

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTOS ACADÊMICOS DE ELETRÔNICA E MECÂNICA
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

ELLEN KEDMA PEREIRA DA COSTA

**MÉTODO DE MODELAMENTO DE UM SISTEMA FLEXÍVEL DE
MANUFATURA NO SOFTWARE RobotStudio®**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2016

ELLEN KEDMA PEREIRA DA COSTA

**MÉTODO DE MODELAMENTO DE UM SISTEMA FLEXÍVEL DE
MANUFATURA NO *SOFTWARE* RobotStudio®**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso de Curso Superior de Tecnologia em Mecatrônica Industrial, dos Departamentos Acadêmicos de Eletrônica e Mecânica da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Luis Paulo Laus.

CURITIBA
2016

TERMO DE APROVAÇÃO

ELLEN KEDMA PEREIRA DA COSTA

MÉTODO DE MODELAMENTO DE UM SISTEMA FLEXÍVEL DE MANUFATURA NO *SOFTWARE* RobotStudio®

Este trabalho de conclusão de curso foi apresentado no dia 6 de julho de 2016, como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial, outorgado pela Universidade Tecnológica Federal do Paraná. O aluno foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Milton Luiz Polli
Coordenador de Curso
Departamento Acadêmico de Mecânica

Prof. Me. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Luiz Carlos de Abreu Rodrigues
UTFPR

Prof. Esp. Joel Gonçalves Pereira
UTFPR

Prof. Dr. Luis Paulo Laus
Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

À Hadassa, corajosa guerreira moderna e mãe do lar – cuja paciência e compreensão ultrapassam os níveis comuns ao ser humano.

À minha irmã Ellen Mara, exemplo vivo de determinação e iniciativa, influência certa para aqueles que almejam alcançar um objetivo.

Ao meu pai Moaci, militar estrategista e pai generoso, sempre disposto a compartilhar sua sabedoria em prol do bem maior.

AGRADECIMENTOS

Antecipo meus pedidos de desculpa àqueles que não estão presentes entre as palavras deste agradecimento, mas deixo registrado minha gratidão pela colaboração, mesmo indireta, no processo de desenvolvimento deste trabalho.

Manifesto meu sincero respeito pelo Professor Dr. Luis Paulo Laus pela sua dedicação, paciência, e orientação neste trabalho e, através dele, extendo minha reverência a toda a comunidade da Universidade Tecnológica Federal do Paraná (UTFPR) pelo apoio incondicional.

A todos os colegas pelo convívio e troca de experiências acadêmicas, sem os quais a formação educacional não seria completa.

Agradeço aos pesquisadores e professores da banca examinadora pela atenção e contribuição dedicadas a este estudo.

Desejo também externar o meu reconhecimento à minha família, pelo carinho, amor, compreensão, e dedicação no longo processo de minha educação, sem os quais não conseguiria manter a firmeza neste propósito.

“Dizer que a tecnologia é uma ciência aplicada (...) sugere que a aplicação é posterior a aquisição de um conhecimento confiável sobre seus possíveis efeitos (...). Assim, há poucas possibilidades de se produzirem surpresas desagradáveis, já que o conhecimento científico prévio é a melhor ferramenta para controlar as consequências de uma aplicação tecnológica, pois não se trata de um processo cego de ensaio e erro e sim de uma intervenção no mundo, baseado no conhecimento teórico e do método experimental próprio da ciência moderna.”

(BAZZO, Walter e SILVEIRA, Rosemari, 2005)

RESUMO

COSTA, Ellen K.P.C. **Método de modelamento de um sistema flexível de manufatura no software RobotStudio®**. 2016. 116 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Devido à crescente competitividade na área industrial, torna-se inevitável a busca pela automação de atividades repetitivas e cansativas, de forma a otimizar os processos de manufatura industrial. Com o advento da robótica, a utilização de robôs tornou-se fator indispensável e, conseqüentemente, a obrigatoriedade de monitoramento e controle de seus comportamentos em ambiente fabril através de simulações virtuais realizadas por *softwares* de engenharia, garantindo maior confiabilidade e segurança aos sistemas. Neste trabalho, foi desenvolvido o modelamento de um Sistema Flexível de Manufatura real no *software* RobotStudio® no intuito de simular o comportamento de manipuladores robóticos integrados a um *layout* virtual desse sistema, utilizando-se da orientação de manuais e tutoriais disponibilizados pelos desenvolvedores do *software*, bem como usuários experientes com a ferramenta. A disposição virtual do *layout* foi alcançada satisfatoriamente, no entanto, a exequibilidade de movimentação automática e programação dos robôs mostraram-se comprometidas devido a discrepâncias e diversidade de informações nos manuais e tutoriais para os fins necessários. Ainda assim, a exigência meticulosa de informações de entrada, bem como a necessidade de execução das ações na plataforma do *software* em ordens específicas, indicaram ser esta uma poderosa ferramenta de alto nível para o estudo de sistemas robóticos, com o potencial de fornecer resultados precisos e detalhados para satisfazer a demanda do mercado tecnológico.

Palavras chave: Robôs. Automação. Simulação virtual. Sistemas flexíveis de manufatura.

ABSTRACT

COSTA, Ellen K.P.C. **Método de modelamento de um sistema flexível de manufatura no software RobotStudio®**. 2016. 116 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Mecatrônica Industrial), Departamentos Acadêmicos de Eletrônica e Mecânica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Due to the increasing competitiveness in the industrial scene, it comes the time for the inevitable seek of automation for repetitive and tiring activities in order to optimize industrial manufacturing processes. With the advent of robotics, the use of robots has become an indispensable factor, and therefore the obligation to monitor and control their behavior in the factories through virtual simulations powered by engineering softwares, ensuring greater reliability and security to the systems built on the factories. In this work, the modeling of a real Flexible Manufacturing System in the software RobotStudio® was developed in order to simulate the behavior of robotic manipulators integrated to a virtual layout of this system, using the guidance of the manuals and tutorials provided by the software developers, as well as users experienced with the tool. The virtual arrangement of the layout was satisfactorily achieved, however, the feasibility of automatic movement and robot programming proved to be compromised due to discrepancies and diversity of information in the manuals and tutorials for the necessary purposes. Nevertheless, the meticulous requirement of input information, as well as the need to execute actions on the software platform in specific orders, indicated that this is a powerful high-level tool for the study of robotic systems, with the potential to provide accurate and detailed solutions to satisfy the demand of the technological market.

Keywords: Robots. Automation. Virtual simulation. Flexible manufacturing systems.

LISTA DE ILUSTRAÇÕES

1	EXEMPLO DE PADRÃO DE UM AMBIENTE VIRTUAL.....	21
2	TELA DE ESCOLHA DA LINGUAGEM DURANTE INSTALAÇÃO.....	21
3	PROCESSO DE INSTALAÇÃO DO <i>SOFTWARE</i>	22
4	CONTEÚDO PROGRAMÁTICO DE UM DOS TUTORIAS DISPONIBILIZADOS PELA ABB™ NA INTERNET.....	23
5	TELA DE INÍCIO: “ <i>OPEN SOLUTION WITH EMPTY STATION</i> ”.....	24
6	PLATAFORMA DE CRIAÇÃO DE UMA NOVA SIMULAÇÃO.....	24
7	ESCOLHENDO UM ROBÔ DA BIBLIOTECA ABB™ <i>LIBRARY</i>	25
8	POSICIONAMENTO DO ROBÔ.....	25
9	INSTRUÇÃO PARA <i>ZOOMING</i> DO ROBÔ.....	25
10	INSTRUÇÃO DE ROTACIONAMENTO DO ROBÔ.....	26
11	VISTA INFERIOR TRASEIRA DO MODELO IRB 1200, APÓS O ROTACIONAMENTO.....	26
12	CRIAÇÃO E NOMEAÇÃO DE UM NOVO SISTEMA A PARTIR DE UM <i>LAYOUT</i>	27
13	INSTALAÇÃO DOS CONTROLADORES NO <i>SOFTWARE</i>	27
14	INSTALAÇÃO DE CONTROLADORES OK.....	27
15	MINIFÁBRICA DO LABORATÓRIO A-007.....	28
16	SOFTWARE DE VISÃO DE MÁQUINA.....	29
17	SISTEMA DE SUPERVISÃO E CONTROLE ATUAL.....	29
18	UMA CENA DA APRESENTAÇÃO, MOSTRANDO TRÊS ROBÔS.....	31
19	UMA CENA DA APRESENTAÇÃO, MOSTRANDO OS ROBÔS EM REBELIÃO.....	31
20	ESTRUTURA DE UM ROBÔ INDUSTRIAL.....	34
21	JUNTAS LINEARES, ROTATIVAS, E PRISMÁTICAS.....	35
22	ROBÔ CARTESIANO.....	38
23	ROBÔ CILÍNDRICO.....	38
24	ROBÔ ESFÉRICO.....	39
25	ROBÔ ARTICULADO.....	39
26	ROBÔ SCARA.....	40
27	DEFINIÇÃO DOS ÂNGULOS DE ORIENTAÇÃO RPY.....	41
28	BUSCA EM BIBLIOTECA (<i>BROWSE LIBRARY</i>).....	44
29	IMPORTAR OBJETOS DA BIBLIOTECA DO USUÁRIO (<i>IMPORT OBJECTS</i>).....	45
30	IMPORTAR EQUIPAMENTOS DA BIBLIOTECA (<i>IMPORT EQUIPMENTS</i>).....	47
31	PRINCIPAIS ELEMENTOS IMPORTADOS PARA O AMBIENTE VIRTUAL DA ABB™.....	47
32	BARRA DE FERRAMENTAS PARA MODELAMENTO DE OBJETOS 3D.....	48
33	ESCOLHENDO GERAR OBJETOS SÓLIDOS.....	49

34	OPÇÕES PARA A CRIAÇÃO DE OBJETOS SÓLIDOS 3D NO ROBOTSTUDIO®.....	49
35	<i>PALLET</i> DE MADEIRA.....	50
36	BASE SUPORTE PARA CABINE DE CLP E SISTEMA DE ARMAZENAGEM.....	50
37	ARMÁRIOS DE MADEIRA GERADOS PELA FERRAMENTA 3D	51
38	OBJETOS DIVERSOS GERADOS PELA CONSTRUÇÃO SÓLIDA CÚBICA 3D.....	51
39	ELEMENTOS BÁSICOS DO CENTRO DE TORNEAMENTO CRIADOS EM 3D.....	52
40	ELEMENTOS BÁSICOS DO CENTRO DE USINAGEM CRIADOS EM 3D.....	52
41	FORMATOS CILÍNDRICOS E CÚBICOS GERADOS NA FERRAMENTA 3D	53
42	OBJETOS DO SISTEMA DE VISÃO.....	53
43	TARUGO DE LATÃO GERADO EM 3D.....	54
44	BASE CILÍNDRICA DAS CASTANHAS DO EIXO ÁRVORE NO CENTRO DE TORNEAMENTO.....	54
45	CENTRO DE POSICIONAMENTO DE PEÇAS NAS CASTANHAS DO CENTRO DE TORNEAMENTO.....	55
46	CENTRO DE POSICIONAMENTO DE PEÇAS NA MORSA DO CENTRO DE FRESAMENTO.....	55
47	PARÂMETROS PARA CRIAR UM OBJETO 3D CÚBICO.....	56
48	PARÂMETROS PARA CRIAR UM OBJETO 3D CÔNICO.....	57
49	PARÂMETROS PARA CRIAR UM OBJETO 3D CILÍNDRICO.....	57
50	RECURSO PARA IMPORTAÇÃO DE OBJETOS 3D DESENVOLVIDOS EM CAD.....	58
51	OPÇÕES PARA A IMPORTAÇÃO DE OBJETOS 3D DESENVOLVIDOS EM CAD....	58
52	LOCALIZANDO O ARQUIVO DA PEÇA DESENVOLVIDA EM CAD.....	59
53	ESCOLHENDO O ARQUIVO 3D DO CAD SOLIDWORKS®.....	59
54	IMPORTANDO A PEÇA 3D DESENVOLVIDA EM CAD SOLIDWORKS®.....	60
55	PEÇA 3D DESENVOLVIDA EM CAD SOLIDWORKS® IMPORTADA PARA O AMBIENTE VIRTUAL DO ROBOTSTUDIO®.....	60
56	RENAMEANDO UM COMPONENTE 3D RECÉM IMPORTADO PARA O AMBIENTE VIRTUAL DO ROBOTSTUDIO®.....	61
57	COMO POSICIONAR UM OBJETO NO ROBOTSTUDIO®.....	62
58	COMO POSICIONAR UM OBJETO NO ROBOTSTUDIO® UTILIZANDO O TCP.....	62
59	VISTA PANORÂMICA DO <i>LAYOUT</i> BÁSICO DO LABORATÓRIO FINALIZADO.....	63
60	BIBLIOTECA DE MODELOS ROBÓTICOS DOS MANIPULADORES DA ABB™.....	64
61	MODELO IRB 1200 – MANIPULADOR ROBÓTICO DA ABB™.....	64
62	MODELO IRB 1200 – JUNTA 1 COM 340 GRAUS DE FLEXIBILIDADE.....	65
63	MODELO IRB 1200 – JUNTA 2 COM 230 GRAUS DE FLEXIBILIDADE.....	65
64	MODELO IRB 1200 – JUNTA 3 COM 276 GRAUS DE FLEXIBILIDADE.....	65
65	MODELO IRB 1200 – JUNTA 4.....	66
66	MODELO IRB 1200 – JUNTA 5 COM 260 GRAUS DE FLEXIBILIDADE.....	66
67	ESCOLHA DO ELEMENTO FINAL DO ROBÔ NA BIBLIOTECA DO ROBOTSTUDIO®.....	67

68	SMART GRIPPER – ELEMENTO FINAL DO ROBÔ NA BIBLIOTECA DO ROBOTSTUDIO®.....	67
69	FIXAÇÃO DA GARRA À EXTREMIDADE DO ROBÔ.....	68
70	GARRA FIXADA À EXTREMIDADE DO ROBÔ.....	68
71	MODELO IRB 1200 – JUNTA 6 COM 800 GRAUS DE FLEXIBILIDADE.....	69
72	POSICIONAMENTO DO ROBÔ SERVENTE AO CENTRO DE TORNEAMENTO.....	70
73	POSICIONAMENTO DO ROBÔ SERVENTE AO CENTRO DE USINAGEM (FRESADORA).....	70
74	VISTA DA ESTAÇÃO DE TRABALHO APÓS O POSICIONAMENTO DOS MANIPULADORES ROBÓTICOS.....	71
75	SISTEMAS DE COORDENADAS EM UM <i>WORKOBJECT</i>	74
76	PROCEDIMENTO DE CRIAÇÃO DE UM <i>WORKOBJECT</i>	75
77	RENOMEAR E ALTERAR VALORES DE <i>USER FRAME</i> (<i>FRAME</i> DO USUÁRIO)....	75
78	AQUISIÇÃO DAS COORDENADAS DO <i>USER FRAME</i> (<i>FRAME</i> DO USUÁRIO)....	76
79	AQUISIÇÃO DAS COORDENADAS DO <i>OBJECT FRAME</i> (CENTRO DA PEÇA).....	77
80	<i>FRAMES</i> DE UM <i>WORKOBJECT</i>	78
81	DISPONIBILIZAÇÃO AUTOMÁTICA DAS COORDENADAS DA POSIÇÃO DE BUSCA AO ALCANCE DO ROBÔ.....	79
82	TESTE DE ALCANCE (<i>REACHABILITY</i>).....	80
83	GERAÇÃO DE UMA TRAJETÓRIA.....	82
84	EDIÇÃO DAS PROPRIEDADES DE UMA TRAJETÓRIA CRIADA.....	83
85	CRIAÇÃO DE UM MÓDULO DE PROGRAMA.....	84
86	EDIÇÃO DO MÓDULO DE PROGRAMA.....	85
87	SINCRONIZAÇÃO COM O CONTROLE VIRTUAL.....	86
88	GERAÇÃO AUTOMÁTICA DE PROGRAMA <i>RAPID</i>	87
89	INICIALIZAR SIMULAÇÃO DE MOVIMENTAÇÃO DOS ROBÔS.....	87
90	ACIONAMENTO DOS MOTORES E UTILIZAÇÃO DO MODO AUTOMÁTICO.....	88
91	PROGRAMAÇÃO E EDIÇÃO DA SIMUL. DE MOVIMENTAÇÃO DOS ROBÔS.....	89
92	SINCRONIZAÇÃO DO PROG. GERADO COM A ESTAÇÃO DE TRABALHO.....	89
93	SALVANDO O MÓDULO DE PROGRAMA <i>RAPID</i>	90
94	INSTALAÇÃO DE UM CONTROLE VIRTUAL COM BASE NO <i>LAYOUT</i>	91
95	INSTALAÇÃO DO VC – <i>ROBOTWARE</i> E MECANISMOS.....	92
96	INICIALIZAÇÃO DO CONTROLE VIRTUAL.....	92
97	EXEMPLO DE SOLUÇÃO PARA PROBLEMAS DE ACESSIBILIDADE POR APROXIMAÇÃO.....	93
98	MONITORAMENTO DA TRAJETÓRIA DE MOVIMENTAÇÃO.....	95
99	RECURSOS DE MONITORAMENTO.....	95
100	RECURSO PARA GERAÇÃO DE GRÁFICOS COM DADOS PARA RELATÓRIOS...	96
101	RECURSO DE GRAVAÇÕES DE APLICATIVOS, SIMULAÇÃO, E GRÁFICOS.....	96

102	ESCOLHA DE UM CONVEYOR E EDIÇÃO DO TCP.....	97
103	CONFIGURAÇÃO INTELIGENTE DA ESTEIRA.....	99
104	DIFERENÇAS DOS RESULTADOS NO MECANISMO CONVEYOR.....	99
105	CONFIGURAÇÃO DE UM PLANO SENSOR DE DETECÇÃO.....	100
106	PLANO SENSOR E CAIXAS TRANSPORTADAS.....	100
107	INTEGRAÇÃO COM O SERVIDOR DE UMA CÂMERA VIRTUAL.....	102
108	PROGRAMAÇÃO DE COMUNICAÇÃO DE UMA CÂMERA VIRTUAL.....	102

LISTA DE TABELAS

TABELA 1.	CLASSIFICAÇÃO DOS ROBÔS.....	37
-----------	------------------------------	----

LISTA DE ACRÔNIMOS

ABB™	<i>Asea Brown Boveri</i>
CAD	<i>Computer Aided Design</i> (Desenho Assistido por Computador)
CLP	Controlador Lógico Programável
CNC	Comando Numérico Computadorizado
DAMEC	Departamento Acadêmico de Mecânica
FMS	<i>Flexible Manufacturing System</i> (Sistemas Flexíveis de Manufatura)
LASIM	Laboratório de Automação e Sistemas Inteligentes da Manufatura
OPC	<i>Open Platform Communications</i> (Comunicações de Plataforma Aberta)
PC	<i>Personal Computer</i> (Computador Pessoal)
RS	RobotStudio®
SCADA	<i>Supervisory Control And Data Acquisition</i> (Controle de Supervisão e Aquisição de Dados)
TCP	<i>Tool Center Point</i> (Ponto Central da Ferramenta)
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1	INTRODUÇÃO	17
1.1	PROBLEMA.....	18
1.2	JUSTIFICATIVA.....	19
1.3	OBJETIVOS.....	20
1.3.1	OBJETIVO GERAL.....	20
1.3.2	OBJETIVOS ESPECÍFICOS.....	20
2	PROCEDIMENTOS METODOLÓGICOS	21
2.1	INSTALAÇÃO DO SOFTWARE ROBOTSTUDIO® DA ABB™.....	21
2.2	ESTUDO DO SOFTWARE.....	23
2.3	INICIALIZAÇÃO DE UM AMBIENTE DE SIMULAÇÃO	23
2.4	DESENVOLVIMENTO DE UM MODELO DE SIMULAÇÃO	24
2.5	DESENVOLVIMENTO DE UM AMBIENTE VIRTUAL.....	28
2.6	ESTRUTURA DO TRABALHO.....	30
3	FUNDAMENTAÇÃO TEÓRICA	31
3.1	INTRODUÇÃO À ROBÓTICA INDUSTRIAL.....	31
3.2	HISTÓRICO.....	31
3.3	DESENVOLVIMENTO DOS ROBÔS INDUSTRIAIS	32
3.4	VANTAGENS DA ROBÓTICA INDUSTRIAL	32
3.5	CONCEITOS BÁSICOS DE UM ROBÔ	33
3.5.1	BRAÇO MECÂNICO.....	33
3.5.2	SENSORIAMENTO E PROGRAMAÇÃO.....	34
3.5.3	JUNTAS ROBÓTICAS.....	35
3.5.4	PRECISÃO E REPETIBILIDADE.....	36
3.6	PRINCIPAIS APLICAÇÕES DOS ROBÔS INDUSTRIAIS	36
3.7	CLASSIFICAÇÃO DOS ROBÔS.....	37
3.7.1	GEOMETRIA DO ROBÔ	37
3.8	PROGRAMAÇÃO DE ROBÔS.....	41
4	DESENVOLVIMENTO	44
4.1	INICIALIZAÇÃO DA CONSTRUÇÃO DE UM SISTEMA VIRTUAL.....	44
4.1.1	LAYOUT.....	44
4.1.2	LIBRARY (“BIBLIOTECA”).....	44
4.1.3	CRIAÇÃO DE OBJETOS 3D	48
4.1.3.1	CRIAÇÃO DE OBJETOS 3D NO ROBOTSTUDIO ®.....	48

4.1.3.2	IMPORTANDO OBJETOS 3D CAD.....	58
4.1.4	POSICIONAMENTO E AJUSTES DE OBJETOS NO AMBIENTE.....	61
4.2	IMPLEMENTAÇÃO DE MANIPULADORES ROBÓTICOS.....	63
4.2.1	ESCOLHA DOS MODELOS.....	63
4.2.2	POSICIONAMENTO DOS ROBÔS.....	69
5	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS.....	72
5.1	GERAÇÃO DE MOVIMENTAÇÃO E TRAJETÓRIA.....	72
5.1.1	CRIAÇÃO DE <i>WORKOBJECTS</i>	74
5.1.2	CRIAÇÃO DE <i>TARGETS</i> (“PONTOS ALVOS”)	76
5.1.3	GERAÇÃO DE TRAJETÓRIAS.....	81
5.1.4	MOVIMENTAÇÃO DOS ROBÔS	84
5.1.5	SIMULAÇÃO.....	88
5.2	DIFICULDADES ENCONTRADAS	90
5.2.1	INSTALAÇÃO DO CONTROLADOR.....	91
5.2.2	ERROS DE ALCANCE (<i>REACHABILITY</i>).....	93
5.2.3	BUSCA POR SOLUÇÕES.....	94
5.3	ANÁLISE DO POTENCIAL DE UTILIZAÇÃO	94
5.3.1	SIMULAÇÃO VIRTUAL.....	95
5.3.2	MOVIMENTAÇÃO DE ESTEIRAS	97
5.3.3	CÂMERA DE INSPEÇÃO INTEGRADA	101
5.3.4	CONFIGURAÇÃO OPC	103
5.3.4.1	COMUNICAÇÃO CLP - ROBÔS	103
5.3.4.2	CÂMERA INTEGRADA AO FMS	104
6	CONSIDERAÇÕES FINAIS.....	105
	REFERÊNCIAS.....	107
	ANEXO A – FIGURAS DETALHANDO O PROCEDIMENTO DE INSTALAÇÃO DO SOFTWARE LICENCIADO.....	110
	ANEXO B – FIGURAS REPRESENTANDO TODAS AS MOVIMENTAÇÕES DE JUNTAS DE UM ROBÔ NO AMBIENTE VIRTUAL DO SOFTWARE ROBOTSTUDIO®, UTILIZANDO AS FERRAMENTAS DE MANIPULAÇÃO DE JUNTAS “JOGS”.....	114
	ANEXO C – AVISO DE ISENÇÃO DE RESPONSABILIDADES POR MUDANÇAS DE INFORMAÇÕES, SEM AVISO PRÉVIO, NOS MANUAIS DA ABB.....	116

1 INTRODUÇÃO

Um Sistema Flexível de Manufatura – cuja definição específica é controversa; não sendo a sua discussão o principal objetivo do presente trabalho – trata-se genericamente de um sistema de manufatura que possui certa flexibilidade para reagir a mudanças esperadas ou inesperadas em um dado processo de fabricação. A principal vantagem de um sistema FMS (*Flexible Manufacturing System*) é sua considerável flexibilidade em administrar recursos essenciais para a eficiência da manufatura, tais como tempo e esforço para a manufatura de novos produtos. A melhor aplicação de FMS dá-se na produção de pequenos a médios lotes de produtos. Um Sistema Flexível de Manufatura, para ser considerado como tal, é composto de um controle central – normalmente exercido por um controlador lógico programável (CLP); máquinas controladas numericamente por computador (CNC); sistema de inspeção; sistema de transporte; bem como sistema de armazenamento e estocagem dos materiais. (ROSÁRIO, 2005).

A partir de meados dos anos 70, a introdução de manipuladores robóticos tornou-se largamente relevante na indústria para que um FMS tenha maior eficácia no seu desempenho, reduzindo-se a necessidade de intervenção humana entre o sistema de transporte e as máquinas de produção para posicionamento dos materiais a serem trabalhados, adquirindo-se, portanto, maior **agilidade** e **precisão** nos processos. Tendo em vista que estes últimos fatores são de determinante importância para o desempenho industrial, torna-se extremamente necessária a melhoria constante na execução das tarefas realizadas pelos manipuladores robóticos, para que os processos de fabricação encontrem-se em constante estado de otimização. O estudo da otimização do desempenho dos robôs é realizado através de *softwares* de engenharia específicos, tais como o **RobotStudio®** - desenvolvido pela ABB™ para simular e analisar o comportamento dos seus modelos de robôs catalogados em quaisquer ambientes virtuais de indústria.

O principal objetivo deste trabalho foi o de realizar-se, virtualmente, o estudo do possível desempenho de manipuladores robóticos da ABB™ em uma plataforma que simulasse, da forma mais aproximada possível, o ambiente da Minifábrica FMS no Laboratório de Automação e Sistemas Inteligentes de Manufatura (A-007) do Departamento Acadêmico de Mecânica da Universidade Tecnológica Federal do Paraná – localizada no presente momento na sede Central do Campus Curitiba –

através de modelamento da interação entre os robôs da ABB™ e a minifábrica no *software RobotStudio®*.

1.1 PROBLEMA

A minifábrica do laboratório A-007 é equipada com dois (2) robôs manipuladores da marca **Mitsubishi** do modelo **RV-M1** compondo o sistema flexível de manufatura para fins didáticos: um robô manipulador na estação de torneamento, e um segundo robô manipulador junto à estação de fresamento de peças; fazendo-se assim necessária a integração dos mesmos com o restante do sistema para que tenham efetiva participação no processo de fabricação. Atualmente, a programação dos robôs se dá em duas fases: na primeira fase realiza-se a **programação off-line** dos robôs através do *teaching box*, uma interface de programação para aprendizado (gravação) de pontos (posição/orientação); na segunda fase realiza-se então a **comunicação** dos mesmos com o controle central CLP através do **Hyper Terminal Applet** (v. 5.1 Copyright © 2007 Microsoft Corporation). A integração dos robôs no processo produtivo se dá pela troca de sinais usando uma interface digital 0 - 24V similar e compatível com as encontradas em CLP.

Desejou-se verificar como se comportariam os robôs caso fossem utilizados modelos do catálogo da ABB™, uma vez que o *software* de modelamento e simulação RobotStudio® encontra-se atualmente sob posse licenciada desta instituição. Nesta verificação buscava-se, tanto de forma empírica como através de metodologia, analisar certas particularidades, como: a) a forma necessária de integração dos novos modelos com o restante do sistema, tal como foi elaborado; b) quais medidas seriam necessárias para que tais modelos pudessem ser implantados no sistema; ou c) caso suspeitar-se, através de simulação virtual apenas, a não viabilidade de implementação dos mesmos no sistema, analisar e explicar os motivos averiguados.

Houve, portanto, o interesse em verificar-se a compatibilidade entre os modelos de robôs e um sistema de fabricação, cuja configuração e *layout* de montagem fosse semelhante ao existente no Sistema Flexível de Manufatura disponível no laboratório A-007 para fins didáticos e de fabricação.

1.2 JUSTIFICATIVA

A programação *off-line* é uma das melhores formas de maximizar o retorno de investimentos feitos em sistemas robotizados. Tanto a programação *off-line* quanto a simulação dos robôs em um *software* como o RobotStudio®, da ABB™, podem ser realizados de um PC de escritório em uma fábrica, sem que haja a necessidade de interromper uma dada produção. (ROBOTSTUDIO DA ABB, 2016).

Através do *software* RobotStudio® é possível adquirir ferramentas para incrementar a rentabilidade de um sistema robótico, permitindo a realização de tarefas tais como treinamento, programação e otimização sem gerar distúrbios na produção – fato que proporciona inúmeros benefícios, tais como:

- Redução de risco;
- Inicialização (*start-up*) mais rápida;
- *Changeover* com tempo mais reduzido (termo em manufatura que define o processo de mudança de uma linha de produção, ou máquina, em outra linha de produção de produtos);
- Produtividade incrementada;

O RobotStudio® é desenvolvido em um controle virtual da ABB™ - uma cópia exata do verdadeiro *software* que executa os robôs na produção. Isto permite simulações muito realistas das performances a serem realizadas pelos robôs, utilizando os reais programas e configurações de arquivo idênticas às utilizadas no chão de fábrica (ROBOTSTUDIO DA ABB™, 2016).

Tendo em vista que o objetivo didático principal da minifábrica é o de simular na prática o funcionamento, manuseio, preparação e otimização de um Sistema Flexível de Manufatura real, faz-se necessária a constante atualização de melhorias que possam ser inseridas ou estudadas na estrutura física e virtual da mesma. Os possíveis ganhos com um estudo voltado ao modelamento em *software* de manipuladores robóticos de alto desempenho na minifábrica do laboratório A-007 são o de criar novas possibilidades na geração de habilidades e conhecimento na área tecnológica envolvendo sistemas flexíveis de manufatura e sua constante utilização no mercado industrial, agregando valores aos sistemas através da diminuição dos tempos de produção, e do tempo de execução devido à maior agilidade e precisão de elementos como manipuladores robóticos de última geração.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

Desenvolver e descrever um método de modelamento que simule o desempenho de um robô da ABB™ no ambiente virtual da minifábrica FMS A-007 através do *software* RobotStudio® da ABB™.

1.3.2 OBJETIVOS ESPECÍFICOS

- 1) Estudar o *software* através dos tutoriais disponibilizados pelos autores - ABB™ ou outros autores experientes com em relação ao *software* – em formato audiovisual ou escrito;
- 2) Identificar os modelos de manipuladores robóticos disponíveis no catálogo do *software* RobotStudio® que mais atendam às necessidades das estações de torneamento e fresamento do FMS do laboratório A-007;
- 3) Desenvolver um ambiente virtual na plataforma do *software* o mais próximo possível do *layout* físico da minifábrica existente no laboratório A-007;
- 4) Integrar os modelos virtuais de manipuladores robóticos ao restante do sistema virtual da minifábrica;
- 5) Levantar dados e fatores relevantes que viabilizem a implementação dos modelos de manipuladores robóticos ao atual sistema da minifábrica;
- 6) Analisar a possibilidade da supervisão direta do processo de fabricação através do próprio *software*, uma vez que se tenha alcançado a implementação virtual dos manipuladores robóticos da ABB™ ao sistema integrado;
- 7) Gerar relatório detalhando quais medidas seriam necessárias para que tais modelos de manipuladores robóticos da ABB™ possam ser hipoteticamente implantados no sistema FMS atual do laboratório, bem como os motivos de sua não viabilidade, caso constatados.

2 PROCEDIMENTOS METODOLÓGICOS

Buscando-se alcançar o objetivo do presente estudo, abordou-se a seguinte metodologia:

2.1 INSTALAÇÃO DO SOFTWARE RobotStudio® da ABB™

Instalou-se o *software* RobotStudio® da ABB™, no laboratório de Automação e Sistemas Inteligentes de Manufatura (A-007), cujo PC comporta o Windows 10 como sistema operacional. A figura 1 mostra um exemplo visual da potencialidade de utilização do *software*.

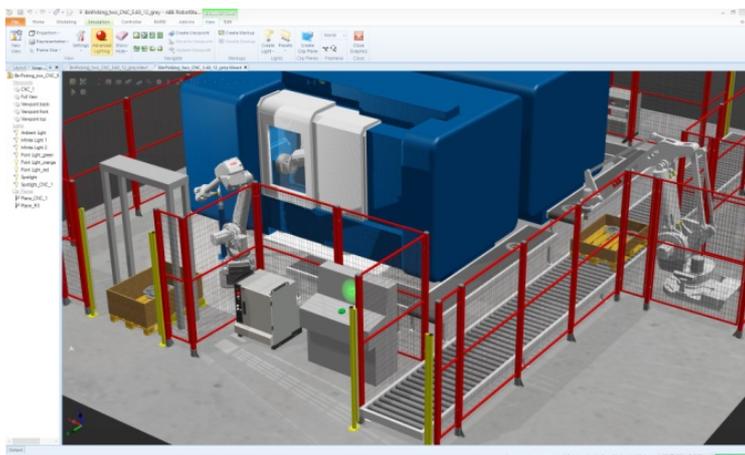


Figura 1. Exemplo padrão de um ambiente virtual desenvolvido em *software* da ABB™. Fonte: RobotStudio® da ABB™ - Site Oficial.

