunun deđiştirilmesi

Toolun uç noktasının koordinatı sabit kalmak şartıyla, toolun hangi eksen etrafında dönerek yeniden pozisyonlandırılması isteniyorsa, ilgili ok istenilen yönde döndürülmelidir. Bu şekilde toolun ucu sabit konumda kalırken, toolun pozisyonu değişmektedir.

2.2.5. İş parçası oluşturulması

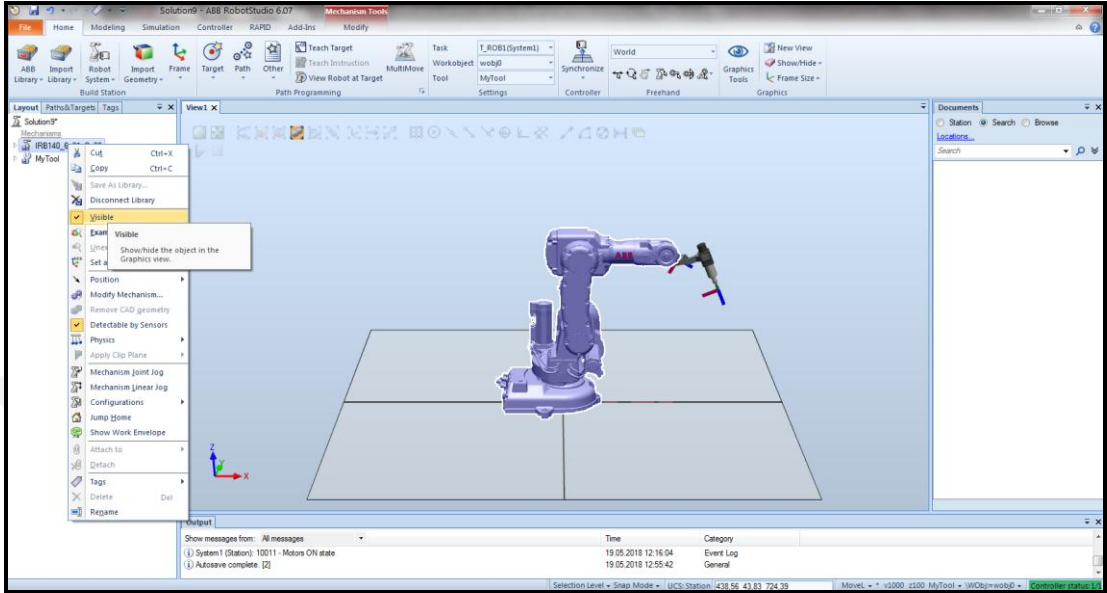
Toolun üzerinde çalışacağı iş parçasını oluşturmak için, Şekil 2.14.'te görüldüğü gibi “Modeling” sekmesi altında, “Solid” sekmesi altındaki ürünler kullanılabilir. Bununla birlikte daha önceden tanımlı olup çalışma kütüphanesine eklenmiş bir cisim varsa, yine “Modeling” sekmesi altında, “Import Geometry” sekmesi altından seçilip çağırılabilir. Ancak bu çalışmada, konuyu basitçe tariflemek için “Solid” sekmesi altından “Box” ürünü kullanılmıştır.



Şekil 2.14. İş parçası oluşturulması

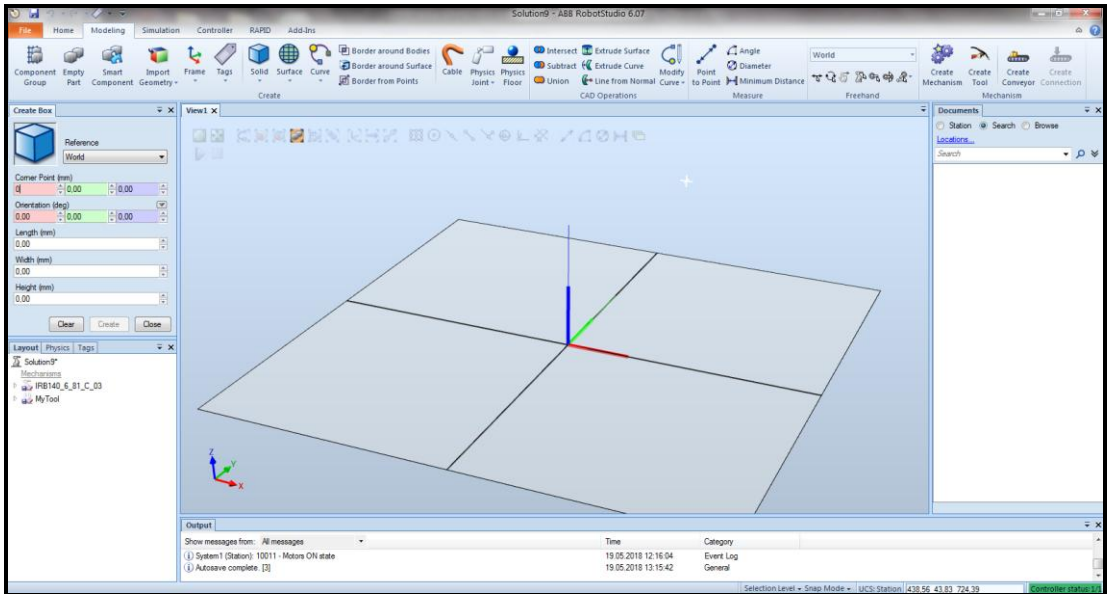
“Box” ürünü çağırıldığında koordinat düzleminin merkezine geleceği için robotun içinde gibi görünecek ve dolayısıyla bu durum, cismin boyutlarını belirlemekte programcı açısından zorluk yaratacaktır. Bu sebeple cismi çağırmadan önce Şekil 2.15'te görüldüğü gibi, “Layout” kısmında robotun adının üzerine sağ tıklanıp, burada “Visible” seçimi kaldırılmalıdır.

Robot görünmüyorken toolun hava boşta gibi görünmesi istenmiyorsa, aynı şekilde “Layout” kısmında toolun da adının üzerine sağ tıklanıp “Visible” seçimi kaldırılarak tool da görünmez yapılabilir.



Şekil 2.15. Robotun görünmez hale getirilmesi

Robot ve tool görünmez hale getirildikten sonra, Şekil 2.14’te görüldüğü gibi “Box” seçimi yapıldığında, ekranın sol tarafında Şekil 2.16’da görüldüğü gibi bir “Create Box” alanı oluşmaktadır.



Şekil 2.16. İş parçasının boyutları ve konumunun belirlenmesi

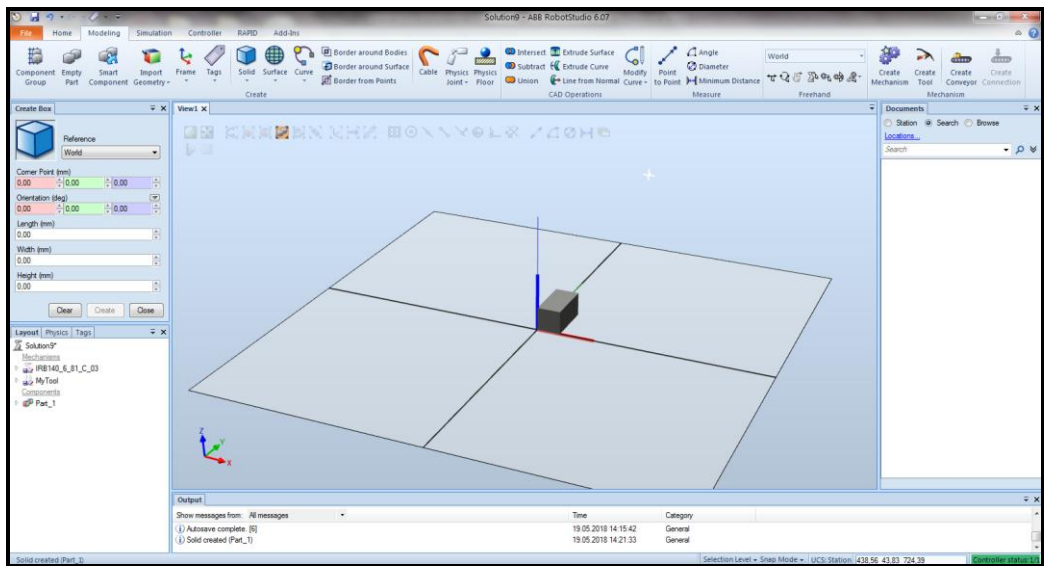
Bu alanda, istenilen cismin boyutlarının ve koordinat düzleminin merkezine göre konumunun belirlenmesi istenmektedir. Bu durumda cismi her pozisyondan görebilmek için aşağıdaki tuş seçimleriyle ekranda oynama yapılabilmektedir:

- Mouse dönen tuş: Ekranı yaklaşıp uzaklaşmasını sağlamaktadır.
- Ctrl + Mouse sol tuş: Ekrandaki düzlemin istenilen yöne kaydırılmasını sağlamaktadır.
- Ctrl + Shift + Mouse sol tuş: Ekrandaki düzlemin istenilen eksen etrafında dönmesi sağlanmaktadır.

“Create Box” alanındaki değerler aşağıda tanımlanmıştır:

- Corner Point (mm): Cismin köşe noktasının, koordinat düzlemine olan uzaklığı x, y ve z eksenlerine göre burada belirtilmektedir.
- Orientation (deg): Cismin x, y ve z eksenlerine göre açısal durumu burada belirtilmektedir.
- Length (mm): Cismin uzunluğu (x ekseninde) belirtilmektedir.
- Width (mm): Cismin genişliği (y ekseninde) belirtilmektedir.
- Height (mm): Cismin yüksekliği (z ekseninde) belirtilmektedir.

Görüldüğü gibi, tüm uzunluk değerleri mm cinsinden girilmektedir. Burada köşe noktasını koordinat düzleminin üzerinde alıp, açısal olarak da herhangi bir değişiklik yapmayıp, uzunluğu ve yüksekliğini 100 mm, genişliğini 200 mm olarak belirleyip “Create” tuşuna tıklandığında cisim Şekil 2.17’deki gibi oluşmaktadır.



Şekil 2.17. 100*200*100 mm boyutlarında iş parçası

Bundan sonra başka bir cisim daha oluşturulması isteniyorsa, açıkta duran “Create Box” ekranında istenilen değerler tekrar girilerek, yine aynı yolla yeni cisim veya cisimler oluşturulabilmektedir. Cisim oluşturma işi bittiğinde, bu alan “Close” tuşu ile kapatılmalıdır.

Robotu tekrar görünür yapmadan önce, cisim “Jog Linear” komutu ile düzlem üzerinde istenilen noktaya taşınabilmektedir. Ardından robot ve tool için “Visible” seçimleri aktifleştirilerek tekrar görünür hale gelmeleri sağlanabilmektedir.

2.2.6. Nokta izleme uygulaması

Bu çalışmada, kütüphanede tanımlı bir çalışma sehpası istasyona eklenip, sehpanın üzerindeki iş parçasının köşe noktaları robota hedef olarak tanımlanmış ve robotun bu noktaları izlediği bir uygulamanın simülasyonu yapılmıştır. Bu uygulamada yapılan işlemler sırasıyla aşağıdaki gibi tariflenebilmektedir:

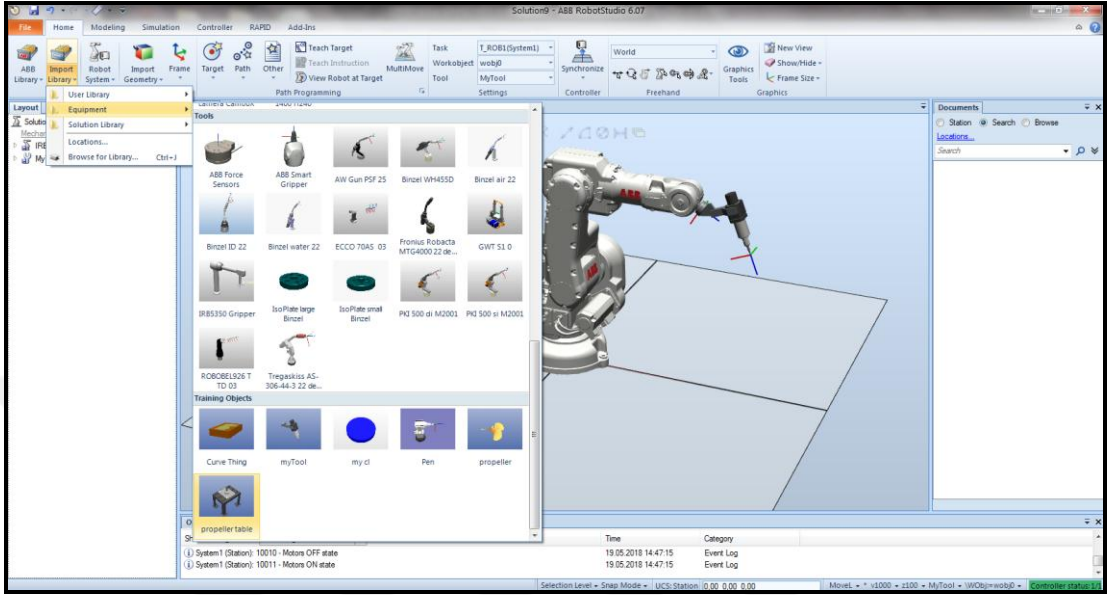
- İstasyon oluşturulur.
- İstasyona robot eklenir.
- Robota sistem kurulur.
- Tool eklenip robota bağlanır.
- İstasyona sehpa ve iş parçası eklenir.
- Sehpanın üzerine yeni bir “workobject” tanımlanır.
- İş parçasının üzerinde robota hedef noktaları tanımlanır.
- Hedef noktalar kullanılarak robotun izleyeceği yol tanımlanır.
- Simülasyon yapılır.

İstasyon oluşturulması, robot eklenmesi, robota sistem kurulması ve tool eklenip robota bağlanması kısımları daha önce tarif edildiği için, direkt olarak istasyona sehpa ve iş parçası eklenmesi adımından devam edilmiştir.

2.2.6.1. İstasyona sehpa ve iş parçası eklenmesi

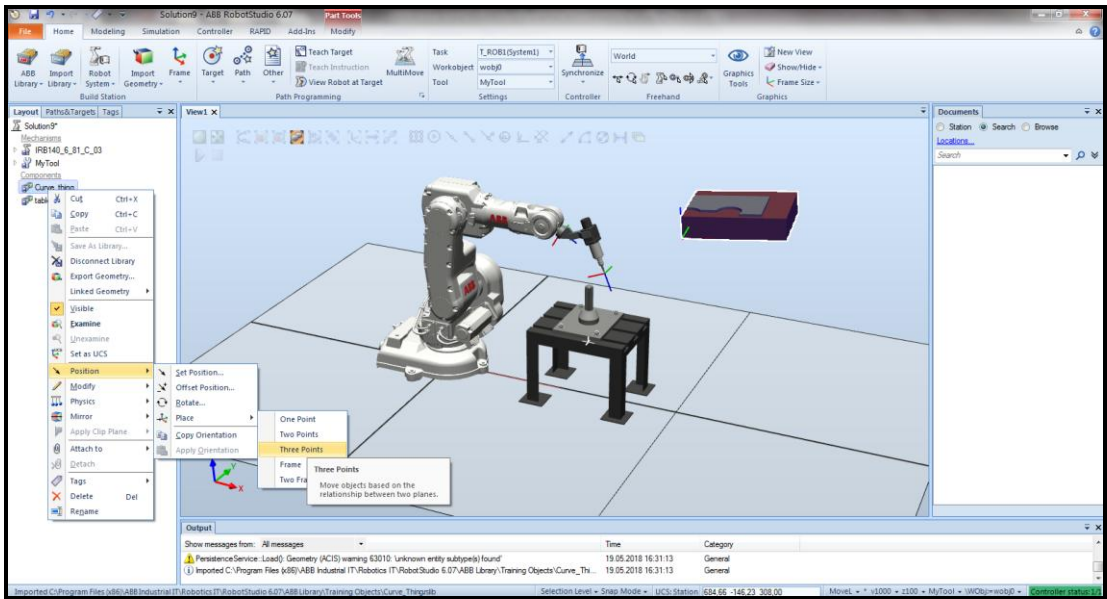
Sehpa eklemek için Şekil 2.18.’de görüldüğü gibi, “Import Library” sekmesinde “Equipment” başlığı altında, “Training Objects” kısmında “propeller table” seçilmiştir. “Propeller table”, programda eğitim amaçlı kullanılacak, üzerinde bir iş parçası olan basit bir sehpadır.

Ardından iş parçası için de, yine Şekil 2.18.'de görüldüğü gibi, “Training Objects” kısmında “Curve Thing” seçilmiştir. Bunun sonrasında, iş parçasının sehpa üzerine konumlandırılması gerekmektedir.



Şekil 2.18. İstasyona sehpa ve iş parçası eklenmesi

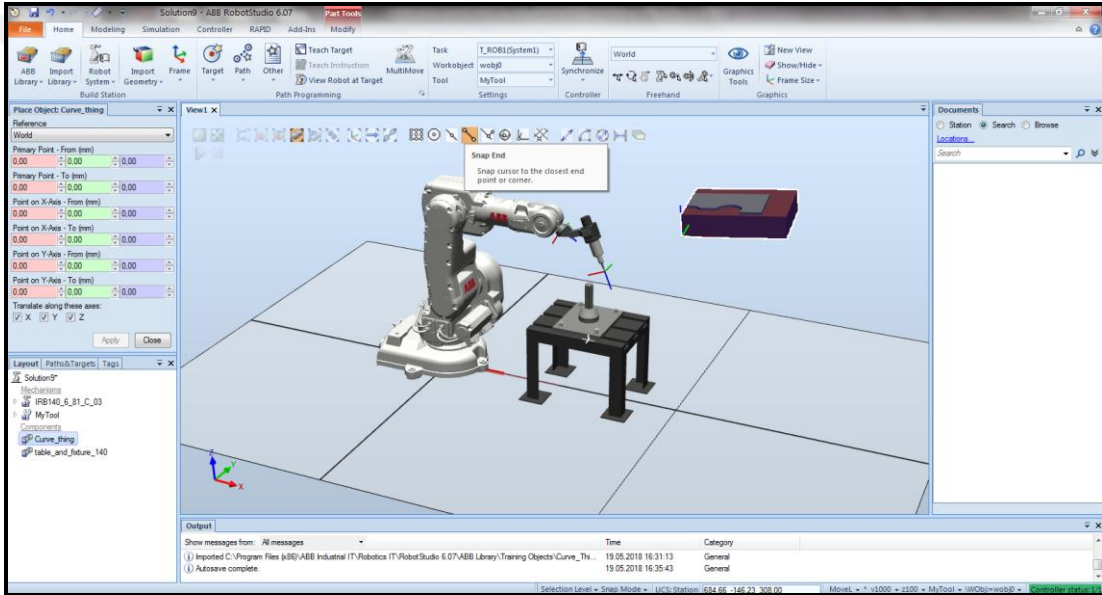
İş parçasını sehpa üzerine konumlandırabilmek için Şekil 2.19.'da görüldüğü gibi, “Layout” kısmı altında “Curve Thing” üzerine sağ tıklanıp açılan pencerede “Position”, buradan açılan pencerede “Place” ve buradan açılan pencerede “Three Points” tıklanır.



Şekil 2.19. İş parçasının sehpa üzerine konumlandırılması – 1

Burada üç noktadan konumlandırmanın tercih edilmesinin sebebi, iş parçasının sehpa göre paralel düzlemde olmamasıdır. İki noktanın eşleştirilmesiyle iş parçası ile sehpanın birer kenarı örtüştürülürken, üçüncü noktanın da eşleştirilmesiyle iş parçasının sehpa ile aynı düzleme konumlanması sağlanmaktadır.

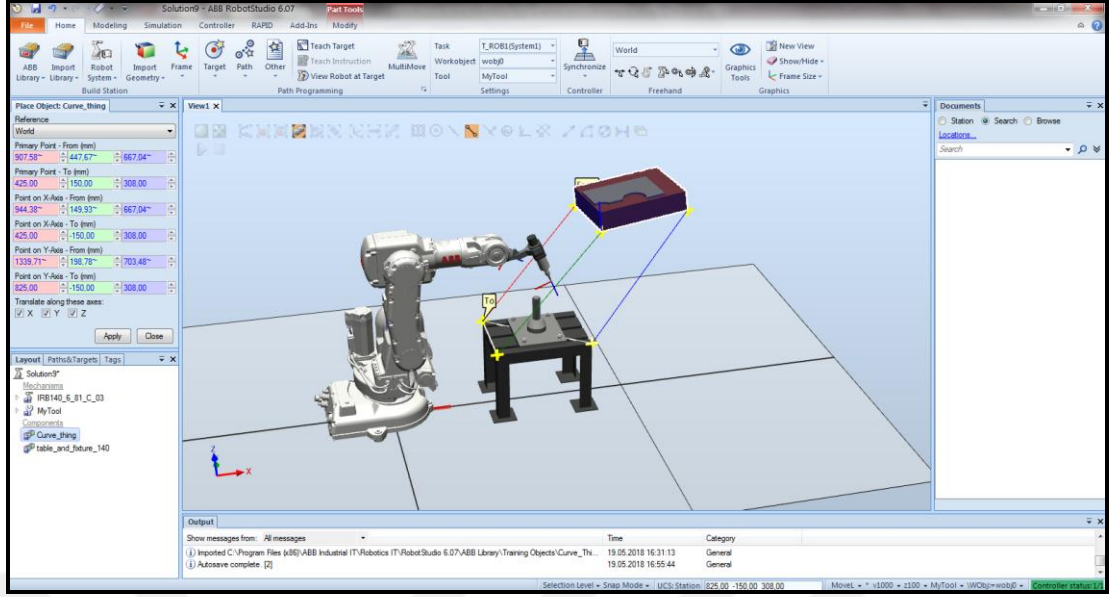
“Three Points” seçimi yapıldıktan sonra Şekil 2.20’de görüldüğü gibi, ekranın sol kısmında birleştirilecek noktaların seçileceği alan gelmektedir. Bu esnada noktaları doğru seçebilmek için gerekli seçim araçları, istasyon ekranının üst kısmında sıralı simgeler halinde bulunmaktadır. Köşe noktaları seçebilmek için burada “Snap End” seçeneği seçilmelidir.



Şekil 2.20. İş parçasının sehpa üzerine konumlandırılması – 2

Birleştirilecek noktaların tanımlandığı alanda görüldüğü gibi, noktaların tanımlarında yukarıdan aşağıya doğru sırasıyla “from” ve “to” ifadeleri vardır. İlk noktayı tanımlamak için “Primary Point – From (mm)” kısmındaki x eksenini belirten kırmızı kutucuğun içine bir kez tıklanmalıdır. Ardından iş parçasının taşınacak ilk noktası seçildiğinde, bu noktanın x, y ve z eksenlerindeki konumlarının “Primary Point – From (mm)” kısmına otomatik olarak doldurulduğu görülmektedir.

Seçilen noktanın sehpa üzerinde taşınacağı noktayı seçmek için de “Primary Point – To (mm)” kısmındaki x eksenini belirten kırmızı kutucuğun içine bir kez tıklanıp, ardından sehpa üzerindeki nokta seçilmelidir. Aynı işlem diğer iki nokta için de yapıldığında Şekil 2.21’deki gibi bir görüntü ortaya çıkmaktadır.