Durante a instalação, escolheu-se o inglês como língua padrão, considerando-se, dentre todas as disponíveis, ser esta a de maior familiaridade do pesquisador, conforme mostra a figura 2.



Figura 2. Tela de escolha da linguagem durante a instalação do *software* RobotStudio®. Fonte: O autor.

No Anexo A encontra-se detalhadamente a figura 3, mostrando o procedimento de instalação do *software* licenciado.

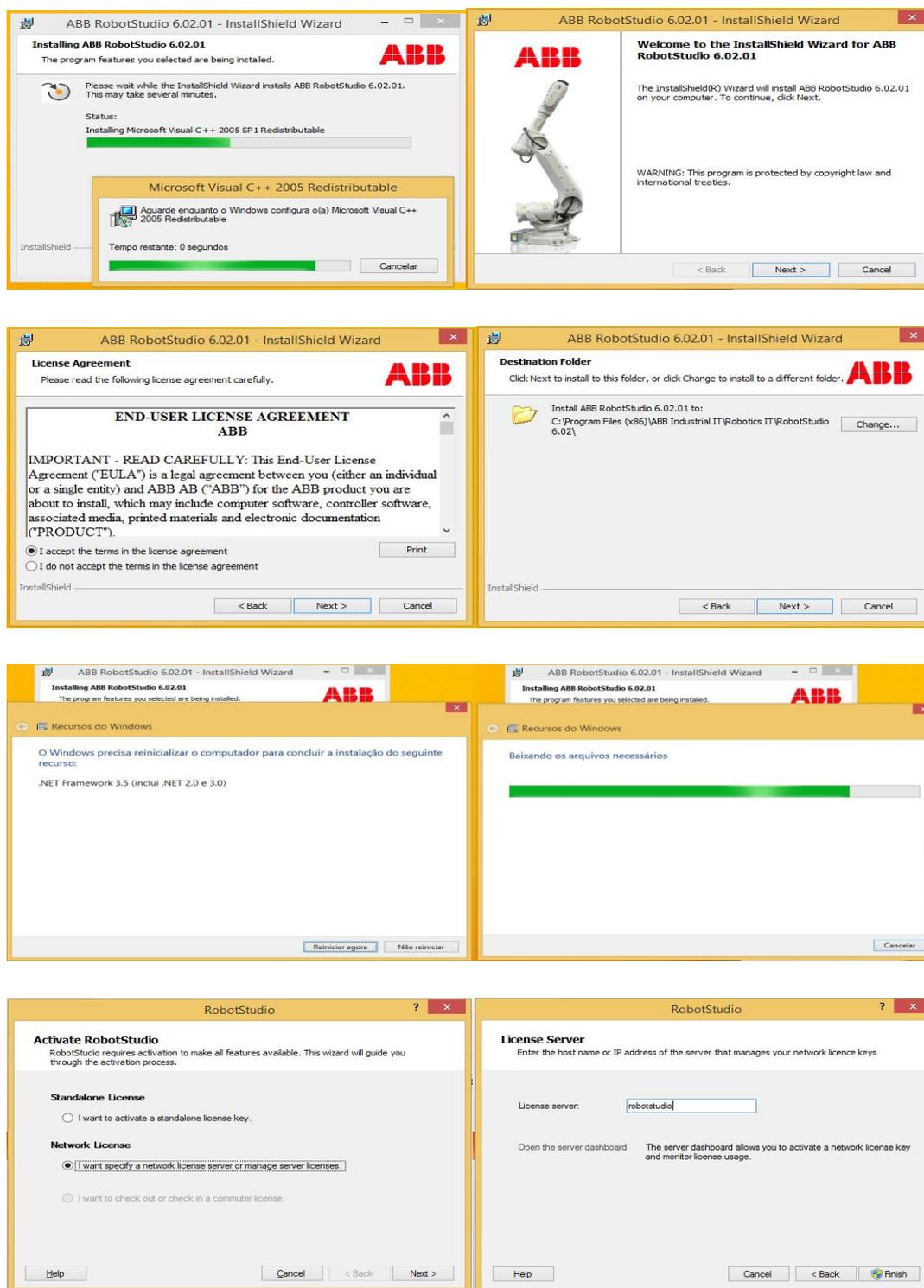
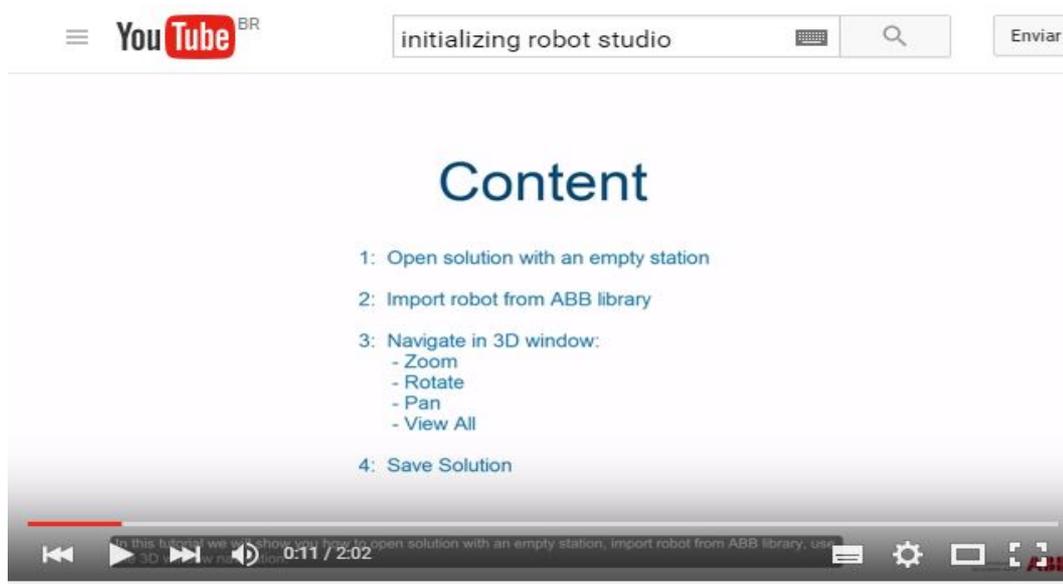


Figura 3. Processo de Instalação do *Software*.
Fonte: O autor.

2.2 ESTUDO DO SOFTWARE

Finalizado o processo de instalação, iniciou-se o processo de familiarização do pesquisador com o *software* através de tutoriais – sendo estes em formatos audiovisuais ou escritos – que estejam disponíveis pela ABB™ ou outros autores experientes, tais como os vídeos tutoriais explicativos encontrados na internet.



RobotStudio Tutorial Getting Started 1

Figura 4. Conteúdo programático de um dos tutoriais disponibilizados pela ABB™ na internet. Fonte: RobotStudio® Tutorial Getting Started 1, 2015.

Em meios de informação tais como o mostrado na figura 4, podem-se encontrar materiais de estudo através de canais desenvolvidos pela própria ABB™ com vídeos oficiais de explicações básicas de inicialização e desenvolvimento de simulações. A princípio, optou-se por este meio de pesquisa.

2.3 INICIALIZAÇÃO DE UM AMBIENTE DE SIMULAÇÃO

Tendo-se concluída a instalação do *software* com sucesso, e com o auxílio de um tutorial, deu-se início à criação de um ambiente virtual para estudo abrindo-se um novo projeto, como o mostrado na figura 5 a seguir.

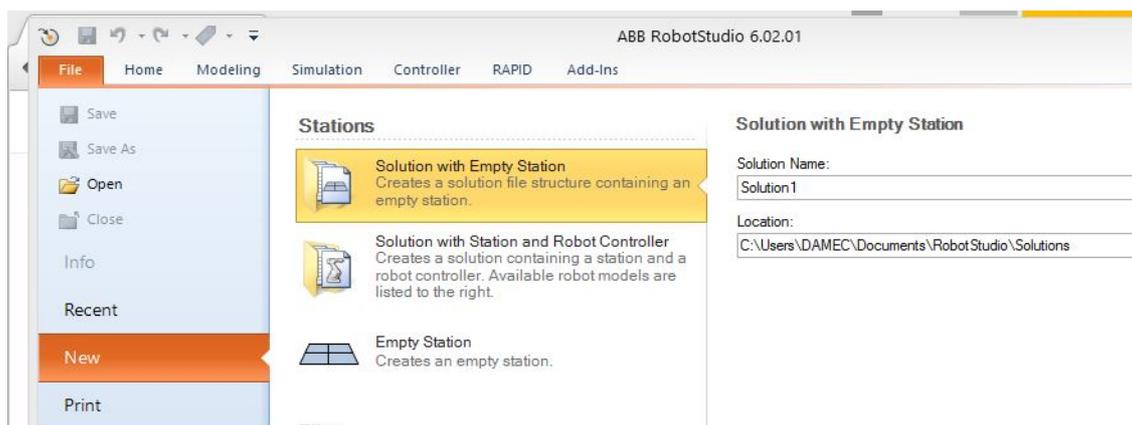


Figura 5. Tela de início: “Open Solution with Empty Station” (“Abrir Solução com Estação Vazia” em tradução livre). Fonte: O autor.

2.4 DESENVOLVIMENTO DE UM MODELO DE SIMULAÇÃO

Abrindo-se uma nova estação vazia no simulador (figura 5), pode-se escolher um modelo de manipulador robótico da ABB™ através do menu da biblioteca.

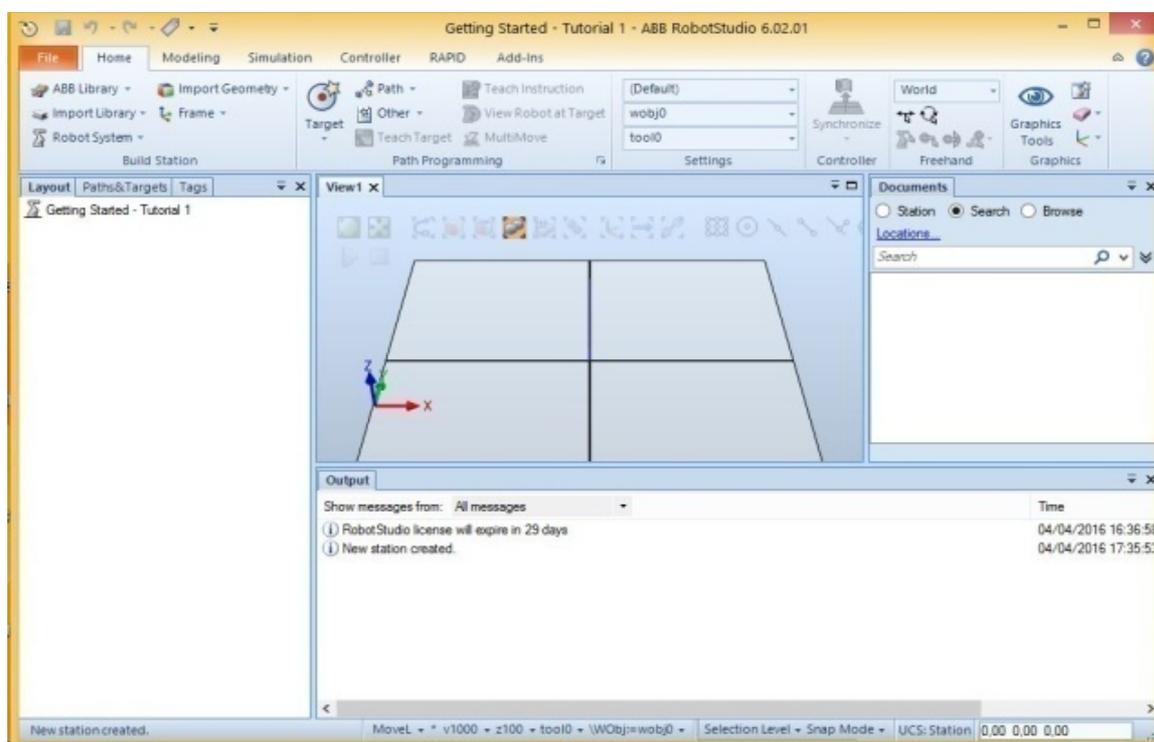


Figura 6. Plataforma de criação de uma nova simulação. Fonte: O autor.

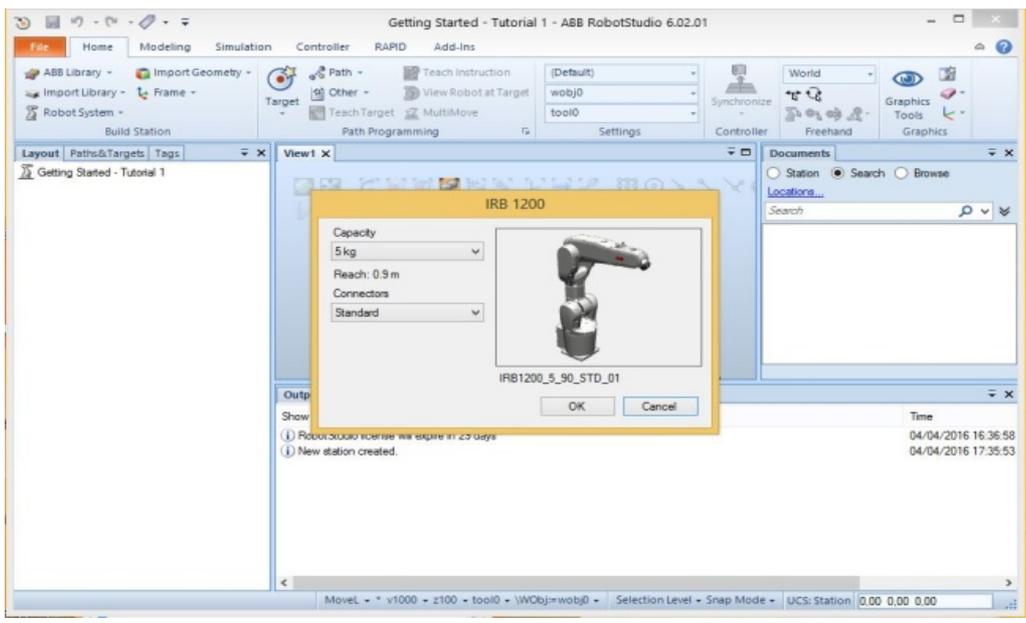


Figura 7. Escolhendo um robô da biblioteca ABB™ Library.
Fonte: O autor.

Após o posicionamento do robô na plataforma, aprendeu-se o método de manipulação do protótipo, tal como *zooming* e visualização de suas vistas.

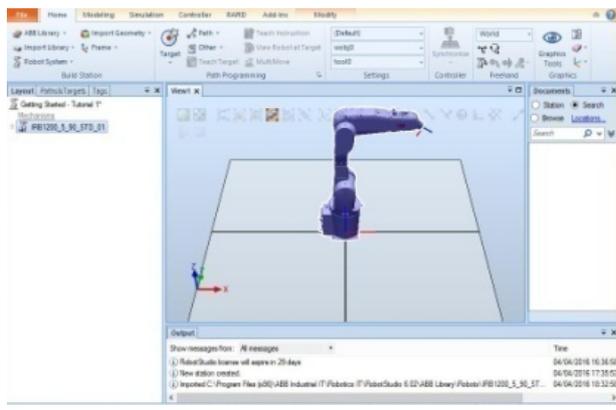


Figura 8. Posicionamento do robô. Fonte: O autor.

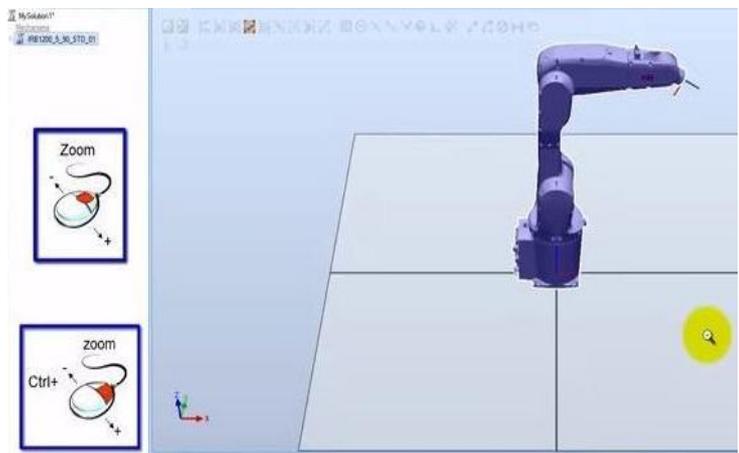


Figura 9. Instrução para *zooming* do robô. Fonte: O autor.

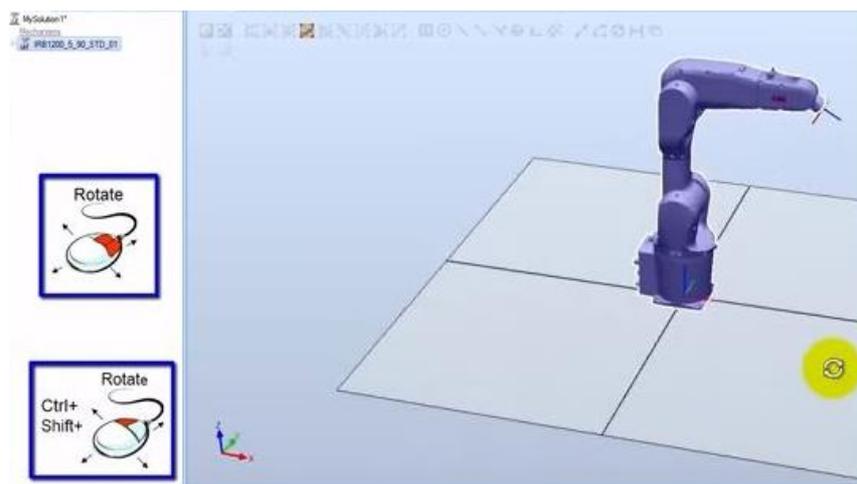


Figura 10. Instrução de rotacionamento do robô.
Fonte: O autor.

Através do rotacionamento, por exemplo, é possível o acesso visual às diversas partes de um manipulador robótico, como mostrado na figura 11.

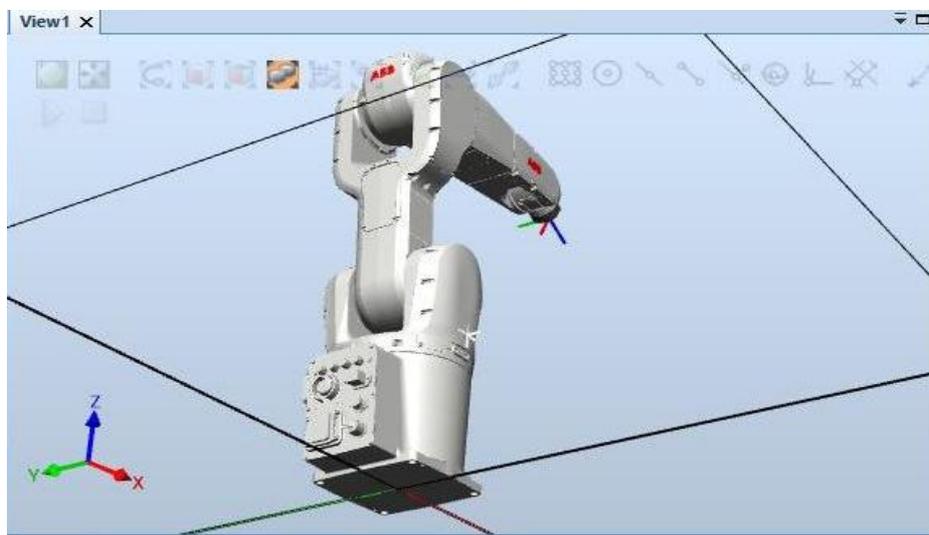


Figura 11. Vista inferior traseira do modelo IRB 1200, após o rotacionamento.
Fonte: O autor.

Tendo-se escolhido o modelo de manipulador robótico adequado ao sistema, criou-se um novo sistema (*System Name and Location*) no qual foi inserido o robô escolhido, de acordo com o procedimento mostrado na figura 12 a seguir.

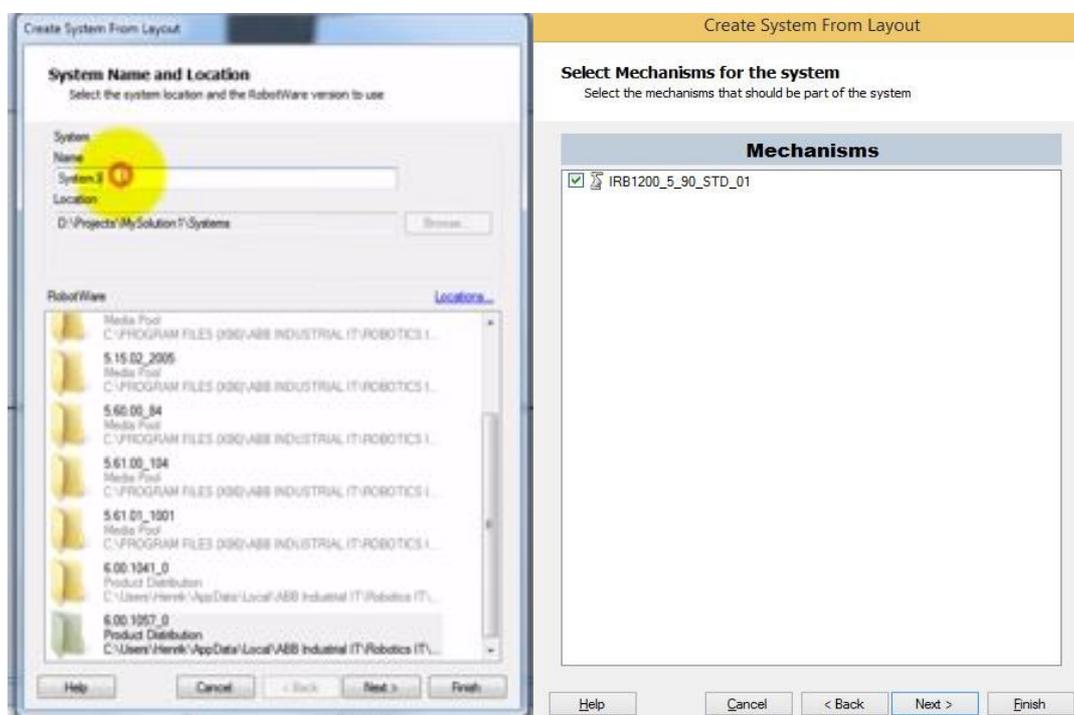


Figura 12. Criação e nomeação de um novo sistema a partir de um *layout*.
Fonte: RobotStudio® Tutorial Getting Started 2, 2015.

O sistema criado deve, portanto, ser uma réplica virtual na qual o comportamento dos robôs simule exatamente o comportamento verdadeiro. Realizou-se, também, a instalação dos controladores (figuras 13 e 14) dentro do *software*, por meio dos quais são realizadas as movimentações das juntas dos robôs, conforme mostrado nas figuras do Anexo B.

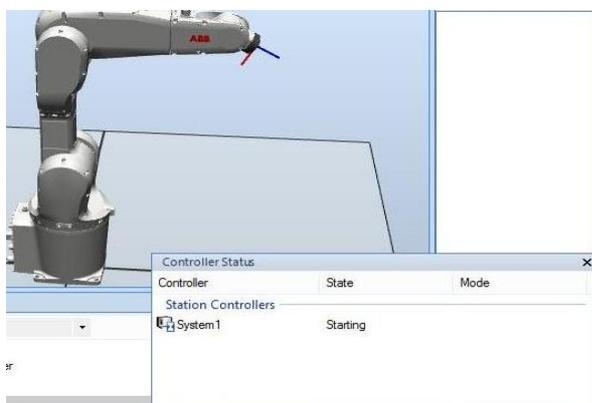


Figura 13. Instalação dos controladores no *software*.
Fonte: O autor.

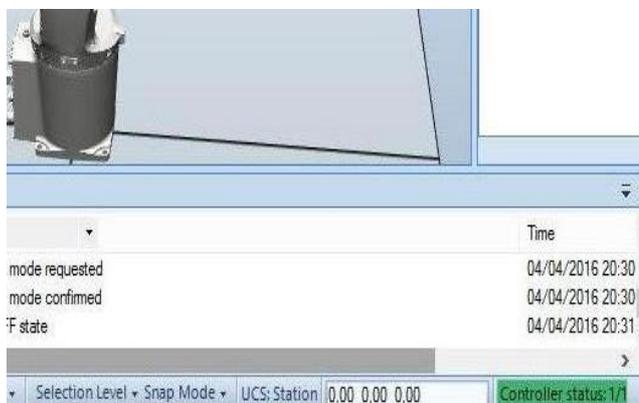


Figura 14. Instalação de controladores OK.
Fonte: O autor.

2.5 DESENVOLVIMENTO DE UM AMBIENTE VIRTUAL

O ambiente virtual que foi criado no *software* teve como base o *layout* e o maquinário do sistema flexível de manufatura disponível na minifábrica do laboratório A-007, que consiste dos seguintes elementos, conforme explica a figura 15:



(a)



(b)



(c)



(d)



(e)



(f)

Figura 15. Minifábrica do laboratório A-007: (a) armazém automático; (b) esteira de transporte; (c) centro de torneamento CNC; (d) centro de fresamento CNC; (e) manipuladores robóticos; (f) câmera de visão de máquina. Fonte: fotos adquiridas do painel físico do LASIM – DAMEC.

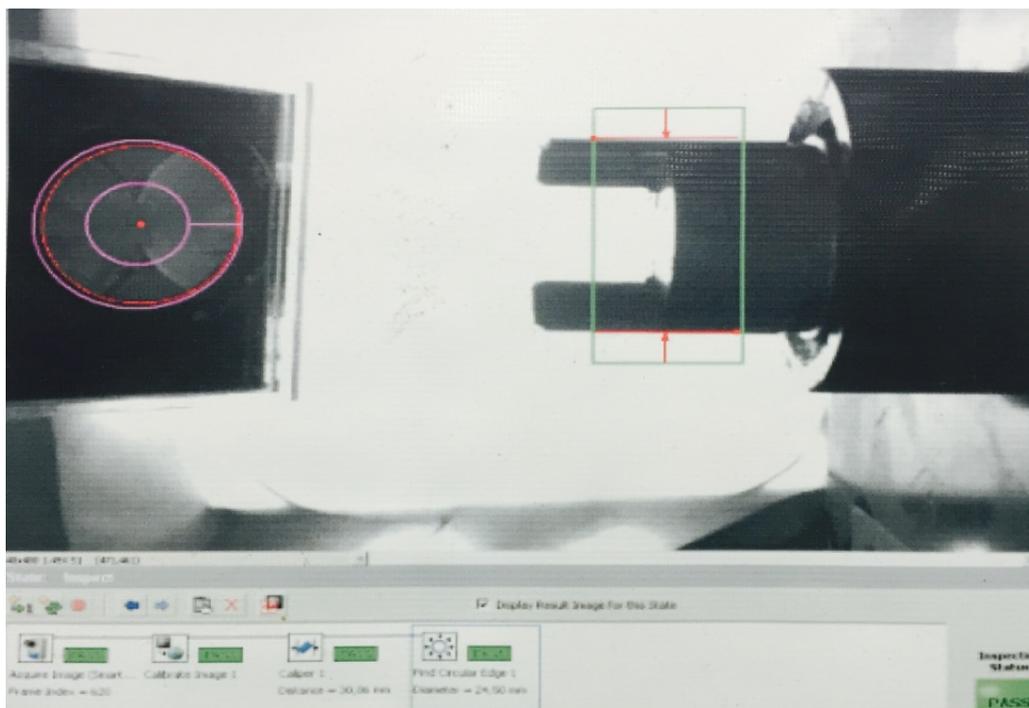


Figura 16. Software de visão de máquina.
 Fonte: foto adquirida do painel físico do LASIM – DAMEC.

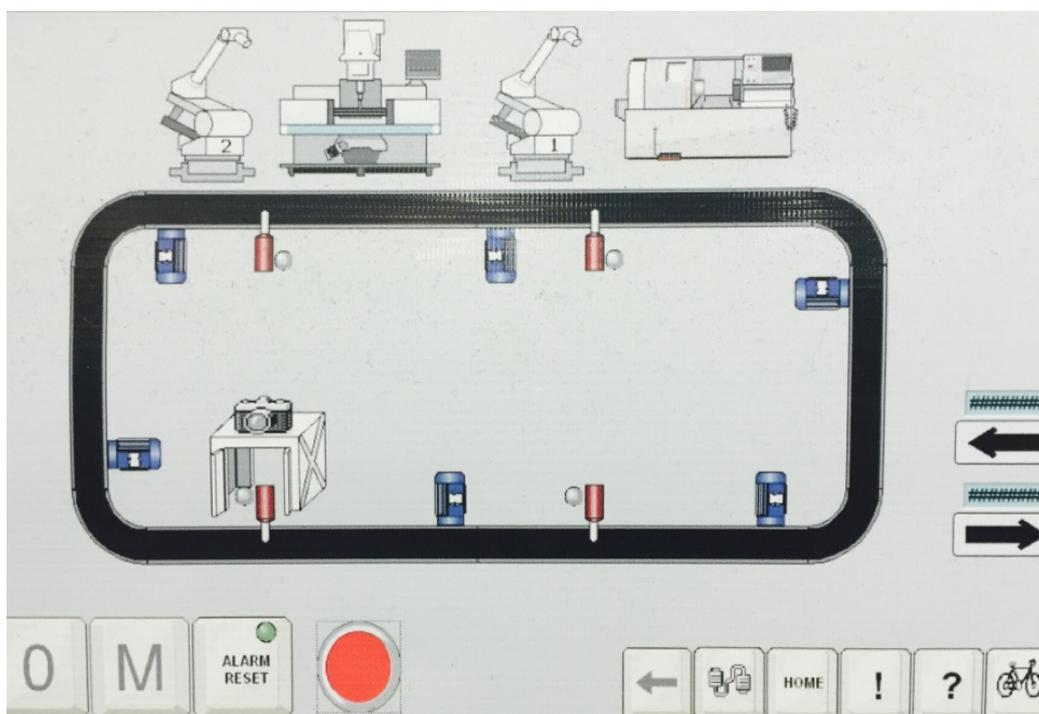


Figura 17. Sistema de supervisão e controle atual.
 Fonte: foto adquirida do painel físico do LASIM – DAMEC.

2.6 ESTRUTURA DO TRABALHO

A apresentação do trabalho terá estruturação conforme abaixo.

Capítulo 1 - Introdução: neste capítulo serão apresentados o tema, as delimitações da pesquisa, o problema, os objetivos da pesquisa, a justificativa, os procedimentos metodológicos, a fundamentação teórica, e a estrutura geral do trabalho.

Capítulo 2 – Procedimentos Metodológicos: neste capítulo será apresentada a metodologia utilizada na realização deste trabalho.

Capítulo 3 – Fundamentação Teórica: serão expostos os conceitos, estudos anteriores e conhecimentos científicos acerca do tema a ser explorado neste trabalho.

Capítulo 4 – Desenvolvimento: serão apresentadas as etapas do desenvolvimento do ambiente virtual no *software* semelhante ao *layout* físico da minifábrica existente no laboratório A-007.

Capítulo 5 – Apresentação e Análise dos Resultados: após o desenvolvimento do sistema e a realização de testes com base nos procedimentos metodológicos, serão realizadas as devidas análises e descritos os resultados obtidos.

Capítulo 6 – Considerações Finais: neste capítulo serão retomados a pergunta de pesquisa e os seus objetivos, apontando quais foram solucionados e como foram solucionados, respondidos e atingidos por meio do trabalho realizado. Serão também sugeridos trabalhos futuros que poderão ser realizados a partir deste trabalho realizados.

3 FUNDAMENTAÇÃO TEÓRICA

3.1 INTRODUÇÃO À ROBÓTICA INDUSTRIAL

Conforme o que se encontra na literatura, sobre o início da utilização dos robôs, nos primórdios dos anos 60, no objetivo de substituir o homem em atividades envolvendo condições tediosas ou desagradáveis tais como altos níveis de calor, ruídos, gases tóxicos, ou esforço físico extremo (ROSÁRIO, 2005, p. 142), esta prática tornou-se tendência, parte disso como consequência ao constante aumento dos níveis salariais de empregados e ao avanço tecnológico na área de computação, induzindo a uma redução relativa dos preços dos robôs – em comparação com os preços dos primeiros robôs a serem comercializados – e melhorando o desempenho significativamente, garantindo a evolução dos robôs ao longo dos últimos 30 anos.

3.2 HISTÓRICO

De acordo com a história, o termo ‘robô’, com o significado de “servo”, foi inicialmente utilizado em 1923 por Karel Capek, um escritor tcheco que escreveu a peça de teatro chamada R.U.R (*Rosumovi Univerzální Roboti*, ou “Os Robôs Universais de Rossum” em tradução livre do tcheco). Cenas da peça são mostradas nas figuras 18 e 19.