Şekil 2.21. İş parçasının sehpa üzerine konumlandırılması – 3

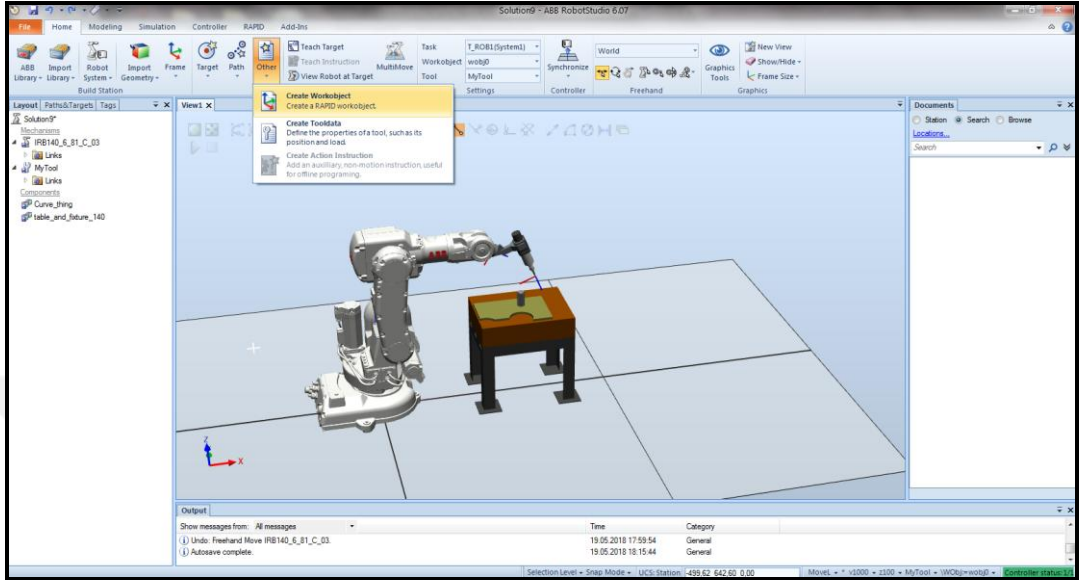
Ardından noktaların tanımlandığı alandaki “Apply” butonu tıklandığında, iş parçası sehpanın üzerine yerleşmektedir. Yeni bir taşıma işlemi yapılmayacaksa bu alan “Close” butonuna tıklanarak kapatılmalıdır.

2.2.6.2. Sehpanın üzerine yeni bir “workobject” tanımlanması

Bir iş parçasının konumunu tanımlamak için kullanılan koordinat sistemi, “workobject” olarak adlandırılmaktadır. İstasyonun açılışında tanımlı olarak gelen ana koordinat sistemi, standart olarak “wobj0” adıyla tanımlanmıştır ve bu isim değiştirilemez. İstasyon üzerinde tanımlı her bir cismin veya noktanın, workobjectin merkezine göre x, y ve z eksenlerinde tanımlı bir konumu vardır.

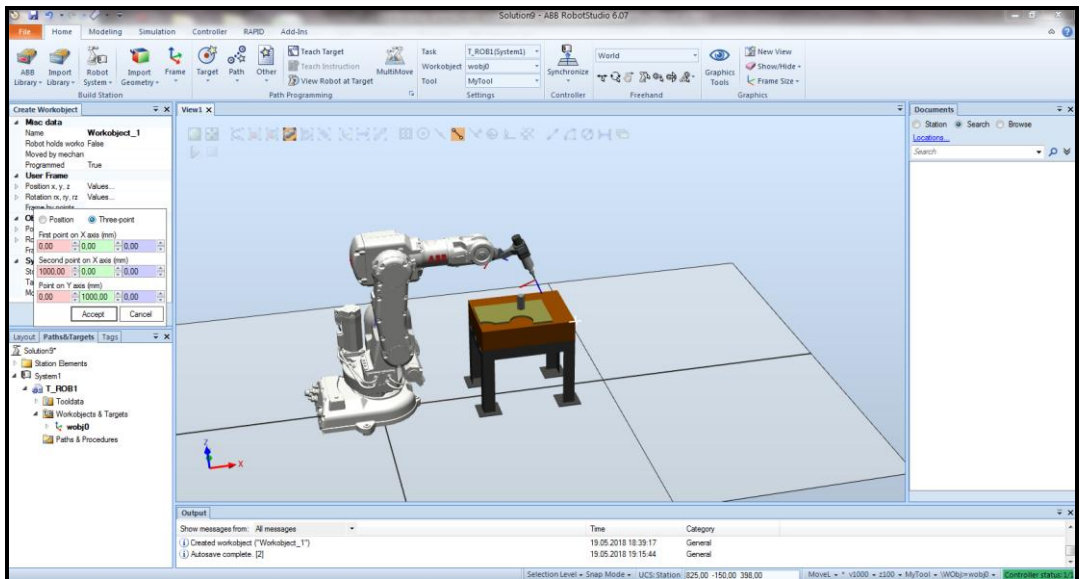
Robot, bir sehpa üzerindeki bir iş parçası ile çalışacaksa, o sehpanın herhangi bir köşesini merkez alan yeni bir workobject tanımlanması tercih edilmektedir. Bu sayede sehpa üzerinde çalışırken robota hedef olarak belirlenecek her bir nokta, standart wobj0 yerine, sehpa üzerine tanımlanan yeni workobjectin merkezine göre konumlanmaktadır. Burada belirtmek gerekir ki her bir workobject, “user frame” ve “object frame” olmak üzere iki merkezli olarak tanımlanır. Workobject üzerinde tanımlı tüm noktalar object frame merkezine bağlıdır. Aynı şekilde object frame, içinde tanımlı olduğu user frame merkezine ve user frame de wobj0 olarak adlandırılmış ana koordinat sisteminin merkezine bağlıdır.

Sehpanın bir köşesini merkez alacak bir workobject tanımlamak için önce Şekil 2.22.'de görüldüğü gibi “Home” sekmesi altında, “Other” sekmesi altındaki “Create Workobject” seçeneği tıklanmalıdır.



Şekil 2.22. Sehpanın üzerine yeni bir workobject tanımlanması – 1

Bu seçim yapıldıktan sonra, oluşturulacak workobjectin merkezinin tanımlanabilmesi için, ekranın sol tarafında “Create Workobject” başlıklı bir alan gelmektedir. Bu alanda “User Frame” kısmı altında, “Frame by Points” seçeneğinin yanındaki ok tıklanınca, çalışma düzlemini tanımlayabilmek için üç noktanın belirleneceği, Şekil 2.23.'te görüldüğü gibi bir kutucuk ortaya çıkmaktadır.



Şekil 2.23. Sehpanın üzerine yeni bir workobject tanımlanması – 2

Açılan kutucukta düzlemi üç noktadan tanımlayabilmek için “Three Points” seçeneği tıklanmalıdır. Bundan sonra tıpkı nokta taşıma işleminde yapıldığı gibi, bu kez “First point on X axis – From (mm)” kısmındaki x eksenini belirten kırmızı kutucuğun içine bir kez tıklanmalıdır. Ardından tanımlanacak workobjectin merkezi olması istenen nokta (sehpanın merkez olması istenen köşe noktası), sehpanın x eksenindeki ilk noktası olarak seçildiğinde, bu noktanın x, y ve z eksenlerindeki konumlarının “First point on X axis – From (mm)” kısmına otomatik olarak doldurulduğu görülmektedir.

Aynı işlem sehpanın x eksenindeki ikinci noktası (merkez köşenin x eksenindeki komşu köşesi) ve y eksenindeki noktasının (merkez köşenin y eksenindeki komşu köşesi) seçimleri için de yapılmalıdır. Ardından “Accept” tıkladığında nokta belirleme kutucuğu kapanmaktadır.

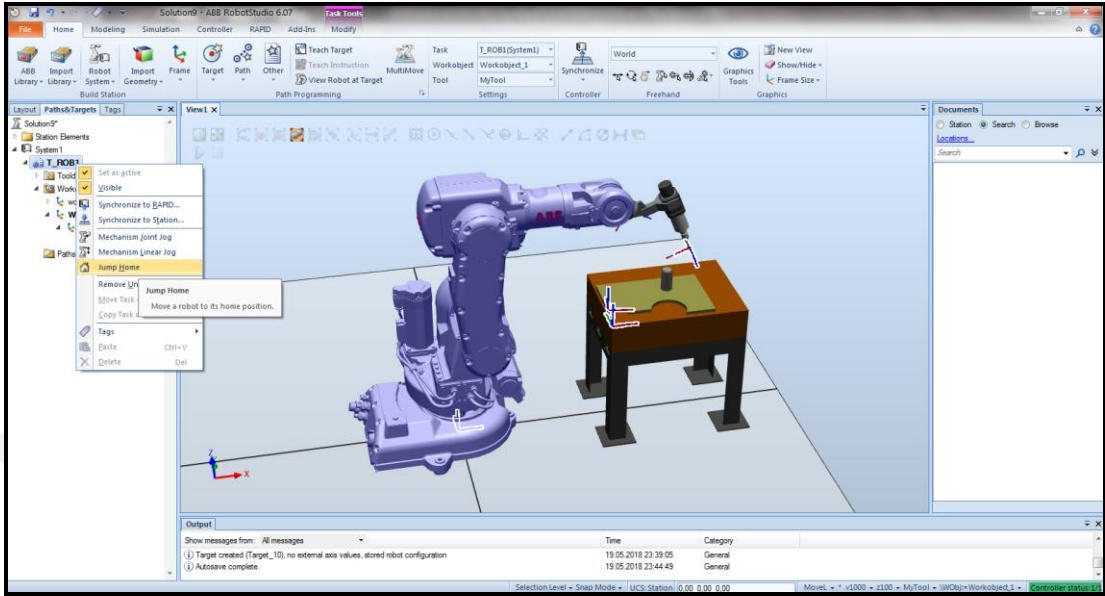
Bu işlemden sonra nokta seçme işlemi bu kez “Object Frame” kısmında yapılmalıdır. Object frame olarak da, sehpanın üzerindeki iş parçasının merkez olması istenen üç köşe noktası belirlenmelidir. Object frame de oluşturulduktan sonra “Apply” seçimi yapılarak workobject oluşturma işlemi tamamlanmalıdır. Ardından “Close” seçimiyle “Create Workobject” alanı kapatılmalıdır.

2.2.6.3. İş parçasının üzerinde robota hedef noktaların tanımlanması

Robota hedef noktası tanımlamak için “Home” sekmesi altındaki “Teach Target” seçimi kullanılabilir. Önceki bölümlerde tariflenmiş olan yollarla toolun ucu istenilen pozisyona çekildikten sonra, “Teach Target” seçimi tıklanarak toolun ucu hedef nokta olarak tanıtılabilmektedir. Bu esnada wobj0 veya yeni tanıtılmış workobjectlerden hangisi aktif durumdaysa hedef onun altına tanımlanmaktadır.

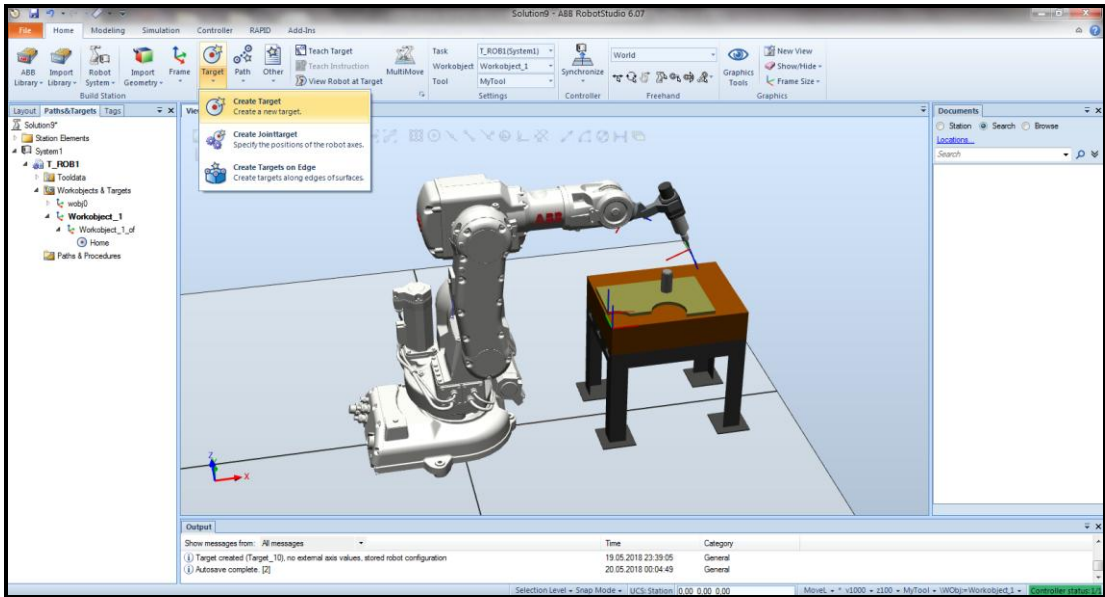
İlk önce bir home pozisyonu tanıtmak için, Şekil 2.24’te gösterildiği gibi “Path&Targets” sekmesi altında, robotu ifade eden T-ROB1 ifadesi üzerine sağ tıklanıp, burada “Jump Home” seçimi yapıldığında robot başlangıçtaki konumuna gitmektedir. Burada “Teach Target” tıklanıp bu nokta tanıtıldıktan sonra, T-ROB1 ifadesi altında, workobject1 altında “Target_10” olarak bu noktanın geldiği görülmektedir.

Bundan sonra Target_10 ifadesi üzerine sağ tıklayıp “Rename” seçimi yapılıp, bu noktanın ismi “Home” olarak değiştirilmesi tercih edilebilmektedir.



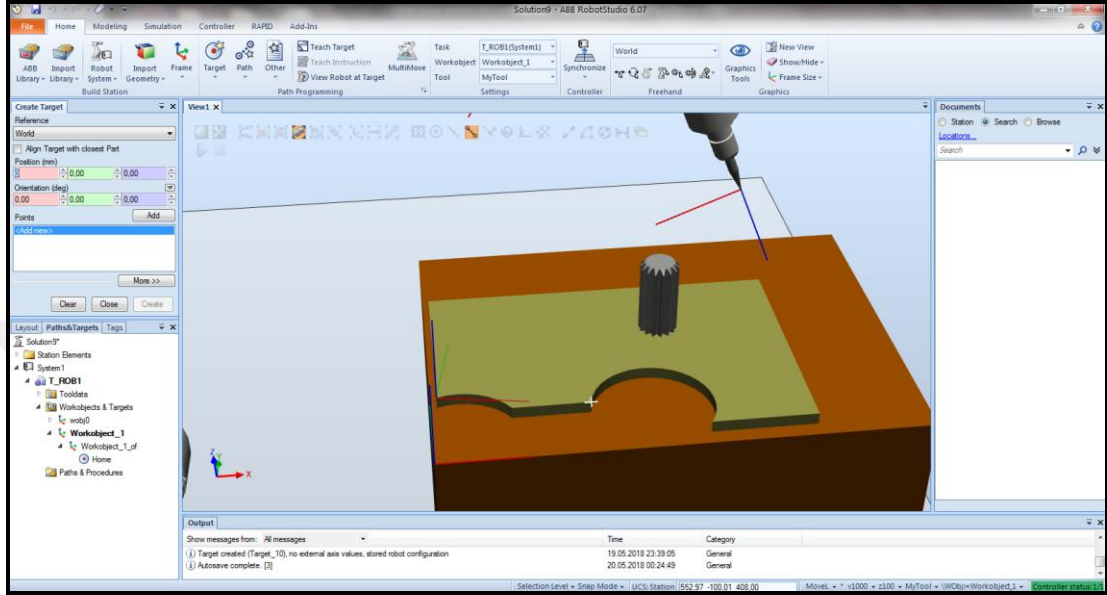
Şekil 2.24. Robotun home pozisyona çekilmesi

İş parçasının üzerindeki hedefleri tanıtmak için ise toolu her bir noktaya el ile çekmeye gerek yoktur. Bunun yerine noktaları seçerek tanıtabilmek için Şekil 2.25.'te görüldüğü gibi “Home” sekmesi altında, “Target” başlığı altında “Create Target” seçimi yapılmalıdır.



Şekil 2.25. Robota hedef nokta tanıtmak – 1

“Create Target” seçimi yapıldıktan sonra Şekil 2.26’da görüldüğü gibi, ekranın sol tarafında, hedef noktaların tanımlandığı “Create Target” başlıklı bir alan gelmektedir. Bu esnada noktaları daha iyi seçebilmek için iş parçasının üzerinde görüntü yakınlaştırma tercih edilebilmektedir.

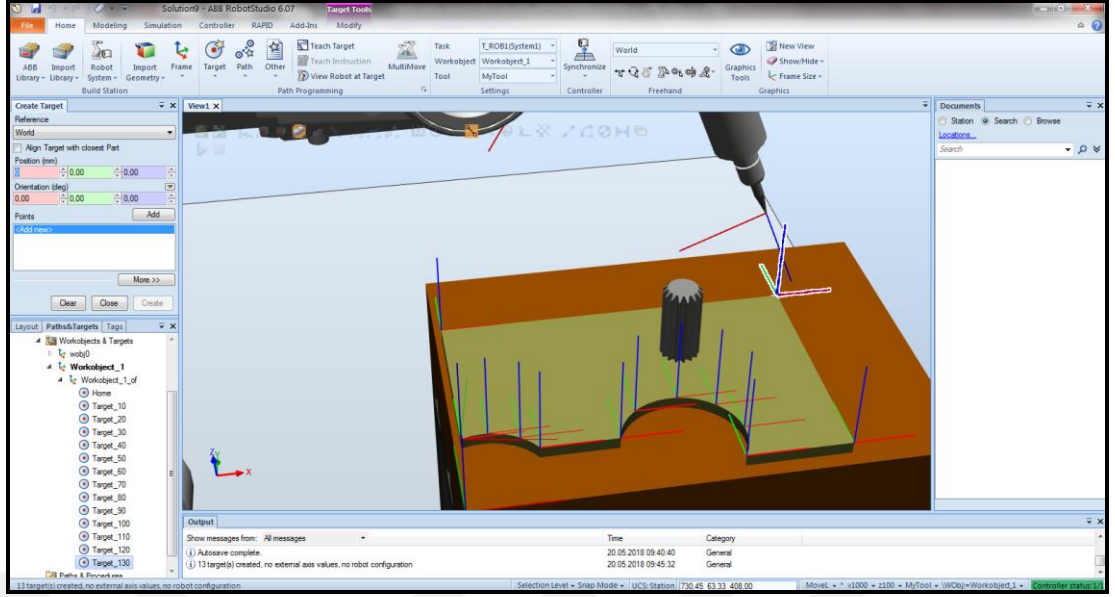


Şekil 2.26. Robota hedef nokta tanıtmak – 2

İlk noktayı tanımlamak için “Position (mm)” kısmındaki x eksenini belirten kırmızı kutucuğun içine bir kez tıklanmalıdır. Ardından iş parçasının üstündeki ilk hedef noktası seçildiğinde, bu noktanın x, y ve z eksenlerindeki konumlarının “Position (mm)” kısmına otomatik olarak doldurulduğu görülmektedir. Bu işlem, hedef olarak tanıtılacak her bir nokta için uygulandıktan sonra, seçilmiş olan noktalar bu alandaki “Points” kutucuğu içinde görülebilmektedir.

Nokta seçimi yapılırken önemli bir nokta şudur ki, daha önceki kısımlarda söz edilmiş olan “Snap End” nokta seçimi, bir doğru parçası üzerindeki ilk ve son noktaları seçmeye yaramaktadır. Ancak bu iş parçası üzerinde olduğu gibi eğri bölgeler üzerinde de hedef tanımlanması istendiğinde, o hedefler için “Snap Edge” nokta seçimi kullanılmalıdır.

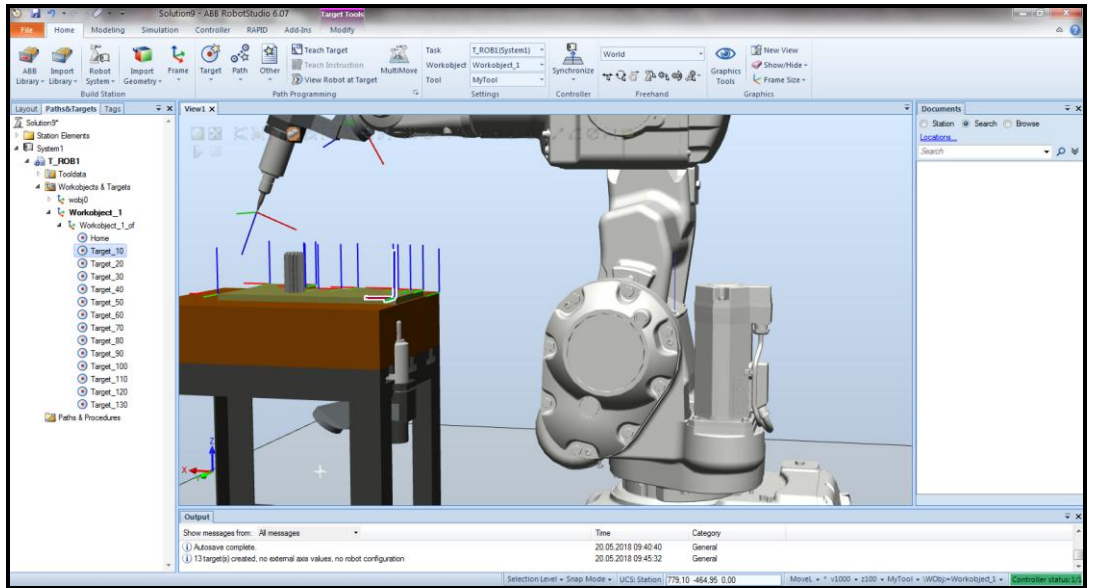
Tüm noktaların seçimi tamamlandıktan sonra “Create” seçimi yapıldığında, seçilen hedef noktalar Şekil 2.27.’deki gibi görünmektedir. Bundan sonra “Close” seçimiyle bu alan kapatılmalıdır.



Şekil 2.27. Robota hedef nokta tanıtmak – 3

Hedef noktalar tanımlandıktan sonra, toolun bu noktalarda nasıl konumlandığı kontrol edilmelidir.