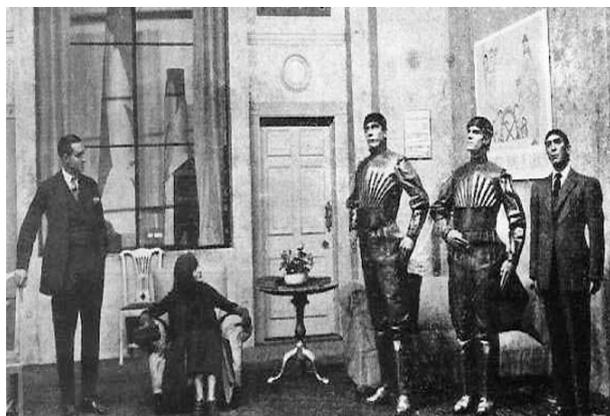


Figura 18. Uma cena da apresentação, mostrando três robôs.
Fonte: R.U.R (Wikipedia, 2016).



Figura 19. Uma cena da apresentação, mostrando os robôs em rebelião.
Fonte: R.U.R (Wikipedia, 2016).

A idéia da necessidade de construção de robôs para aumentar a produtividade industrial e melhorar a qualidade dos produtos ganhou força a partir do século XX. Desde que os sistemas de microcomputadores passaram a oferecer recursos diversos de tecnologia, a robótica apenas tende a crescer, levando a crer que em um curto espaço de tempo, o desenvolvimento de robôs cada vez mais inteligentes será uma realidade (ROSÁRIO, 2005, p. 142).

3.3 DESENVOLVIMENTO DOS ROBÔS INDUSTRIAIS

Com o lançamento dos primeiros robôs industriais, apenas algumas empresas de países desenvolvidos, como Japão e Estados Unidos, adquiriram acesso devido aos altíssimos preços. Este cenário, no entanto, modificou-se a partir de 1976 uma vez que os preços começaram a diminuir rapidamente, devido ao desenvolvimento veloz de recursos nas áreas de informática, robótica e microeletrônica. O *Eniac*, que foi o primeiro computador à válvula desenvolvido em 1950, por exemplo, teve toda a capacidade concentrada em uma pastilha de silício de menos de 0,5 cm² porém com velocidade de processamento muito superior e menor custo. Rosário (2005) cita ainda que os microprocessadores “influenciaram diretamente a capacidade de todas as máquinas industriais, com impacto decisivo nas tecnologias associadas à robótica, pois isso permitiu que a capacidade de processamento de informações se multiplicasse exponencialmente” (ROSÁRIO 2005, p. 145).

3.4 VANTAGENS DA ROBÓTICA INDUSTRIAL

Os benefícios gerados pelo crescimento da tecnologia relacionada à robótica possibilitaram grandes incrementos na produtividade do trabalho e, conseqüentemente, ao atendimento das necessidades básicas da população, pois além do aumento da produção, os equipamentos automatizados uniformizaram e melhoraram a qualidade dos produtos, evitando perdas e refugos. Possibilitou-se também a eliminação de tempos mortos, pois as máquinas poderiam trabalhar 24 horas por dia sem gerar tempo ocioso na linha de produção, gerando crescimento na rentabilidade dos investimentos; fator que ficaria comprometido caso houvesse a insistência em manter operadores humanos em determinadas funções, pois analisando logicamente, são necessárias paradas no trabalho para atendimento das

necessidades do operador como troca de turno, horário de almoço, férias, entre outros.

Posteriormente concebeu-se a possibilidade de que os sistemas mecânicos seriam controlados por operações numérico-aritméticas por máquinas CNC (comando numérico computadorizado), ou seja, máquinas de operação cuja velocidade e posicionamento seriam controlados através de computadores conectados aos seus motores.

A utilização de robôs industriais proporciona vantagens numerosas, dentre as quais: aumento da produtividade; melhoria e consistência na qualidade final de um produto; minimização de operações; menor demanda de contratação de mão-de-obra especializada (difícil de encontrar); contabilidade no processo; facilidade na programação e utilização dos robôs em operações desagradáveis e repetitivas ou em ambientes difíceis e perigosos, bem como a capacidade de trabalho por longos períodos sem interrupção (ROSÁRIO, 2005, p.145).

3.5 CONCEITOS BÁSICOS DE UM ROBÔ

Com respeito à estrutura, um robô é um sistema mecânico composto por corpos rígidos, articulados entre si, destinados a sustentar e posicionar/orientar a ferramenta terminal que, dotado de garra mecânica ou ferramenta especializada, entra em contato direto com o processo. A mobilidade do manipulador é resultado de uma série de movimentos elementares, independentes entre si, denominados *graus de liberdade*. A mobilidade de um robô depende do número de articulações que possui.

3.5.1 BRAÇO MECÂNICO

O braço do robô executa movimentos no espaço, transferindo objetos e ferramentas de um ponto para outro, instruído pelo controlador e informado sobre o ambiente por sensores. De acordo com ROSÁRIO (2005, p. 148), um manipulador robótico consiste, basicamente, de um braço mecânico motorizado programável que apresenta algumas características antropomórficas e um computador que controla seus movimentos, conforme mostrado na figura 20. O computador armazena em sua memória um programa que detalha a trajetória a ser seguida pelo braço. Quando o programa está em modo de execução, o computador envia sinais ativando motores

que acionam o braço e a carga final na extremidade dele, mantida sob controle pelo atuador.

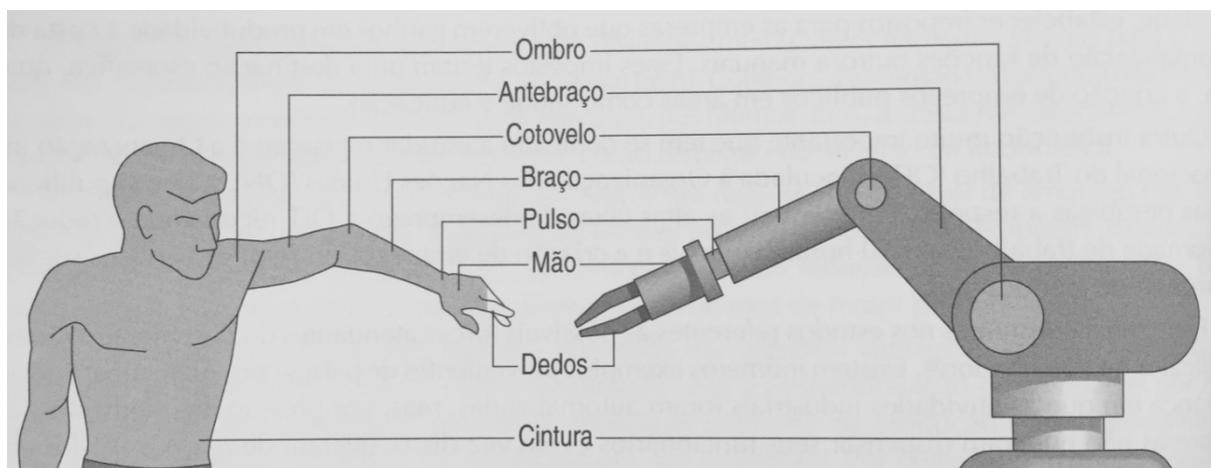


Figura 20. Estrutura de um robô industrial em uma analogia com o ser humano. (BAISE, 2016)

O sensoriamento e a programação são dois aspectos fundamentais para o funcionamento de um braço mecânico.

3.5.2 SENSORIAMENTO E PROGRAMAÇÃO

Os modelos de robôs avançados estão equipados com sensores, e seu controle é realizado por meio da programação de um computador, apresentando as seguintes características:

- Memória para armazenar os programas;
- Transmissões para os controladores dos motores;
- Conexões para a entrada e saída de dados e para ativar os programas operacionais;
- Unidade de comunicação controlada por um humano.

3.5.3 JUNTAS ROBÓTICAS

O número de juntas determina os graus de liberdade do robô; a maioria deles possui de três a seis eixos.

A maioria dos robôs industriais tem entre quatro e seis graus de liberdade, (três para o posicionamento da ferramenta terminal e três para orientação). A título comparativo, um ser humano tem sete graus de liberdade do ombro até o pulso.

A figura 21 mostra que os braços robóticos podem ser constituídos por juntas:

- a) Lineares: permitem o movimento linear entre dois vínculos. É composto de dois vínculos alinhados um dentro do outro, onde um vínculo interno escorrega pelo externo, dando origem ao movimento linear (a).
- b) Rotativas: cuja transmissão possibilita movimentos de rotação entre dois vínculos. Os dois vínculos são unidos por uma dobradiça comum, com uma parte podendo se mover em relação à outra parte (b).
- c) Prismáticas: comportam-se como uma combinação de três juntas de rotação, permitindo movimentos de rotação em torno dos três eixos (c).

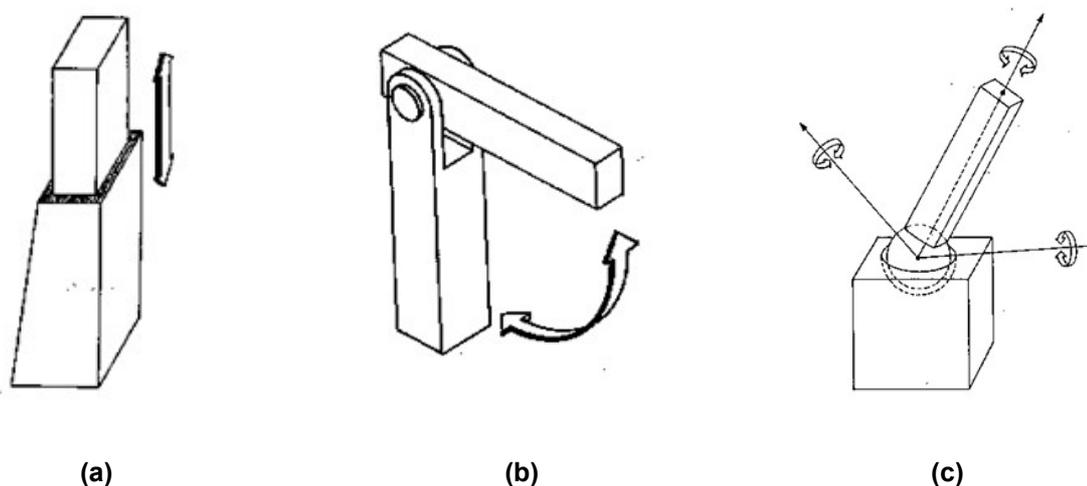


Figura 21. Juntas (a) lineares, (b) rotativas, e (c) prismáticas.
(BAISE, 2016)

3.5.4. PRECISÃO E REPETIBILIDADE

Os robôs são caracterizados também por meio de dois parâmetros: a **precisão** e a capacidade de repetição, ou **repetibilidade**. Por *precisão*, podemos entender como sendo a capacidade de um robô de ir a uma posição desejada em um sistema de referência fixo (geralmente a base do robô) com um erro determinado, como por exemplo, de ± 1 mm. Trata-se de “precisão em posicionamento absoluto” (ROSÁRIO, 2005, p.169).

Quanto à *capacidade de repetição*, entendemos como sendo a capacidade do robô de, uma vez conhecida e alcançada certa posição, e partindo-se sempre da mesma condição inicial, voltar a ir – ou seja, repetir – a posição com um erro determinado. Na maioria dos manuais dos robôs são fornecidas informações relacionadas à repetibilidade dos mesmos, mas não sobre sua precisão absoluta, pois é muito mais difícil de obter (ARMADA, 1995).

3.6 PRINCIPAIS APLICAÇÕES DOS ROBÔS INDUSTRIAIS

As aplicações dos robôs nas indústrias brasileiras são diversas. Segundo Rosário (2005, p. 150) em termos percentuais, por exemplo, os robôs da ABB™ são utilizados para soldagem por resistência por pontos (33%), manipulação de materiais/paletização (25%), soldagem por arco (18%), pintura (10%) e em outras aplicações como corte a jato de água, corte por gás, acabamento e montagem (14%).

De acordo com os dados apresentados por Rosário (2005) em “*Princípios da Mecatrônica*” (Prentice Hall, 2005):

Aproximadamente 65% dos robôs existentes no Brasil, em 1998, eram utilizados na indústria automobilística, principalmente na soldagem por resistência por pontos. [...] O nível de automatização da indústria automobilística em âmbito mundial já é de atualmente 90%, embora algumas tarefas de produção já tenham alcançado o nível máximo de automatização total. (ROSÁRIO; JOÃO MAURÍCIO, 2005, p.150).

3.7 CLASSIFICAÇÃO DOS ROBÔS

Os robôs industriais podem ser classificados como mostra a tabela 1 abaixo (ENGELBERGER, 1995):

Classificação dos Robôs	
	Sub categorias
1. Número de Juntas	
2. Tipo de Controle	
3. Tipo de Acionamento	Por movimento (<i>drivers</i>) de: <ul style="list-style-type: none"> • rotação • deslizamento
	Por acionamento (<i>driver</i>): <ul style="list-style-type: none"> • elétrico • hidráulico • pneumático
	Por conexão (<i>drivers</i>): <ul style="list-style-type: none"> • <u>Direto</u>: motor montado diretamente na junta que irá mover-se. • <u>Indireto</u>: usando-se elementos de transmissão (corrente, correias, diferenciais e engrenagens)
4. Geometria	

Tabela 1. Classificação dos robôs.
Fonte: O autor.

3.7.1 GEOMETRIA DO ROBÔ

Em sua explicação sobre a classificação dos robôs quanto à sua geometria (uma, dentre várias formas de classificação), as cinco classes ou geometrias principais de um robô são cartesiana, cilíndrica, esfera (polar), de revolução (articulada), e SCARA (*Selective Compliant Articulated Robot for Assembly*), (ROSÁRIO, 2005, p. 157):

- **Robô de Coordenadas Cartesianas:** pode movimentar-se em linha reta, em deslocamentos horizontais e verticais. As coordenadas cartesianas especificam um ponto do espaço em função de suas coordenadas x , y , e z . Esses robôs têm três articulações deslizantes e são codificados como PPP.

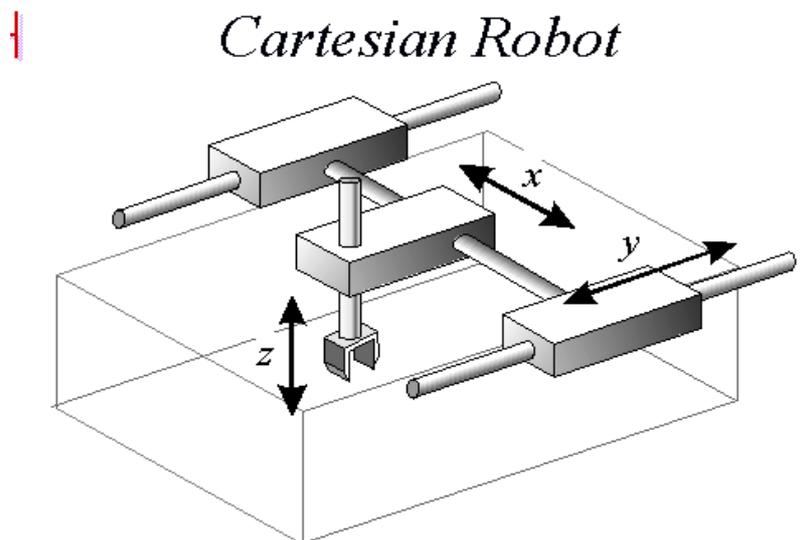


Figura 22. Robô cartesiano (ZALZALA, 2016).

- **Robô de Coordenadas Cilíndricas:** combina movimentos lineares com movimentos rotacionais. Normalmente, possuem um movimento rotacional e dois lineares, codificado como RPP. Sua área de trabalho é maior do que a dos cartesianos, mas rigidez mecânica ligeiramente inferior.

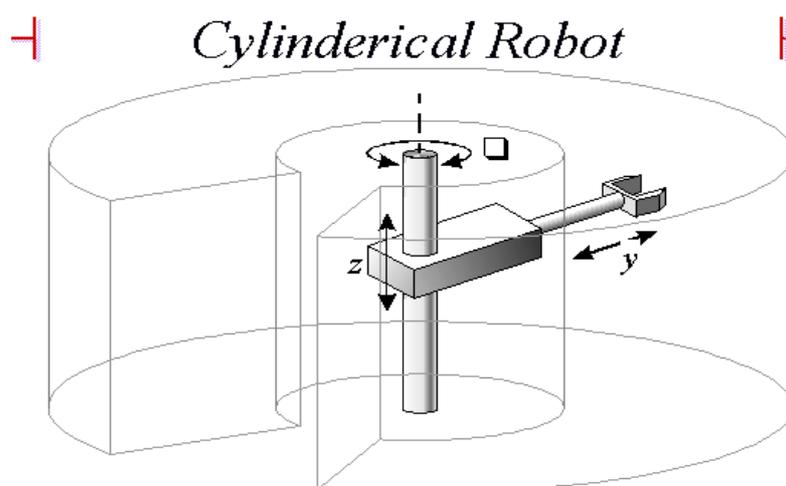


Figura 23. Robô cilíndrico (ZALZALA, 2016).

- **Robô de Coordenadas Esféricas (Polares):** possui dois movimentos rotacionais e um linear. Esses robôs são codificados como RRP. Sua área de trabalho é maior do que a dos modelos cilíndricos, porém sua rigidez mecânica é menor.

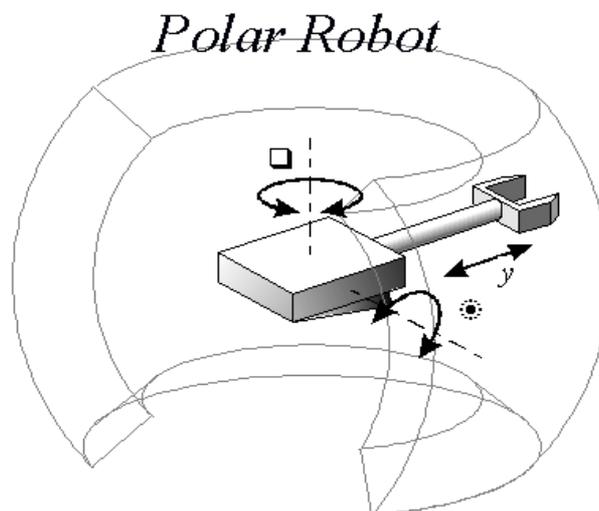


Figura 24. Robô esférico (ZALZALA, 2016).

- **Robô de Coordenadas de Revolução (Articulado ou Antropomórfico):** assemelha-se a um braço humano, caracteriza-se por possuir três juntas de revolução e são codificados como RRR. Sua área de atuação é maior do que a de qualquer tipo de robô, porém apresentam baixa rigidez mecânica. Seu controle é complicado e difícil em razão das três juntas de revolução e das variações no momento de carga e no de inércia.

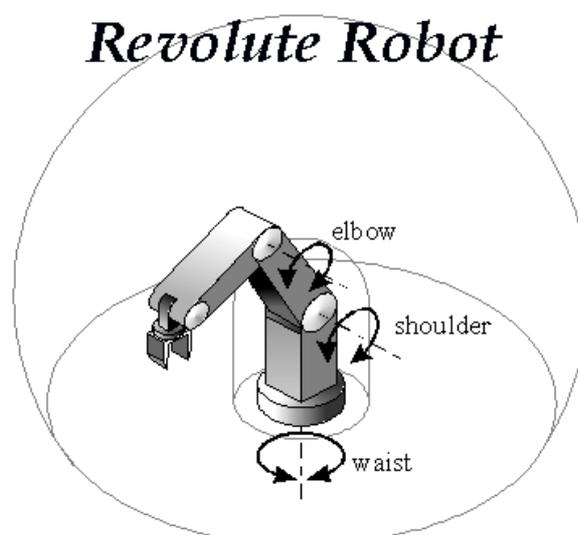


Figura 25. Robô articulado (ZALZALA, 2016).

- **Robô SCARA:** utilizado para tarefas de montagem, é codificado como RRP, possuindo duas juntas de revolução e uma deslizante. Os robôs SCARA têm área de atuação menor do que a do modelo esférico e são apropriados para operações de montagem devido ao movimento linear vertical do terceiro eixo.

✦ *SCARA Robot*

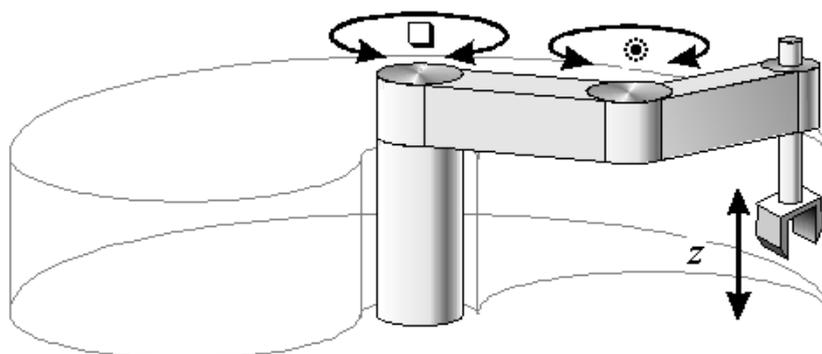


Figura 26. Robô SCARA (ZALZALA, 2016).

São através das articulações que o robô realiza o movimento do atuador para qualquer ponto na sua área de atuação, porém não habilita o controle da orientação do atuador no espaço. As características da movimentação são importante, não apenas devido a necessidade de alcance da peça, mas também devido a necessidade de condução do atuador a uma certa altitude em relação à peça, tarefa esta que pode ser realizada com a adição de articulações ao pulso do braço, proporcionando-lhe um grau de liberdade maior, o que possibilita ao robô a execução dos seguintes movimentos, conforme mostrados na figura 27, e descritos a seguir.

- *Roll (R)*: movimento de rotação no sentido horário e anti-horário;
- *Pitch (P)*: movimento para cima e para baixo;
- *Yaw (Y)*: movimento para a esquerda e para a direita.

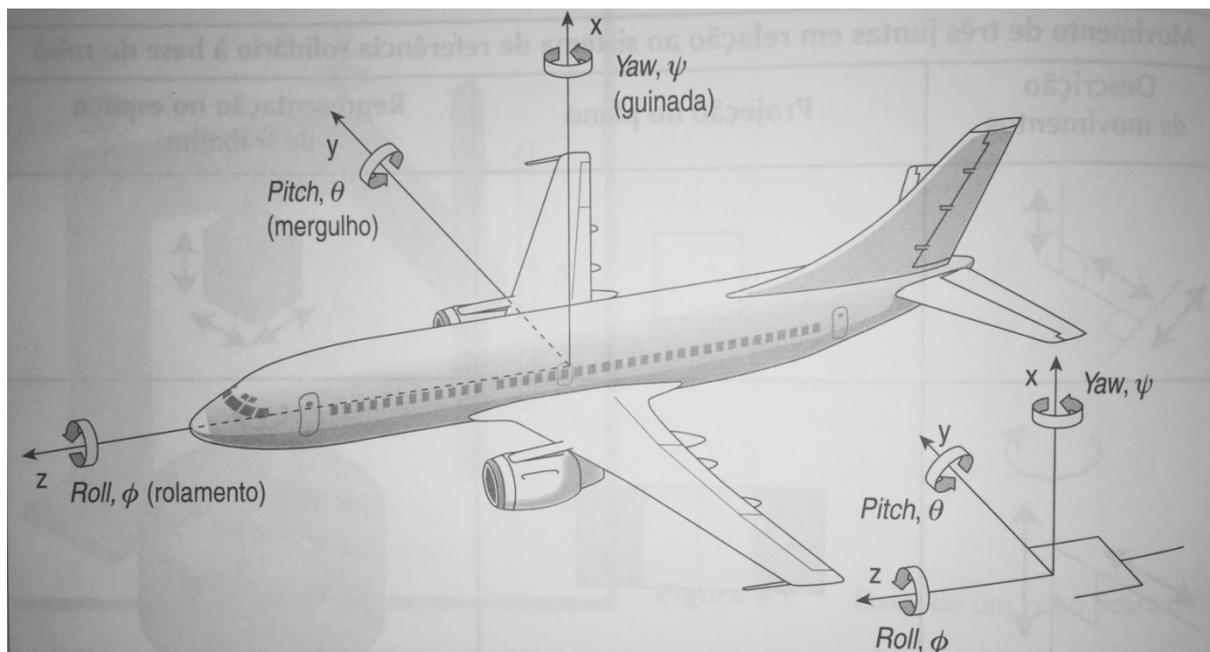


Figura 27. Definição dos ângulos de orientação RPY (BAISE, 2016).

3.8 PROGRAMAÇÃO DE ROBÔS

Robôs industriais são equipamentos reprogramáveis de múltiplas funcionalidades com grande flexibilidade de operação. Nos dias atuais, a programação é realizada por meio de um 'controle remoto' responsável por conduzir o robô pelas posições críticas do ciclo de operação. O robô, neste caso, é utilizado durante o período de programação, não sendo possível um controle muito preciso sobre a **trajetória** da garra ou ferramenta. Trata-se, numa situação como a descrita, de uma programação *on-line*.

A programação de tarefas *off-line* não apresenta tais inconvenientes por ser realizada em computadores, necessitando apenas de um modelo matemático. Esse modelo contém informações sobre a cinemática e a dinâmica do robô. Uma das principais vantagens da programação *off-line* é a ausência de necessidade da disponibilização dos equipamentos para o desenvolvimento do programa, necessitando-se, a princípio, apenas de um desenho da planta e das especificações funcionais e tecnológicas dos elementos da célula, para posteriormente gerar os movimentos do manipulador e integrá-lo dentro de uma célula automatizada, a partir da lógica de programação e cálculos.

Utilizando-se ferramentas de programação *off-line*, permite-se ainda a depuração de programas e o uso de ferramentas que servem de suporte para o processo de programação (por exemplo, para a otimização do processo de soldagem, montagem ou pintura).

No robô industrial, os seus diferentes graus de liberdade podem ser associados a diversos sistemas de coordenadas, cada um deles correspondendo a um grau de liberdade e servindo para descrever o movimento do grau de liberdade associado a ele.

O modelo geométrico, segundo ROSÁRIO (2005, p. 177), é aquele que expressa a posição e a orientação da garra em relação a um sistema de coordenadas fixo à base do robô em função de suas coordenadas generalizadas (angulares, no caso de juntas rotacionais) e essa relação “pode ser expressa matematicamente por uma matriz, denominada **matriz de passagem homogênea do robô**, que relaciona o sistema de coordenadas da base com o sistema de coordenadas do último elemento”.

A programação lógica pode ser então desenvolvida *off-line*, com facilidade de depuração e simulação efetivas. Os comandos de movimentos principais podem ser criados *off-line*, podendo-se reutilizar dados de CAD (*Computer Aided Design*) e pela interação do programador. Os comandos que envolvam movimentos para localização e colocação de peças dentro de células robotizadas são melhor e mais seguramente programados *on-line*. Verificando-se, posteriormente, o programa do robô na simulação, este deve ser descarregado na unidade de controle, um CLP por exemplo.

A programação das tarefas de robôs industriais envolve, ainda de acordo com as conclusões de ROSÁRIO (2005, p.194), a integração de três fatores básicos:

- 1) A coordenada do ponto de movimento deve ser identificada e armazenada na unidade de controle. O ponto pode ser armazenado como coordenada de eixo de junta individual ou como coordenada geométrica do tipo que o robô possa armazenar.
- 2) As funções, executadas nos pontos específicos, devem ser identificadas e gravadas.
- 3) Os dados dos pontos e funções são organizados em um caminho de seqüência lógica e de subseqüências. Isso inclui pressupor que caminhos devem ser feitos sobre condições específicas e o momento em que serão executadas várias supervisões de estado.

A utilização de todo o potencial que um software de programação *off-line* tem a oferecer permite que células robotizadas, constituídas de dispositivos automatizados integrados, sejam implementadas de modo rápido e consistente, proporcionando ainda maior segurança.

Ao investir-se na aquisição dessas ferramentas, tais como o RobotStudio®, permite-se, ao longo do tempo, que o uso da integração de processos tenham maior credibilidade e que seus conceitos sejam disseminados sob forma de treinamento e adequação do contexto, somando esforços dos fabricantes de robôs que importam soluções de seus países para atendimento dos clientes (ROSÁRIO, 2005, p.194).

4 DESENVOLVIMENTO

4.1 INICIALIZAÇÃO DA CONSTRUÇÃO DE UM SISTEMA VIRTUAL

4.1.1 LAYOUT

Buscou-se, primeiramente, localizar e gerar o ambiente virtual com todos os elementos e componentes possíveis, que representassem - tanto visualmente quanto por finalidade específica – os objetos e aparelhos presentes no laboratório físico da sala A-007. A realização desta construção pode acontecer de duas formas, conforme o desejo do projetista:

- Busca de objetos prontos na biblioteca do *software* (*Library*);
- Importar objetos 3D desenvolvidos em outro software CAD;

4.1.2 LIBRARY (“BIBLIOTECA”)

Para procurar um objeto na biblioteca do RobotStudio®, é necessário clicar no botão “*Import Library*” (Importar Biblioteca), localizado no canto superior esquerdo da tela, para buscar (*Browse*) diretamente um componente localizado em determinado diretório (*Folder*), conforme mostra a figura 28 - ou clicando na seta imediatamente ao lado do botão para localizar, visualmente, os elementos presentes na biblioteca (figura 29).

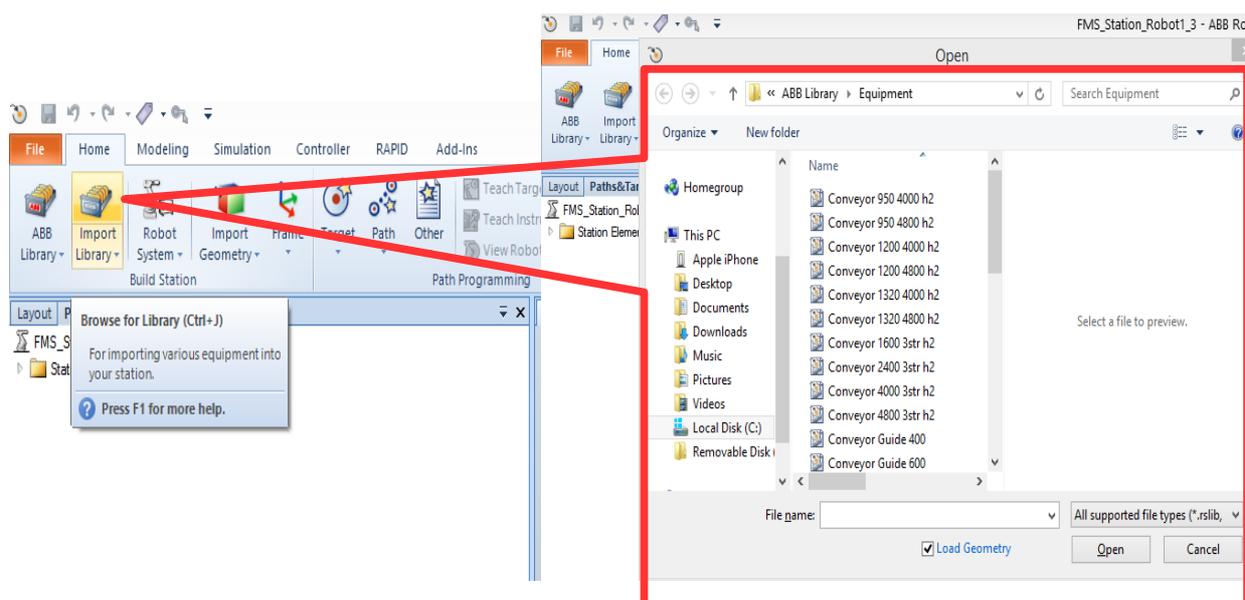


Figura 28 Busca em Biblioteca (*Browse Library*).
Fonte: O autor.

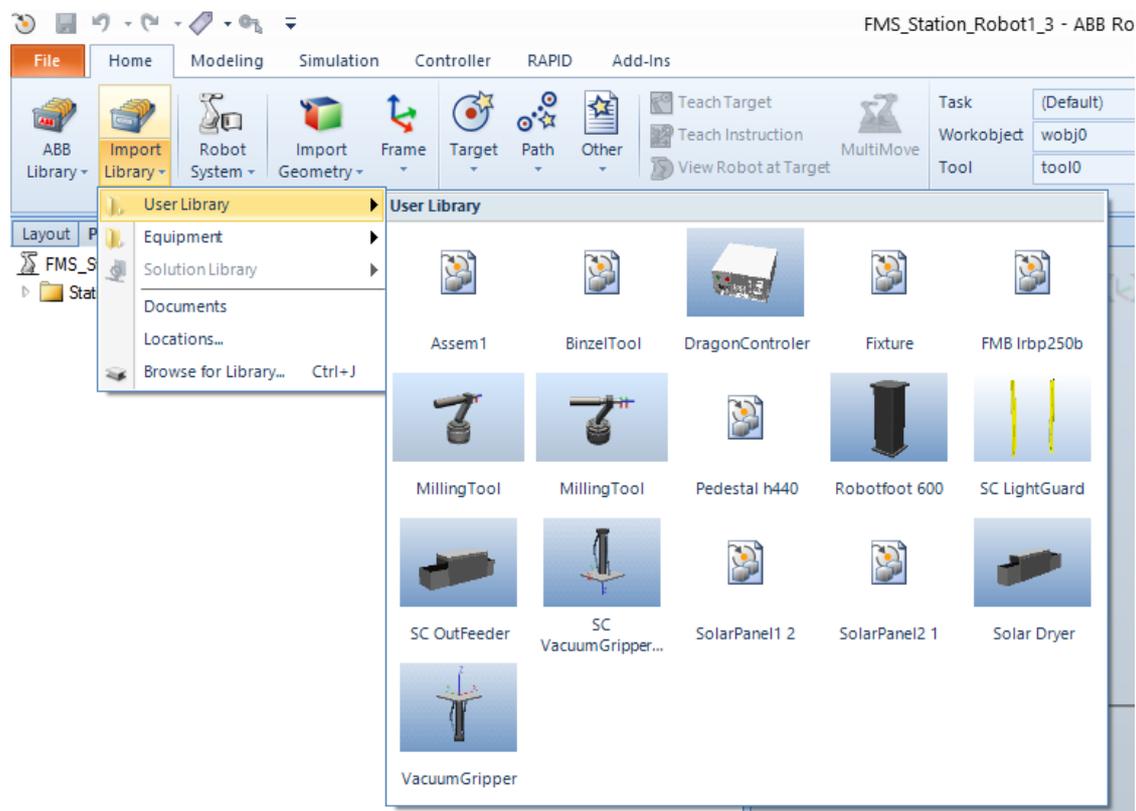
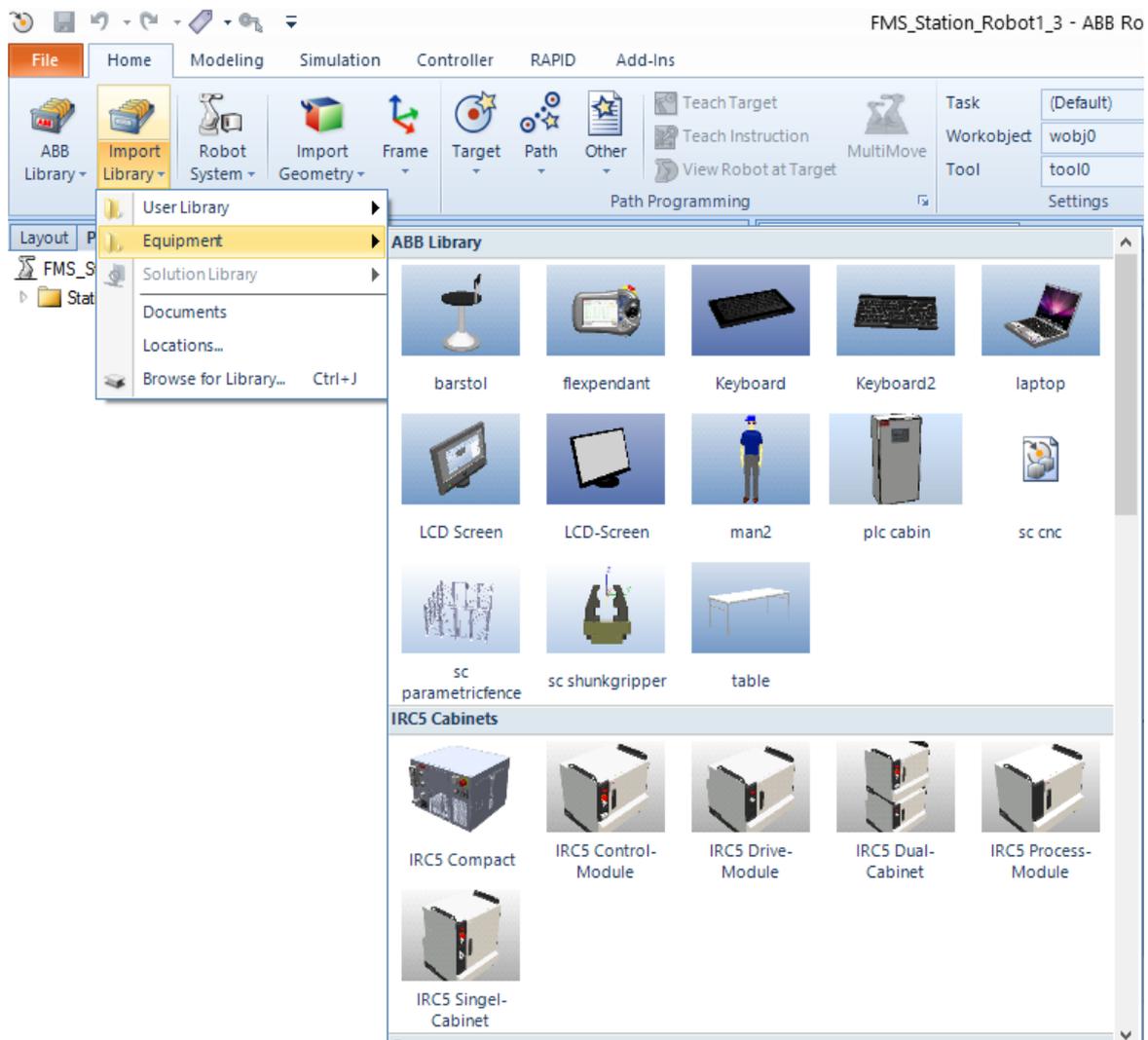
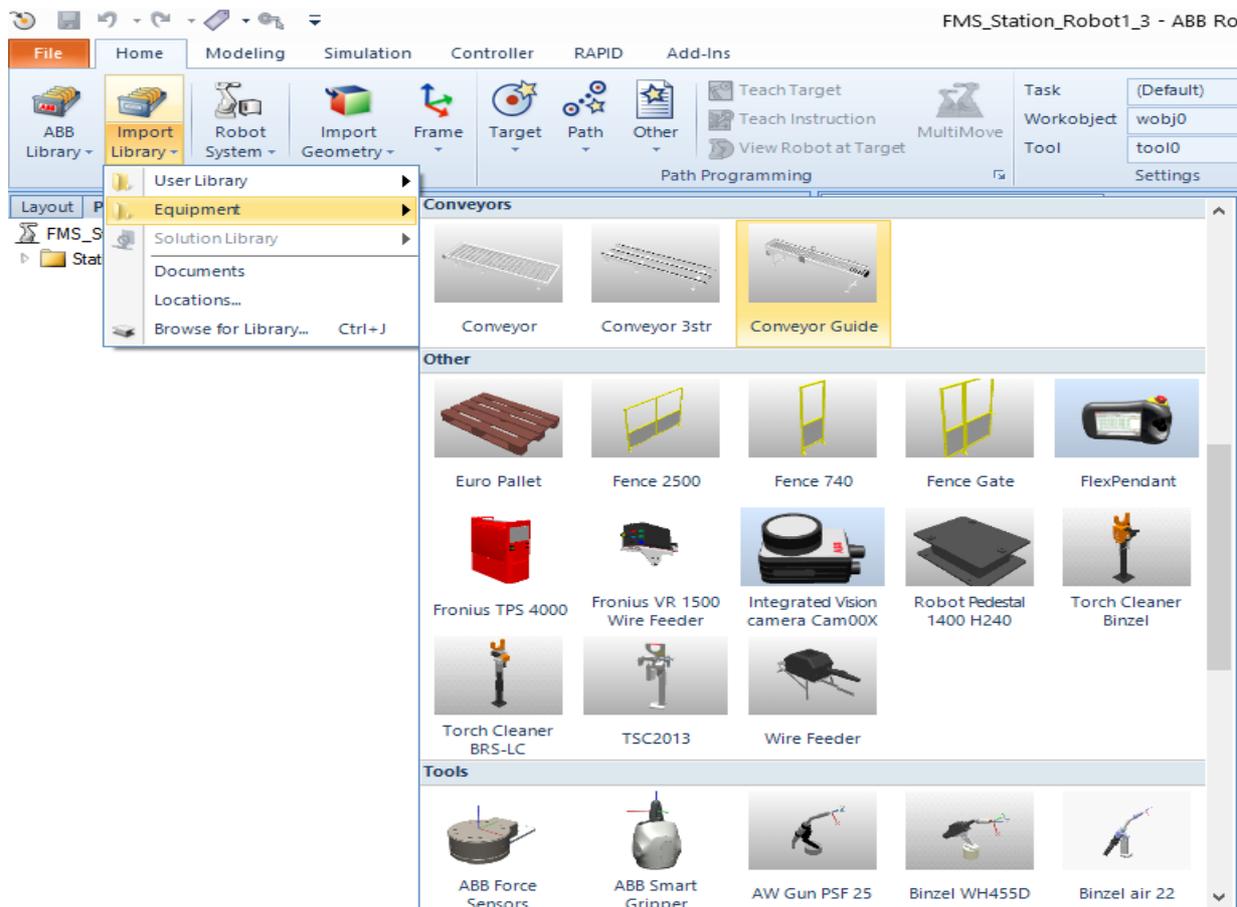


Figura 29. Importar Objetos da Biblioteca do Usuário (*Import Objects from User Library*).
Fonte: O autor.

Desta forma, realizou-se a busca de componentes de sistema como: (a) esteiras (*conveyors*), suportes para máquinas e equipamentos, gabinetes para equipamentos e controladores, mesas, cadeiras, bem como equipamentos eletrônicos: computadores, teclados, e câmera de inspeção (b) conforme mostram as figuras 30 e 31.



(a)



(b)

Figura 30. Importar Equipamentos da Biblioteca (*Import Equipments from Library*), (a) controladores e cabines, (b) esteiras e objetos em geral. Fonte: O autor.



Figura 31. Principais elementos e componentes importados para o ambiente virtual da ABB™. Fonte: O autor.

4.1.3 CRIAÇÃO DE OBJETOS 3D

Tendo em vista que nem todos os equipamentos ou objetos necessários são encontrados dentro da plataforma de construção, já que cada sistema apresenta *layouts* específicos de acordo com a necessidade do projetista, é possível, através do software, acessar ou gerar objetos 3D que representem o elemento desejado, existindo duas formas de adquirir esses objetos customizados:

- Ferramenta de modelamento 3D, contida dentro no próprio *software* RobotStudio®;
- Importação de objetos 3D gerados por outro software CAD.

4.1.3.1 CRIAÇÃO DE OBJETOS 3D NO RobotStudio®

Para a criação de Objetos 3D utilizando a própria ferramenta de modelamento do RobotStudio®, mostrada na figura 32, é necessário abrir a aba “*Modeling*” (Modelamento), e escolher a forma na qual se deseja construir o objeto. A princípio, apesar de haver uma diversidade de recursos para a geração de objetos 3D, neste trabalho utilizou-se esta ferramenta apenas para a criação de elementos simples e de prática aplicação, tais como pallets, suportes, tarugos, entre outros. Para tal, optou-se pela construção de elementos completamente sólidos e bem definidos, através da escolha do botão “*Solid*”.

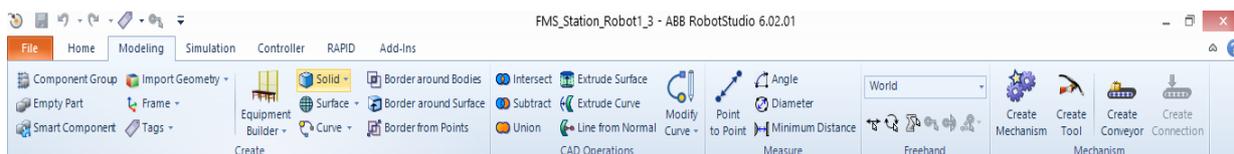


Figura 32. Barra de Ferramentas para Modelamento de Objetos 3D.
Fonte: O autor.

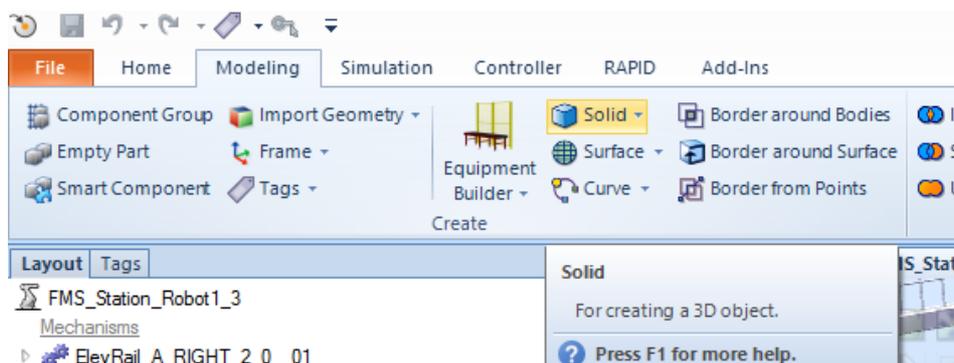


Figura 33. Escolhendo gerar objetos sólidos.
 Fonte: O autor.

Nota-se, conforme mostram as figuras 33 e 34, que é possível gerar diversas formas sólidas, sejam elas cúbicas, cônicas, cilíndricas, piramidais ou esféricas.

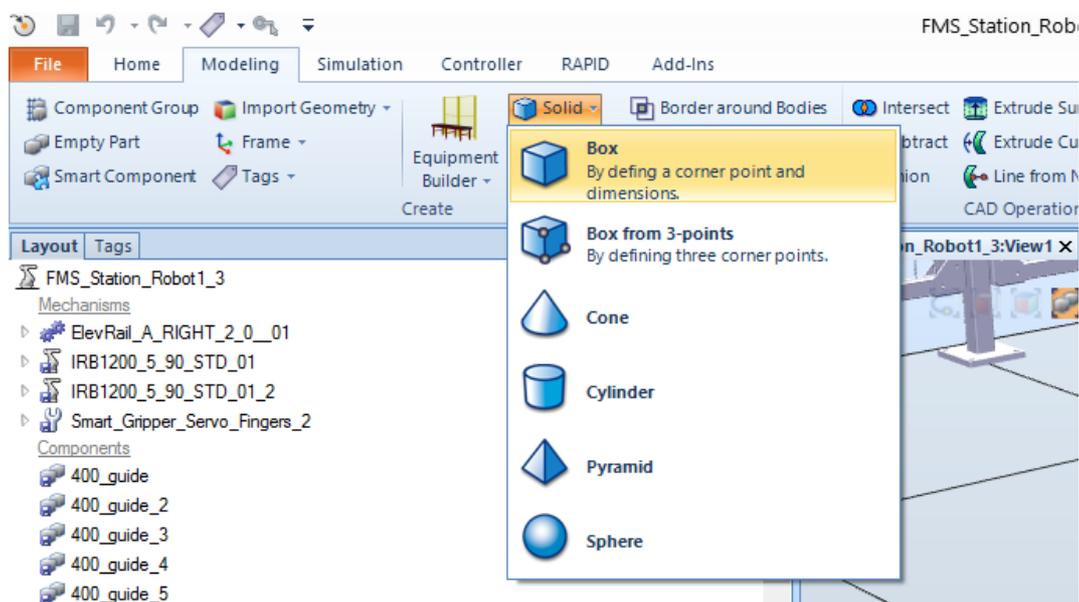


Figura 34. Opções para a criação de objetos sólidos 3D no RobotStudio®.
 Fonte: O autor.

Para o presente trabalho utilizou-se, em grande parte, a criação de objetos 3D em formatos cúbicos para a construção dos *pallets* de madeira de 2mm de espessura (figura 35), assim como suportes básicos para máquinas CNC e Armazém (figura 36).

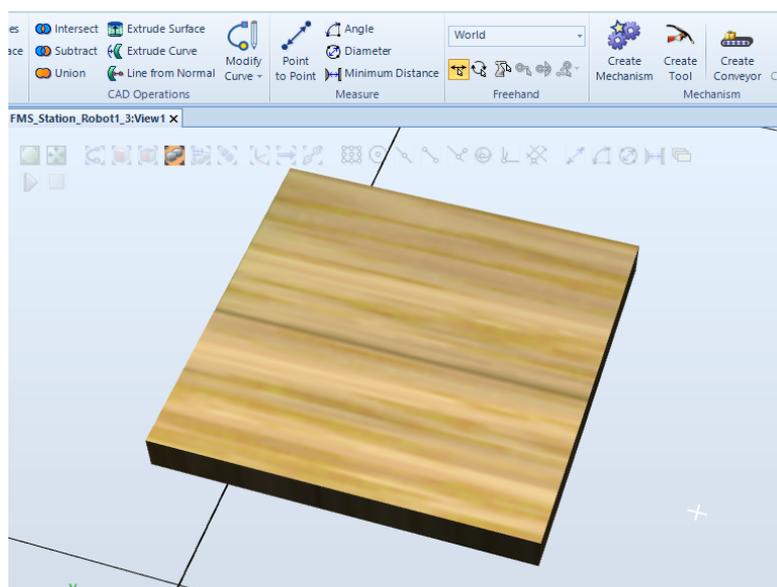


Figura 35. Pallet de madeira.

Fonte: O autor.

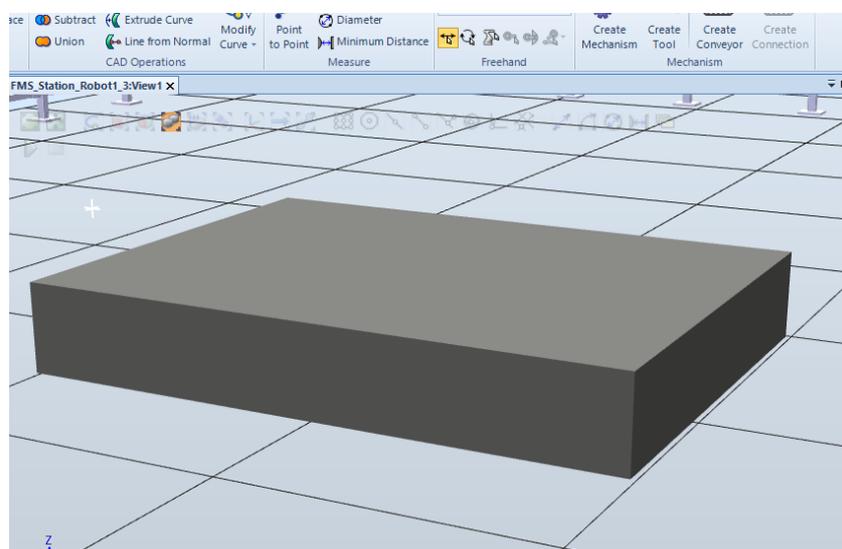


Figura 36. Base suporte para cabine de CLP e sistema de armazenagem.

Fonte: O autor.

Foram também construídos armários de madeira (figura 37), base da mesa móvel (figura 38) e as castanhas do centro de torneamento (figura 39) bem como a morsa do centro de usinagem/fresamento (figura 40), conforme mostra-se a seguir.

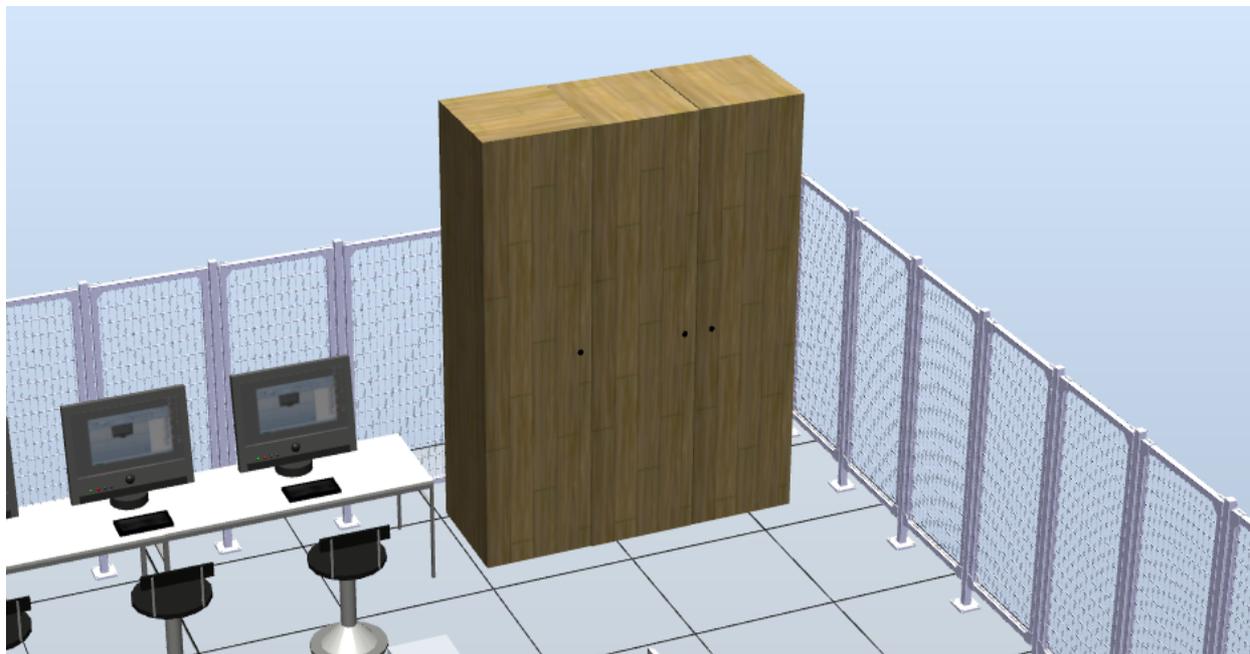


Figura 37. Armários de madeira gerados pela ferramenta 3D do RobotStudio®.
Fonte: O autor.

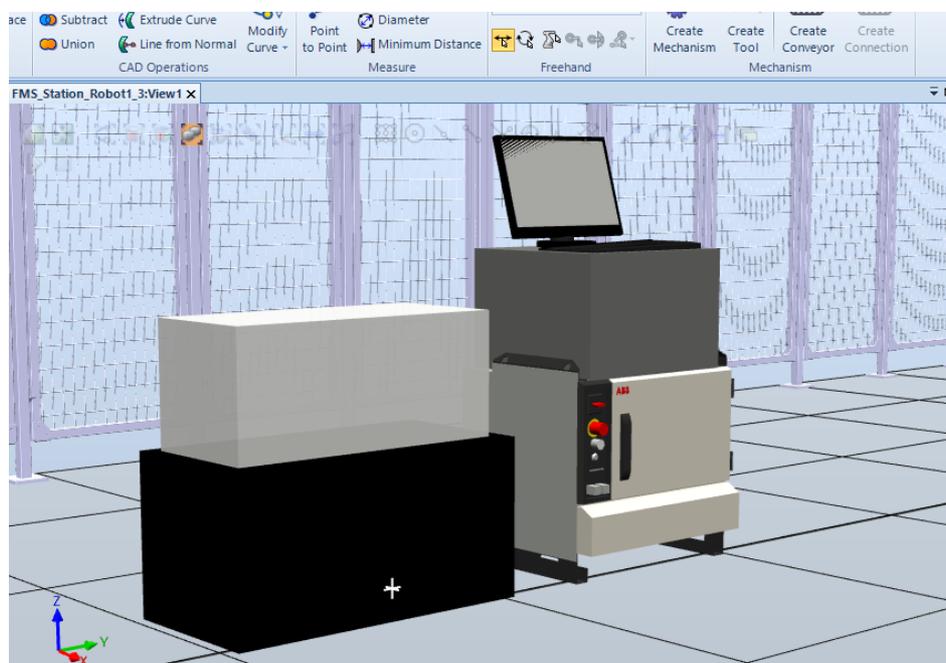


Figura 38. Objetos diversos gerados pela construção sólida cúbica 3D.
Fonte: O autor.



Figura 39. Detalhamento dos elementos básicos do Centro de Torneamento criados em 3D.
Fonte: O autor.

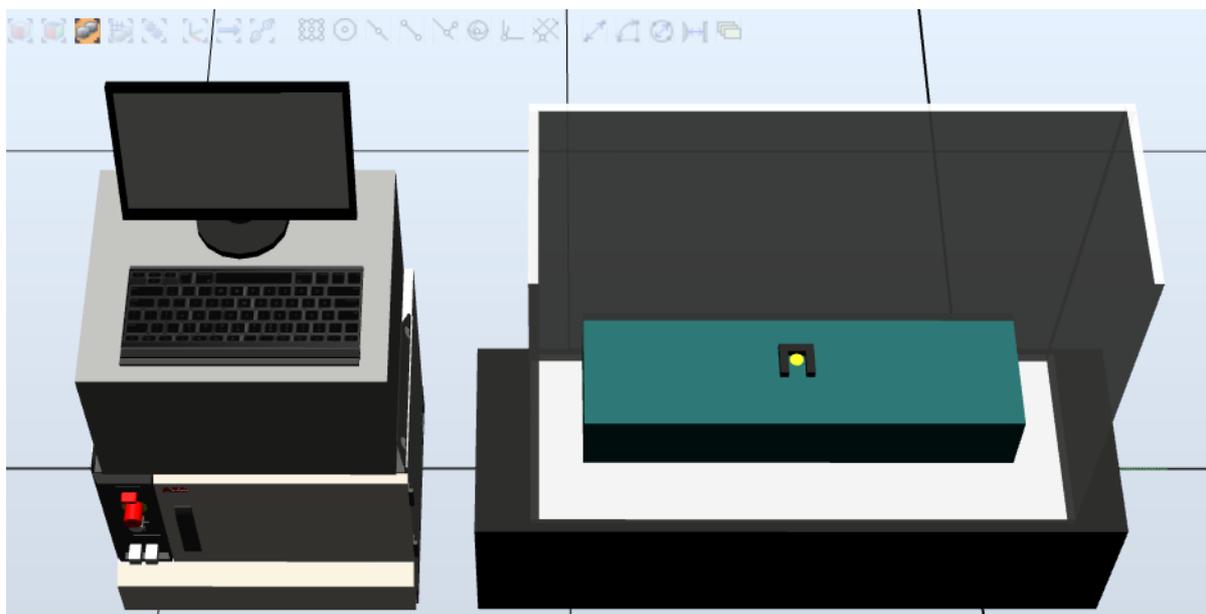


Figura 40. Detalhamento dos elementos básicos do Centro de Usinagem criados em 3D.
Fonte: O autor.

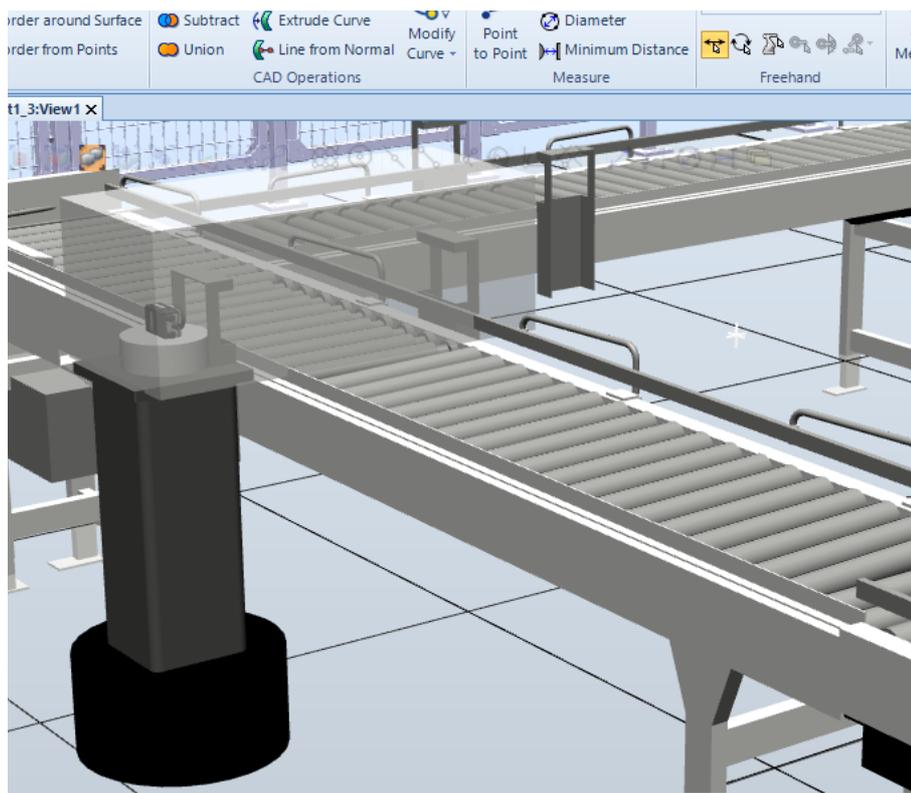


Figura 41. Formatos cilíndricos e cúbicos gerados para o Sistema de Visão na ferramenta 3D do RobotStudio®. Fonte: O autor.

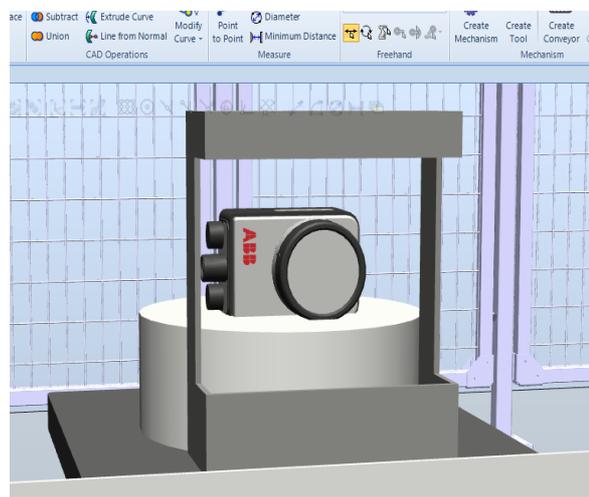
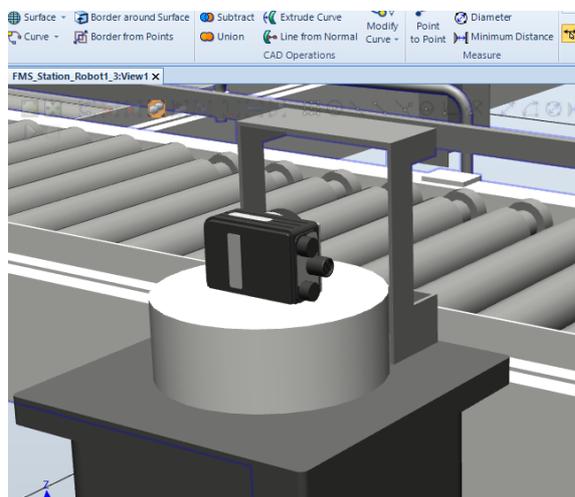
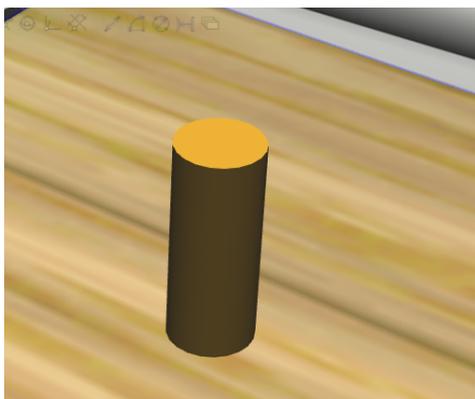


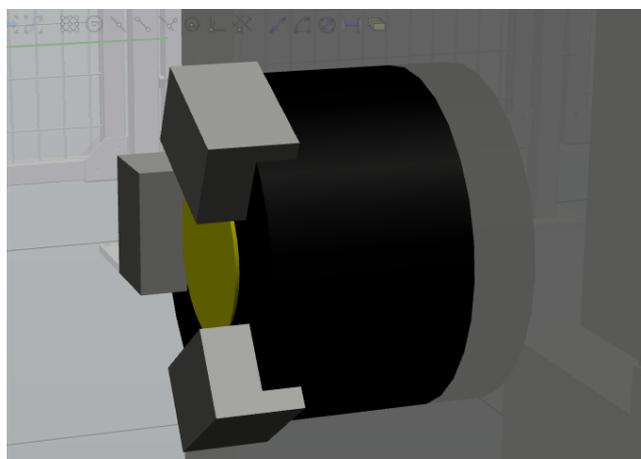
Figura 42. Objetos do Sistema de Visão. Fonte: O autor.

Formatos 3D cilíndricos foram construídos para a base de suporte da câmera integrada de inspeção (figuras 41 e 42), para tarugos de diâmetro 31.7 mm (figura 43) e para a base cilíndrica das castanhas do centro de torneamento (figura 44).

O recurso de modelamento do RobotStudio® também permite definir o material do objeto 3D criado, em suas propriedades, conforme o tarugo definido com material de latão, mostrado na figura 43.



**Figura 43. Tarugo de latão de 31,7 mm de diâmetro e 80mm de altura gerado em 3D.
Fonte: O autor.**



**Figura 44. Base cilíndrica das castanhas do eixo árvore no centro de torneamento.
Fonte: O autor.**

Em formatos cônicos, foram construídos alguns elementos de auxílio para determinarem as áreas de centralização para depósito de peças dentro do centro de torneamento e fresamento (figuras 45 e 46). Esses “elementos de posicionamento” foram gerados arbitrariamente pelo projetista do presente trabalho, como forma simbólica de localização do posicionamento das peças nas castanhas e na morsa, o

que facilitaria o monitoramento visual durante futura simulação, bem como auxiliar na criação dos *targets*, ou “alvos”, os quais serão posteriormente discutidos.

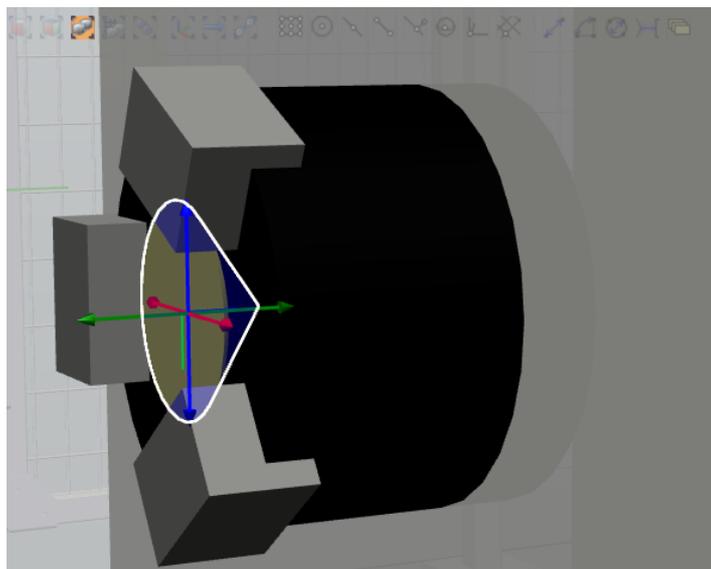


Figura 45. Centro de posicionamento de peças nas castanhas do centro de torneamento.
Fonte: O autor.