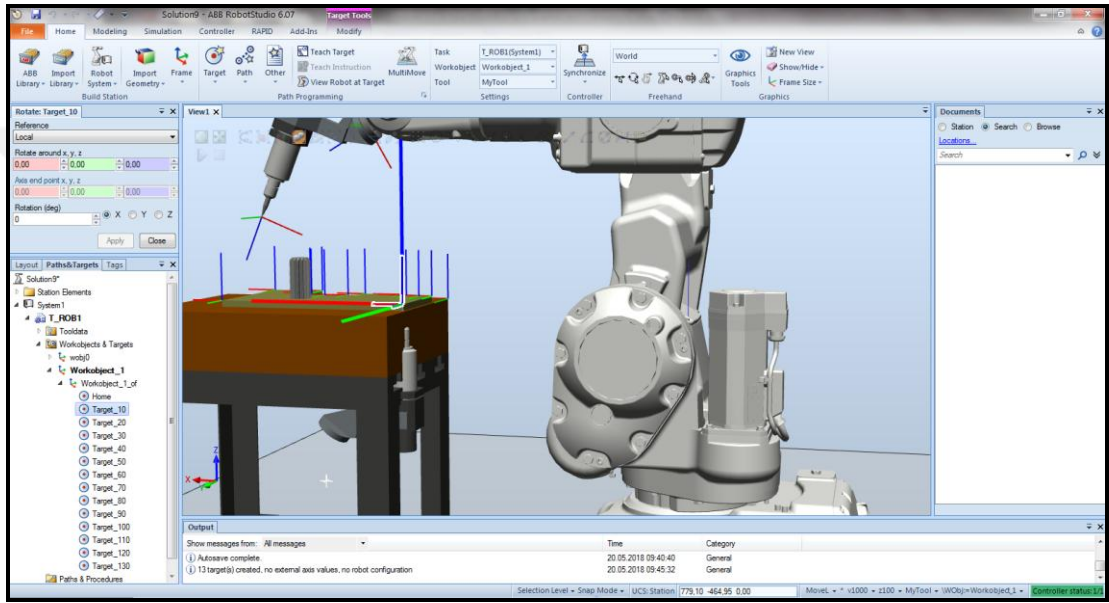
Bunun için ilk olarak tool zaten home pozisyonda olduğu için, Target_10 üzerinde bir deneme yapılmalıdır. Workobject_1 altında Target_10 üzerine sağ tıklayıp, “View Tool at Target” ve ardından “myTool” seçimi yapıldığında, toolun Target_10’daki pozisyonu Şekil 2.28’deki gibi görülmektedir.



Şekil 2.28. Toolun hedef noktadaki pozisyonunun gözlenmesi

Ancak robotun mevcut yerleşiminde toolu bu şekilde pozisyonlayabilmesi mümkün görünmemektedir. Bu durumda hedef noktanın koordinat sistemine göre yeri sabit kalmak şartıyla, noktayı kendi etrafında döndürmek suretiyle üzerindeki eksen çizgilerinin yönü tool ile uyumlu hale getirilmelidir.

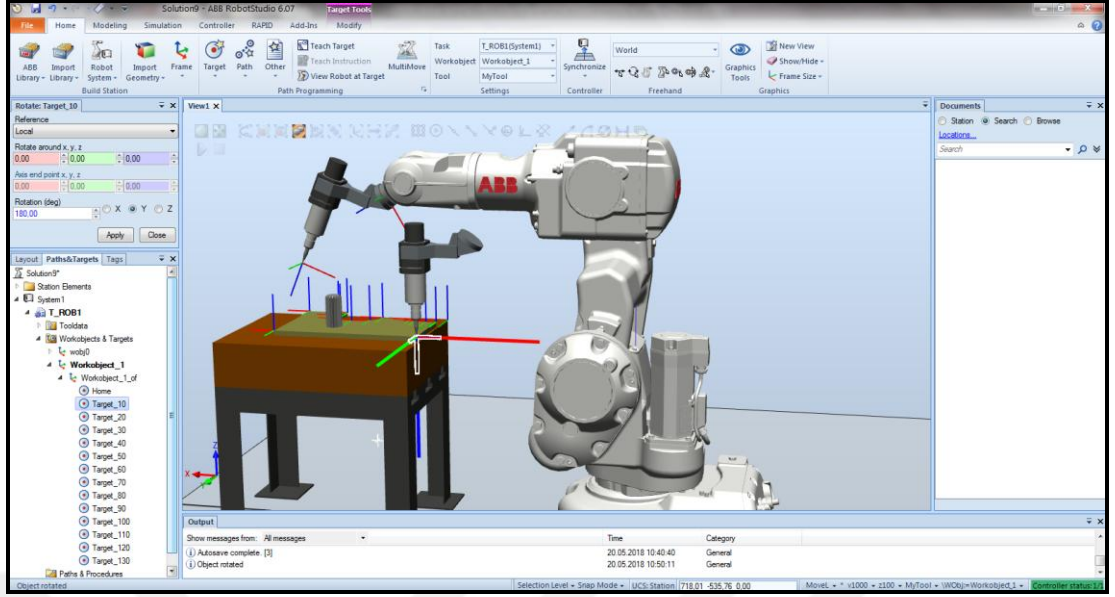
Hedef noktayı kendi etrafında döndürebilmek için Target_10 sağ tıklanıp, “Modify Target”, buradan da “Rotate” seçimi yapıldığında, ekranın sol tarafında Şekil 2.29’daki gibi bir alan gelmektedir.



Şekil 2.29. Hedef noktayı kendi etrafında döndürmek – 1

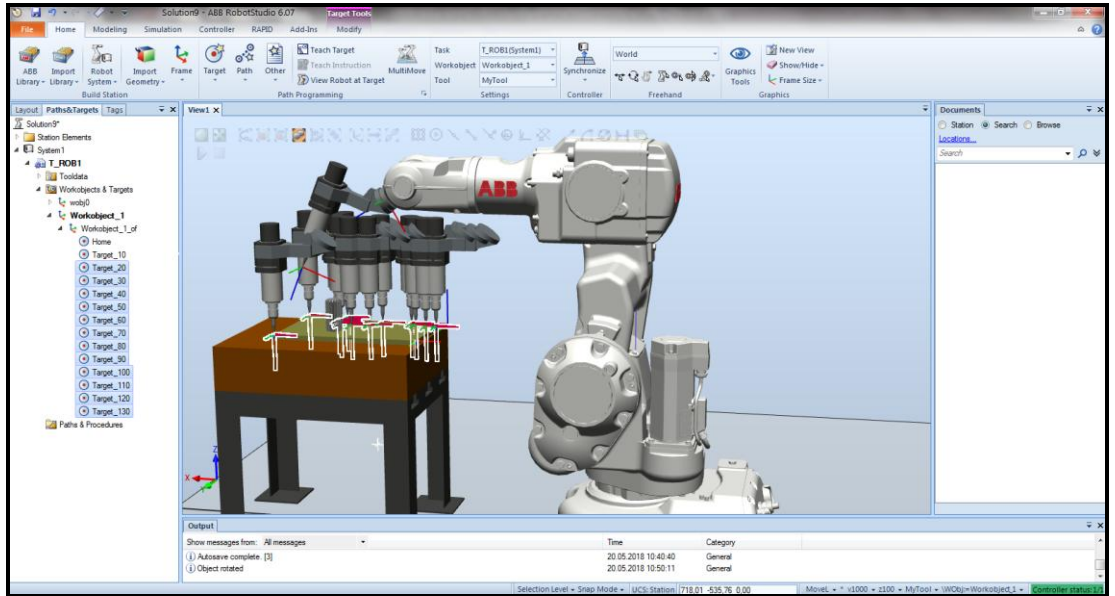
Burada hedef noktanın y eksenini etrafında 180 derece döndürülmesi, toolu uygun bir pozisyona getirecektir. Bunun için “Rotation (deg)” kısmına 180 yazılıp, yanındaki eksenler arasında Y eksenini seçildikten sonra “Apply” seçimi yapılmalıdır.

Hedef noktanın Y eksenini etrafında 180 derece döndürülmesi sonrasında toolun bu noktadaki pozisyonu Şekil 2.30’da görülmektedir. Ardından “Close” seçimiyle bu alan kapatılmalıdır.



Şekil 2.30. Hedef noktayı kendi etrafında döndürmek – 2

Bundan sonra bu düzenlemenin diğer noktalara da aynı şekilde uygulanması gerekmektedir. Bunun için Target_10 üzerine sağ tıklanıp “Copy Orientation” seçimi yapılmalıdır. Ardından diğer hedef noktalar toplu halde seçilip sağ tıklanıp “Apply Orientation” seçimi yapıldığında, Şekil 2.31’de görüldüğü gibi, Target_10 için yapılan hedef nokta düzenlemesi, diğer tüm noktalara aynı şekilde uygulanmakta ve tool bu noktalarda da robotun erişebileceği bir şekilde pozisyonlanmaktadır.

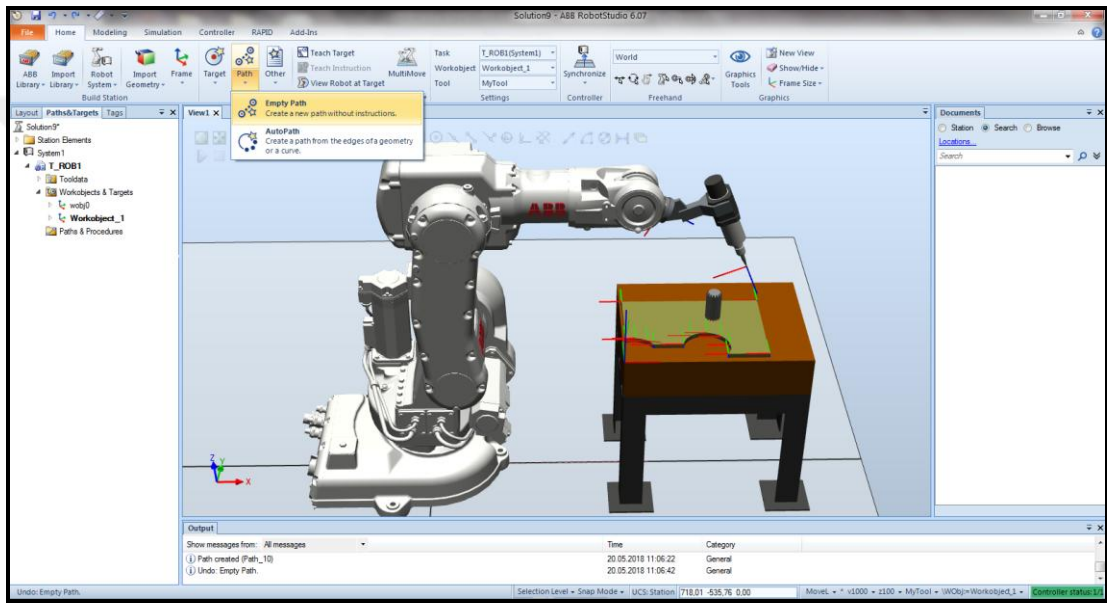


Şekil 2.31. Hedef noktayı kendi etrafında döndürmek – 3

Ardından yine tüm hedef noktalar birlikte seçilip herhangi bir üzerine sağ tıklanıp, “View Tool at Target” içindeki “myTool” seçimi kaldırıldığında, toolun hedef noktaların üzerindeki görüntüsü kaybolmaktadır.

2.2.6.4. Hedef noktalar kullanılarak robotun izleyeceği yolun tanımlanması

Robotun izleyeceği yolun tanımlanabilmesi için Şekil 2.32.’de görüldüğü gibi “Home” sekmesi altında, “Path” başlığı altında “Empty Path” seçimi yapılmalıdır. Bu seçim yapıldığında, aktif olan workobject altında “Path_10” oluşmaktadır.



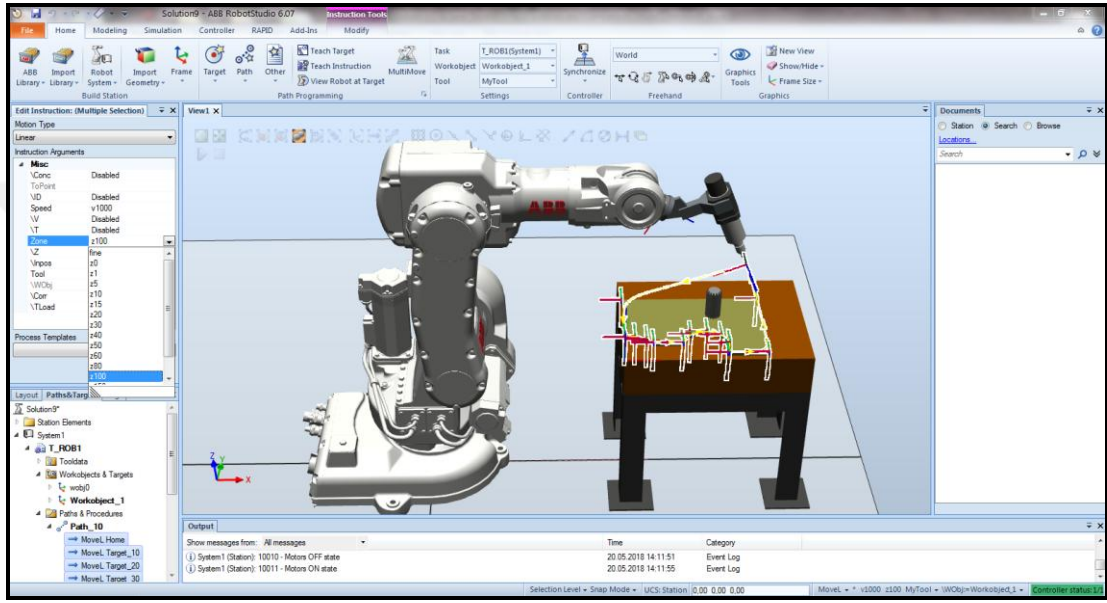
Şekil 2.32. Robotun izleyeceği yolun oluşturulması

Robotun izlemesi istenen güzergahı tanımlamak için, daha önce oluşturulmuş olan hedef noktalar, istenen sırayla sürükleyip bırak yöntemiyle Path_10’un içine bırakılmalıdır.

Hedef noktalar Path_10 içine bırakıldıktan sonra, toolun izleyeceği güzergah görünür olmaktadır. Ancak bu noktada bazı geliştirmelerden bahsedilmelidir. Ekranın sağ alt köşesindeki z100 ifadesi, daha önce de ifade edildiği gibi, robotun hedef noktalara yaklaşma hassasiyetini belirtmektedir. Toolun, hedef noktaların tam üzerinden geçmesi isteniyorsa, Şekil 2.33’te görüldüğü gibi, tüm hareket komutları birlikte seçilip sağ tıklanıp “Edit Instruction” seçimi yapılmalıdır. Açılan “Edit Instruction” alanında, Zone kısmının yanındaki oka tıklanıp, başlangıçta z100 olan hassasiyet

değeri istenildiği gibi değiştirilebilmektedir. Bu çalışmada, Zone değeri fine olarak değiştirilmiştir.

Yine “Edit Instruction” alanında, robotun hız değeri de bulunmaktadır. Başlangıçta v1000 olarak seçili gelen hız değeri değiştirilmek istenirse, yanındaki oka tıklanıp, açılan kutucuktan seçim yapılmalıdır. Bu çalışmada, Speed değeri v500 olarak seçilmiştir.

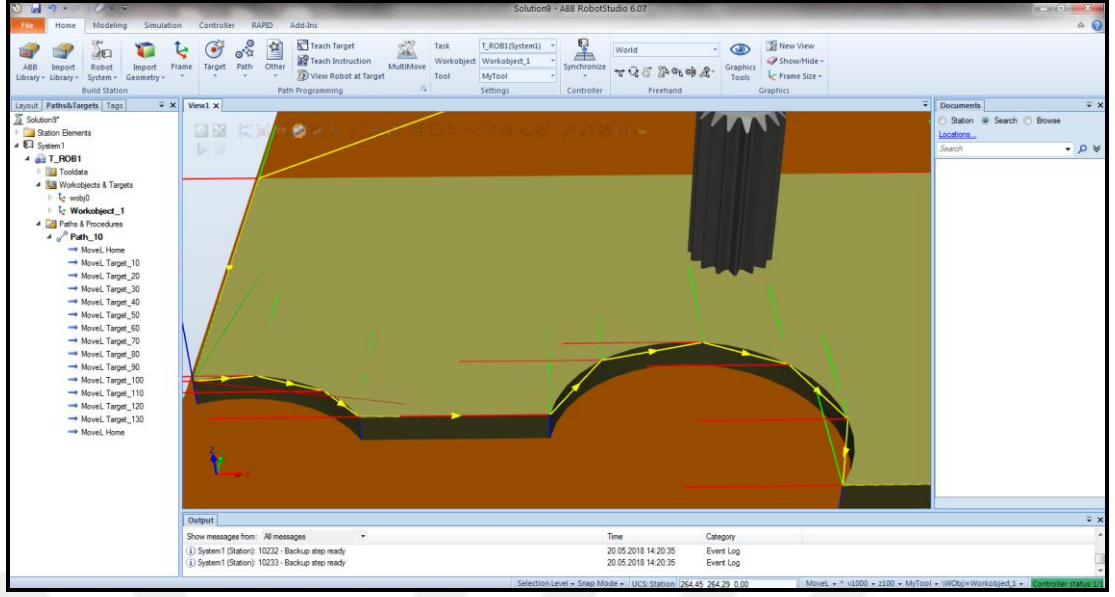


Şekil 2.33. Robotun Zone hassasiyetinin değiştirilmesi

Değişiklikler yapıldıktan sonra “Apply” seçimi yapılmalıdır. Ardından “Close” seçimi ile bu alan kapatılmalıdır.

Hedef noktalar ve hareket ayarları tanımlanmıştır ancak Şekil 2.34.’te görüldüğü gibi, dairesel bölgeler üzerinde tanımlanan noktalar arasındaki yollar da doğrusal olarak gösterilmektedir.

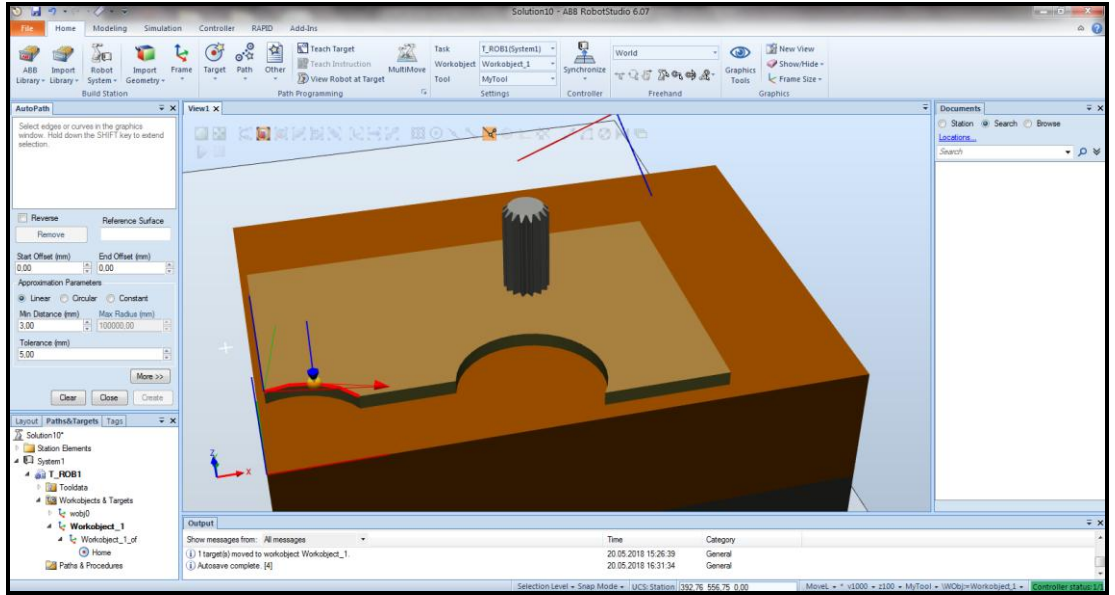
Bu sorunu gidermek ve toolun dairesel bölgedeki noktalar arasında dairesel hareket yapmasını sağlamak için, dairesel bölge içinde birçok nokta tanımlanması gerekmektedir. Ancak bunun yerine, dairesel bölgeler için geliştirilmiş olan “AutoPath” özelliği tercih edilmektedir.



Şekil 2.34. Hedef noktalar arasında çizilen yol

Yeni bir AutoPath tanımlamadan önce, daha önceden tanımlanmış olan tüm hedef noktalar ve güzergahlar silinmiştir. Yalnızca Home noktası, aynı şekilde kalması tercih edildiği için silinmemiştir.

Yeni bir AutoPath tanımlamak için, “Home” sekmesi altında, “Path” başlığı altında “AutoPath” seçimi yapılmalıdır. Bu seçim yapıldığında, Şekil 2.35.’te görüldüğü gibi, ekranın sol tarafında “Autopath” alanı açılmaktadır.



Şekil 2.35. AutoPath oluşturulması

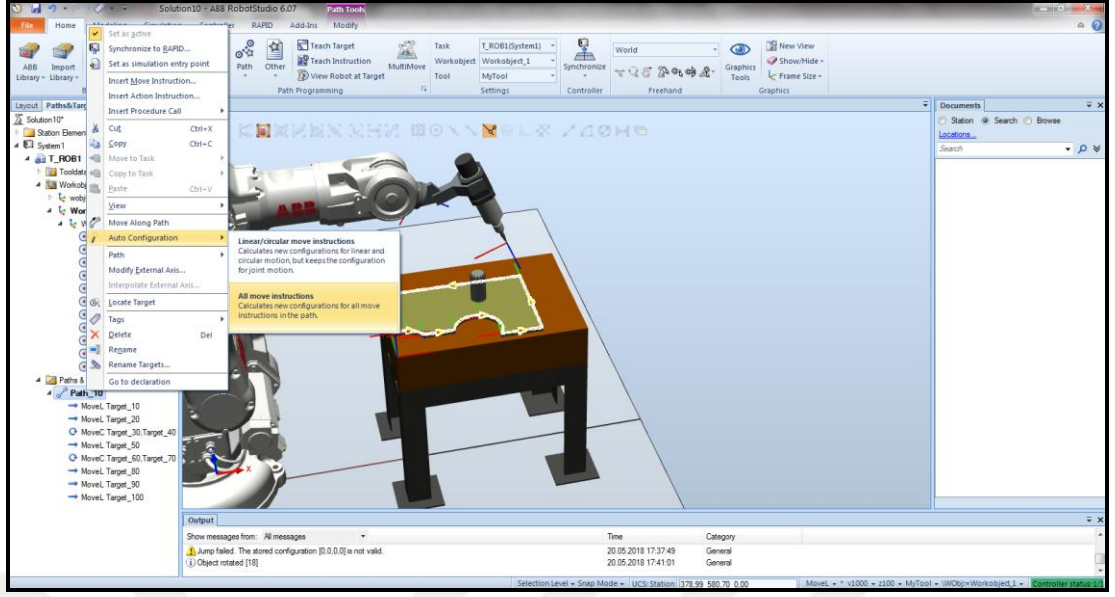
Bu alan açıldığında, köşe seçimi yapmaya gerek olmadan, izlenmesi istenen yol üzerinde birer adet nokta seçilmelidir. Bu seçim yapıldıktan sonra, Autopath alanındaki “Linear” seçimi yerine “Circular” seçimi yapılmalıdır. Ardından “Create” butonu tıklandığında rota oluşmaktadır.

AutoPath yönteminde hedef noktalar ayrı ayrı seçilmek yerine direkt olarak rota tanımlanmaktadır ancak yine de seçili workobject altında hedef noktalar otomatik olarak oluşmaktadır. Hedef noktalar yeni oluştuğu için bu aşamada, daha önce tarif edildiği gibi, “Rotation” komutuyla gerekli eksen etrafında döndürülüp toolun pozisyonlanmasına uygun duruma getirilmelidir.

Hedef noktalar da uygun şekilde döndürüldükten sonra bu çalışma için, Zone değeri tekrar “fine” olarak ve Speed değeri de tekrar v500 olarak değiştirilmiştir. İlave olarak da Path_10 rotasının başına ve sonuna birer adet Home pozisyonu eklenmiştir.