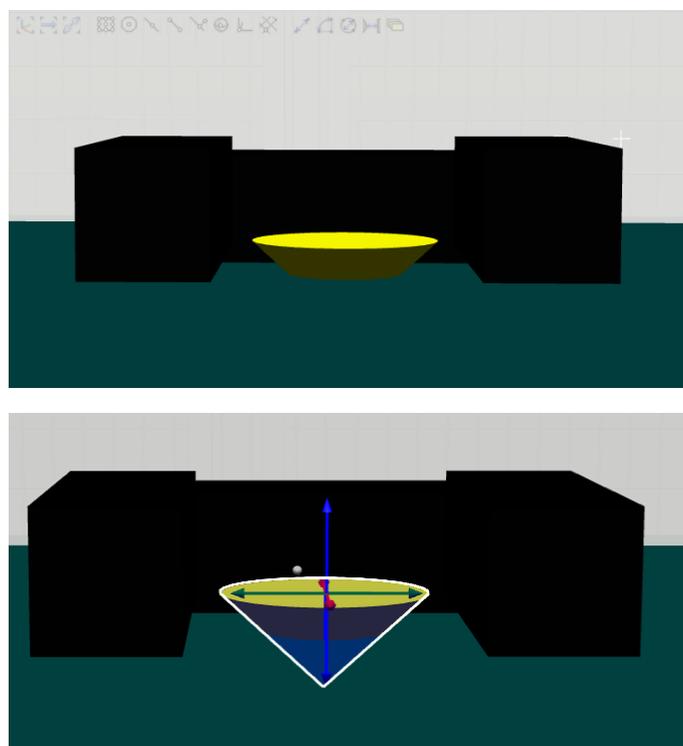


Figura 46. Centro de posicionamento de peças na morsa do centro de fresamento.
Fonte: O autor.

Após a escolha do formato sólido para a construção virtual desses objetos, depara-se com uma janela solicitando os parâmetros das dimensões dos objetos. Para um objeto cúbico, é necessário informar o Comprimento (*Length*), a Largura (*Width*), e a Altura (*Height*) desejada do objeto em questão (figura 47).

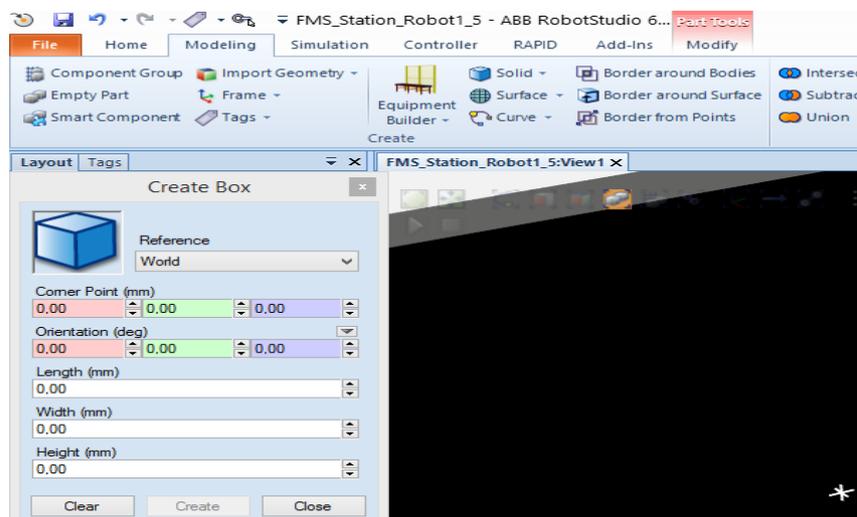


Figura 47. Parâmetros para criar um objeto 3D cúbico.

Fonte: O autor.

Nota-se que os parâmetros de medida estão, neste exemplo, determinados em milímetros (mm). No entanto, pode-se definir outros sistemas de medição, como polegadas (*inches*), por exemplo, ao definir a configuração na etapa inicial de criação da estação de trabalho.

Na sequência de criação dos objetos 3D, determinou-se o diâmetro e a altura dos objetos cônicos, que variavam de 15 a 20 mm de altura, e 40 mm de diâmetro (figura 48).

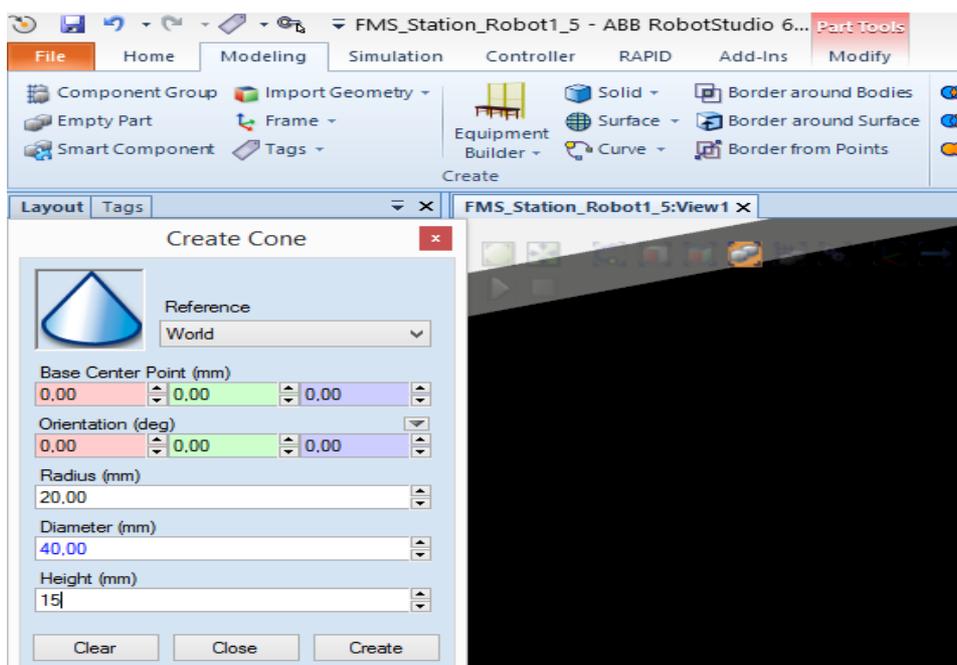


Figura 48. Parâmetros para criar um objeto 3D cônico.
Fonte: O autor.

O mesmo procedimento deu-se para o dimensionamento dos objetos cilíndricos, no qual é necessário informar o diâmetro ou raio, e a altura dos mesmos, conforme mostrado na figura 49.

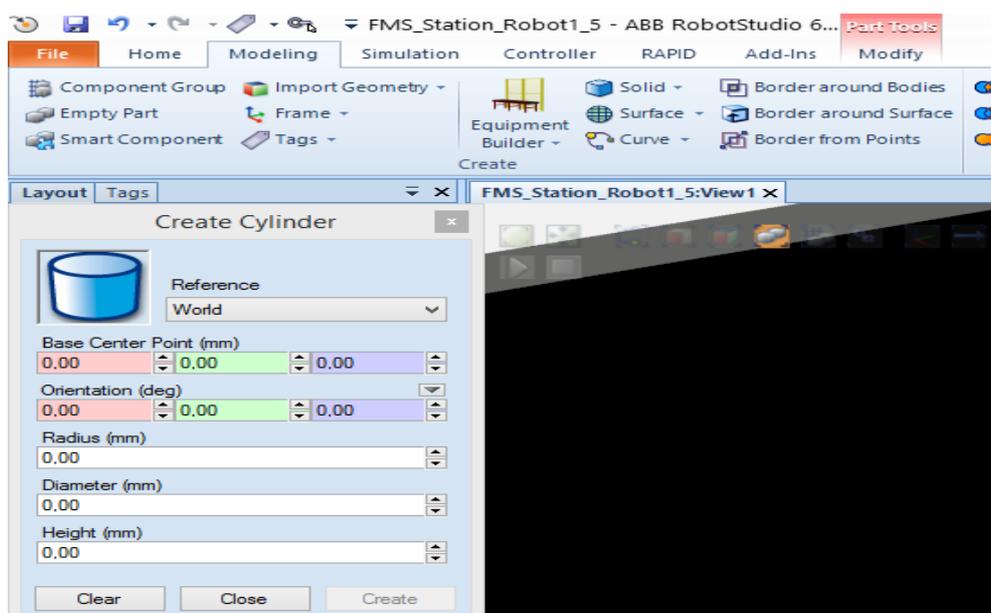


Figura 49. Parâmetros para criar um objeto 3D cilíndrico.
Fonte: O autor.

4.1.3.2 IMPORTANDO OBJETOS 3D CAD

Caso haja interesse em desenvolver objetos mais sofisticados na representação 3D em CAD, pode-se, alternativamente, importar esses objetos provenientes de outros *softwares*, como os desenvolvidos em Solidworks®, por exemplo. Faz-se necessário, portanto, procurar o diretório no qual o elemento está armazenado.

Na aba “Home”, conforme indicado na figura 49, escolhe-se a opção “Import Geometry” (ou, “Importar Geometria”, em tradução livre), no canto superior esquerdo da barra de ferramentas (figura 50).

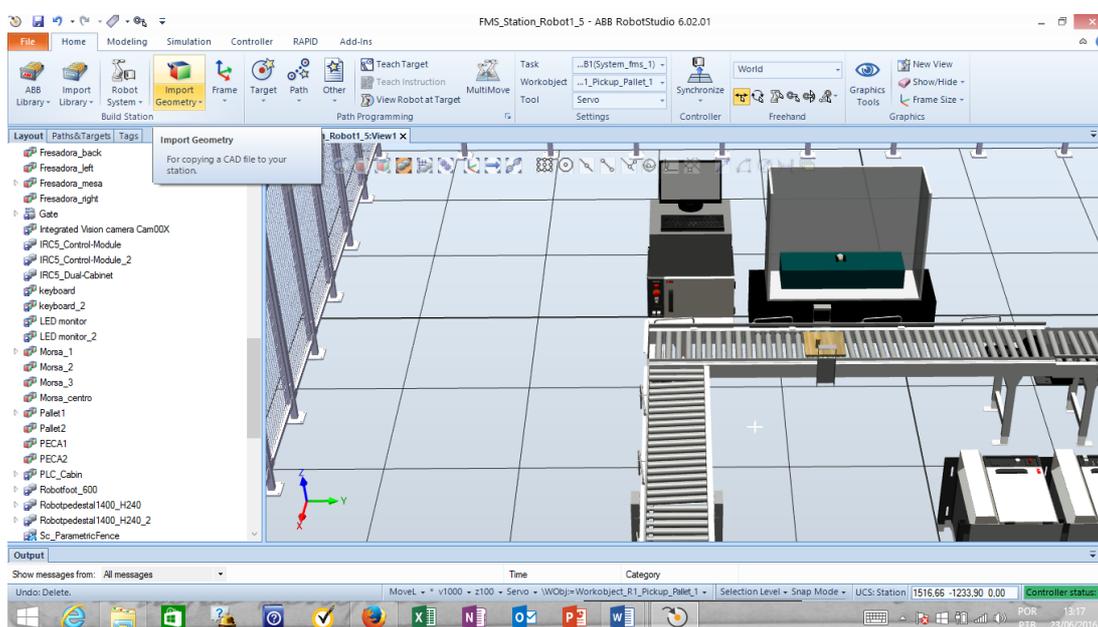


Figura 50. Recurso para importação de objetos 3D desenvolvidos em CAD.

Fonte: O autor.

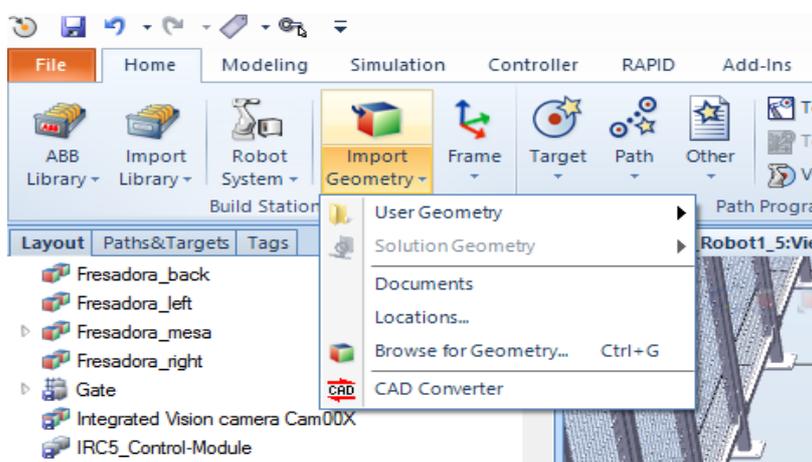


Figura 51. Opções para a importação de objetos 3D desenvolvidos em CAD.

Fonte: O autor.

Pode-se notar, conforme mostrado na figura 51, que existem opções variadas para o manejo de um objeto 3D advindo de outro *software* CAD. É possível localizar uma criação do usuário em “*User Geometry*”, ou buscar geometria em determinado diretório com “*Browse for Geometry*” (figura 52), bem como converter um arquivo CAD para a plataforma do RobotStudio®.

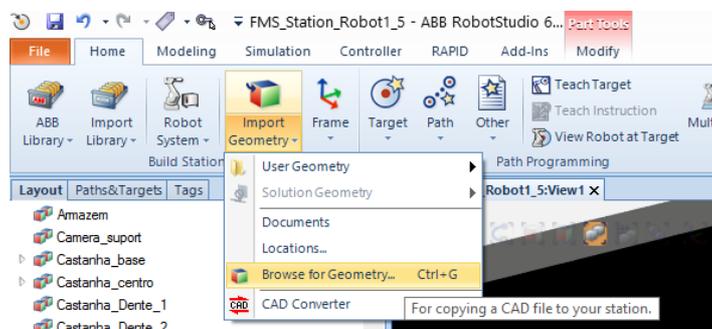


Figura 52. Localizando o arquivo da peça desenvolvida em CAD.
Fonte: O autor.

Para fins de pesquisa, buscou-se importar um componente de peça anteriormente projetado no CAD Solidworks®, localizando-o facilmente em seu diretório de comum armazenamento. O arquivo importado, neste caso, possuía a extensão “.SLDPRT”, e processou a peça sem apresentar dificuldades, conforme mostram as figuras 53 e 54.

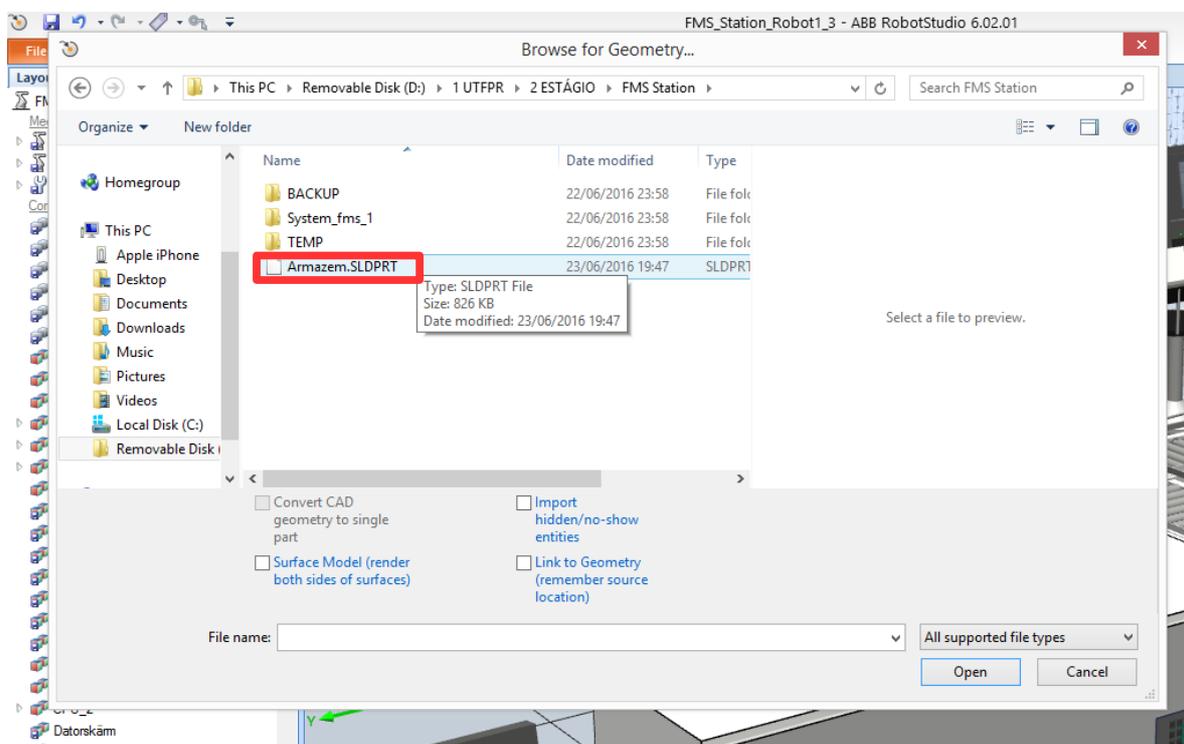


Figura 53. Escolhendo o arquivo 3D do CAD Solidworks®. Fonte: O autor.

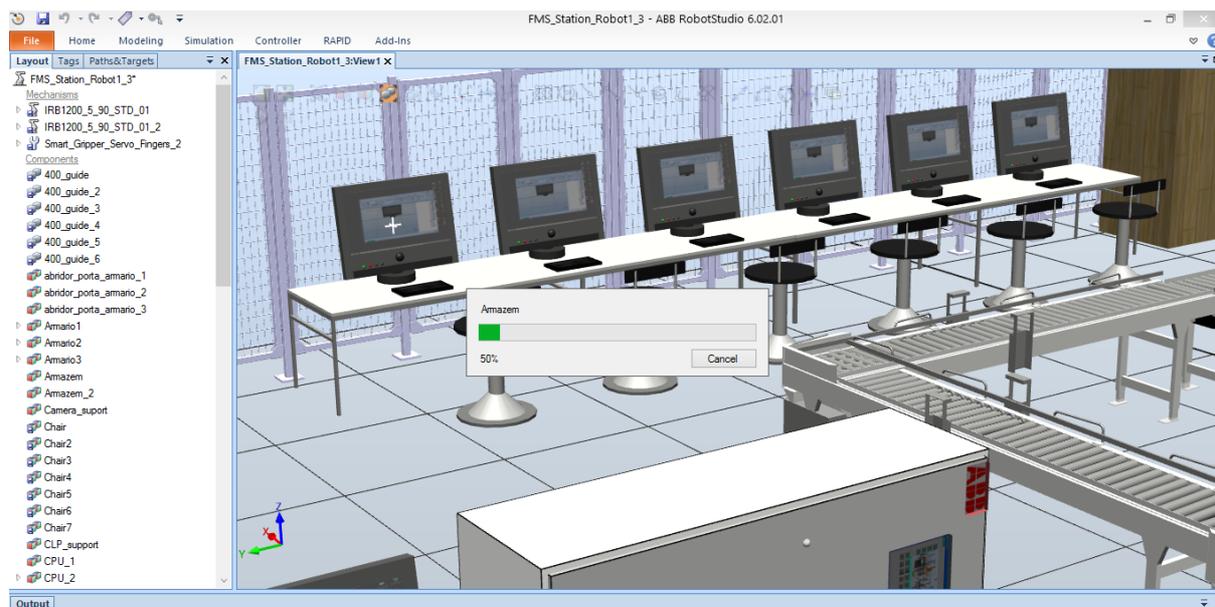


Figura 54. Importando a peça 3D desenvolvida em CAD Solidworks®.
Fonte: O autor.

Uma vez que o objeto CAD 3D foi importado para o ambiente virtual do RobotStudio®, com seu real dimensionamento físico, este fica listado no registro de componentes, como parte integrante do Layout do sistema, conforme mostra a figura 55.

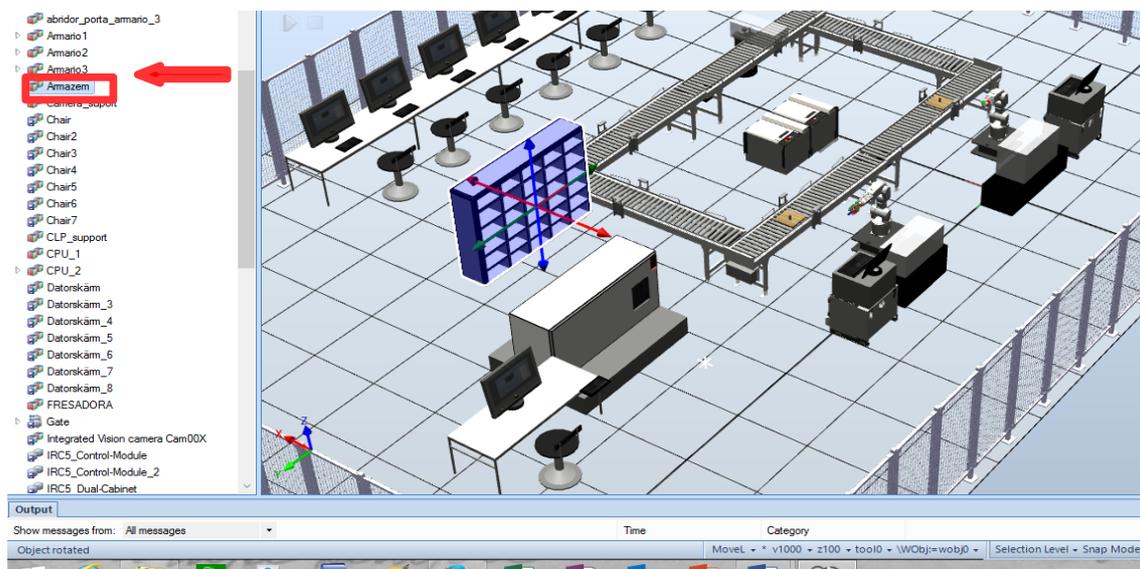


Figura 55. Peça 3D desenvolvida em CAD Solidworks® importada para o ambiente virtual do RobotStudio®.
Fonte: O autor.

A partir desse momento, esse novo componente pode ser renomeado de acordo com a sua definição no sistema (figura 56).

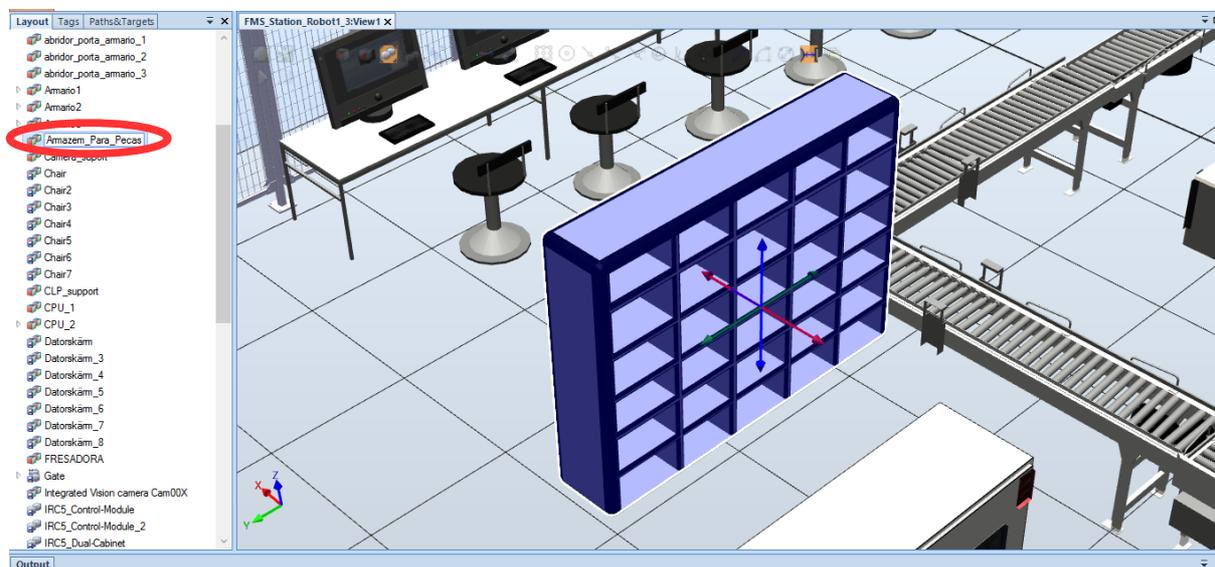


Figura 56. Renomeando um componente 3D recém importado para o ambiente virtual do RobotStudio®. Fonte: O autor.

O sistema de coordenadas exibido na peça ao ser selecionada representa o ponto central da peça, chamado de **TCP** ou sistema de coordenadas “**Tool Center Point**”, e com ele tem-se a possibilidade de movimentar e posicionar a peça no local desejado.

4.1.4 POSICIONAMENTO E AJUSTES DE OBJETOS NO AMBIENTE

No ambiente virtual do RobotStudio®, pode-se movimentar os objetos ao longo dos 3 eixos de orientação das coordenadas cartesianas global (*World*) ou local (*Local*). Clicando-se com o botão direito em cima do objeto listado à esquerda da tela, escolhendo-se “*Positions*”, e após, “*Set Position*” (figura 57) – sendo possível entrar manualmente com os valores, bem como a orientação, caso necessário.

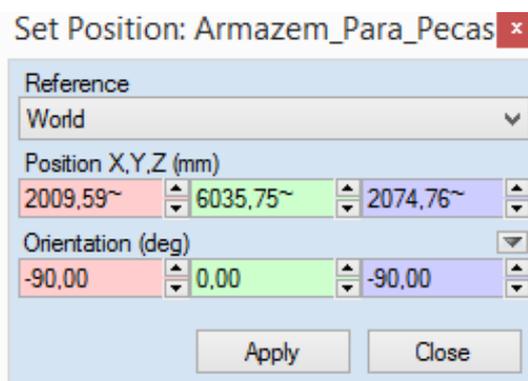
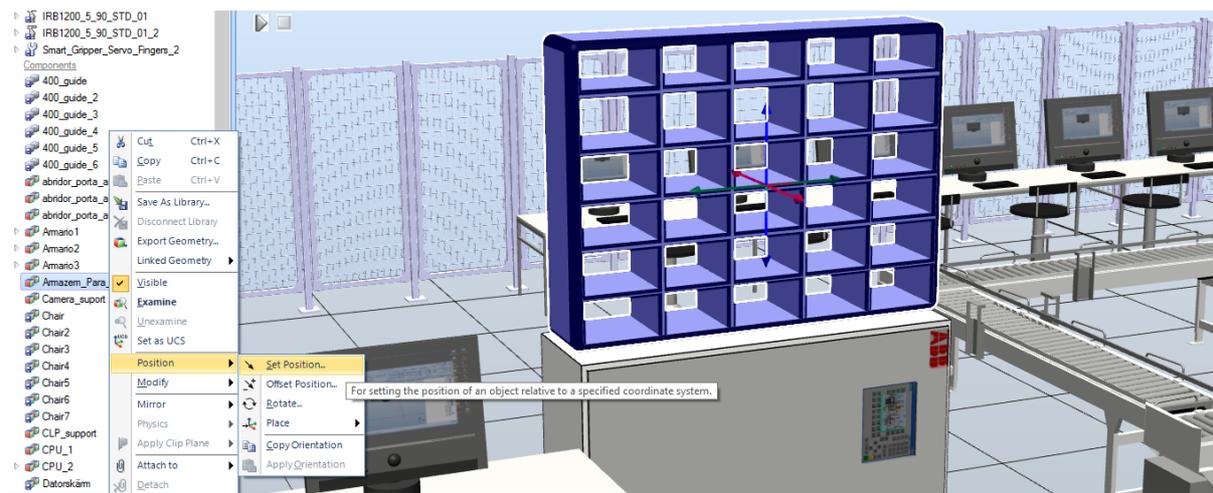


Figura 57. Como posicionar um objeto no RobotStudio®.
Fonte: O autor.

Alternativamente, pode-se também realizar o reposicionamento, linear apenas, atuando diretamente no objeto, clicando nele e “arrastando” os eixos das coordenadas do TCP do mesmo na direção desejada, mostrado na figura 58.

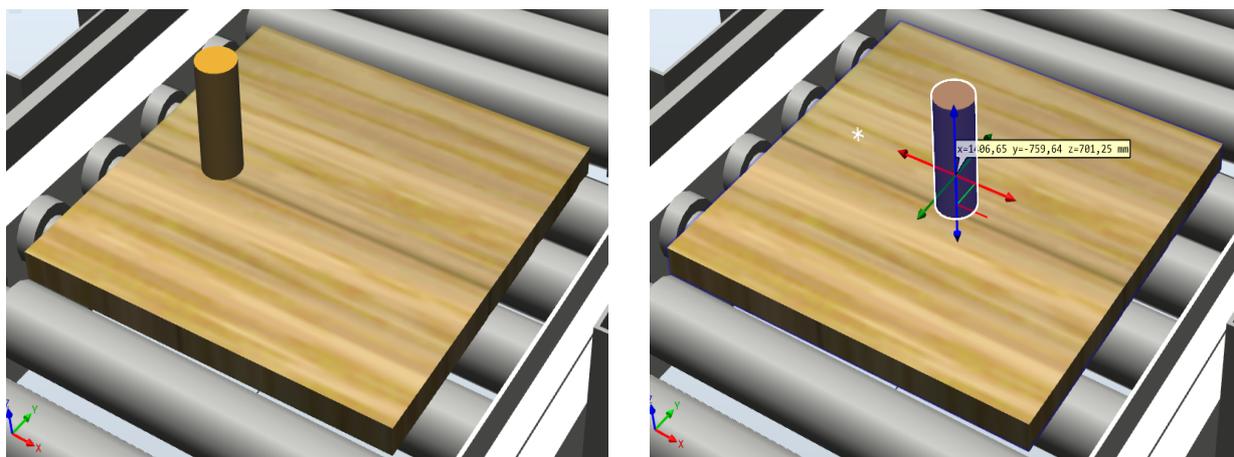


Figura 58. Como posicionar um objeto no RobotStudio® utilizando o TCP.
Fonte: O autor.

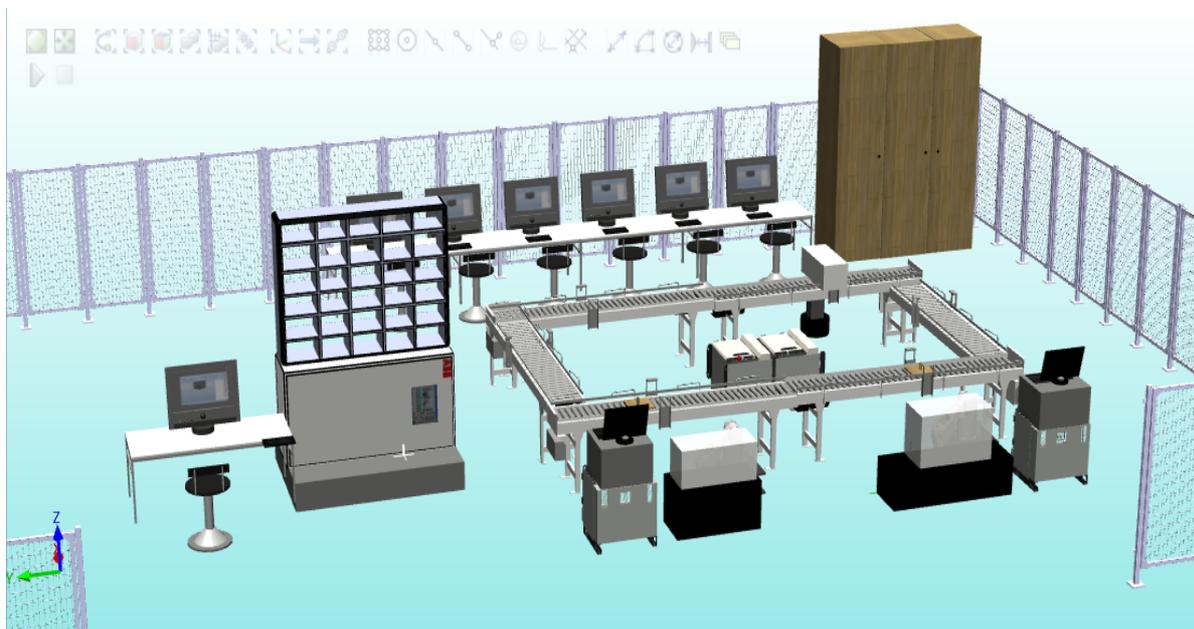


Figura 59. Vista panorâmica do *layout* básico do laboratório finalizado.

Fonte: O autor.

Com base na metodologia utilizada até a presente explanação, finalizou-se a projeção virtual de *layout* físico do laboratório de automação A-007 do Departamento de Mecânica da UTFPR (figura 59) – com exceção do mecanismo automático do Sistema de Armazenamento, pelo fato de não haver uma representação semelhante disponível na biblioteca da plataforma; ficando esta parte para ser analisada logo que a implementação dos robôs tenha sido alcançada, já que estes últimos são o foco principal de estudo deste trabalho.