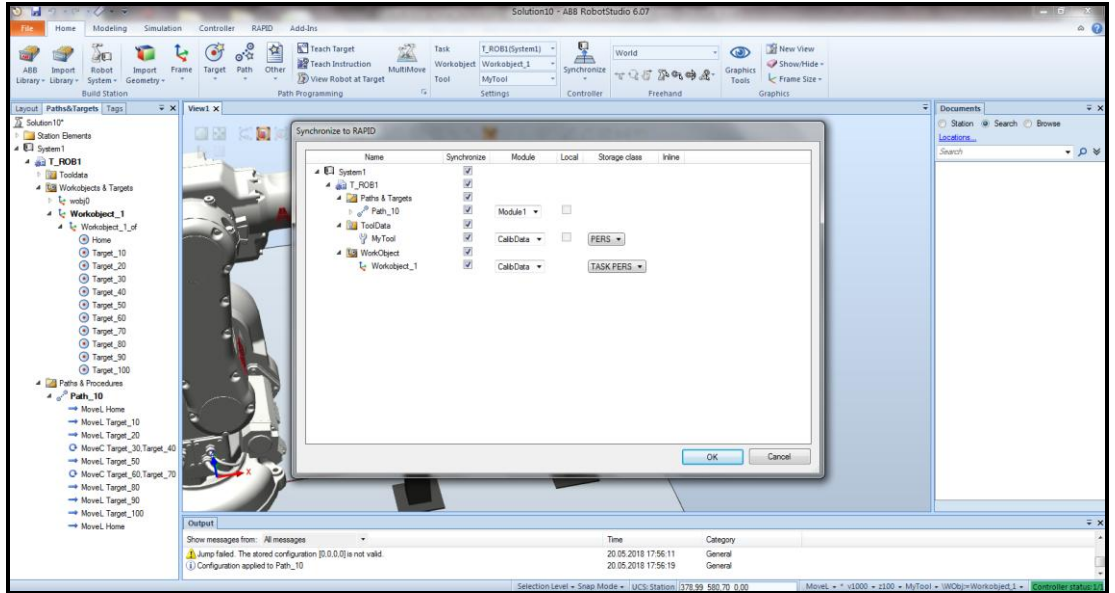
Tüm bu işlemlerin ardından güzergahın tanıtılması için bir konfigürasyon yapılması gerekmektedir. Bunun için de Şekil 2.36.’da görüldüğü gibi, Path_10 sağ tıklanıp “Auto Configuration” altından “All move instructions” seçimi yapılmalıdır. Bu seçim yapıldığında robotun, belirlenen güzergah üzerinden bir kez hızla geçmesi beklenmektedir. Bu işlem gerçekleşirse rota doğru tanımlanmıştır.

Ancak bir veya daha fazla eksenin zorlanması dolayısıyla konfigürasyon tamamlanamazsa, ekranın altındaki mesaj bölümünde bir hata mesajı gösterilmektedir. Hata mesajında robotun, rotanın hangi kısmında zorlandığı belirtilmektedir. Bu durumda, rotanın bu kısmında toolun pozisyonu incelenip, toolun buradaki pozisyonunda robotun eksenlerini zorlamayacak daha farklı pozisyonlar denenmelidir. Her bir denemenin ardından hatasız konfigürasyona ulaşana kadar konfigürasyon işlemi tekrarlanmalıdır. Bu çalışmada konfigürasyon hatasız olarak yapılmıştır.



Şekil 2.36. Rota için konfigürasyon yapılması

Hatasız yapılan konfigürasyonun ardından, tüm yapılan işlemlerin RAPID program altyapısına aktarılması gerekmektedir. Bunun için “Home” sekmesi altında “Synchronize” başlığı altında “Synchronize to RAPID...” seçimi yapılmalıdır. Bu seçim yapıldığında, Şekil 2.37.’de görüldüğü gibi bir ekran ortaya çıkmaktadır.



Şekil 2.37. Yapılan işlemlerin RAPID ile senkronize edilmesi

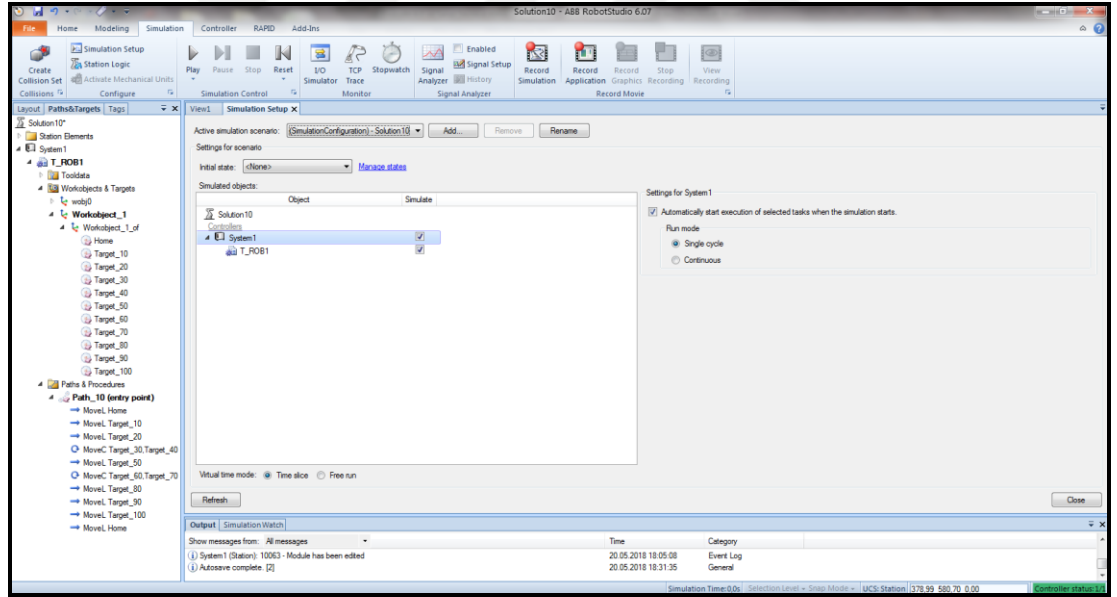
Bu ekranda RAPID ile senkronize edilmesi istenen seçimler yapılmalı ve OK tıklanmalıdır.

2.2.6.5. Simulasyon yapılması

Simulasyon yapılırken rota çizgilerinin ekranda görünmemesi için Path_10 üzerine sağ tıklanıp “View” altında, “Visible” seçimi kaldırılmalıdır.

Simulasyonun hazırlanması için “Simulation” sekmesi altında “Simulation Setup” tıklandığında, Şekil 2.38.’de görülen ekran açılmaktadır. Robotun tanımlanmış olan rotayı bir kez tamamlayıp işini bitirmesi isteniyorsa bu ekranın sol tarafında “Simulation Objects” altında “System 1” üzerine bir kez tıklayıp, ekranın sağ tarafında “Settings for System 1” altındaki “Run Mode” kısmında “Single Cycle” seçimi yapılmalıdır.

Robotun tanımlanan rota üzerinde sürekli yol alması isteniyorsa burada “Continuous” seçimi yapılmalıdır. Bu çalışmada tek bir çevrim tercih edilmiş ve “Single Cycle” seçimi yapılmıştır.



Şekil 2.38. Simulasyon tercihlerinin yapılması

Simule edilecek rotayı seçmek için ise “Simulation Objects” altında, “System 1” altında “T_ROB1” üzerine bir kez tıklayıp, “Settings for T_ROB1” altındaki “Entry point” kısmında simule edilmesi istenen rota seçilmelidir. Bu çalışmada “Path_10” rotası seçilmiştir. Ardından “Simulation Setup” ekranı kapatılabilir.

Bundan sonra “Simulation” sekmesi altında “Play” tuşuna basıldığında, robot seçilen rota üzerinde hareketini simule etmektedir.

2.2.7. Parça alıp bırakma uygulaması

Robotun parça alıp bırakma uygulaması, otomotiv sektöründe çokça kullanımı olan bir uygulamadır. Bu uygulama için gripper isimli tool kullanılmaktadır. Robot, herhangi bir istasyonda işi biten bir parçayı gripper ile alıp başka bir istasyona bırakabilmektedir.

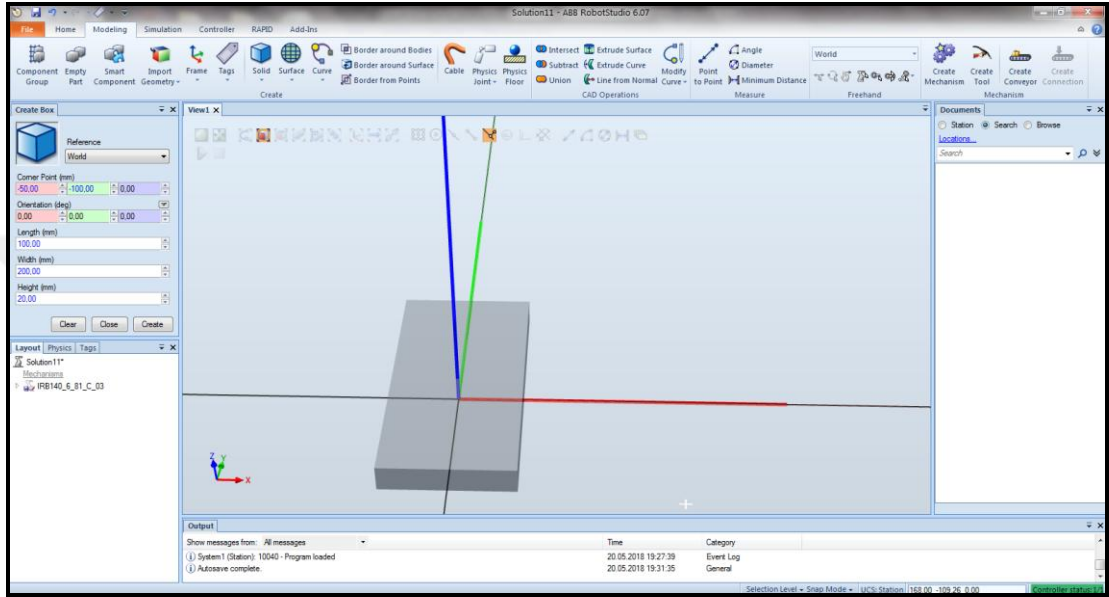
Bu çalışmada, önce yeni bir sehpa ve gripper dizayn edilip kütüphaneye eklenmiş, ardından bir sehpa üzerindeki kutunun gripper ile alınıp, diğer bir sehpa üzerine bırakıldığı, sonra bırakılan parçanın tekrar alınıp önceki sehpanın üzerine bırakıldığı bir uygulamanın simülasyonu yapılmıştır. Bu uygulamada yapılan işlemler sırasıyla aşağıdaki gibi tariflenebilmektedir:

- İstasyon oluşturulur.
- İstasyona robot eklenir.
- Robota sistem kurulur.
- Gripper dizayn edilip bir mekanizma olarak tanımlanır.
- Sehpa dizayn edilir.
- Gripper ile taşınacak iş parçanın oluşturulur.
- SmartComponent tanımlanır.
- Robot sistemi içinde gerekli I/O sinyalleri tanımlanır.
- Station Logic kısmında gerekli konfigürasyon yapılır.
- Robotun izleyeceği yol tanımlanır.
- RAPID altyapısında gerekli kodlar yazılır.
- Simülasyon yapılır.

İstasyon oluşturulması, robot eklenmesi ve robota sistem kurulması kısımları daha önce tarif edildiği için direkt olarak gripper dizayn edilmesi adımından devam edilmiştir.

2.2.7.1. Gripper dizayn edilip bir mekanizma olarak tanımlanması

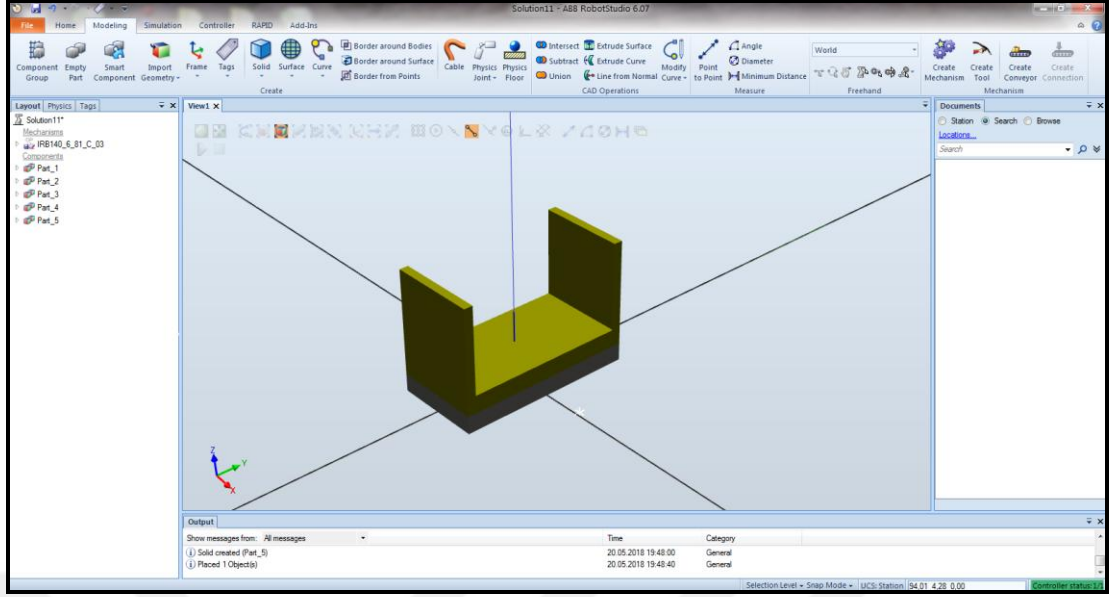
Gripper dizayn etmek için önce Şekil 2.39.'da görüldüğü gibi bir taban parçası dizayn edilmiştir. Bu parçanın orta noktasını, koordinat sisteminin merkezine konumlamak önemlidir. Lakin daha sonra gripperın tabanı olacak bu parça, gripper robotun ucuna bağlanırken merkez noktasından bağlanacaktır.



Şekil 2.39. Gripper için taban parçası oluşturulması

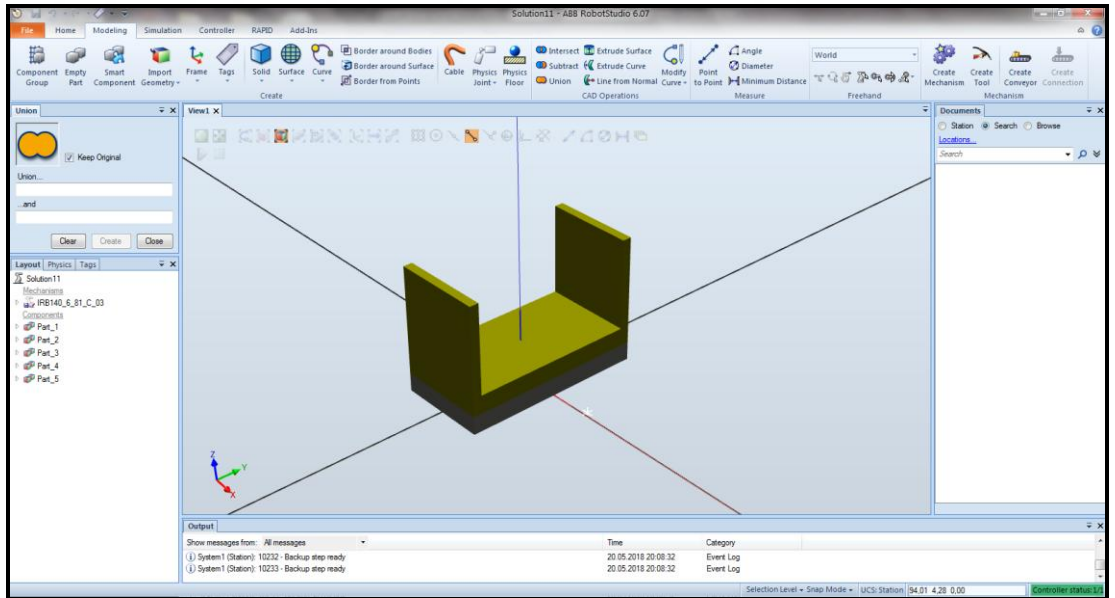
Ardından tabanın üzerinde yanlara doğru açılması istenen iki parça, taban parçasının üzerine konumlanacak şekilde oluşturulmuştur. Tabanın üzerinde açılıp kapanacak olan parçalar tabandan fark edilebilmeleri için farklı renkte dizayn edilmiştir. Bunun için Layout kısmında parçanın ismi üzerine sağ tıklayıp (Part_3 örneği gibi), “Modify” kısmında “Set Color...” seçimi yapılmalıdır. Burada istenilen renk seçilip OK tıkladığında parçanın rengi değişmektedir.

Bundan sonra da Şekil 2.40.'ta görüldüğü gibi açılıp kapanan parçaların üzerine, iş parçasını tutacak olan parçalar dizayn edilip yerleştirilmiştir. Ardından bu parçalar da, açılıp kapanan parçalarla aynı renge dönüştürülmüştür.



Şekil 2.40. Gripper için hareketli parçaların oluşturulması

Bu aşamada, açılıp kapanan parça ile üzerindeki tutucu parçanın birleştirilip tek parça yapılması gerekmektedir. Bunun için “Modeling” sekmesi altında “Union” seçimi yapıldığında, ekranın sol tarafında Şekil 2.41.’de görüldüğü gibi bir “Union” alanı ortaya çıkmaktadır.



Şekil 2.41. Union özelliği ile parçaların birleştirilmesi

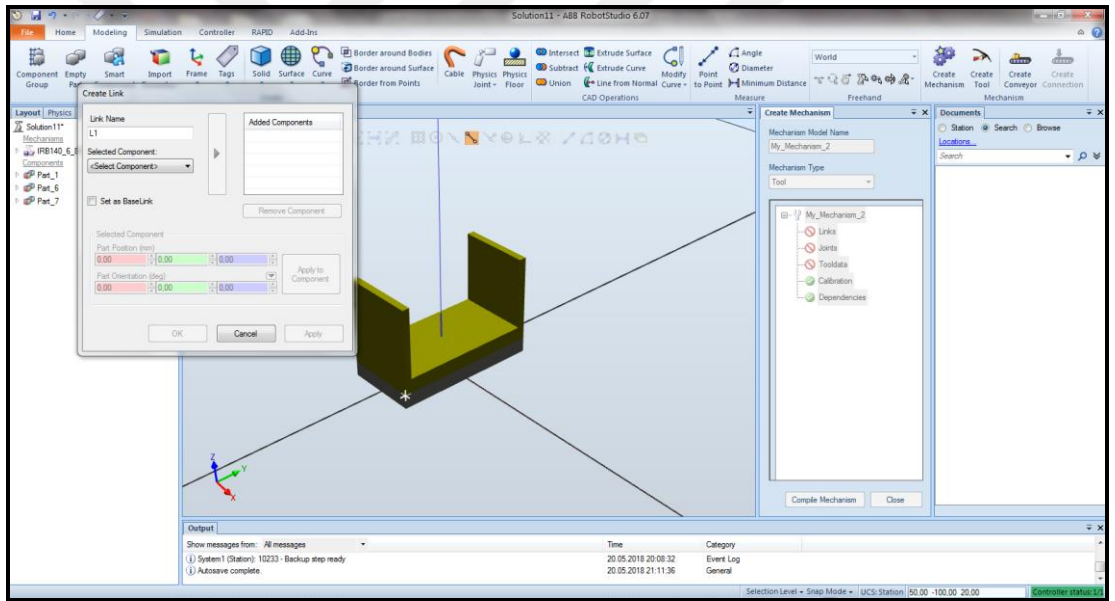
Union ile iki parçanın birleştirilmesi sağlanmaktadır. Bunun için “Keep Original” üzerindeki seçim kaldırılmalıdır. Ardından “Union...” kutucuğunun içine tıklanıp ilk

parça seçilmelidir. Ardından “...and” kutucuğunun içine tıklanıp ikinci parça seçilmelidir. Sonra “Create” tıklanınca birleşme sağlanmaktadır.

Dizayn edilen bu parçayı hareketli bir mekanik objeye dönüştürebilmek için “Modeling” sekmesi altında “Create Mechanism” seçimi yapılmalıdır. Bu seçim yapıldığında “Create Mechanism” başlıklı bir sayfa açılmaktadır. Bu sayfada mekanizmanın ayarları yapılmaktadır.

Burada önce mekanizmanın adı, “My_Mechanism_2” olarak değiştirilmiştir. Ardından “Mechanism Type” kısmında “Tool” seçilmiştir.

Sonrasında, aksamaları tanımlamak için “Link” seçimi tıklandığında, Şekil 2.42.’de görüldüğü gibi “Create Link” adlı bir seçim ekranı açılmaktadır.

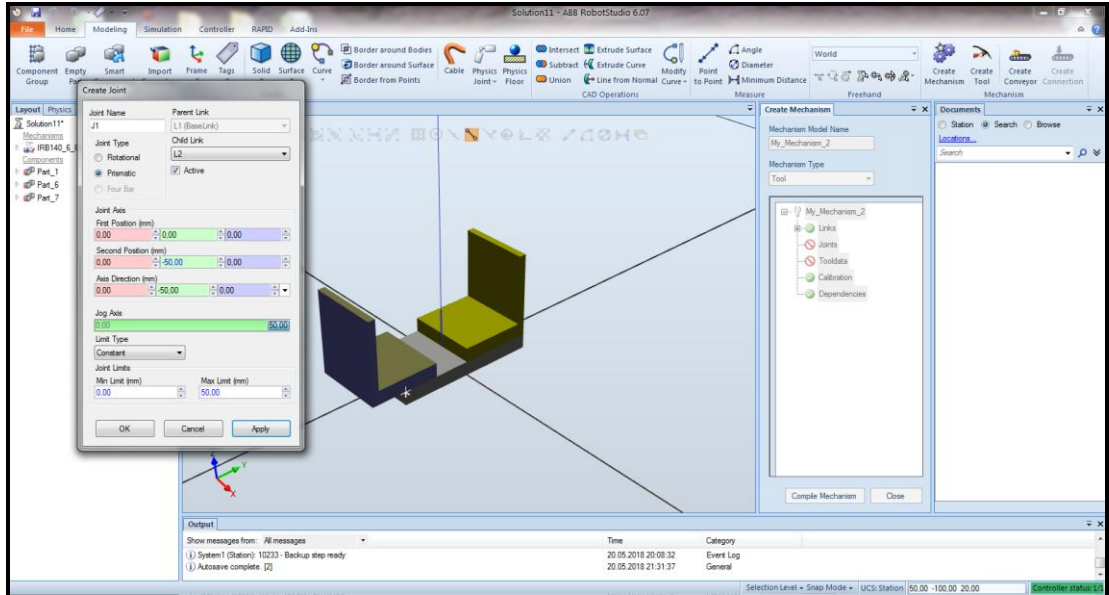


Şekil 2.42. Aksamaların tanımlanması

Burada “Link Name” kutucuğunda önce L1 isimli birinci aksam tanımlanmaktadır. Altındaki “Selected Component” kutucuğunda “Part_1” isimli taban parçası seçilmiş ve altında “Set as BaseLink” seçimi yapılarak taban parçası BaseLink olarak belirlenmiştir. Ardından “Apply” butonu tıklandığında L1 aksamının seçimi tamamlanmakta ve L2 aksamı gelmektedir. L2 ve L3 aksamaları için de hareketli parçalar tanımlandıktan sonra OK butonu tıklanıp aksamaların belirlenmesi tamamlanmaktadır.

Ardından hareketli aksamaların hareket limitlerini tanımlamak için “Joints” seçimi tıklandığında, “Create Joint” adlı bir seçim ekranı açılmaktadır. Burada “Joint Name” kutucuğunda önce J1 isimli birinci hareket tanımlanmaktadır. Bu örnekteki hareket ileri geri olduğu için, “Joint Type” kutucuğunda “Prismatic” seçimi yapılmalıdır. Burada sabit olan aksam, “Parent Link” kısmında “L1 (BaseLink)” olarak görünmektedir ve bu değiştirilemez. “Child Link” kutucuğunda ise hareketi tanımlanacak olan aksam seçilmelidir. Burada ilk olarak L2 aksamının hareketi tanımlanacağı için bu aksam seçilmiştir.

“Joint Axis” kısmında “First Position (mm)” sabit tutularak “Second Position (mm)” kutucuğunda yalnızca y ekseninde -50 değeri girilmiştir. Ardından “Joint Limits” kısmında “Min Limit (mm)” değeri 0 olarak ve “Max Limit (mm)” değeri 50 olarak belirtilmiştir. Bu şekilde L2 aksamına mevcut konumundan y ekseninde dışarıya doğru 50mm lik bir hareket tanımlanmıştır. “Jog Axis” kısmındaki kutucuk Şekil 2.43.’teki gibi hareket ettirilerek, parçanın hareketi gözlenmiştir. Ardından “Apply” butonu tıklandığında J1 hareketi tanımlanmakta ve J2 hareketi gelmektedir. J2 hareketi de aynı şekilde tanımlandıktan sonra OK butonu tıklayıp hareketlerin tanımlanması tamamlanmaktadır.

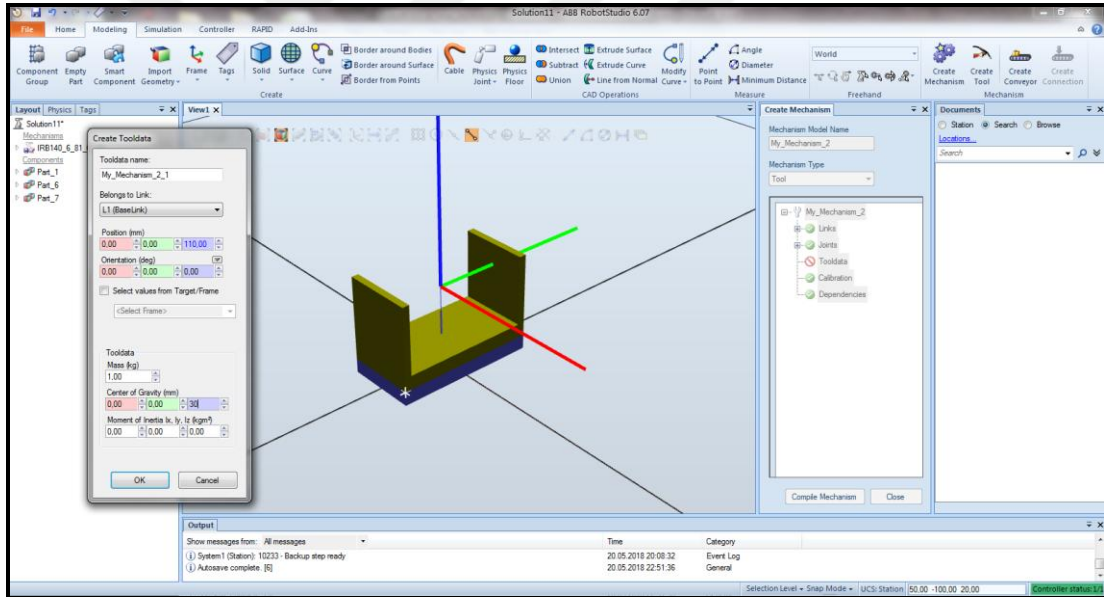


Şekil 2.43. Hareketlerin tanımlanması ve denenmesi

Ardından mekanizmanın uç noktası, ağırlığı, ağırlık merkezi gibi temel özelliklerini tanımlamak için “Tooldata” seçimi tıklandığında, “Create Tooldata” adlı bir seçim

ekranı açılmaktadır. Burada söz edilen uç nokta, robota bağlanan herhangi bir toolda olduğu gibi, mekanizmanın herhangi bir hedef noktaya ulaşabilmesi için üzerinde olması gereken kılavuz noktasıdır.

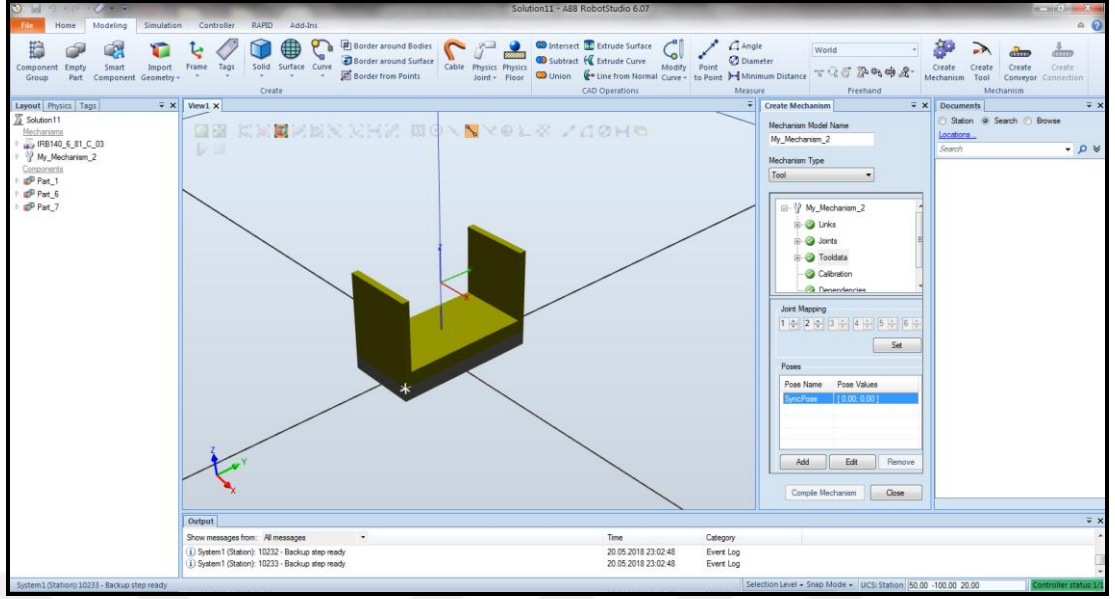
Burada Şekil 2.44.'te görüldüğü gibi, “Belongs to Link” kısmında “L1 (BaseLink)” seçimi yapılarak bu noktanın taban aksamına bağlı olması belirtilmiştir. Konum olarak da x ve y eksenleri merkez nokta ile aynı tutulup, z ekseninde merkez noktadan 110 mm yukarıya tanımlanmıştır. Aşağıda “Tooldata” kısmında ise “Mass (kg)” için 1 değeri girilmiş ve toolun ağırlığı 1 kg olarak belirlenmiştir. Ardından “Center of Gravity (mm)” kısmında ağırlık merkezi olarak x ve y eksenleri yine merkez nokta ile aynı tutulup, z ekseninde merkez noktadan 30 mm yukarıya seçilmiştir. Ardından OK butonu tıklanınca “Tooldata” oluşturulmuş ve “Create Tooldata” seçim ekranı kapanmıştır.



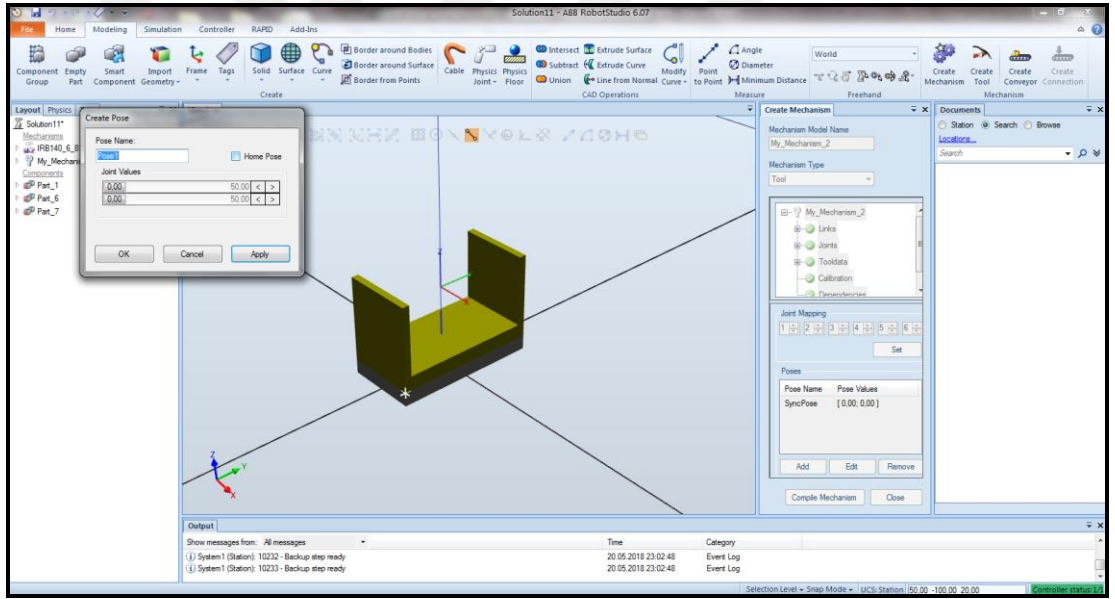
Şekil 2.44. Tooldata değerlerinin belirlenmesi

Bundan sonra “Create Mechanism” ekranının sol alt köşesindeki “Compile Mechanism” butonu tıkladığında, bu ekranda mekanizmanın durumlarını belirleyebilmek için Şekil 2.45.'te görüldüğü gibi “Poses” isimli kutucuk ortaya çıkmaktadır.

Öncelikle hareketli parçalar kapalı haldeykenki durumu Home pozisyon olarak belirleyebilmek için, burada “Add” butonu tıklanmış ve Şekil 2.46.'da görülen “Create Pose” ekranı ortaya çıkmıştır.



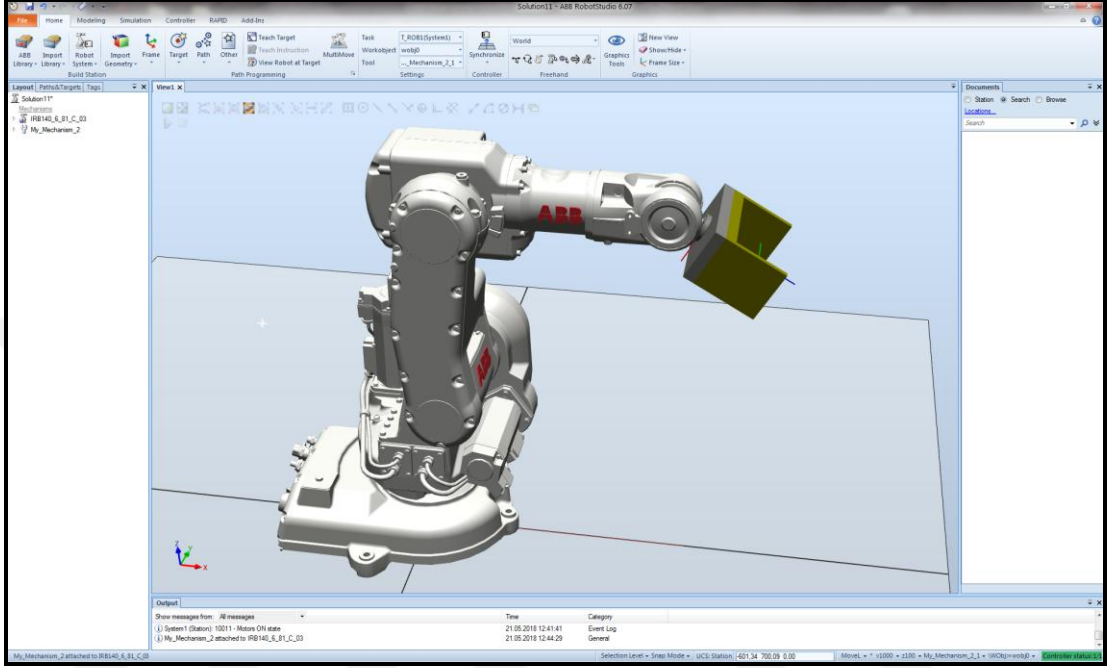
Şekil 2.45. Toolun durumlarının belirlenmesi – 1



Şekil 2.46. Toolun durumlarının belirlenmesi – 2

Bu ekranda “Home Pose” kutucuğu seçilip “Apply” butonuna tıklandığında toolun hareketli aksamalarının kapalı haldeki durumu, başlangıç durumu olarak “HomePose” ismiyle tanımlanmıştır. Ardından yine aynı ekranda, bu kez “Joint Values” kısmındaki iki gösterge de son konumuna kadar açılıp, toolun hareketli aksamalarının açık haldeki durumu, yine “Apply” butonuna tıklanarak, “EndPose” ismiyle toolun son durumu olarak tanımlanmıştır. Ardından “Close” butonuna tıklanıp “Create Mechanism” ekranı kapatılmıştır.

Bu şekilde mekanizmanın tanımlanma işlemi tamamlanmıştır. Ardından robot tekrar görünür hale getirilip gripper mekanizması robota bağlandığında, Şekil 2.47.'de görüldüğü gibi gripper tam merkez noktasından robotun son eksenine üzerine konumlanmıştır.

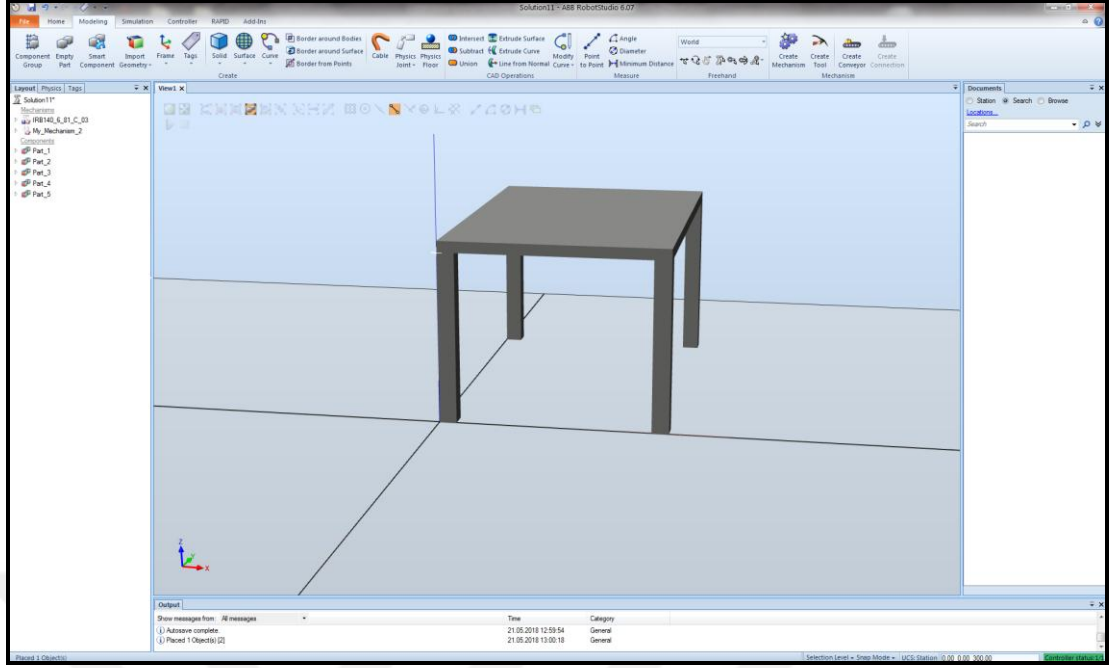


Şekil 2.47. Tanımlanmış olan gripperın robota bağlanması

2.2.7.2. Sehpa dizayn edilmesi

Bunun için önce sehpanın yüzeyi ve ayakları dizayn edilmiştir. Ardından Şekil 2.48.'de olduğu gibi yüzey, ayakların üzerine yerleştirilmiştir. Sonrasında Union işlemi ile yüzey ve dört adet ayaktan oluşan toplam 5 parçanın birleştirilerek bir adet sehpa olarak tanımlanması sağlanmıştır.

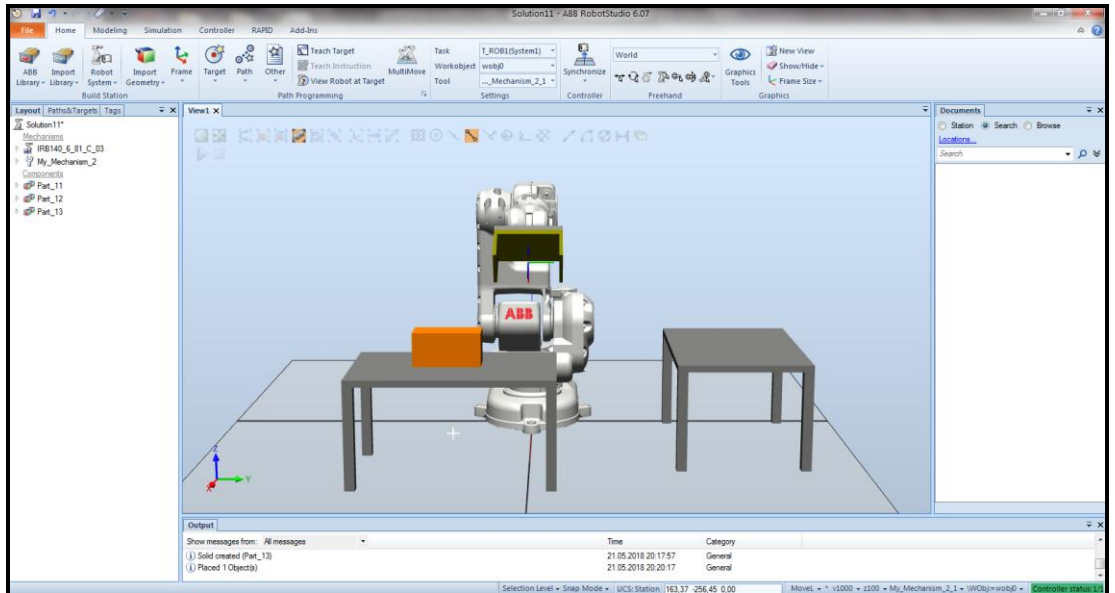
Bunun ardından robot tekrar görünür yapılmış ve bir adet daha sehpa eklenerek, sehpa robotun çevresine uygun konumda yerleştirilmiştir.



Şekil 2.48. Sehpa dizayn edilmesi

2.2.7.3. Gripper ile taşınacak iş parçasının oluşturulması

İş parçasının boyutlarının daha net görülebilmesi için robot ve tool tekrar görünmez yapılmıştır. Ardından iş parçası oluşturulmuş, fark edilmesi için rengi değiştirilmiş ve Şekil 2.49.'da olduğu gibi, sehpanın üzerine konulmuştur.



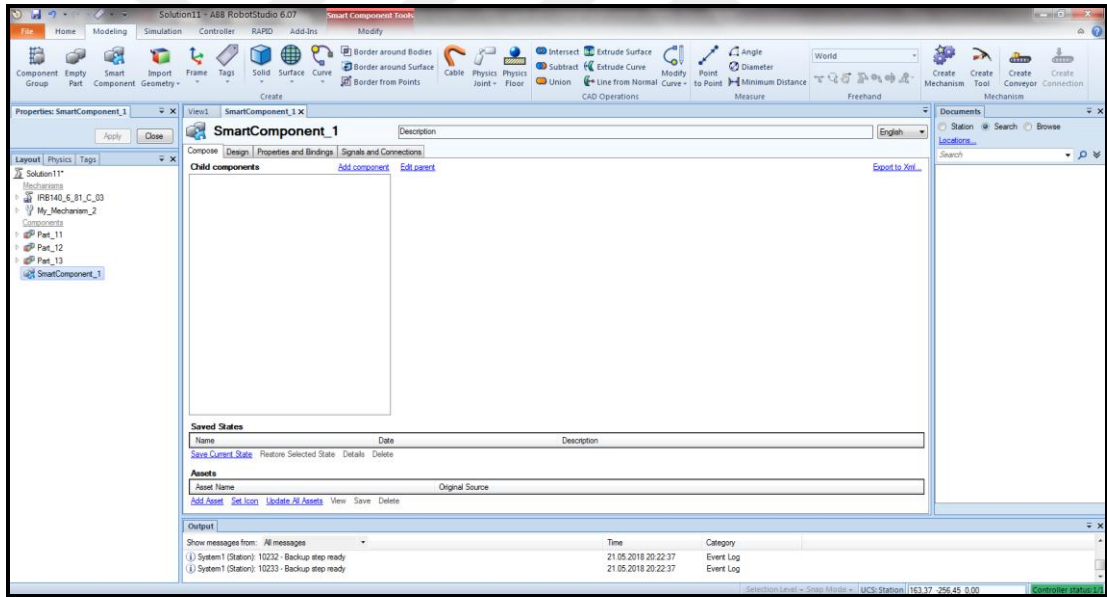
Şekil 2.49. Gripper ile taşınacak iş parçasının oluşturulması

2.2.7.4. Smart Component tanımlanması

Parçaların rutin hareketlerin dışında karmaşık davranışlar göstermesi istendiğinde, “Smart Component” özelliği kullanılmaktadır. Parça alıp bırakma uygulamasında iş parçasının gripper üzerindeki bir sensör ile algılanması, grippera bağlanıp onunla birlikte yol alması gibi davranışlar göstermesi gerekmektedir. Bunun için çeşitli Smart Component’ler tanımlanmıştır.

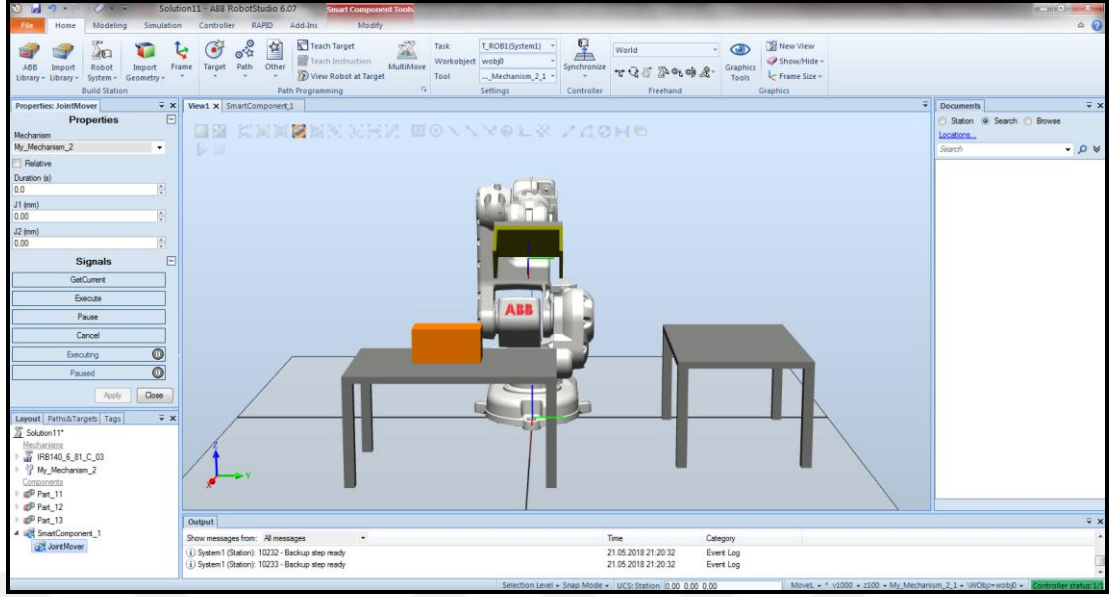
- JointMover

“Modeling” sekmesi altında “Smart Component” seçimi yapıldığında Şekil 2.50.’de görülen ekran açılmaktadır. Burada “Add Component” tıklandığında açılan kısımdan önce, gripperın açma ve kapama hareketlerini tanımlamak için iki adet “JointMover” eklenmiştir.