4.2 IMPLEMENTAÇÃO DE MANIPULADORES ROBÓTICOS

Uma vez que o *layout* da mini-fábrica foi preparado, partiu-se para o tratamento da parte robótica do sistema, a começar pela averiguação e análise dos modelos de manipuladores robóticos disponíveis na biblioteca do *software*.

4.2.1 ESCOLHA DOS MODELOS

Na TAB *Home*, faz-se visível a aba **ABB Library** (figura 60), na qual é possível visualizar todos os modelos robóticos para utilização na plataforma.

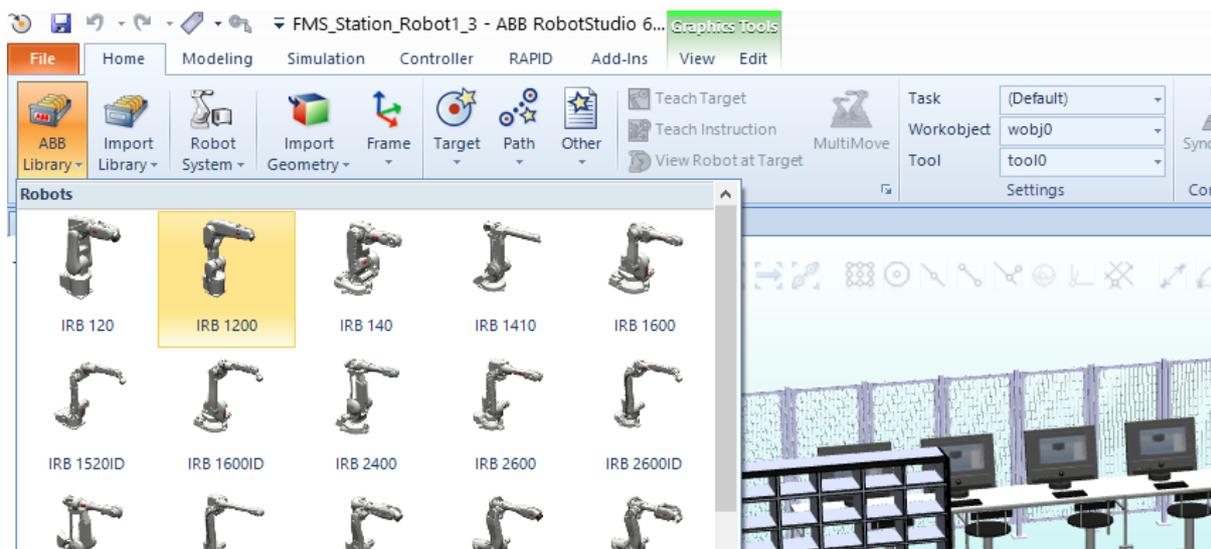


Figura 60. Biblioteca de modelos robóticos dos manipuladores da ABB™.
Fonte: O autor.

Dentre todos os modelos disponíveis, o robô que mais se assemelhou às características físicas e estruturais dos manipuladores RV-M1 da Mitsubishi com 5 (cinco) graus de liberdade já existentes na mini fábrica, foi o modelo **IRB 1200 da ABB™** (figura 61), pois apesar de possuir um grau de liberdade a mais, atenderia aos requisitos básicos para a sua funcionalidade do espaço físico do laboratório, levando-se em consideração o seu tamanho e peso, bem como o seu alcance total.

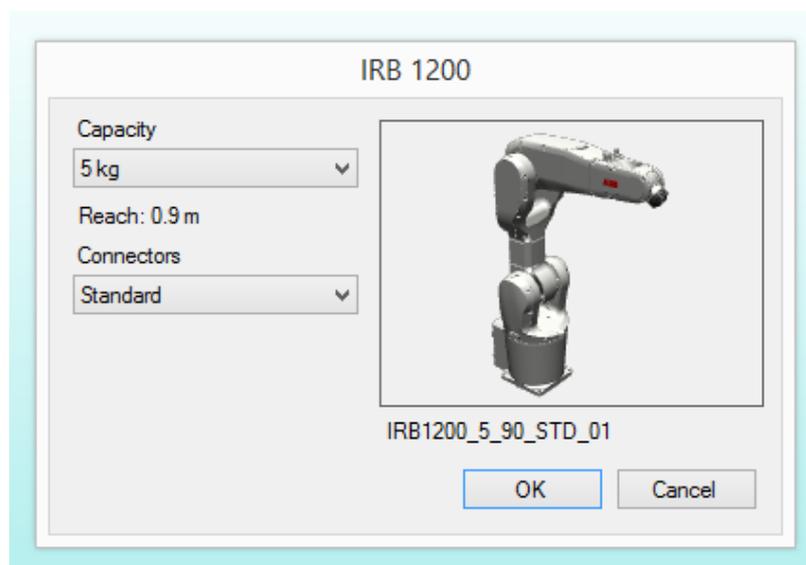


Figura 61. Modelo IRB 1200 – manipulador robótico da ABB™.
Fonte: O autor.

O IRB 1200, pesando 5 kilogramas (kg), tem alcance máximo de 0,9 metros (m) e possui 6 (seis) graus de liberdade, conforme exposto nas figuras a seguir.

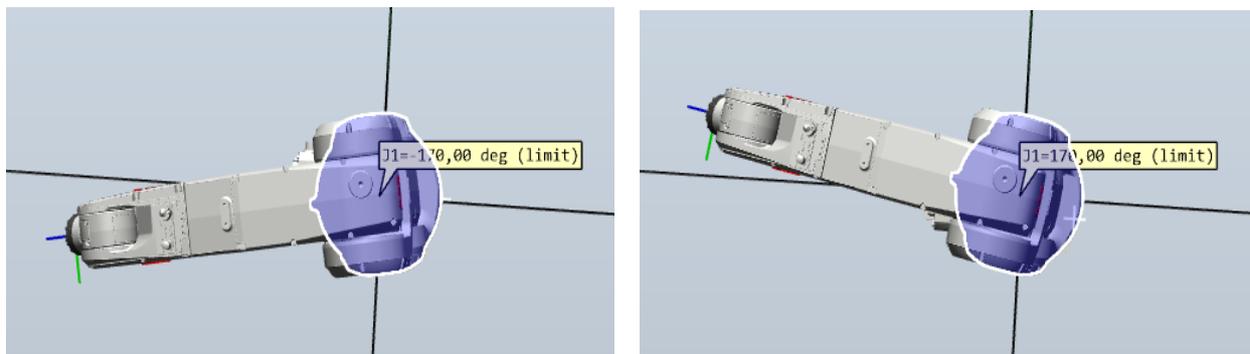


Figura 62. Modelo IRB 1200 – Junta 1 com 340 graus de flexibilidade. Fonte: O autor.

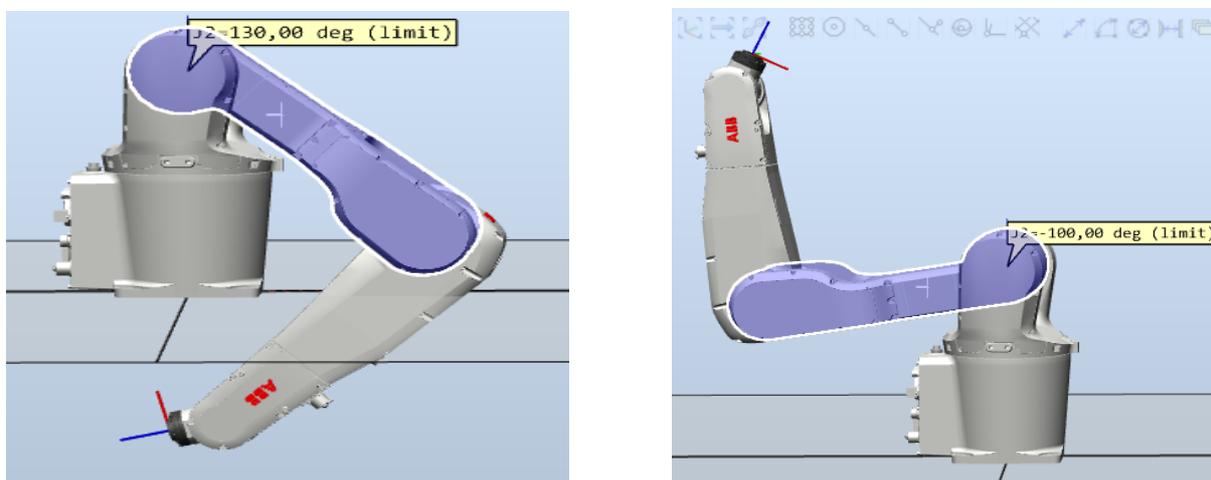


Figura 63. Modelo IRB 1200 – Junta 2 com 230 graus de flexibilidade. Fonte: O autor.

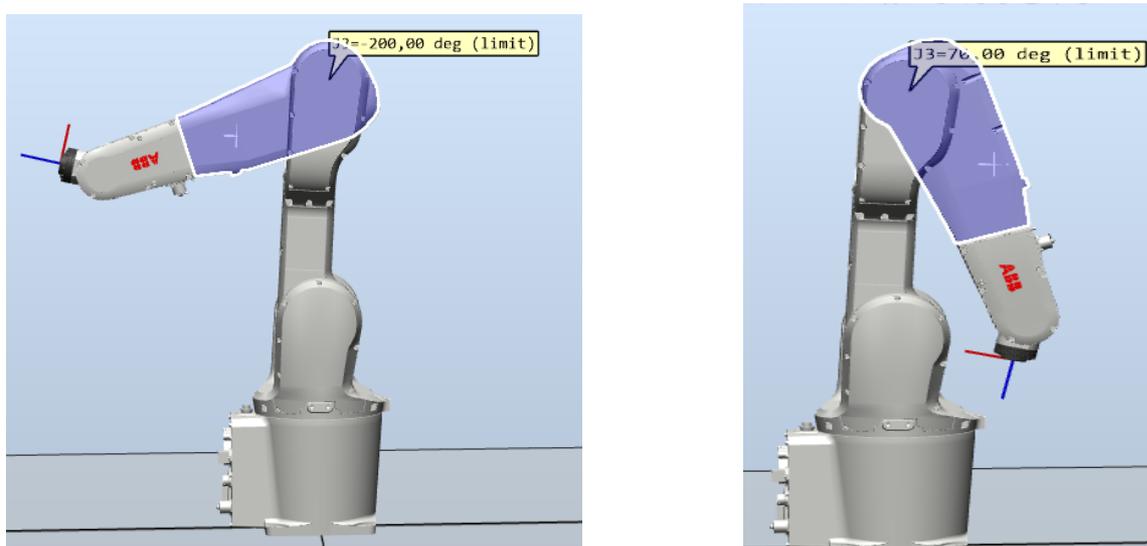


Figura 64. Modelo IRB 1200 – Junta 3 com 276 graus de flexibilidade. Fonte: O autor.

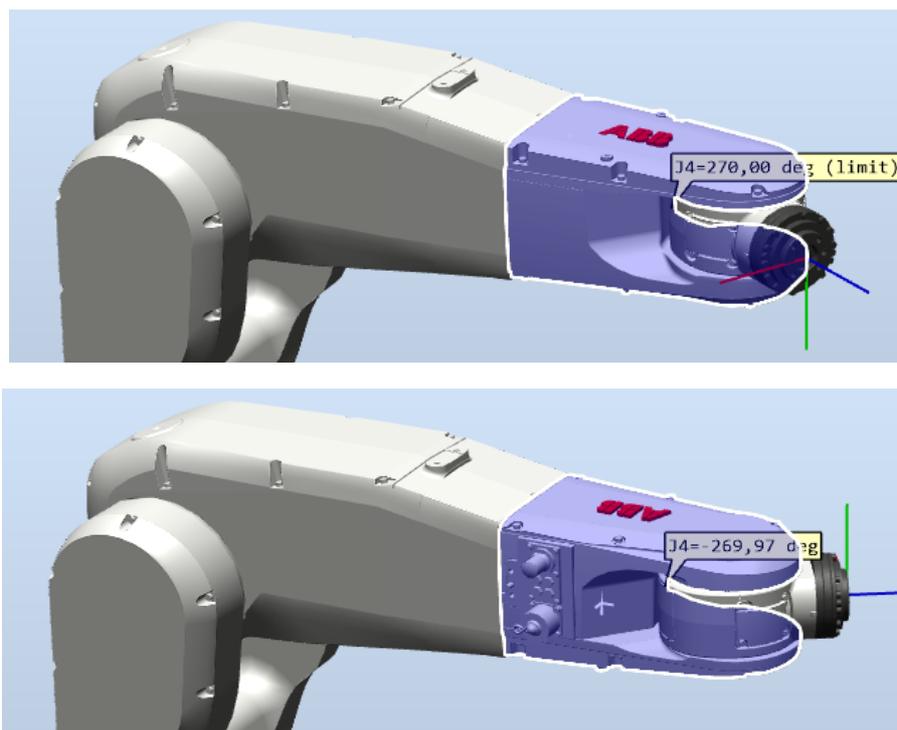


Figura 65. Modelo IRB 1200 – Junta 4. Fonte: O autor.

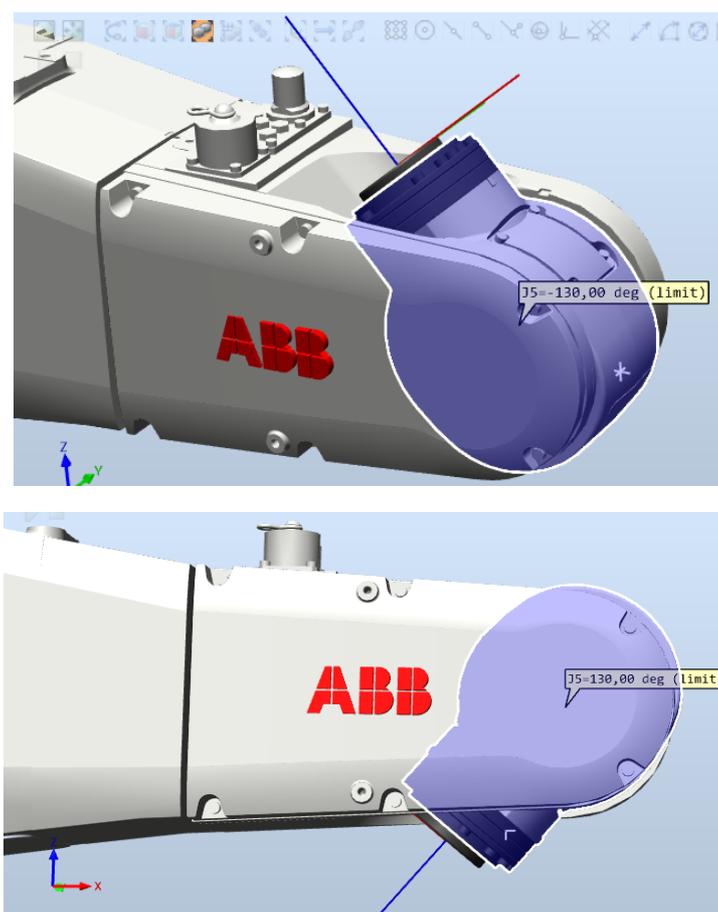


Figura 66. Modelo IRB 1200 – Junta 5 com 260 graus de flexibilidade. Fonte: O autor.

Para melhor visualização do último eixo rotativo (punho), realiza-se a fixação do efetuador. Neste trabalho, fez-se o uso de um “*Smart Component*” disponível na biblioteca (figura 67) que representa a garra do robô, mostrado na figura 68.

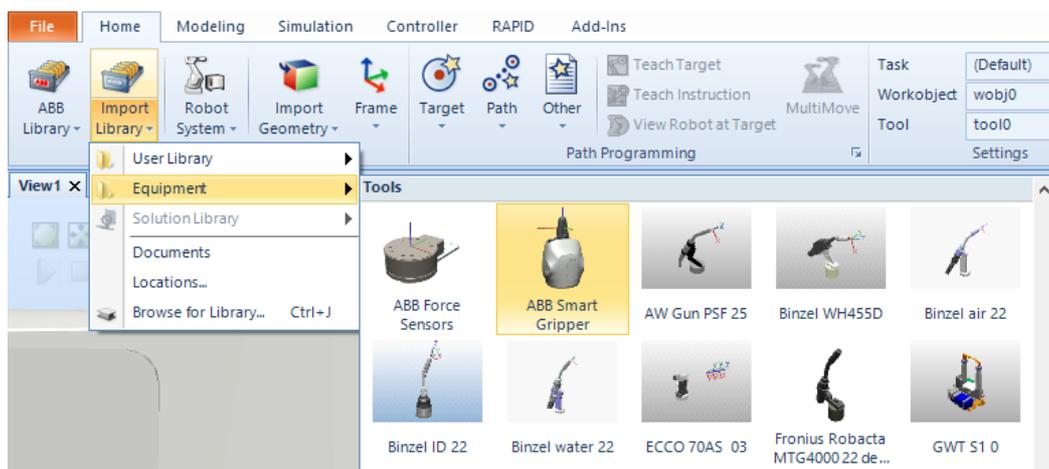


Figura 67. Escolha do elemento final do robô na biblioteca do RobotStudio®.
Fonte: O autor.

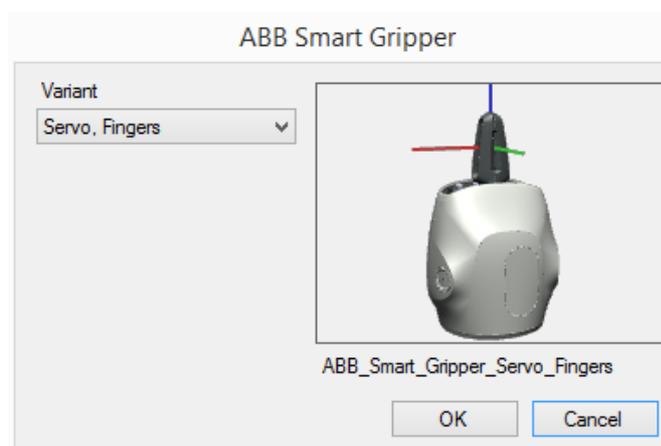


Figura 68. *Smart Gripper* – elemento final do robô na biblioteca do RobotStudio®.
Fonte: O autor.

Para realizar a fixação da garra na extremidade final do robô, clica-se com o botão direito sobre o nome listado à esquerda da tela (figura 69), escolhendo a opção “Attach To” (“fixar à\ao”, em tradução livre), e clicando sobre o nome do robô que se torna visível.

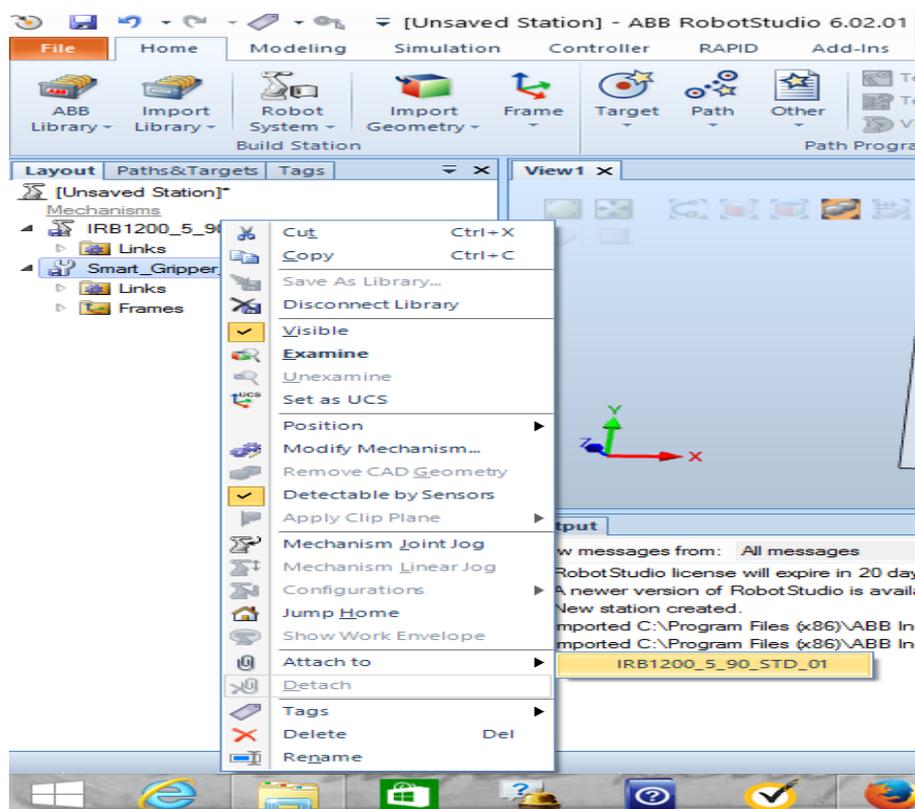


Figura 69. Fixação da garra à extremidade do robô. Fonte: O autor.

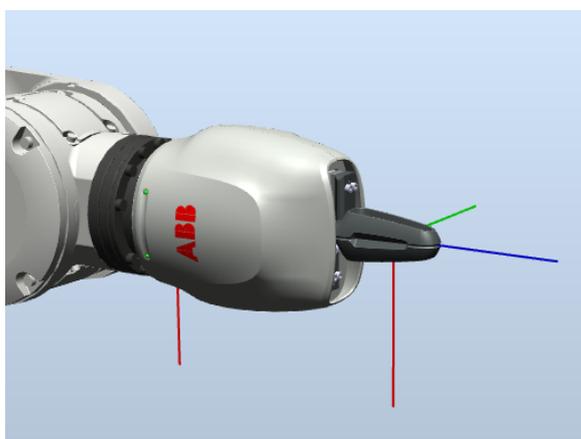


Figura 70. Garra fixada à extremidade do robô. Fonte: O autor.

Observa-se que este elemento foi acoplado ao robô como na figura 70 e, portanto, não é possível realizar a movimentação livre da última junta por meio dele,

sendo necessário realizar a rotação através do TCP rotativo avistado logo que haja o clique sobre a garra, como mostra a figura 71.

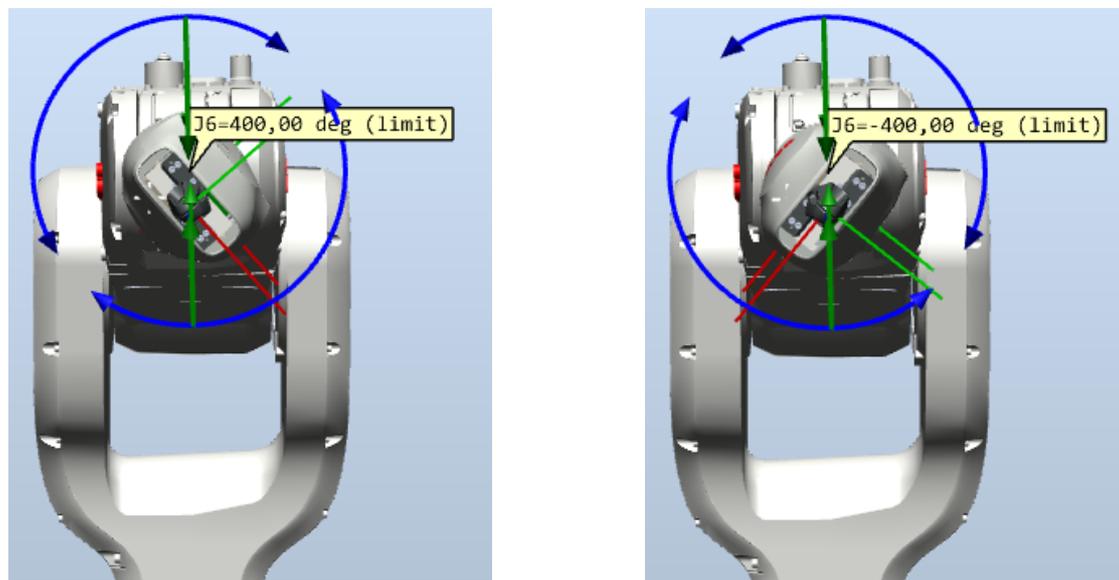


Figura 71. Modelo IRB 1200 – Junta 6 com 800 graus de flexibilidade.
Fonte: O autor.

Com base no que foi observado, leva-se em consideração que a escolha do modelo IRB 1200 possui alcance e flexibilidade suficiente para, teoricamente, superar as expectativas quanto ao seu desempenho nas atividades normalmente realizadas pelo atual robô RV-M1 da Mitsubishi – que possui opções consideravelmente menores de movimentação.

4.2.2 POSICIONAMENTO DOS ROBÔS

O posicionamento dos manipuladores robóticos deu-se após o *layout* de fábrica ter sido preparado, considerando-se que o mesmo deveria ser o mais semelhante possível com a realidade física.

Buscou-se portanto, usando do método de posicionamento de objetos pela TCP dos robôs, posicionar os mesmos de forma que ficassem localizados à distância mais curta possível entre o *pallet* da esteira e o maquinário CNC a que destinam-se realizar os serviços de carregamento e descarregamento, conforme mostram as figuras seguintes.

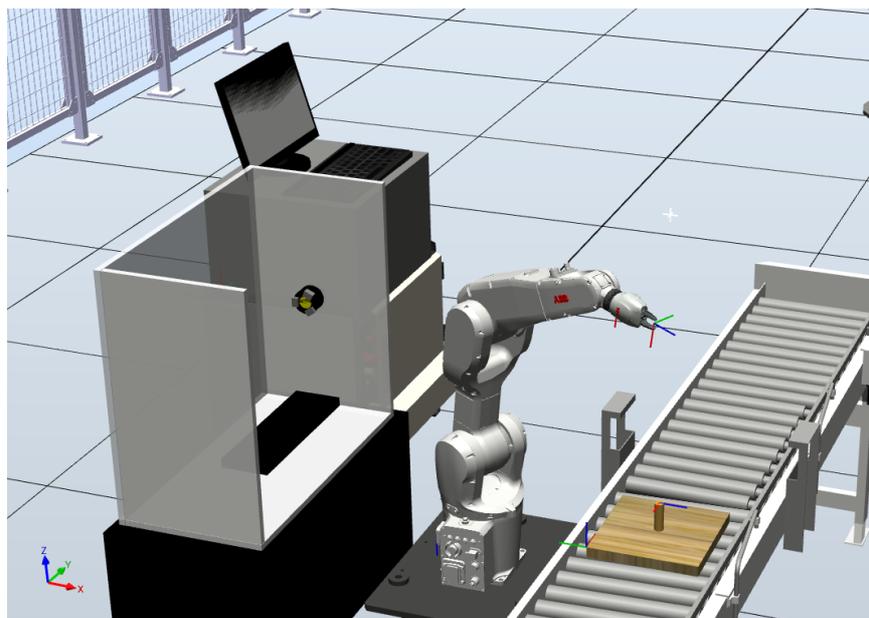


Figura 72. Posicionamento do robô servente ao Centro de Torneamento.
Fonte: O autor.

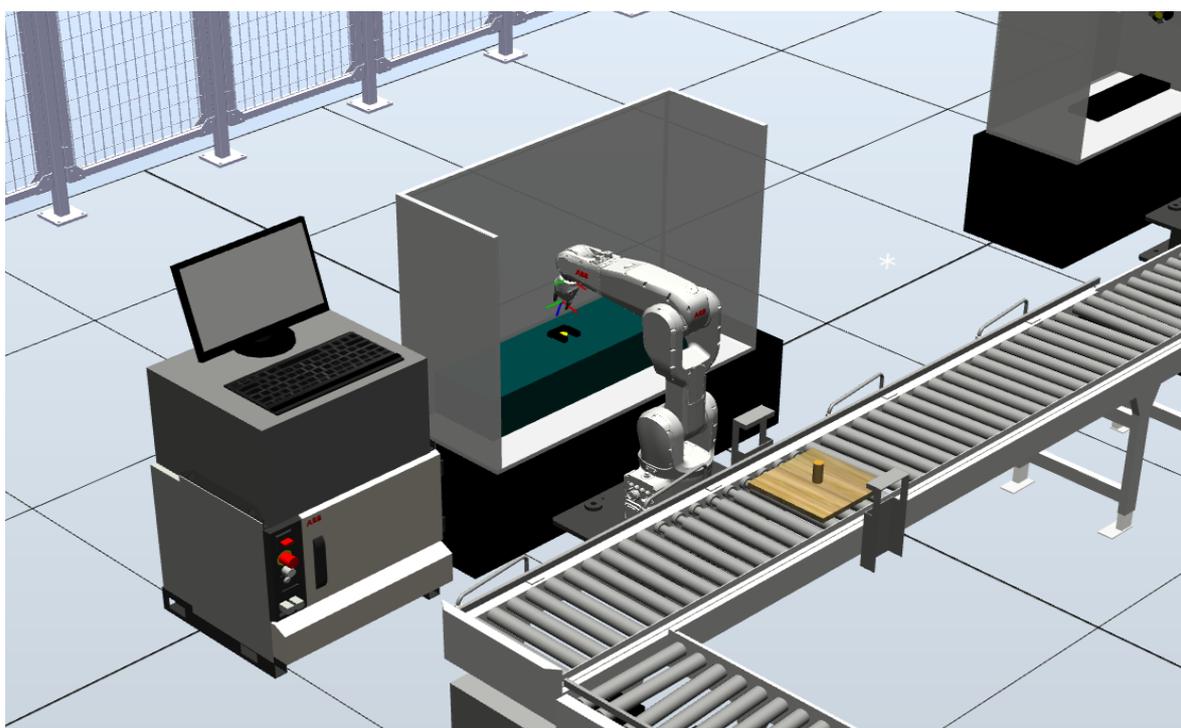


Figura 73. Posicionamento do robô servente ao Centro de Usinagem (Fresadora).
Fonte: O autor.

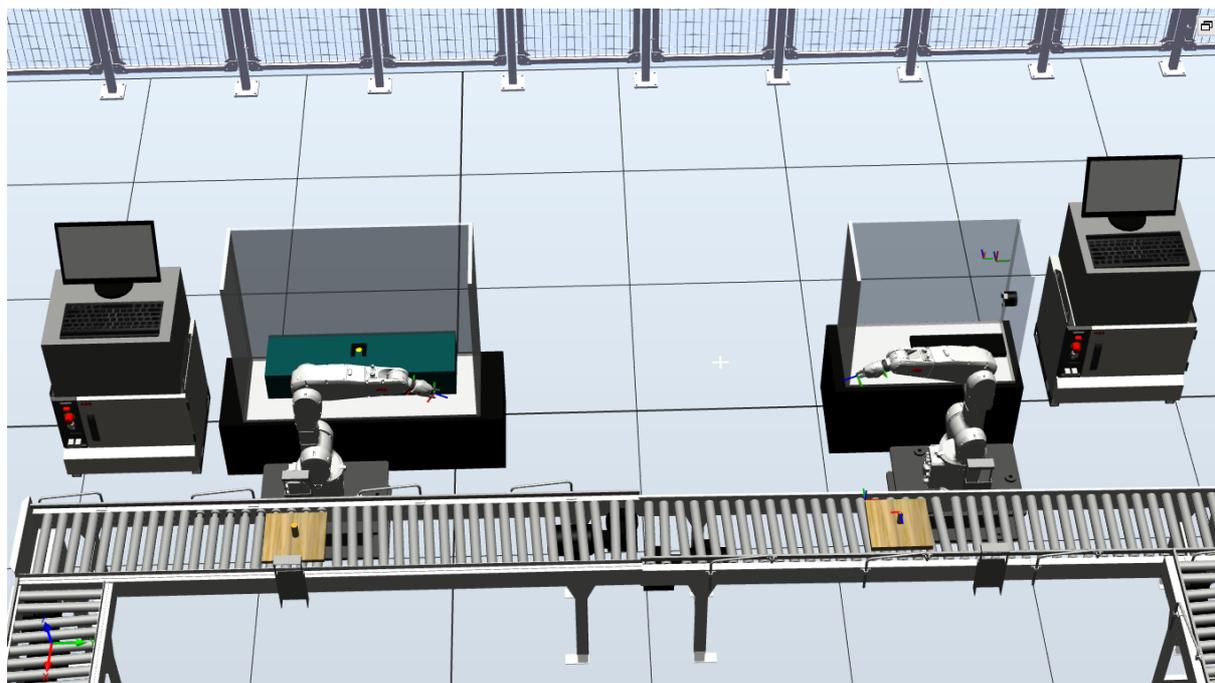


Figura 74. Vista da estação de trabalho após o posicionamento dos manipuladores robóticos.
Fonte: O autor.

O próximo passo no desenvolvimento virtual do FMS seria integrar os modelos virtuais de manipuladores robóticos ao restante do sistema. Para tal, faz-se necessário, primeiramente, determinar as ações e movimentação específica dos robôs no ambiente virtual – etapa esta na qual se manifestaram os primeiros obstáculos de projeção no RobotStudio®. O advento dessas ocorrências dificultou o prosseguimento do plano de execução de implementação, resultando em respostas indesejadas no *software*. Esses resultados serão discutidos no próximo capítulo.

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Neste capítulo estão expostos os resultados obtidos até o presente momento com a metodologia abordada, bem como a elucidação dos motivos para tais resultados, e as possibilidades hipotéticas que poderiam ser exploradas com outras funcionalidades.

5.1 GERAÇÃO DE MOVIMENTAÇÃO E TRAJETÓRIA

Tendo em vista que o objetivo principal do *software* RobotStudio® é o de simular um comportamento, próximo ao idêntico da realidade, dos robôs da ABB™ em um ambiente fabril, utilizam-se largamente seus recursos de projeção de trajetória para gerar movimentação nas juntas dos robôs e ações dos elementos finais dos mesmos, cujos procedimentos são, inclusive, divulgados em forma de tutoriais audiovisuais na internet. De acordo com esses tutoriais, assim como com o manual de instruções disponibilizado pelo RobotStudio®, este procedimento dá-se em cinco (5) etapas, para uma simples ação como a de o manipulador **buscar uma peça, e depositá-la em outro lugar**, por exemplo:

- Criação de *Workobjects*, ou “objetos de serviço”;
- Criação de *Targets*, ou “pontos alvos”;
- Procedimento de geração das trajetórias;
- Procedimento de Movimentação dos Robôs;
- Gerar simulação.

De acordo com o manual de instruções - ***Operating manual RobotStudio®*** (ABB, 2008 - 2010), ou “Manual de Operações RobotStudio®”, em tradução livre - no capítulo 3 - *How to Program Robots*, seção 3.1 - *Workflow for programming Robot* (“Como Programar Robôs”, e “Fluxo de Trabalho para Programação de Robôs”, respectivamente, em tradução livre), determina-se como pré-requisito que “*antes de criar um programa para o seu robô, você deve preparar a*

estação de trabalho, incluindo robôs, objetos e peças de trabalho, e dispositivos elétricos com os quais o robô irá trabalhar” [ABB: Operating Manual, pg. 91].

O manual recomenda, ainda, que o fluxo de execução das tarefas para programar o robô se dá da seguinte forma:

a) Criar **targets** (“alvos”) nas posições necessárias e então gerar **paths** (“trajetórias”), inserindo os *targets* nestes últimos;

b) Verificar a orientação dos *targets*, tendo a certeza de que a orientação dos mesmos encontram-se da forma mais eficiente possível para que as atividades sejam executadas;

c) Verificar a acessibilidade do robô e da ferramenta, se os mesmos conseguem alcançar todos o *targets* da trajetória;

d) Sincronizar o programa gerado ao VC, ou *Virtual Controller* (“Controlador Virtual”), gerando automaticamente o código RAPID proveniente do RobotStudio® que habilita o programa de ser simulado;

e) Realizar edição das instruções de texto ou dados gerados pelo RobotStudio®, utilizando o *Program Editor* (“Editor de Programa”), caso necessário;

f) Detectar Colisões, verificando se o robô ou a ferramenta dele não colide com os equipamentos ao seu redor, caso contrário, ajustar o posicionamento e as orientações até que não ocorram mais colisões;

g) Finalmente, testar o programa, realização movimentação ao longo das trajetórias.

Todos os requisitos supracitados são seguidos com links direcionadores contendo instruções específicas para cada tarefa. No entanto, a estrutura das orientações não segue uma ordem contínua, mas sim abrangendo da forma mais objetiva possível, em tópicos, variadas possibilidades de execução da mesma tarefa, de acordo com a possível situação do projeto e necessidade do programador.

Seguindo-se a orientação dos requisitos, tratou-se primeiramente da criação de um *workobject* para que fosse então realizada a criação de *targets*, conforme a sequência dos tópicos seguintes – nos quais se mostram as tomadas dessas ações bem como as respostas do *software* ao proceder com elas.

5.1.1 CRIAÇÃO DE *WORKOBJECTS* (“Objetos de Trabalho”)

Um *workobject*, de acordo com definição do próprio manual de operações, é nada mais do que um sistema de coordenadas utilizado para descrever a posição de uma peça de trabalho. O *workobject* consiste de dois (2) *frames*: um *frame* do usuário, e um *frame* do objeto. Todas as posições a serem programadas serão relacionadas ao *frame* do objeto (figura 75), o qual é relativo ao *frame* do usuário que, por sua vez, é relativo ao sistema de coordenadas global (ABB, 2008, Operating Manual, p. 92). Percebe-se que o sistema de coordenadas global (*world coordinate system*) é o responsável pelo posicionamento geral ao longo do ambiente virtual como um todo.

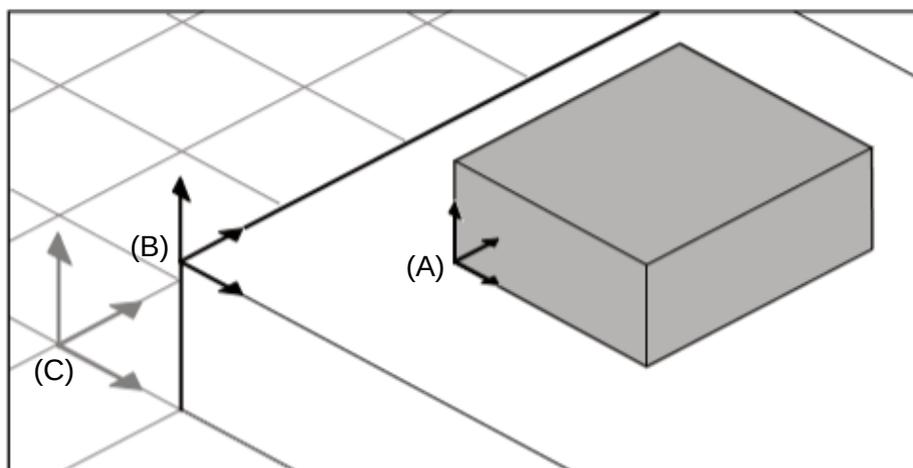


Figura 75. Sistemas de coordenadas em um *workobject*. A) *Frame* do objeto. B) *Frame* do usuário. C) *Frame* global. Fonte: ABB *Operating Manual RobotStudio® 5.13* (manual de operações).

Ao dar-se início à criação do *workobject*, recorre-se ao botão “*Other*” na aba *Home*, e escolhe-se a opção “*Create Workobject*”, conforme mostra a figura 76.

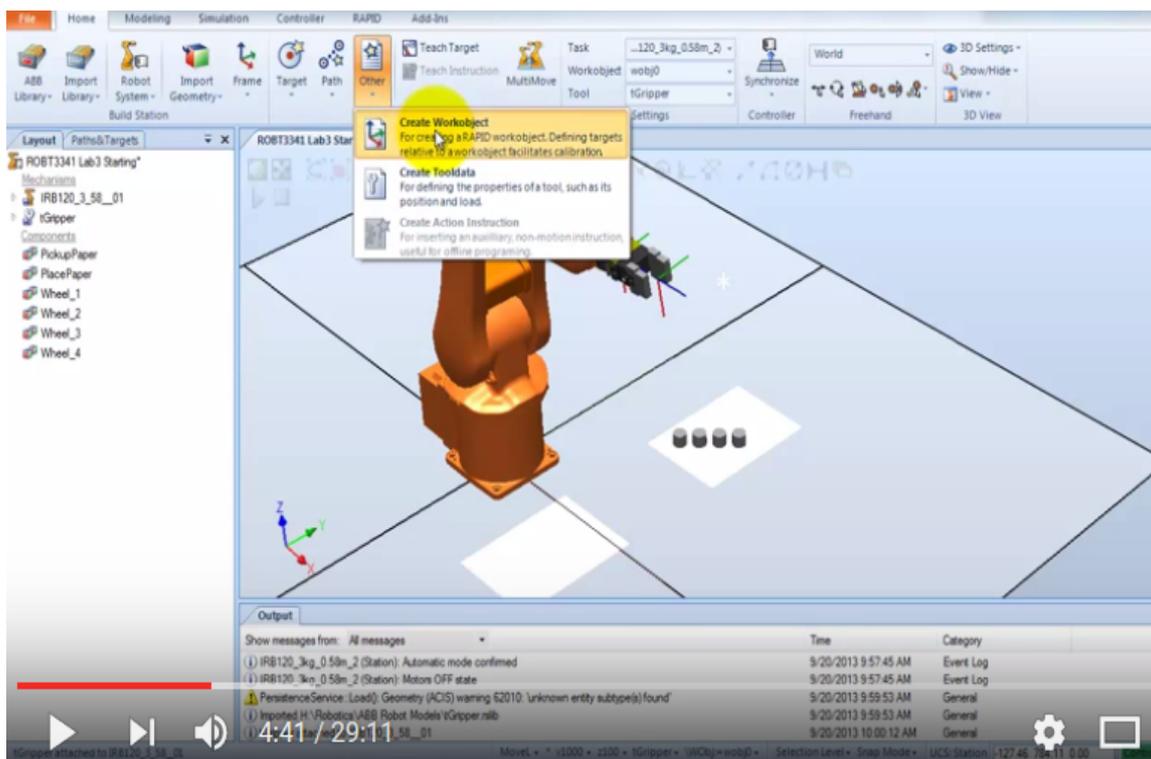


Figura 76. Procedimento de criação de um *workobject*.
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3.

Na janela que se abre, dá-se nome ao *workobject*, e especifica-se a sua localização (figura 77), utilizando-se dos recursos de botões localizados na tela principal de trabalho para garantir precisão de localização.

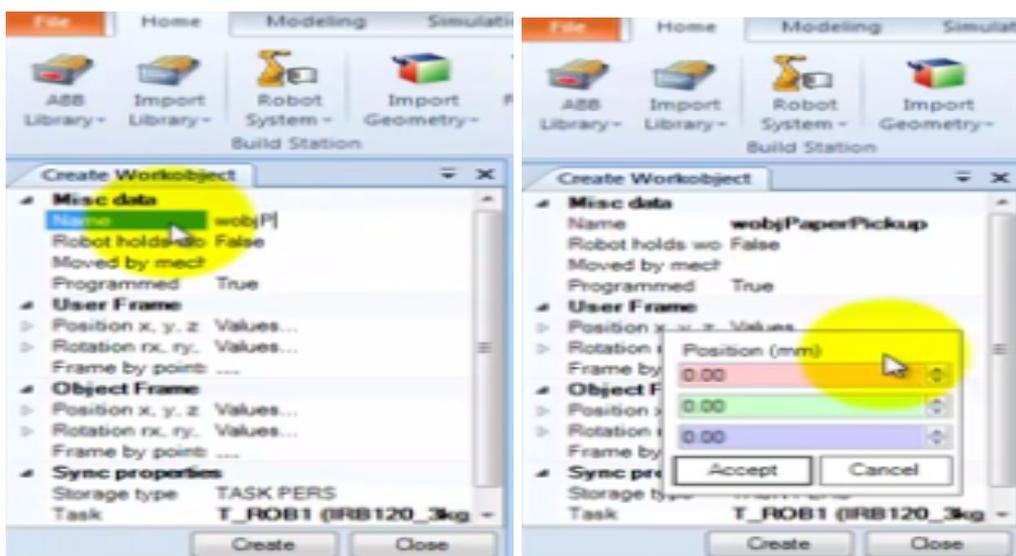


Figura 77. Renomear e alterar valores de *User Frame* (*Frame do Usuário*).
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Como os valores de X, Y e Z são desconhecidos, seleciona-se as opções de botões **Surface Selection** (porque a área de trabalho é uma superfície neste caso) e **Snap End** (porque deseja-se localizar o frame na extremidade do papel), e em seguida basta aproximar o mouse do final da superfície de trabalho, e verificar os valores das coordenadas localizadas na janela de propriedades do *workobject*. Clicar em **Accept** e **Create**, conforme figura 78.

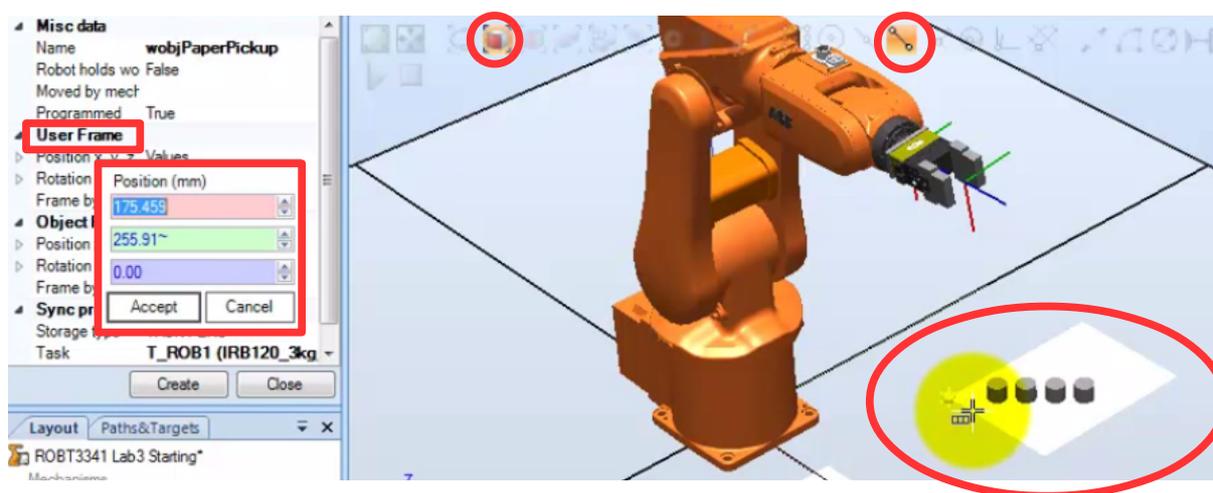


Figura 78. Aquisição das coordenadas do *User Frame* (Frame do Usuário).
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Este mesmo procedimento, para localização da área de busca das peças, deve se repetir também para o local de depósito das peças, seguindo o mesmo procedimento supracitado, tomando o cuidado de definir uma nomeação diferente para o novo *workobject*.

Terminado o segundo procedimento com criação de *workobjects*, pode-se então criar o primeiro *target*, para busca de peça, sendo criado diretamente sobre ela.

5.1.2 CRIAÇÃO DE TARGETS (“Pontos Alvos”)

Ainda na aba *Home*, abre-se o botão “*Target*”, “*Create Target*” (figura 59), e logo que a janela de propriedades se mostra visível, seleciona-se novamente na tela principal a opção **Surface Selection** e, desta vez, **Snap Center** já que a face superior da peça é circular e deseja-se saber a localização exata de seu centro.

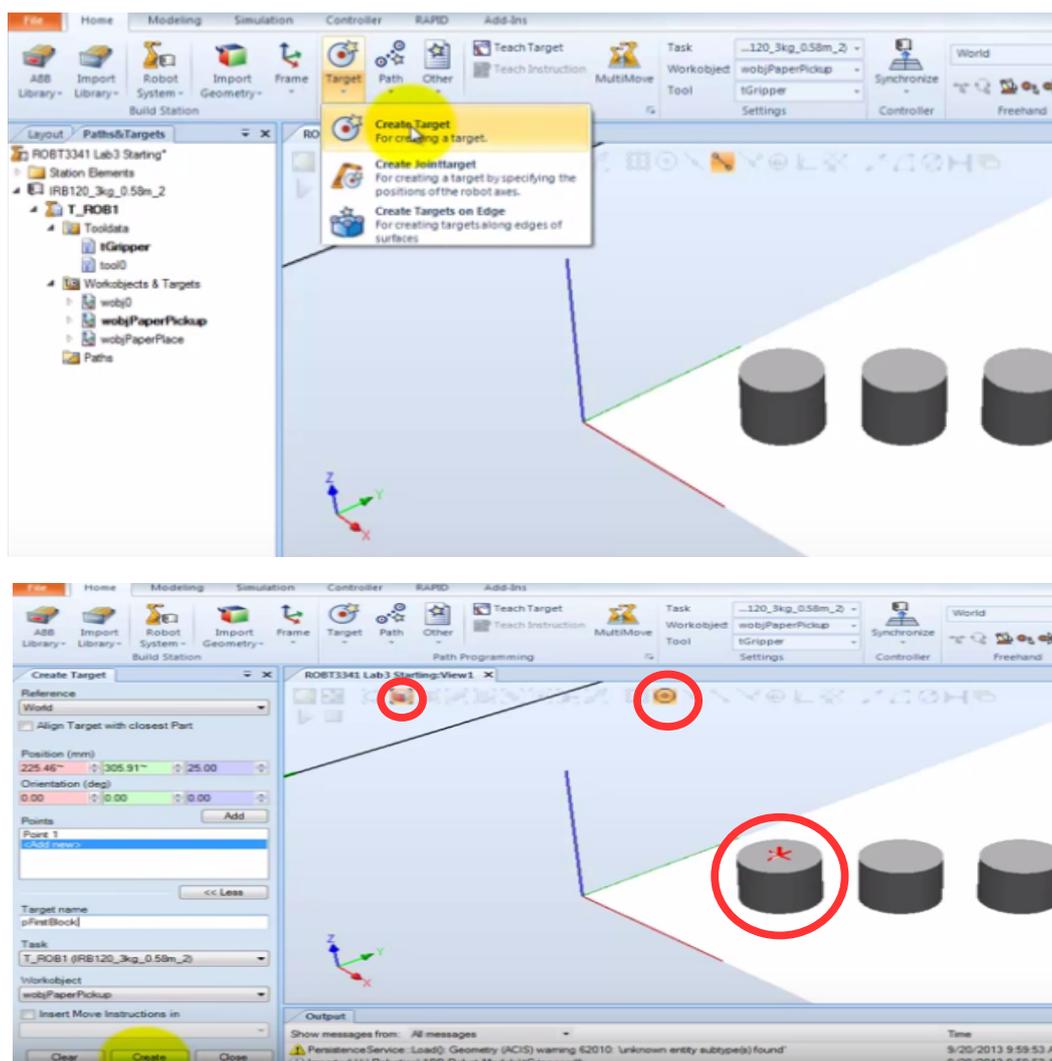


Figura 79. Aquisição das coordenadas do Object Frame (Centro da Peça).
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Mais uma vez, clica-se em cima da superfície da peça e, obtidos os valores na janela de propriedades, renomear o *target* e clicar em *Create*.

Após determinação da localização da peça, os *frames* se dispõem conforme o que mostra a figura 80.

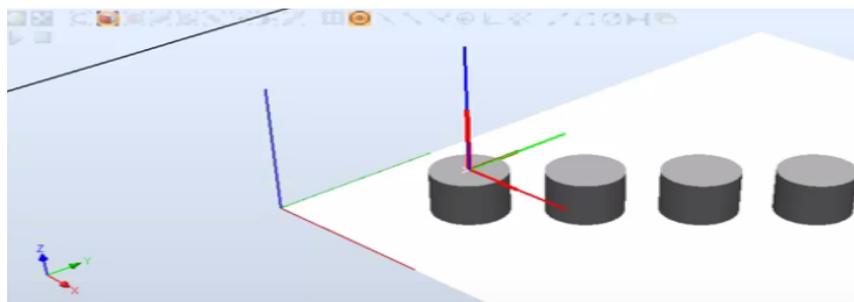


Figura 80. Frames do *workobject*.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Seguindo-se essas orientações, o *software* apresentou o mesmo comportamento até a presente parte. A partir deste ponto, as instruções feitas pelo tutorial não surtiu o mesmo efeito na prática. Tendo em vista que, para que a peça possa ser pega pelo robô, a orientação do *frame* do objeto deve ser a mesma que a orientação da ferramenta do robô – caso contrário, deve-se modificar a orientação do *frame* do *target* na peça até que esta se alinhe com o TCP da ferramenta do robô, conforme orientação do tutorial e mostrado, na prática, nas figuras abaixo, comparativamente. Lembrando que neste ponto, a extremidade da ferramenta ficará perfeitamente alinhada com a face superior da peça, mas será apenas capaz de pegá-la se o posicionamento do TCP da ferramenta for decrementado em “Z”, para que a peça fique entre as garras da ferramenta. Caso as instruções sejam seguidas corretamente, o robô deveria ser capaz de alcançar a peça e buscá-la. Esta possibilidade pode ser verificada clicando-se com o botão direito sobre o nome do *target* listado à esquerda da tela, e selecionando-se “*configurations*”, sendo possível observar os valores de coordenadas necessários para que a ferramenta do robô alcance dada posição. É com esses valores que se realiza a programação da movimentação do robô.

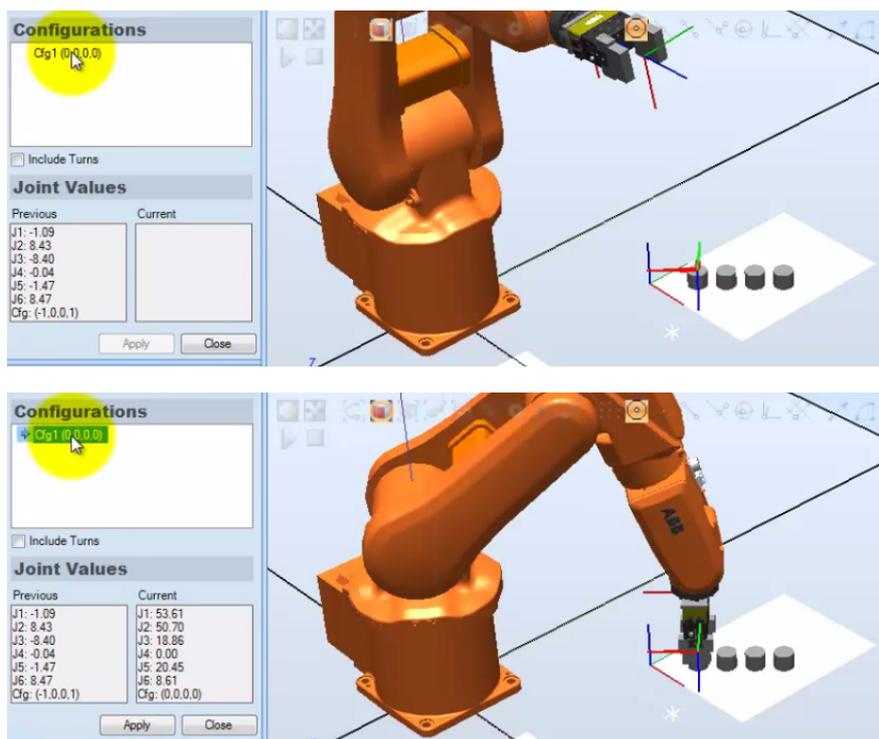


Figura 81. Disponibilização automática das coordenadas da posição de busca ao alcance do robô. BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Na prática, utilizando-se o mesmo procedimento na mini fábrica virtual construída, não houve, no entanto, registro de posições alcançáveis pelo robô, além do recebimento constante de erro, com a justificativa de que não haveria posição válida de alcance para a ferramenta, apesar de teoricamente o robô ser capaz de alcançar dada posição, não havendo obstáculos entre o robô e a peça para possíveis colisões. Também foi realizado o experimento de manipular os eixos do robô manualmente através do botão **Free Hand** na tab **Home**, e pôde-se observar que o robô seria perfeitamente capaz de alcançar a posição, inclusive com a garra invertida, conforme mostrado nas figuras a seguir.

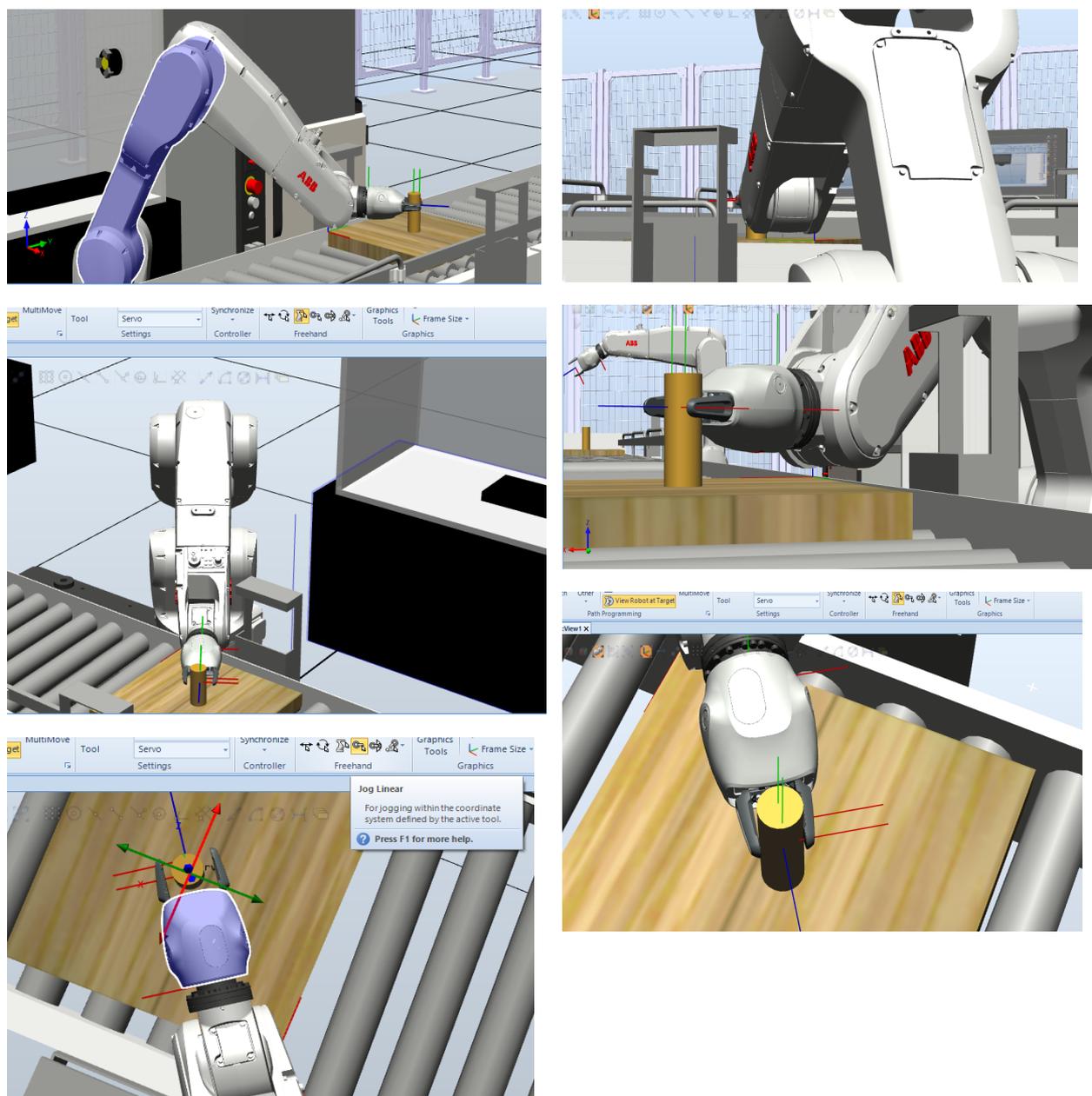


Figura 82. Teste de alcance (*Reachability*).
 Fonte: O autor

Mesmo após a realização do teste mencionado acima, o sistema permaneceu com a acusação de que a posição seria inatingível pela ferramenta do robô. Testou-se também a aproximação do robô do *pallet*; o reposicionamento do *pallet* para fora das proximidades de qualquer possível obstáculo (como o gancho da esteira, visível nas figuras anteriores, por exemplo). Tais procedimentos foram submetidos a diversos testes experimentais, buscando-se soluções, porém sem sucesso.

A partir deste ponto, estudou-se a continuidade do processo, tomando-se como hipótese que o teste não houvesse falhado, de forma a verificar se haveria uma posterior solução ao longo do procedimento. Tomando-se como verdadeira a

funcionalidade do processo realizada até o momento, segue-se com a geração de trajetórias, a partir dos *targets* criados, para nível de informação.

5.1.3 GERAÇÃO DE TRAJETÓRIAS

Tendo sido selecionado um *target* válido para as configurações do robô, é possível gerar uma trajetória a partir dele. Clica-se com o botão direito e seleciona-se a opção **Add to new path** (“Adicionar à nova trajetória”). O nome dessa trajetória pode ser modificado.

Na criação da trajetória, deve-se criar posições intermediárias entre os *targets* inicial e final. Para isso, utiliza-se os recursos **Jog Joint** e **Jog Linear** para reposicionar a ferramenta do robô na posição intermediária. Alcançada a posição intermediária satisfatória, pode-se proceder com 3 ações diferentes: criar um novo *target* pra esta posição, ou escolher a opção “*Teach target*” ou “*Teach instruction*” na barra de ferramentas (figura 83). Para o tutorial escolheu-se **Teach Instruction**, e aceita-se a opção de modificar os valores do novo *workobject*. Observa-se que a trajetória se torna visível.

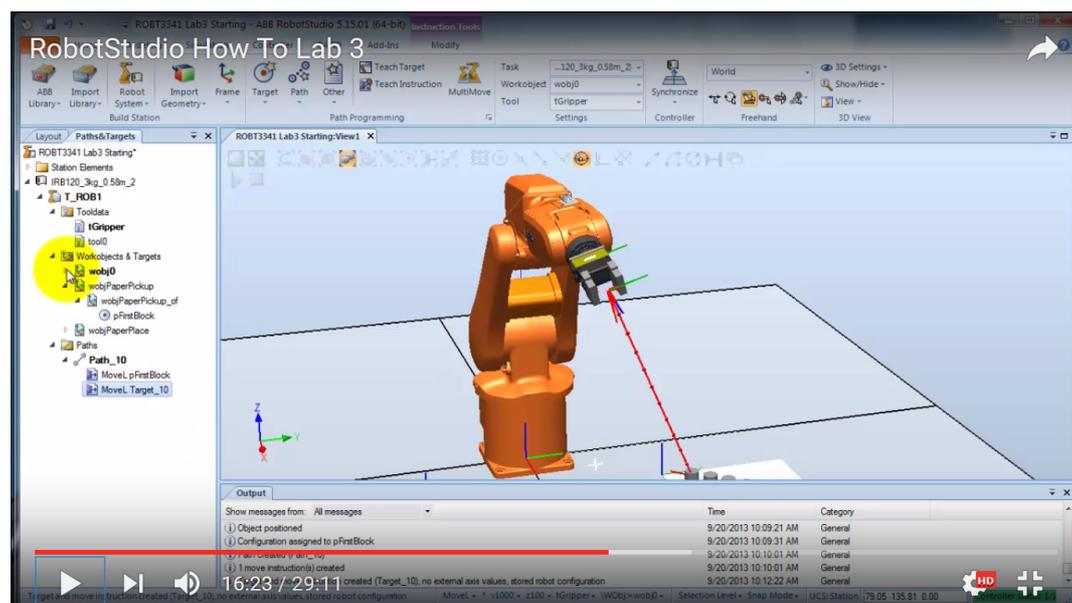
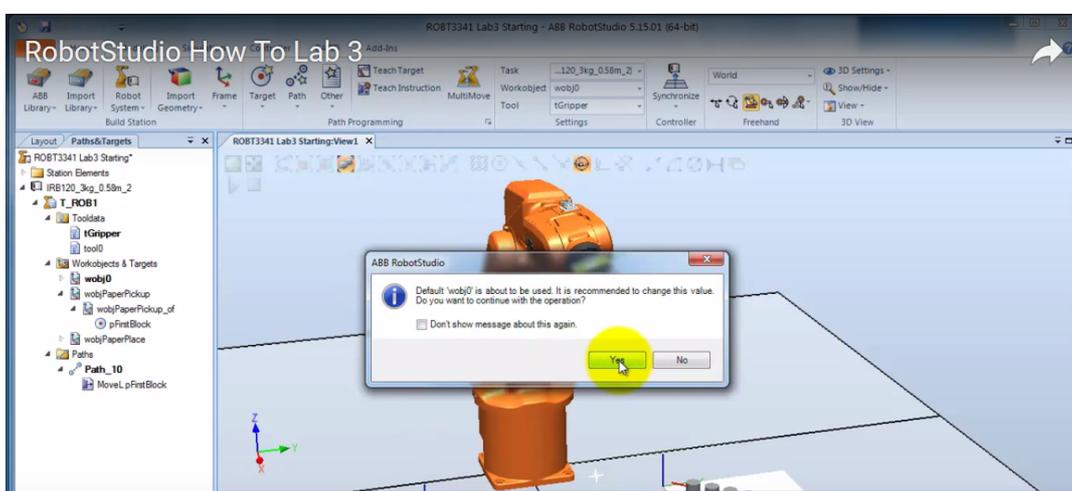
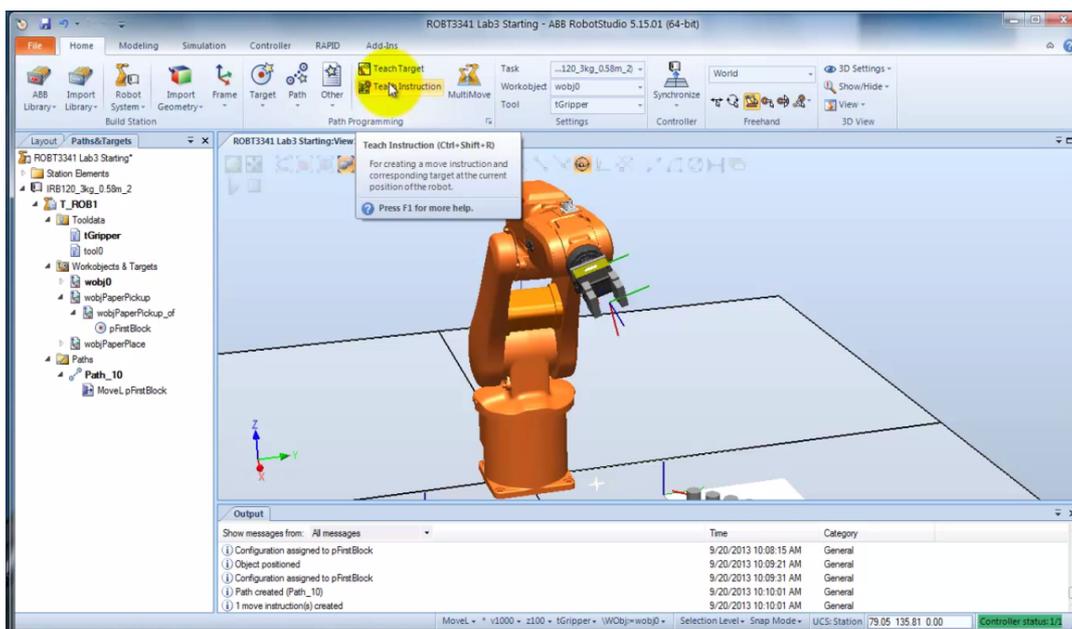


Figura 83. Geração de uma trajetória.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Com o clique do botão direito sobre a trajetória criada, pode-se modificar suas propriedades para que fique mais precisa e em conformidade com o seu objetivo, conforme mostra a figura 84.