Şekil 2.50. Smart Component tanımlanması

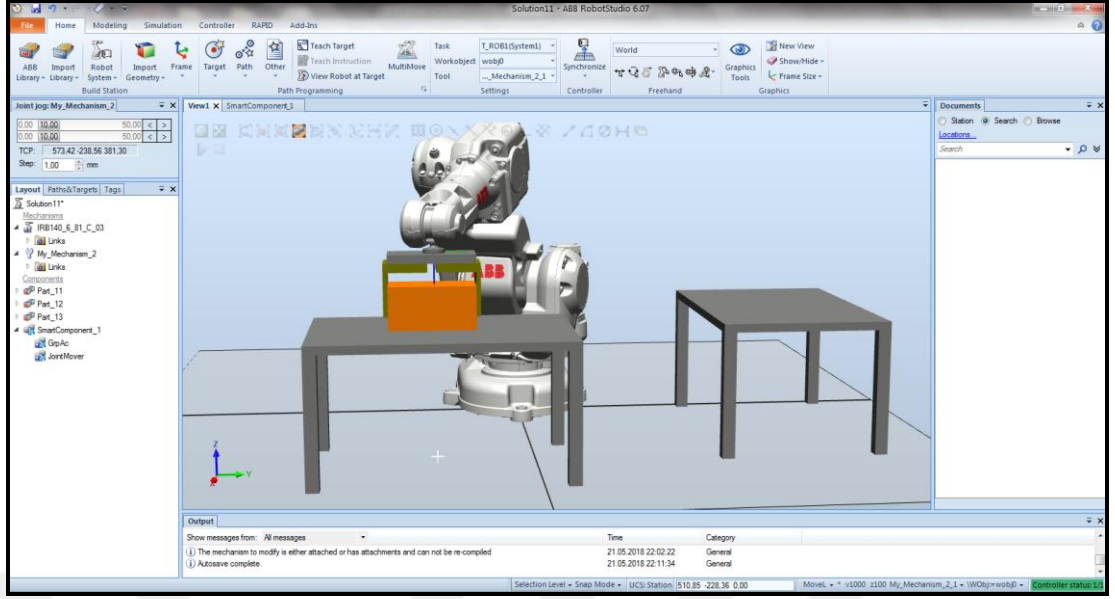
“JointMover” seçimi yapıldığında ekranın sol tarafında, Şekil 2.51.’de görülen “Properties: JointMover” ekranı açılmaktadır. Burada ayarları daha rahat yapabilmek adına, robot ve malzemelerin görüntülediği “View1” ekranına geçiş yapmak tercih edilmektedir.



Şekil 2.51. JointMover ayarlarının yapılması

“Properties: JointMover” ekranında “Mechanism” kısmında gripper seçilmiştir. J1 ve J2 kısımlarında, gripperın açık pozisyonunu tanımlamak için 50 mm değeri girilmiştir. Bunun tercih edilmesinin sebebi, gripper bir mekanizma olarak dizayn edilirken, açık pozisyonda iki hareketli aksamın da en fazla 50 mm açılacak şekilde dizayn edilmiş olmasıdır. Burada gripperın iş parçasının üzerine rahat girebileceği daha küçük açıklık değerleri de tercih edilebilmektedir. Değerler girildikten sonra “Apply” seçimi yapılmıştır. Ardından bu smart componentin adı “GrpAc” olarak değiştirilmiştir.

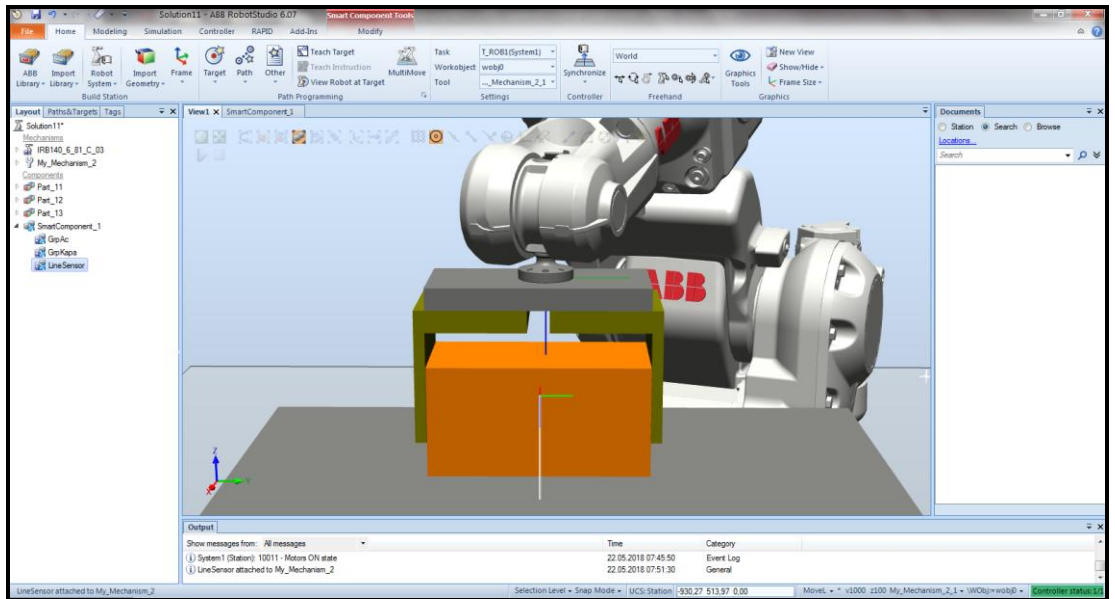
Bunun ardından gripperın kapalı durumunu tanımlamak için ikinci “JointMover” eklenmiştir. Bu kez gripperın iş parçasını alacak şekilde kapalı pozisyonunu tanımlayabilmek için, gripper önce “Jog Linear” ve “Jog Reorient” komutlarıyla açık olarak iş parçasının üzerine getirilmiştir. Ardından “My_Mechanism_2” üzerine sağ tıklayıp “Mechanism Joint Log” seçimi yapılmış ve Şekil 2.52.’de görülen ekranda, iki hareketli aksam hareket ettirilerek, gripper iş parçası üzerinde kapalı duruma getirilmiştir. Ardından “Mechanism Joint Log” kapatılıp “JointMover” tekrar açılmış ve burada “GetCurrent” seçimiyle, gripperın mevcut pozisyonunun tanımlanması sağlanmıştır. Ardından bu smart componentin adı “GrpKapa” olarak değiştirilmiştir.



Şekil 2.52. Mechanism Joint Log ekranında aksamaların hareket ettirilmesi

- LineSensor

Daha sonra, gripperın iş parçasını görebilmesi için “SmartComponent_1” içinden bir adet “LineSensor” eklenmiştir. Öncelikle LineSensor, tıpkı toolun robota bağlandığı gibi, toola bağlanmalıdır. Bunun için de LineSensor üzerine sağ tıklayıp “Attach to” seçimi altında gripper seçilmelidir. Bu seçim yapıldığında LineSensor, Şekil 2.53.’te görüldüğü gibi (beyaz bir çizgi şeklinde), gripperın tanımlanmış olan uç noktasına konumlanmaktadır.

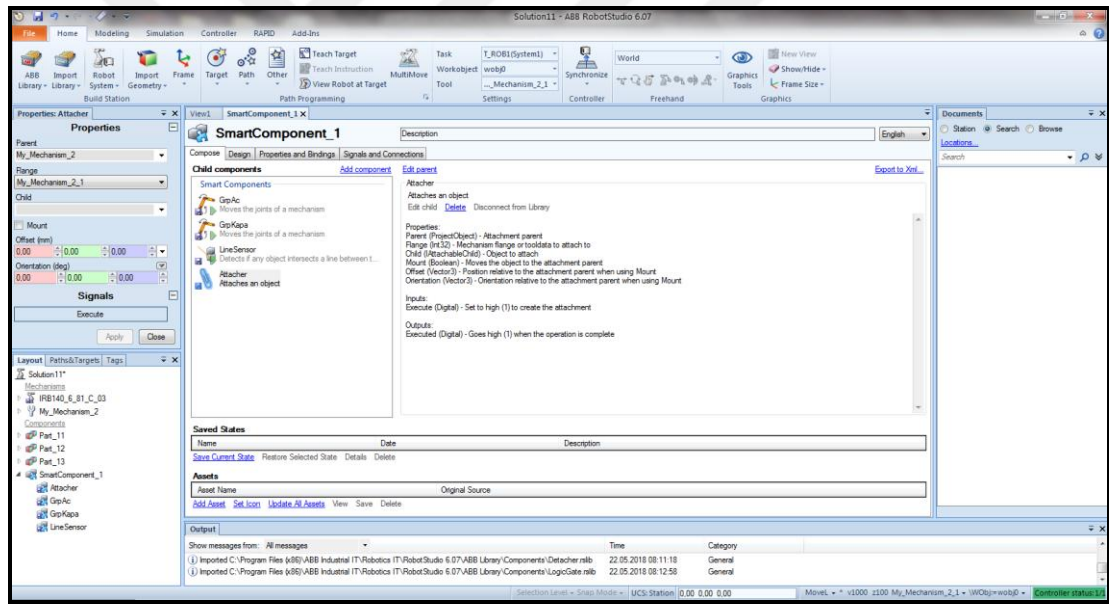


Şekil 2.53. LineSensorun grippera bağlanması

Bu noktada LineSensor test edilmek için ayar penceresinde “Active” butonu tıklandığında, “SensedPart” kısmında “Part_13” yazdığı görülmüştür. Ancak LineSensor Şekil 2.53.’de görülen konumdayken, iş parçasının da altında sehpa gibi başka cisimleri de görebilmesi mümkündür. Bunun önüne geçebilmek için LineSensor sadece iş parçasını görecek şekilde, ayar penceresindeki koordinatlarıyla oynanarak, biraz daha yukarıya konumlandırılmıştır.

- Attacher ve Detacher

Daha sonra, gripperın iş parçasını kendine bağlayıp birlikte hareket edebilmesi için “SmartComponent_1” içinden “Attacher” eklenmiştir. Şekil 2.54.’te görülen Attacher ayar ekranında “Parent” kısmında gripper seçilip bu ekran kapatılmıştır. Diğer ayarlar blok diyagramında yapılabilmektedir.



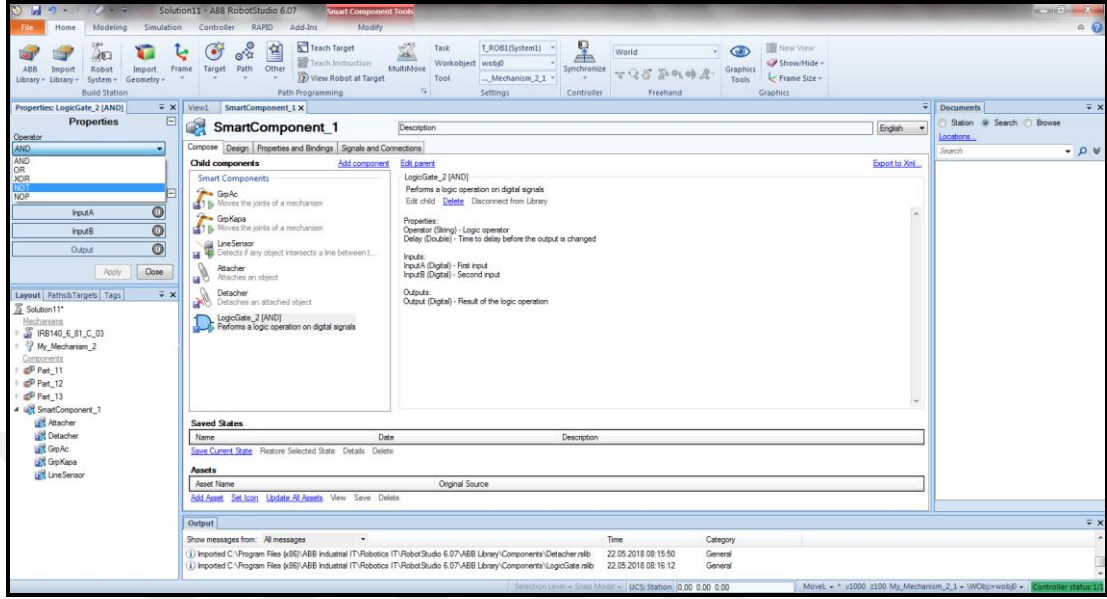
Şekil 2.54. Attacher eklenmesi

Ardından “Detacher” eklenmiş ve herhangi bir ayar yapılmadan, ayar ekranı kapatılmıştır. Bu SmartComponent için de gerekli ayarlar blok diyagramında yapılabilmektedir.

- LogicGate

Daha sonra, blok diyagramda kullanılabilmesi için bir adet “LogicGate” eklenmiş ve sinyali tersleyebilmesi için, Şekil 2.55.’te görüldüğü gibi “Operator” kısmında

“NOT” seçimi yapılmıştır. Bu şekilde LogicGate, bir NOT kapısı olarak tanımlanmıştır.



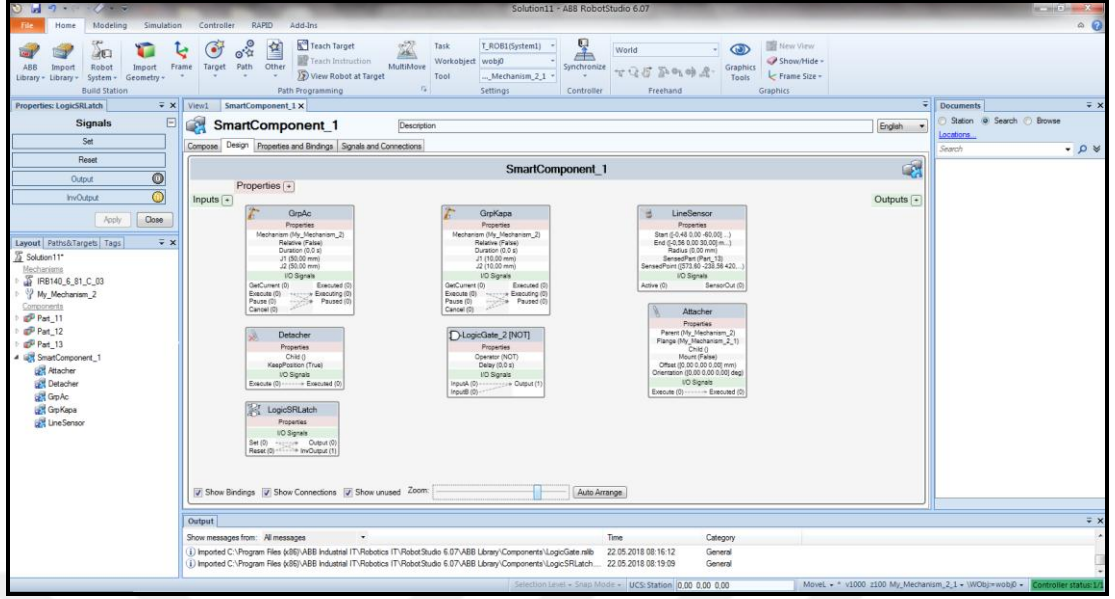
Şekil 2.55. LogicGate eklenmesi ve NOT kapısı olarak seçilmesi

- LogicSRLatch

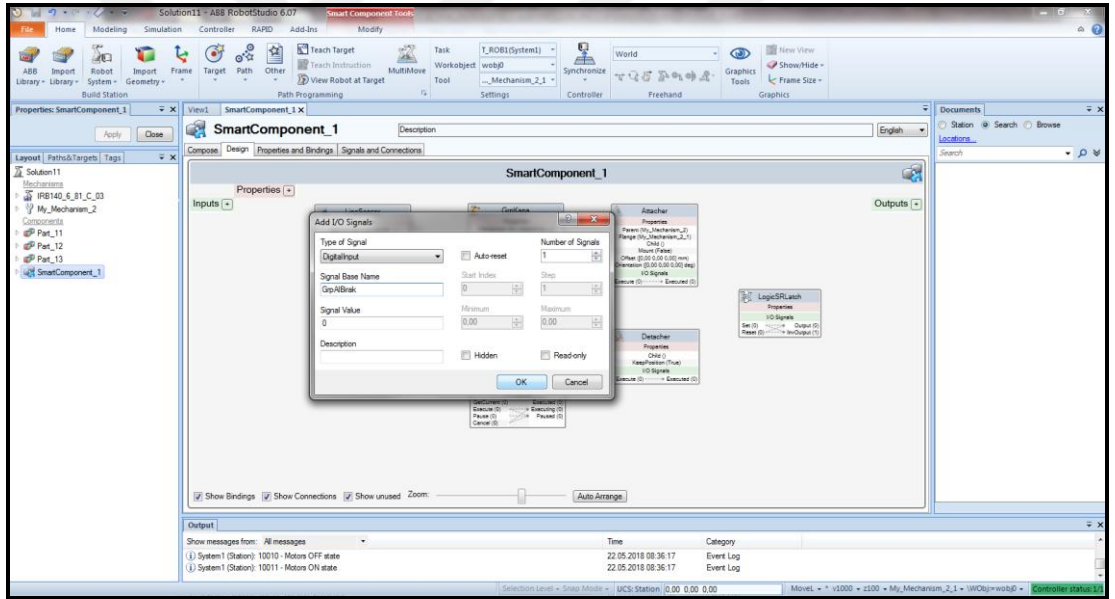
Daha sonra, yine blok diyagramda kullanılabilmesi için bir adet “LogisSRLatch” fonksiyonu eklenmiş ve herhangi bir ayar yapılmadan ayar penceresi kapatılmıştır. Bu bir Set-Reset fonksiyonudur ve bunun da ayarları blok diyagramında yapılabilmektedir.

Daha sonra “SmartComponent_1” ekranında “Design” sekmesine geçiş yapıldığında, Şekil 2.56.’daki blok diyagramı görünmektedir. Blok diyagramı ilk anda, tüm komponentler düzensiz dağılmış halde bulunmaktadır. Bu esnada önce komponentlerin yeri ve sırası düzenlenmiştir.

Bu ekranda, blok diyagramın solunda “Inputs” ve sağında “Outputs” kısmı görünmektedir. Burada SmartComponent için bir giriş ve çıkış sinyali belirlenmesi gerekmektedir. Bunun için “Inputs” yanındaki butona tıklanıldığında Şekil 2.57.’deki ekran gelmektedir.



Şekil 2.56. SmartComponent blok diyagramı



Şekil 2.57. Blok diyagramında Input sinyali tanımlanması

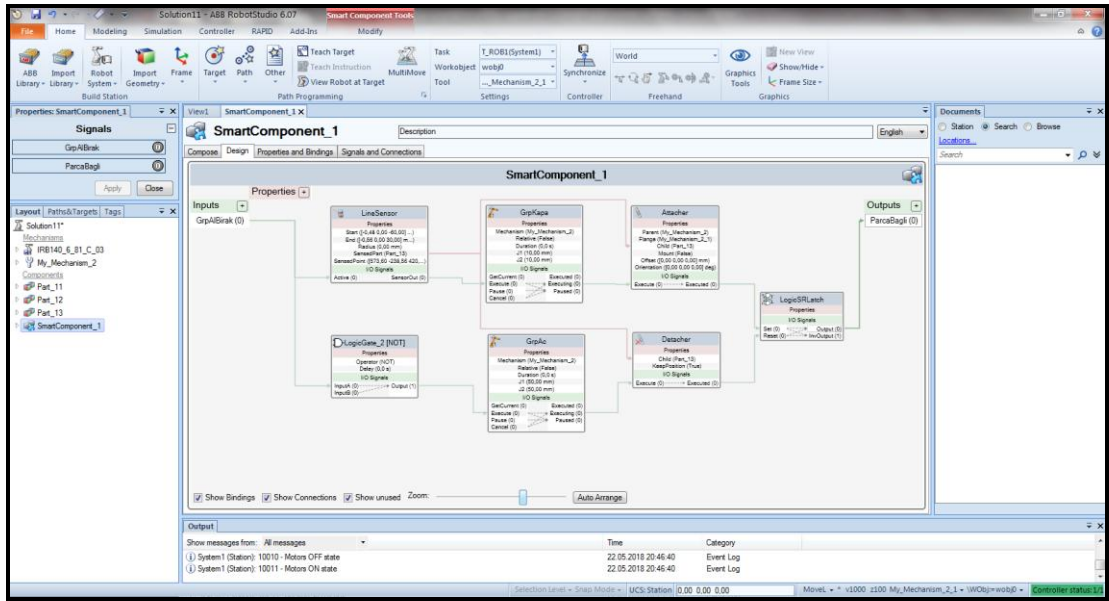
Burada “Type of Signal” kısmında “Digital Input” seçilidir. “Signal Base Name” kısmında ise bu sinyale “GrpAIBirak” isim verilmiştir. “Signal Value” 0 değerinde bırakılıp OK tıklanmış ve Input sinyali tanımlanmıştır. Ardından “Outputs” kısmından aynı şekilde, “ParcaBagli” isimli bir Output sinyali tanımlanmıştır.

Bundan sonra, blok diyagramındaki bağlantılar yapılabilmektedir. Öncelikle “GrpAIBirak” sinyalinden “LineSensor”un “Active” girişine bir bağlantı çekilmiştir. LineSensorun çıkışından GrpKapa komponentinin “Execute” girişine, GrpKapa

komponentinin “Executing” çıkışından da “Attacher” komponentinin “Execute” girişine bağlantı çekilmiştir. İlave olarak da “Attacher” komponentinin hangi cismi bağlayacağını belirtmek için LineSensorun “Sensed Part” kısmından “Attacher” komponentinin “Child” kısmına bir bağlantı çekilmiş ve bu kısma “Part_13” tanımlandığı görülmüştür. “Attacher” komponentinin “Executed” çıkışından “LogicSRLatch” komponentinin “Set” girişine bağlantı çekilmiştir.

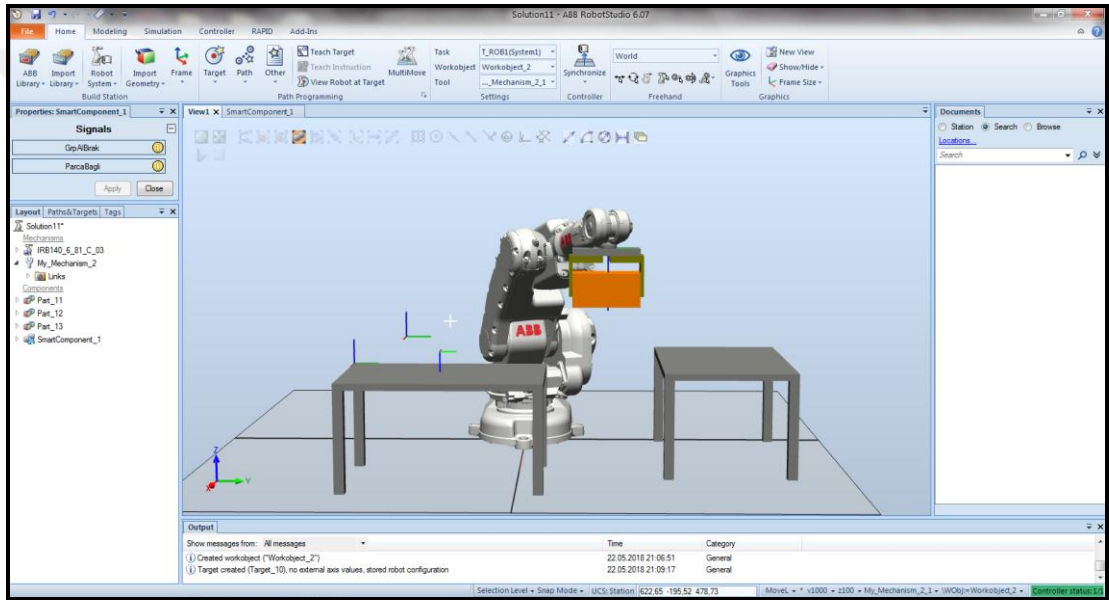
Sonrasında, “GrpAlBirik” sinyalinin 0 olduğu durumlar için de bu sinyalden “NOT” kapısının girişine bir bağlantı çekilmiştir. NOT kapısının çıkışından GrpAc komponentinin “Execute” girişine, GrpAc komponentinin “Executing” çıkışından da “Detacher” komponentinin “Execute” girişine bağlantı çekilmiştir. İlave olarak da “Detacher” komponentinin hangi cismi bırakacağını belirtmek için LineSensorun “Sensed Part” kısmından “Detacher” komponentinin “Child” kısmına bir bağlantı çekilmiş ve bu kısma “Part_13” tanımlandığı görülmüştür. “Detacher” komponentinin “Executed” çıkışından “LogicSRLatch” komponentinin “Reset” girişine bağlantı çekilmiştir.

Son olarak da “LogicSRLatch” komponentinin çıkışı, “ParcaBagli” sinyaline bağlanmış ve Şekil 2.58’de görüldüğü gibi SmartComponent_1 bu şekilde tamamlanmıştır.



Şekil 2.58. SmartComponent blok diyagramının tamamlanması

SmartComponent'in doğru çalıştığının teyit edilebilmesi için "View1" ekranına geri dönmüştür. Burada ekranın solundaki "Properties: SmartComponent_1" ekranında "GrpAlBirik" sinyalinin üzerine tıklanmış ve hem gripperın kapandığı, hem de "ParcaBagli" sinyalinin 1 olduğu ve dolayısıyla iş parçasının grippera bağlandığı görülmüştür. Burada bir deneme yapılmadan önce sehpa üzerine bir workobject tanımlanmış ve mevcut pozisyon, robota hedef nokta olarak tanıtılmıştır. Ardından denemek için "GrpAlBirik" sinyali 1 yapıp, gripper "Jog Linear" komutuyla hareket ettirildiğinde, Şekilde 2.59.'da görüldüğü gibi, gripperın iş parçasıyla birlikte hareket ettiği teyit edilmiştir.

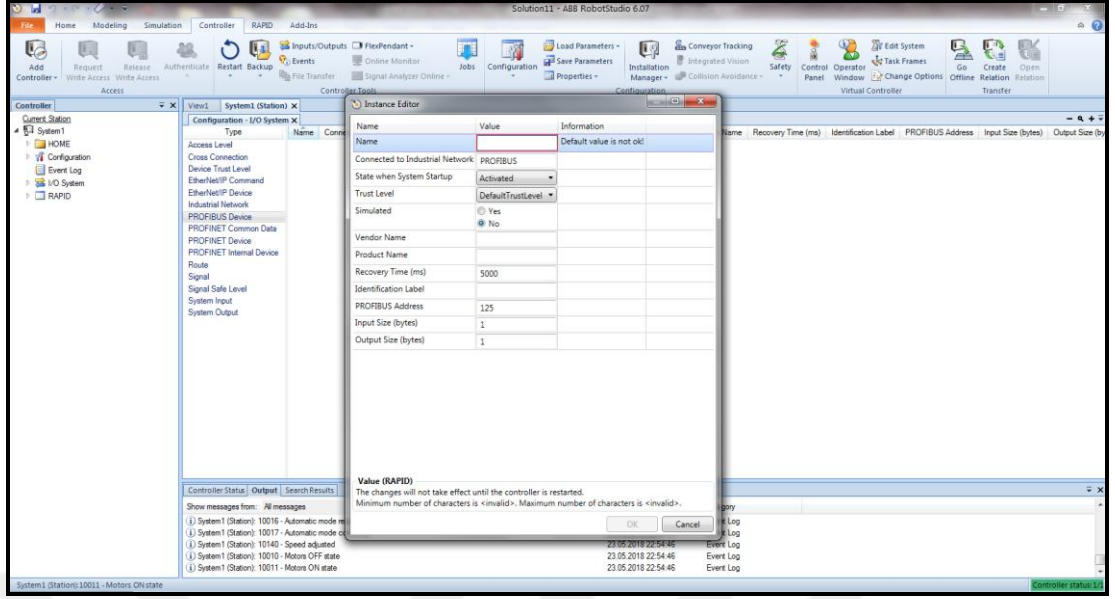


Şekil 2.59. SmartComponentin tut – bırak fonksiyonunun denemesi

2.2.7.5. Robot sistemi içinde gerekli I/O sinyallerinin tanımlanması

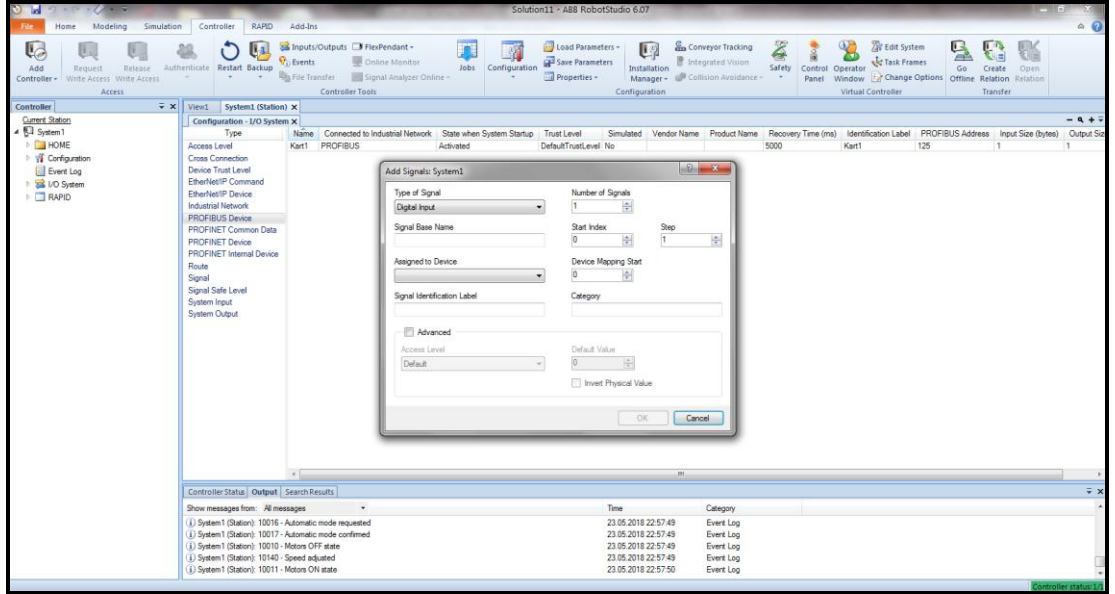
Robot sistemi içine I/O sinyalleri eklemek için "Controller" sekmesi altında "Configuration" başlığı altında "I/O System" seçimi yapılmış ve "System1 (Station)" ekranı açılmıştır. Burada "PROFIBUS Device" üzerine sağ tıklanıp "New PROFIBUS Device" seçimi yapıldığında, Şekil 2.60.'ta görülen pencere açılmıştır.

Bu pencerede "Name" ve "Identification Label" kutucuklarına "Kart1" ismi yazılmış ve OK tıklanarak pencere kapatılmıştır. Ardından bir uyarı penceresi açılmış ve sistem restart edilmeden değişikliklerin geçerli olmayacağı uyarısı yapılmıştır. Bu sebeple değişikliklerin geçerli olabilmesi için "Controller" sekmesi altında "Restart" butonuna tıklanarak sistem restart edilmiştir.



Şekil 2.60. PROFIBUS Device tanımlanması

Bundan sonra I/O sinyallerini eklemek için “Configuration” başlığı altında “Add Signals” seçimi yapılırca, Şekil 2.61.’deki pencere açılmıştır.



Şekil 2.61. PROFIBUS Device üzerine I/O sinyalleri tanımlanması

Bu pencerede “Type of Signal” kısmında “Digital Output” seçimi yapılmıştır. “Signal Base Name” kısmında sinyalin adı “PB_GrpAlBirik” olarak girilmiştir. “Start Index” kısmında 1 değeri girilmiş ve “Assigned to Device” kısmında, yeni tanımlanmış olan “Kart1” seçilmiştir.

Ardından gripperın al bırak durumunu, yani SmartComponent'in çıkış bilgisini robot sisteminde bir giriş olarak okuyabilmek için bir Input eklenmiştir. Bunun için "Type of Signal" kısmında "Digital Input" seçimi yapılmıştır. "Signal Base Name" kısmında sinyalin adı "PB_SmartComponent" olarak girilmiştir. "Start Index" kısmında 1 değeri girilmiş ve yeni bir Input daha girileceği için, "Device Mapping Start" kısmında da 1 değeri girilmiştir. Bu değer, sinyalin kart üzerindeki sırasını göstermektedir. Ardından "Assigned to Device" kısmında, yeni tanımlanmış olan "Kart1" seçilmiş ve OK tıklanarak pencere kapatılmıştır.

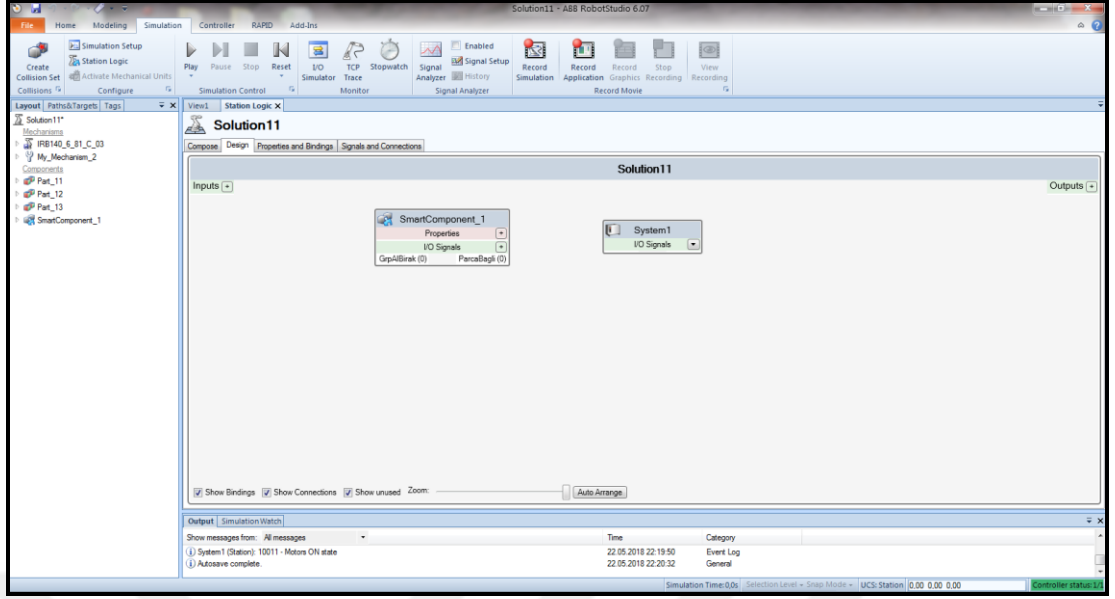
Sonrasında, çevrimi başlatmak için bir başlatma input sinyali tanımlanmıştır. Burada sinyalin adı "Start" olarak girilmiştir. "Start Index" kısmında 1 değeri girilmiş ve "Device Mapping Start" kısmında da 0 değeri girilmiştir. Ardından "Assigned to Device" kısmında, yeni tanımlanmış olan "Kart1" seçilmiş ve OK tıklanarak pencere kapatılmıştır. Robot sisteminde yapılan tüm bu değişikliklerin geçerli olabilmesi için, bu işlemlerin ardından tekrar sistem Restart edilmiştir.

2.2.7.6. Station Logic kısmında gerekli konfigürasyonun yapılması

SmartComponent ve sistem I/O sinyalleri oluşturulduktan sonra, istenen programı çalıştırabilmek için bu sinyallerin eşleştirilmesi gerekmektedir. Bunun için "Simulation" sekmesi altında "Station Logic" tıkladığında ilgili ekran açılmaktadır.

Bu ekranda "Design" sekmesi seçilince, Şekil 2.62.'de görülen blok diyagram açılmaktadır. "System1" blok içindeki OK tıklanarak, açılan pencerede daha önce tanımlanmış olan "PB_SmartComponent" ve "PB_GrpAlBirik" sinyalleri seçilmiştir.

Burada sistemin "PB_GrpAlBirik" çıkışı, SmartComponent'in "GrpAlBirik" girişine bağlanmıştır. SmartComponent'in "ParcaBagli" çıkışı ise, sistemin "PB_SmartComponent1" girişine bağlanmıştır.



Şekil 2.62. Station Logic ekranı

2.2.7.7. Robotun izleyeceği yolun tanımlanması

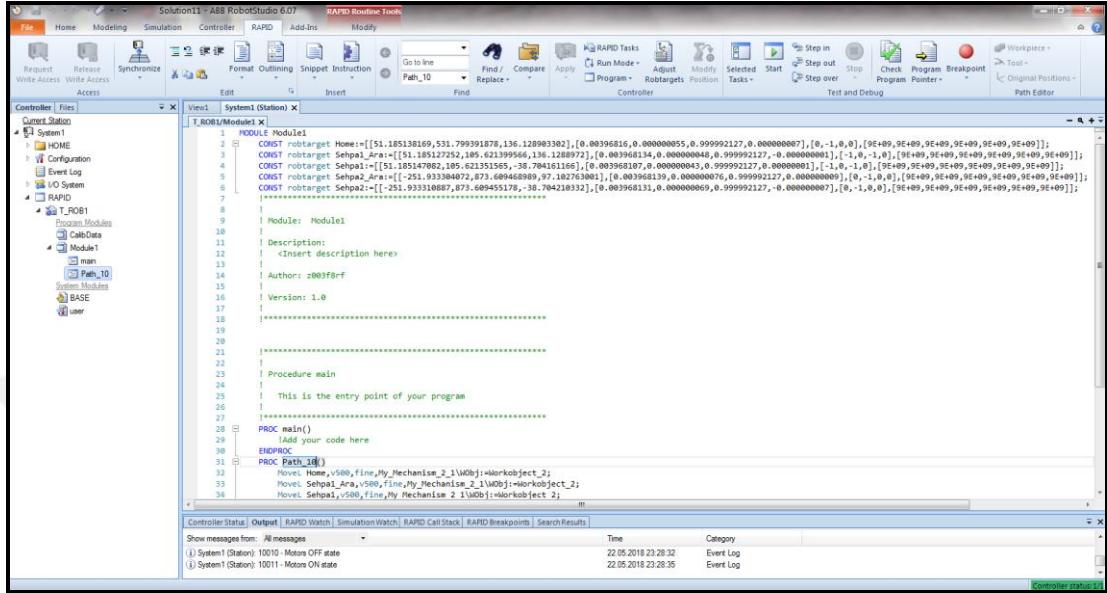
Gripper parçanın üzerindeyken “SmartComponent” üzerine sağ tıklanıp “Properties” seçimi yapılarak sinyal penceresi açılmıştır. Burada “GrpAlBirik” sinyali 1 yapılarak gripperın parçayı tutması sağlanmıştır. Ardından ilk sehpanın üzerindeki, ikinci sehpanın üzerindeki ve home pozisyonlar tanımlanmıştır.

Burada robotun yolu sorunsuz şekilde izleyebilmesi için püf nokta, sehpa üzerinde birer adet ara pozisyon tanımlanmasıdır. Ara pozisyon tanımlanmadan robotun home pozisyondan direkt sehpanın üzerine, parçayı alacak konuma gelmesi şeklinde rota belirlendiğinde, gripper iş parçasına çarpabilmektedir. Bu sebeple robot home pozisyondan sehpa gelmeden önce, parçanın üzerinde bir konumda ara pozisyon tanımlanmış, ardından parçanın üzerine dik inmesi sağlanacak şekilde rota oluşturulmuştur.

2.2.7.8. RAPID altyapısında gerekli kodların yazılması

RAPID, ABB robotun programlama dilidir. Programın bundan sonraki aşamasında, RAPID kod editörü içinde gerekli kodlar yazılmıştır.

Bunun için “RAPID” sekmesi tıklanmış, ekranın sol tarafındaki listede “RAPID” altında “T_ROB1”, altında “Module1”, altında herhangi bir rota sağ tıklanıp “RAPID Editor” seçimi yapıldığında, Şekil 2.63.’te görülen RAPID kod editörü açılmaktadır.



Şekil 2.63. RAPID kod editörü

Ana program “main” altında çalışmaktadır. Çevrimin başlayabilmesi için “Start1” giriş sinyali, bas çek buton mantığıyla, 1 ve ardından 0 vermelidir. Bunun için “WaitDI Start1,1;” ve ardından “WaitDI Start1,0;” komutları kullanılmıştır. Gripperın tutma programı için “Set PB_GrpAIBirak1;” ve bırakma programı için “Reset PB_GrpAIBirak1;” komutları yazılmıştır. Gripper açıldığında ve kapandığında 1 saniye beklemesi için set ve reset komutları ardından “WaitTime 1;” komutu kullanılmıştır. Ardından tutma ve bırakma programları, Path_10 altında gerekli yerlere eklenmiştir. Bu aşamada oluşan programın ekran görüntüsü, Şekil 2.64.’teki gibidir.

Bundan sonra, “RAPID” sekmesi altında “Apply” seçimi tıklanmış ve yazılan program doğrulanmıştır. Ardından RAPID kod editöründeki programın istasyona aktarılması için “RAPID” sekmesi altında, “Synchronize” başlığı altında, “Synchronize to Station” seçimi yapılmış, açılan pencerede tüm kutucuklar seçilmiş ve işlem tamamlanmıştır.

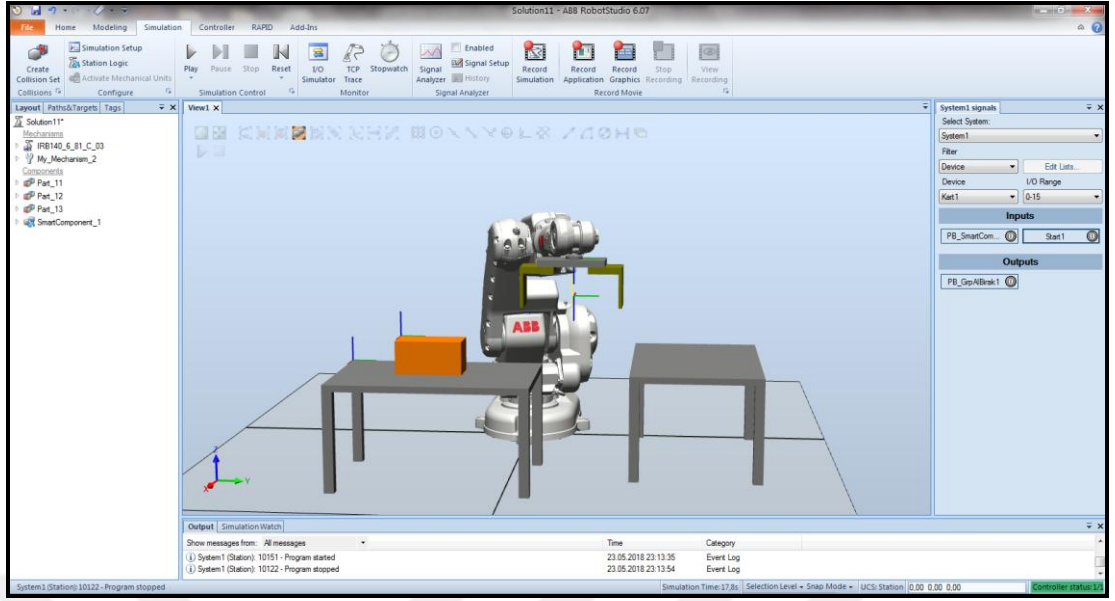
```
View1 System1 (Station) X
T_ROB1/Module1* X
27 !*****
28 PROC main()
29     WaitDI Start1,1;
30     WaitDI Start1,0;
31     Path_10;
32 ENDPROC
33 PROC tut()
34     Set PB_GrpAlBirak1;
35     WaitTime 1;
36 ENDPROC
37 PROC birak()
38     Reset PB_GrpAlBirak1;
39     WaitTime 1;
40 ENDPROC
41 PROC Path_10()
42     MoveL Home,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
43     MoveL Sehpa1_Ara,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
44     MoveL Sehpa1,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
45     tut;
46     MoveL Sehpa2_Ara,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
47     MoveL Sehpa2,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
48     birak;
49     MoveL Sehpa2_Ara,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
50     MoveL Home,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
51     MoveL Sehpa2_Ara,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
52     MoveL Sehpa2,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
53     tut;
54     MoveL Sehpa1_Ara,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
55     MoveL Sehpa1,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
56     birak;
57     MoveL Sehpa1_Ara,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
58     MoveL Home,v500,fine,My_Mechanism_2_1\WObj:=Workobject_2;
59 ENDPROC
60 ENDMODULE
```

Şekil 2.64. RAPID kod editöründe programın yazılması

2.2.7.9. Simulasyon yapılması

“Simulation” sekmesi altında, “I/O Simulator” seçimi yapılmış ve ekranın sağ tarafında, Şekil 2.65.’te görülen sinyal simülör ekranı açılmıştır. Burada “Device” kısmında “Kart1” seçimi yapıldığında, sistem üzerinde sonradan tanımlanmış olan I/O sinyallerinin filtrelendiği gözlenmiştir.

Burada “Simulation” sekmesi altında “Play” tuşuna basılıp simulasyon başlatılmıştır. Ardından “main” programında belirtildiği gibi, çevrimin başlaması için, “Start1” sinyali önce 1 ve ardından 0 yapılmca programın çalıştığı görülmüştür.



Şekil 2.65. I/O Simulator ekranı

3. SANAL GERÇEKLİK UYGULAMASI

ABB RobotStudio programı, HTC Vive sanal gerçeklik sistemi ile uyumludur. Bu sayede programda hazırlanan bir simulasyon, HTC Vive sanal gerçeklik sistemi bağlanarak 3 boyutlu olarak deneyimlenebilmektedir.

Bu teknolojinin avantajları aşağıda sıralanmıştır:

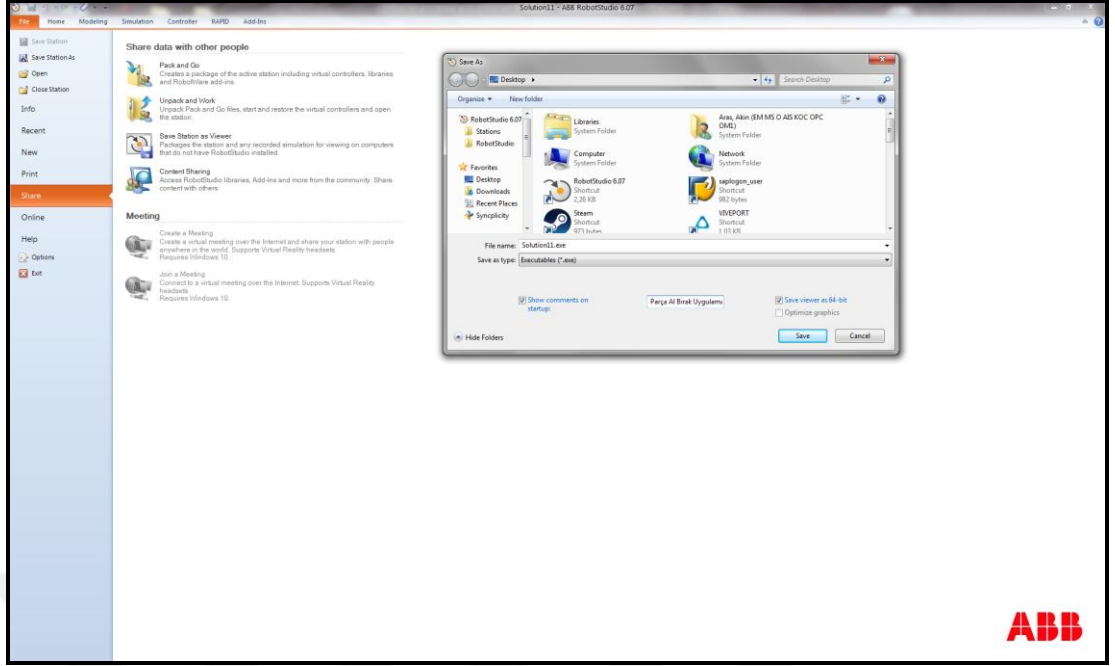
- İş güvenliği riski olmadan robot istasyonunun içi sanal ortamda gezilebilmektedir.
- Bilgisayar ekranına göre çok daha gerçekçi bir biçimde, istasyonun dizaynı gözlenmelenebilmektedir.
- İstasyon fiziksel olarak kurulmadan sanal olarak deneyimlenip, gerçekte neye benzeyeceği çok daha iyi anlaşılabilir.
- Gerçekte girilmesi mümkün olmayan yerlere girilip, istasyonun çalışmasının daha farklı açılardan gözlemlenebilmesine fırsat vermektedir.

Bu çalışma kapsamında, ABB RobotStudio programında yazılıp simulasyonu yapılmış olan parça al bırak uygulaması, sanal gerçeklik ortamında deneyimlenmiştir.

3.1. Station Viewer Çıktısının Oluşturulması

Sanal gerçeklik setiyle görüntülenebilmesi için, istasyonun bir Station Viewer çıktısının alınması gerekmektedir. Sation Viewer çıktısı aynı zamanda, RobotStudio programı kurulu olmayan bilgisiyarlarda da, robot istasyonunun ve simulasyonun görüntülenebilmesini sağlamaktadır.

Bunun için “File” sekmesi altında, “Share” kısmında, “Save Station as Viewer” seçimi yapılmalıdır. Bu seçim yapıldığında, Şekil 3.1.’de görülen pencere açılmaktadır.



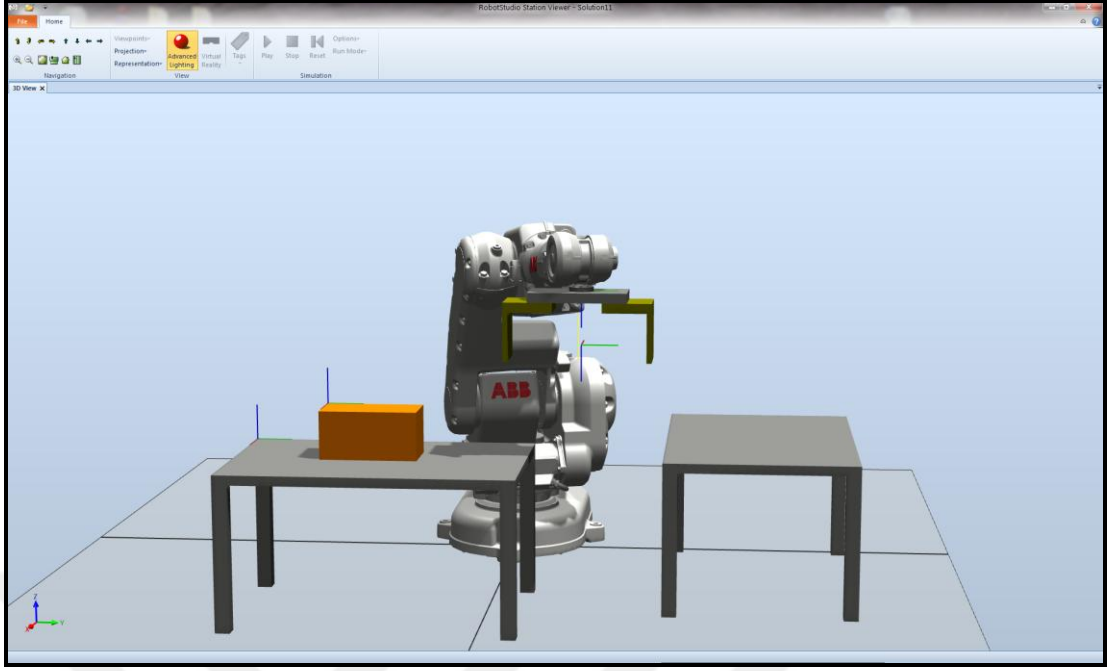
Şekil 3.1. Station Viewer çıktısının oluşturulması

Bu pencerede “Show comments on startup” kısmında, demo açılırken ekranda görüntülenmesi istenen mesaj yazılabilmektedir. Bu uygulama için “Parça al bırak uygulaması” mesajı yazılmıştır. Ardından “Save” butonu tıklandığında, StationViewer çıktısı oluşturulmuştur.

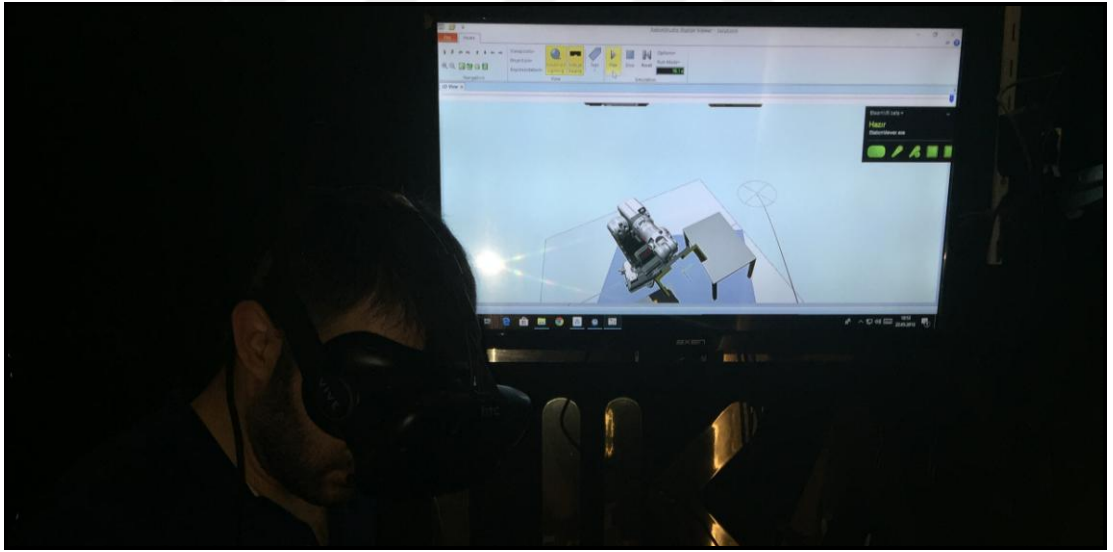
3.2. HTC Vive VR Sistem ile Sanal Gerçeklik Uygulaması

Masaüstüne kaydedilmiş olan Station Viewer çıktısının ekran görüntüsü, Şekil 3.2.’de görüldüğü gibidir.

“Home” sekmesi altında “Virtual Reality” butonu görülmektedir. Bu buton ilk anda aktif değildir. Ancak HTC Vive sanal gerçeklik sistemi bilgisayara bağlanıp çalıştırıldığında, buton aktifleşmektedir. Ardından bu butona tıklanıp gözlük takıldığında, sanal ortamda istasyonun içine girilmektedir. Şekil 3.3.’te görüldüğü gibi, kullanıcı 3 boyutlu olarak her yönden istasyonu gözlemleyebilmektedir.

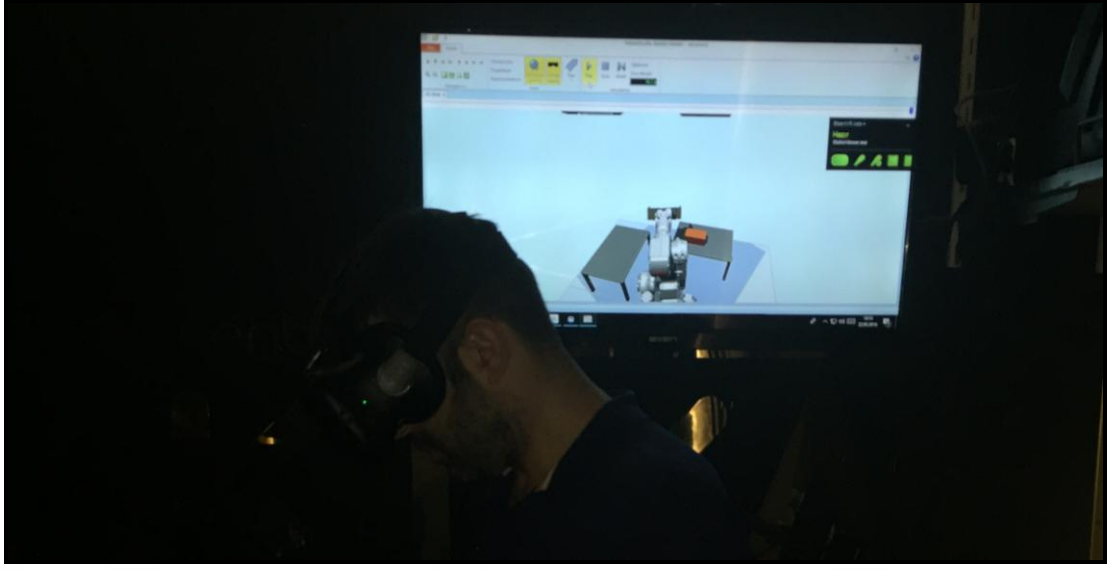


Şekil 3.2. Station Viewer ekran görüntüsü



Şekil 3.3. HTC Vive VR ile robot simülasyonunun izlenmesi – 1

Bununla birlikte, eldeki sensörler ile herhangi bir nokta işaretlendiğinde, kullanıcı sanal ortamda o noktaya ışınlanıp Şekil 3.4.'te görüldüğü gibi, istasyonu o noktadan gözlemleyebilmektedir.



Şekil 3.4. HTC Vive VR ile robot simulasyonunun izlenmesi – 2

4. SONUÇLAR VE ÖNERİLER

Robotlu üretim hatları oldukça yüksek maliyetli yatırımlardır. Bu sebeple, projelendirme aşamasından devreye alma ile prototip ve seri üretim aşamalarına kadar tüm süreçler büyük bir maliyet ve zaman baskısı altında yürümektedir. Mümkün olduğunca hızlı bir şekilde projenin tamamlanması ve istenen kalite seviyesinde çıktı alınması beklenmektedir. Bu bağlamda, kısa sürede ve doğruluğu yüksek bir çevrimdışı programlama yapabilmenin önemi ortaya çıkmaktadır.

Bu çalışmada, çevrimdışı programlama süreçlerini iyileştirebilmek için, Endüstri 4.0 konseptiyle öne çıkan teknolojilerden sanal gerçeklikten faydalanılması tercih edilmiştir. Bunun için önce, otomotiv sektöründe yoğun kullanım alanı olan iki farklı robot uygulaması, ABB RobotStudio yazılımında programlanmış ve simülasyonları yapılmıştır. Ardından bu simülasyonlar, HTC Vive sanal gerçeklik sistemiyle deneyimlenmiş ve programların istendiği şekilde çalıştığı doğrulanmıştır.

Sanal gerçeklik teknolojisi kullanımının en büyük avantajı, dizayn edilen sistemin içine adeta gerçek dünyadaymış gibi girilip, isyasyonun her yönden ve her mesafeden istendiği gibi incelenebilmesi olduğu görülmüştür. Bunu yaparken ise iş güvenliği riski ve fiziksel engeller olmaması, gerçek istasyonda fiziksel olarak yaklaşım gözlenemeyecek kısımların dahi, sistem simülasyonda çalışır haldeyken yaklaşım gözlemlenebilmesini sağlamıştır. Bu şekilde, dizayn edilen sistemin nasıl bir davranış gösterdiği çok daha anlaşılır bir şekilde gözlemlenmiştir. Bunun, çevrimdışı programlamanın süresi ve kalitesine olan olumlu etkisi doğrulanmıştır.

Çevrimdışı robot programlama süreçlerini iyileştirmek ve bu alanda çalışan personelin eğitim ihtiyaçlarına etkin bir şekilde yanıt verebilmek amacıyla, sanal gerçeklik teknolojisinin önümüzdeki süreçte daha yoğun kullanımı öngörülmektedir.

Bu çalışmada, Endüstri 4.0 konseptiyle öne çıkan teknolojilerden sanal gerçekliğin, robot programlama sürecinde kullanımı incelenmiştir. Bundan sonraki süreçte, endüstri ve akademide bu konuyla ilgili çalışmaların artması beklenmektedir. Daha

ileride ise, arttırılmış gerçeklik teknolojisinin robot programlama ve hatta robot bakım süreçlerine entegre edilmesi çalışmalarına yoğunlaşılmasının önemi vurgulanmaktadır.



KAYNAKLAR

- [1] Silva M. F., World Model Creation in ABB RobotStudio 5.15.02, *ISEP - Instituto Superior de Engenharia do Porto, Project: ROBIN - Simulation and Offline Programming of Industrial Robots*, 2016.
- [2] Silva M. F., Targets Creation in ABB RobotStudio 5.15.02, *ISEP - Instituto Superior de Engenharia do Porto, Project: ROBIN - Simulation and Offline Programming of Industrial Robots*, 2016.
- [3] Silva M. F., Program Creation in ABB RobotStudio 5.15.02, *ISEP - Instituto Superior de Engenharia do Porto, Project: ROBIN - Simulation and Offline Programming of Industrial Robots*, 2016.
- [4] Silva M. F., Simulation of the Robot Program in the Virtual Controller, *ISEP - Instituto Superior de Engenharia do Porto, Project: ROBIN - Simulation and Offline Programming of Industrial Robots*, 2016.
- [5] Silva M. F., Mechanism Creation in ABB RobotStudio 5.15.02, *ISEP - Instituto Superior de Engenharia do Porto, Project: ROBIN - Simulation and Offline Programming of Industrial Robots*, 2016.
- [6] Cohal V., A Welding Technology Using RobotStudio, *21st Innovative Manufacturing Engineering & Energy International Conference – IManE&E 2017*, Iasi, Romania, 25-26 May 2017.
- [7] Moura F. M., Silva M., Application for Automatic Programming of Palletizing Robots, *18th IEEE International Conference on Autonomous Robot Systems and Competitions*, Torres Vedras, Portugal, 25-27 April 2018.
- [8] Holubek R., Košťál P., Sobrino D. R. D., Roman R., Offline Programming of an ABB Robot Using Imported CAD Models in the RobotStudio Software Environment, *Applied Mechanics and Materials*, 2014, **693**, 62-67.
- [9] Şekerci C., Sanal Gerçeklik Kavramının Tarihçesi, *Uluslararası Sosyal Araştırmalar Dergisi*, DOI: 10.17719/jisr.20175434681.
- [10] Bayraktar E., Kaleli F., Sanal Gerçeklik ve Uygulama Alanları, *Akademik Bilişim Konferansı*, Kütahya, Türkiye, 31 Ocak – 02 Şubat 2007.
- [11] Tepe T., Kaleci D., Tuzun H., Eğitim Teknolojilerinde Yeni Eğilimler: Sanal Gerçeklik Uygulamaları, *10th International Computer and Instructional Technologies Symposium*, Rize, Türkiye, 16-18 Mayıs 2016.

- [12] Erbaş Ç., Demirer V., Eğitimde Sanal ve Artırılmış Gerçeklik Uygulamaları, Editörler: Akkoyunlu B., İşman A., Odabaşı H. F., *Eğitim Teknolojileri Okumaları 2015*, www.tojet.net, 131-148, 2015.
- [13] Gökoğlu S., Öztürk M., Erdoğan F., Çakıroğlu Ü., Öğrenme Ortamlarında Sanal Dönüşümler: Sanal Gerçeklik Yaklaşımı, Editörler: Odabaşı H. F., Akkoyunlu B., İşman A. , *Eğitim Teknolojileri Okumaları 2017*, www.tojet.net, 423-436, 2017.
- [14] Ekici R., Güven A., Sanal Gerçeklik Teknolojisinin Turizm Endüstrisindeki Rolü, *Avrasya Bilimler Akademisi Sosyal Bilimler Dergisi*, DOI: 10.17740/eas.soc.2017.UTKM.
- [15] Üzümcü E., Akın B., Nergiz H., İnözü M., Çelikcan U., Anksiyete Bozukluklarında Sanal Gerçeklik, *Psikiyatride Güncel Yaklaşımlar*, DOI: 10.18863/pgy.336593.
- [16] Quinn G. C., Galeazzi A., Gengnagel C., Augmented and Virtual Reality Structures, *Proceedings of the IASS Annual Symposium 20 17: "Interfaces: architecture . engineering . science"*, Hamburg, Germany, 25-28 September 2017.
- [17] Yengin D., Virtual Reality as Technology, *Communication and Technology Congress – CTC 2017*, DOI: 10.7456/ctc_2017_05.
- [18] <http://www.whoinventedfirst.com/invented-steam-engine/>, (Ziyaret tarihi: 10 Şubat 2018).
- [19] <http://www.bilgibaba.org/yazi/elektrik-motoru-nedir-nasil-calisir>, (Ziyaret tarihi: 10 Şubat 2018).
- [20] https://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/pages/default_alt.aspx, (Ziyaret tarihi: 10 Şubat 2018).
- [21] Plattform Industrie 4.0, Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0 - Final Report of the Industrie 4.0 Working Group, *Acatech*, 2013.
- [22] <https://www.semiwiki.com/forum/content/6341-industry-4-0-manufacturing-processes.html>, (Ziyaret tarihi: 15 Şubat 2018).
- [23] Ninan S., Gangula B., von Alten M., Snidermann B., Who Owns the Road? The IoT-Connected Car of Today-and Tomorrow, *Deloitte University Press*, 2015.
- [24] <https://3dprintingindustry.com/news/sliced-3d-printing-digest-concept-laser-boeing-bournemouth-university-feetz-stratasys-bell-helicopter-108181/>, (Ziyaret tarihi: 18 Şubat 2018).
- [25] <https://fedotov.co/top-5-3d-printing-resources/>, (Ziyaret tarihi: 18 Şubat 2018).

- [26] <https://www.tctmagazine.com/3d-printing-news/3d-systems-acquires-leading-medical-devices-company/>, (Ziyaret tarihi: 18 Şubat 2018).
- [27] <https://www.indianweb2.com/2017/04/10/meet-cobots-collaborative-robots/>, (Ziyaret tarihi: 18 Şubat 2018).
- [28] United States Postal Service Office of Inspector General, Public Perception of Drone Delivery in the United States, *United States Postal Service*, RARC-WP-17-001, 29, 2016.
- [29] <http://geoawesomeness.com/much-money-will-amazon-save-using-drones-deliveries/>, (Ziyaret tarihi: 22 Şubat 2018).
- [30] Geissbauer R., Vedso J., Schrauf S., Industry 4.0: Building the Digital Enterprise, PWC, www.pwc.com/industry40, (Ziyaret tarihi: 25 Şubat 2018).
- [31] Zäh M. F., Wunsch G., Schnelle Inbetriebnahme von Produktionssystemen, *wt Werkstattstechnik online*, 2005, **95**, 699-704.
- [32] Schmidt P. P., Köslin F., Fay A., Potential of Dynamically Adaptable Simulation Models for Virtual Commissioning, *Simulation Notes Europe SNE*, 2015, **25**(2), 59-68.
- [33] Zäh M. F., Wunsch G., Hensel T., Lindworsky A., Nutzen der Virtuellen Inbetriebnahme: Ein Experiment - Use of Virtual Commissioning: An Experiment, *ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb*, 2006, **101**, 595-599.
- [34] Hoffmann P., Schumann R., Maksoud T., Premier G., Virtual Commissioning of Manufacturing Systems: A Review and New Approaches for Simplification, *24th European Conference on Modelling and Simulation*, Kuala Lumpur, Malaysia, 01-04 June 2010.
- [35] <https://www.plm.automation.siemens.com/global/en/products/tecnomatix/>, (Ziyaret tarihi: 27 Şubat 2018).
- [36] Guerrero L., López V., Mejía J., Virtual Commissioning with Process Simulation (Tecnomatix), *Computer-Aided Design and Applications*, 2014, **11**, 11-19.
- [37] <https://www.cardsplmsolutions.nl/en/plm-software/tecnomatix/process-simulate-23/screenshots>, (Ziyaret tarihi: 27 Şubat 2018).

KİŞİSEL YAYIN VE ESERLER

- [1] **Aras A.**, Ayaz M., Özdemir E., Abut N., Investigation on Industry 4.0 and Virtual Commissioning, *International Journal of Engineering and Technologies - IJET*, 2018, **4**(1).
- [2] **Aras A.**, Erhan K., Özdemir E., Abut N., Otomasyon Eğitiminde Animasyon Temelli Eğitim Setinin Kullanımı, *Otomasyon Dergisi*, Mayıs 2015.



ÖZGEÇMİŞ

Akın ARAS, 19.10.1987 tarihinde Kocaeli’de doğdu. İlk öğrenimini Kocaeli’de çeşitli okullarda tamamladıktan sonra, orta öğrenimini Kocaeli Anadolu Lisesi’nde tamamladı. Lisans öğrenimini Yıldız Teknik Üniversitesi Elektrik Mühendisliği bölümünde tamamladı. Ardından Kocaeli Valiliği bünyesine Elektrik Mühendisi kadrosunda atandı. Ardından özel sektör deneyimi kazanmak amacıyla kamudan ayrılıp, Ford Otosan Gölcük Fabrikası’nda Proje ve Otomasyon Mühendisi olarak çalıştı ve bu süre içinde Yeniköy Fabrikası Gövde Üretim Alanı’nın kurulumu ve devreye alınması projesinde görev aldı. 2014 yılından beri Siemens’te Proje Yöneticisi olarak çalışmaktadır.