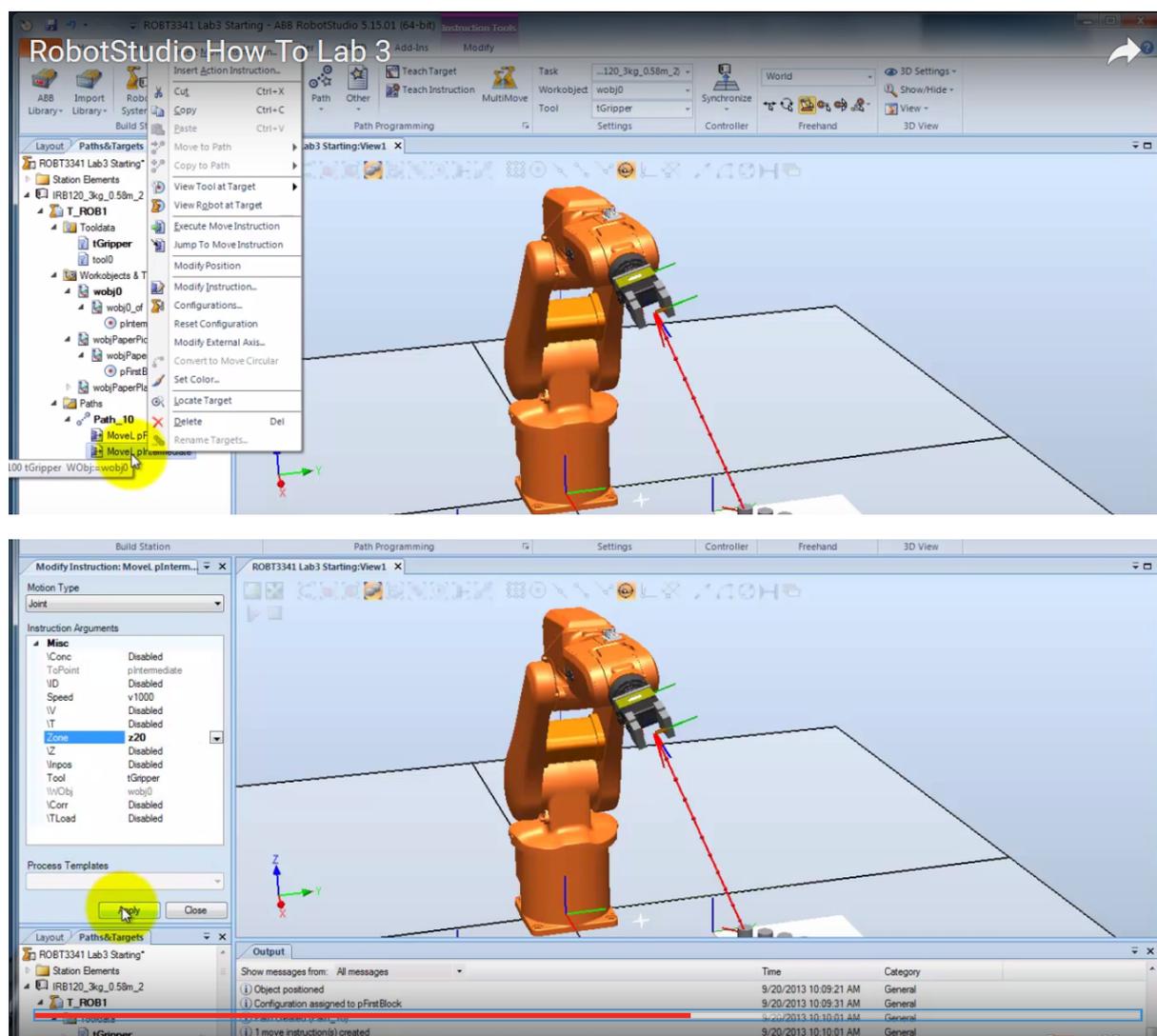


Figura 84. Edição das propriedades de uma trajetória criada.
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Uma vez que a trajetória foi gerada e configurada, é chegado o momento de configurar a movimentação ao longo dela através da programação da movimentação do robô.

5.1.4 MOVIMENTAÇÃO DOS ROBÔS

A movimentação do robô se dá na parte de programação RAPID, que pode ser realizada de diversas formas. Uma delas se encontra na aba *RAPID* da barra de ferramentas. A opção escolhida no entanto – seguindo o tutorial “**RobotStudio® How to Lab 3**” (BCIT, 2014) – foi entrar pela aba **Simulation** e escolher a opção **Simulation Setup**, conforme mostra a figura 85.

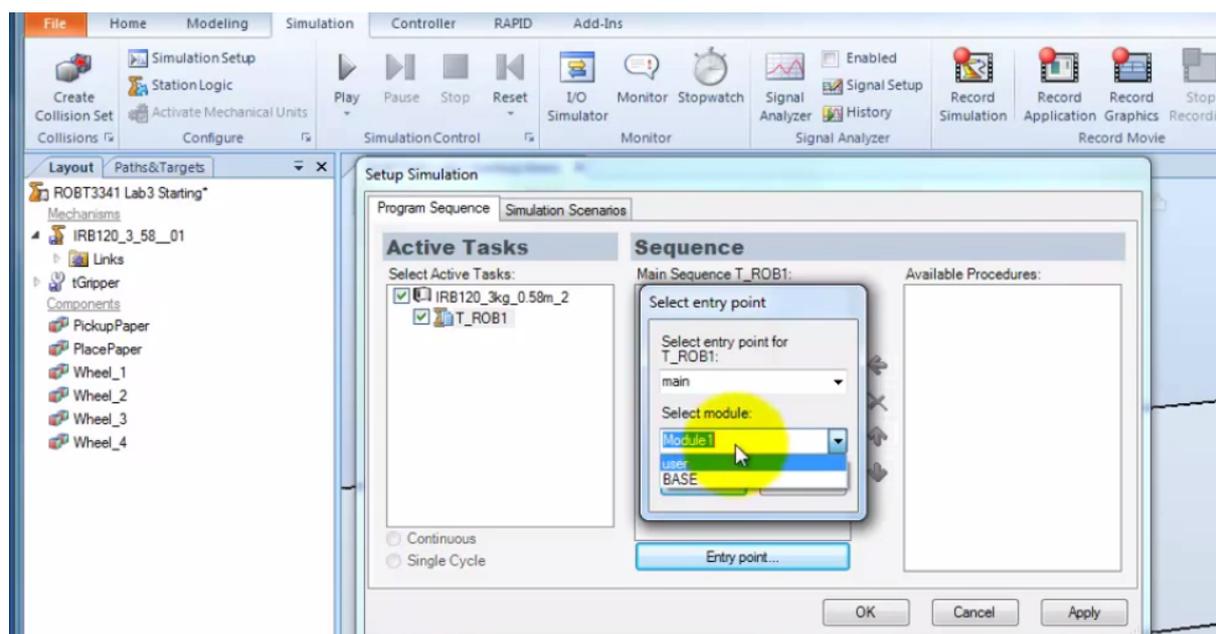


Figura 85. Criação de um módulo de programa.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Ao clicar em **OK**, verifica-se que o novo módulo de programa foi listado na aba *RAPID*. Clica-se duas vezes nesse módulo de programa e o que se vê é a estrutura vazia de um programa, preparado para edição, como na figura 86.

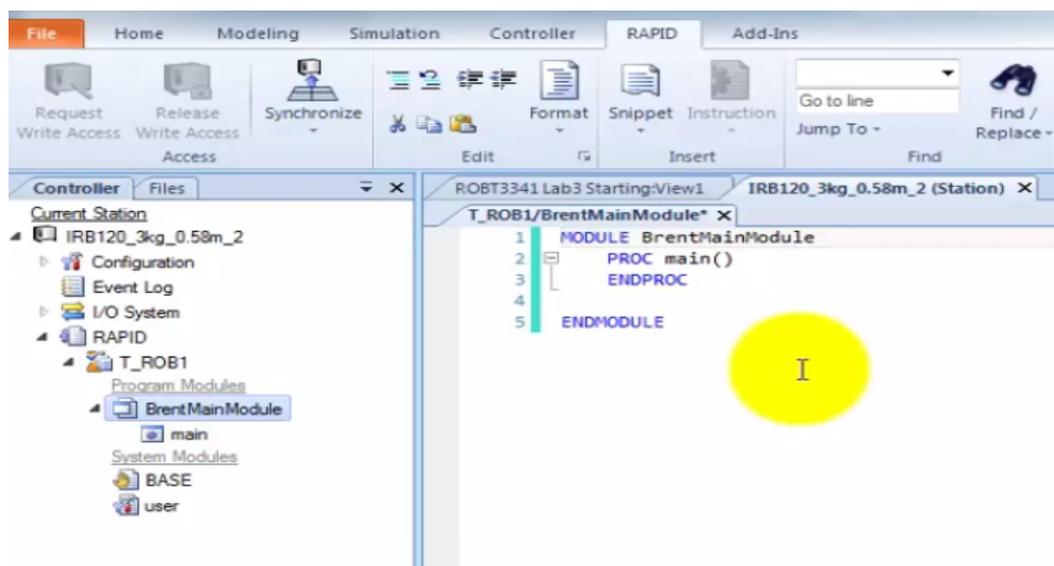


Figura 86. Edição do módulo de programa.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

A partir deste ponto, é realizada a conexão com o controlador virtual (VC). Nota-se que o sistema já possui um controlador virtual em execução vinculado a ele e que nesta parte é feita apenas a sincronização do sistema com ele, pois toda vez que o sistema sofre modificações – sejam físicas ou configurativas – é necessário realizar a sincronização com o VC, conforme mostra a figura 87. Assim, todas as opções escolhidas na janela do Controle Virtual será o nome do módulo de programa criado.

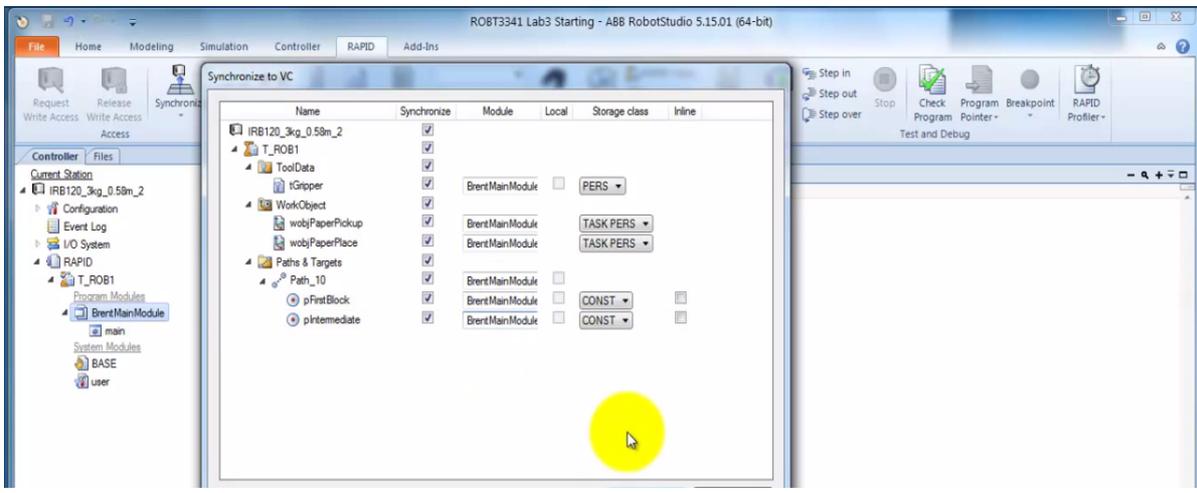
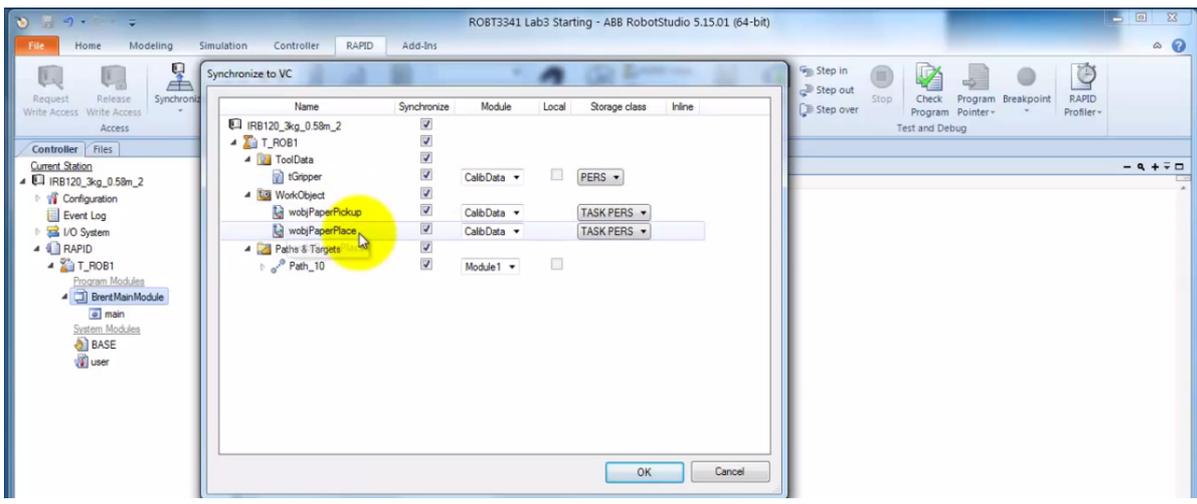
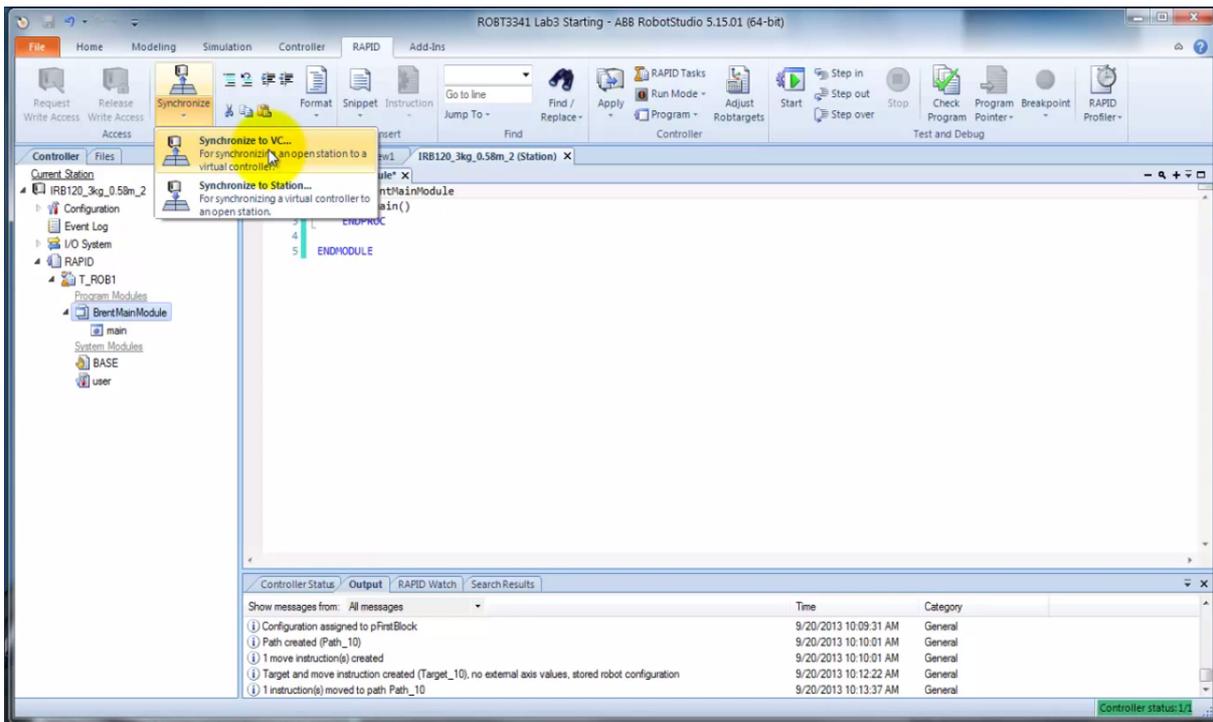


Figura 87. Sincronização com o Controle Virtual.
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Clicando-se em **OK** da janela de sincronização (figura 87), o programa recebe a configuração exibida na figura 88.

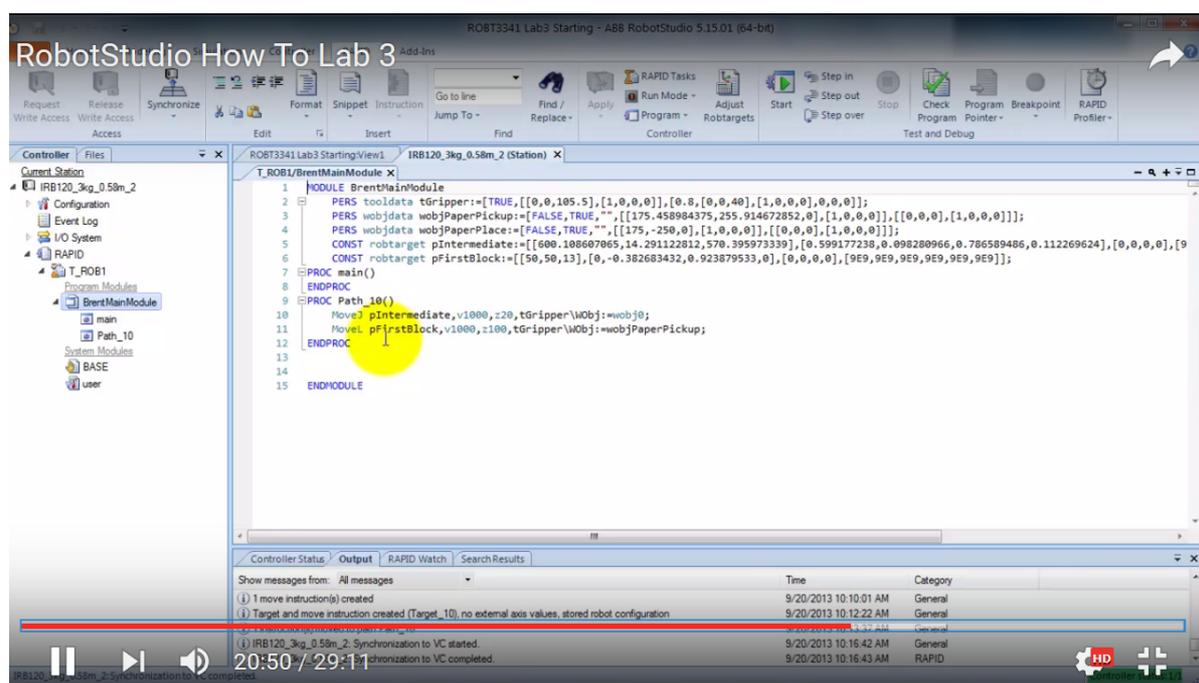


Figura 88. Geração automática de programa RAPID.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Faz-se, nesta parte, todas as configurações necessárias para a simulação da movimentação do robô. Pode-se modificar a estrutura do programa, copiar e colar movimentações idênticas após determinado sequenciamento, alterar valores, alterar nomes de trajetórias, criar novas rotinas se necessário. Finalizado isto, tem-se pronto o programa final, pronto para rodar a simulação, bastando clicar, na barra de ferramentas, em **Apply**, depois em **Check Programa** para realizar o *debug* do programa (verificar na janela de status *Output* se não há a ocorrência de erros), e finalmente clicar em **Start**, mostrado na figura 89.

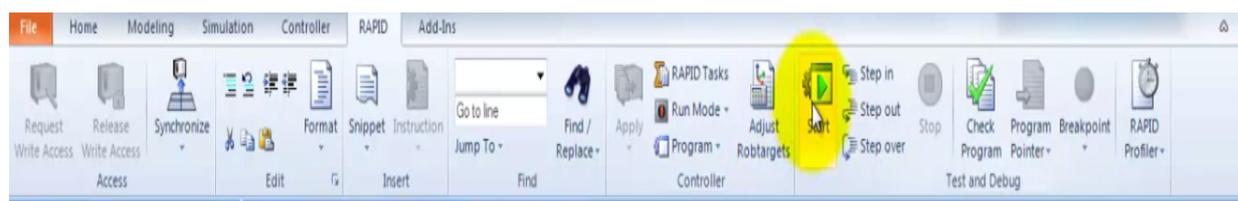


Figura 89. Inicializar simulação de movimentação dos robôs.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

5.1.5 SIMULAÇÃO

Iniciada a tentativa de simulação, o processo de *debug* apresentou erros no status no tutorial. Este problema foi solucionado colocando o sistema em modo automático no *painel de controle*, localizado na tab **Controller**, e acionando-se os motores do robô através deste painel (figura 90).

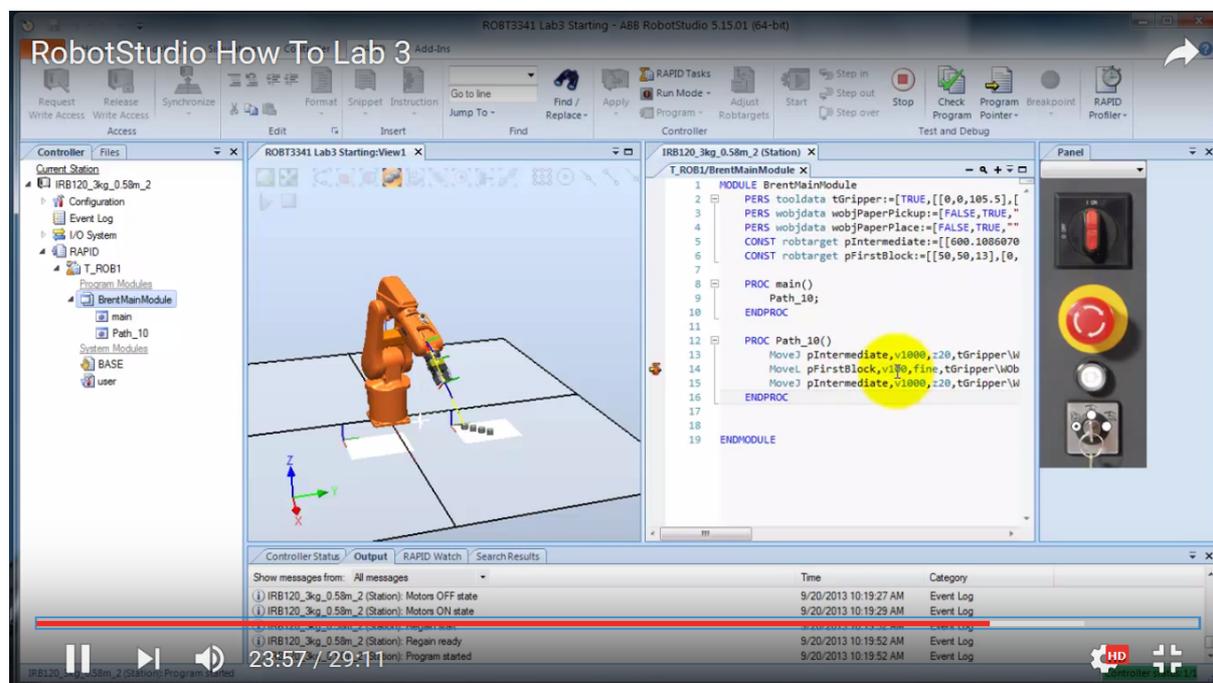


Figura 90. Acionamento dos motores e utilização do modo automático.
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Desta forma, visualiza-se a simulação animada da movimentação do robô ao longo da trajetória. Para verificar quais etapas da programa estão sendo executadas, basta acionar a tela da programação lado a lado com a execução da simulação, através da aba do *RAPID*.

Diversas possibilidades de execução da simulação são possíveis: modo contínuo, execução por linha, execução de um único ciclo, dentre outras opções.

Outras possibilidades importantes que podem ser realizadas para a simulação: editar variáveis **inputs** e **outputs** (I/O) e controlá-las, assim como monitorar as respostas dos dados gerados. Neste caso específico, foi habilitada a opção **I/O** na barra de ferramentas e realização da edição no programa RAPID sobre as condições de seus valores. No exemplo do tutorial, foi determinado um procedimento no qual o robô moveria sua junta a uma velocidade de 1 m/s para alcançar a posição intermediária.

Alcançada esta posição, o robô deveria esperar até que uma entrada digital fosse verdadeira ($DI10_1 = 1$), como sinal para que o robô vá buscar a primeira peça. Uma vez que um botão ligado a essa entrada seja acionado, o robô se desloca até a peça; então o programa seta uma saída $DI13_1$ como verdadeira (igual a 1), induzindo o robô a realizar a movimentação de retorno à posição intermediária. A figura 91 mostra um exemplo de estrutura da programação.

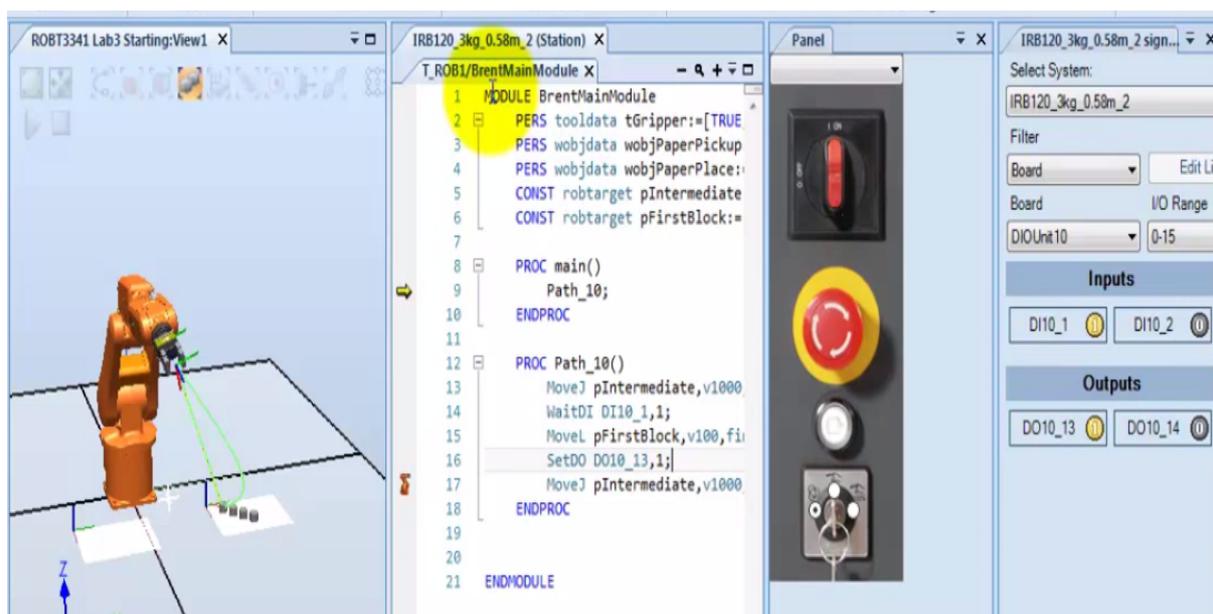


Figura 91. Programação e edição da simulação de movimentação dos robôs.
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Da mesma forma que todas as modificações realizadas na estação de trabalho devem ser sincronizadas com o Controlador Virtual, também todas as mudanças realizadas em alto nível – ou seja, na fase de programação – devem ser sincronizadas com a estação, na *tab* RAPID (figura 92).

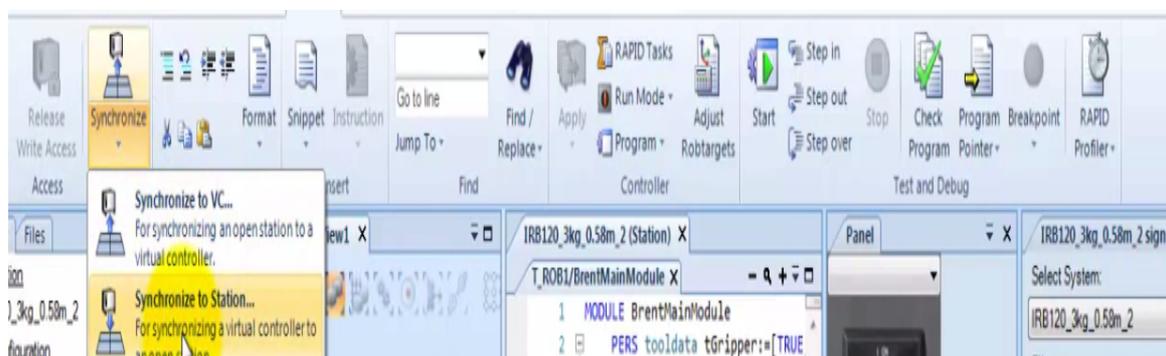


Figura 92. Sincronização do programa gerado com a estação de trabalho.
Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

Finalizado o procedimento, resta salvar o módulo de programa para que possa ser reutilizado em futuras oportunidades, conforme figura 93.

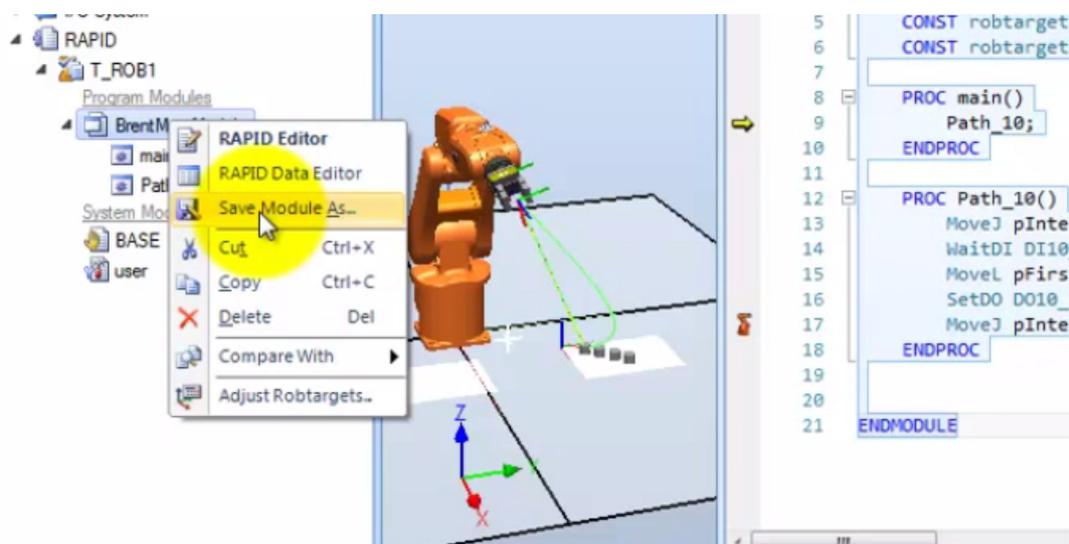


Figura 93. Salvando o módulo de programa RAPID.

Fonte: BCIT Mechatronics and Robotics - RobotStudio® How To Lab 3, 2014.

5.2 DIFICULDADES ENCONTRADAS

Conforme observado anteriormente, presenciou-se, logo na fase inicial do processo ao gerar a primeira trajetória do robô, uma falha prática no sistema, na qual houve a acusação de que o *target* criado não era acessível ao manipulador robótico. Essa falha não pôde ser solucionada pelos procedimentos padrões, agravada pelo fato de o *software* não disponibilizar um diagnóstico específico para a falha, com informações sobre os motivos da ocorrência, exigindo um estudo investigativo das razões para a ocorrência da falha, como será apresentado a seguir.

Os manuais de operações nem sempre possuem informações específicas para os casos encontrados, pois é muito comum ficarem desatualizados, conforme afirmação da própria ABB™ (2008 - 2010), nas páginas introdutórias de todos os seus manuais:

“The information in this manual (...) is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual” [ABB™ Operating Manual RobotStudio®, 2008 – 2010, folha de rosto].

Em tradução livre: “As informações neste manual estão sujeitas à mudanças sem prévio aviso e não devem ser entendidas como um compromisso feito pela ABB.

A ABB não assume responsabilidades por quaisquer erros que possam aparecer neste manual”. Este aviso está disponível, na íntegra, no Anexo C para visualização.

5.2.1 INSTALAÇÃO DO CONTROLADOR

A primeira preocupação frente à tentativa de solucionar o problema, foi o de verificar quais as possíveis características do sistema em construção eram discrepantes com a do sistema do exposto no tutorial, além da diferença no modelo do robô, bem como a diferença do *layout* e dos equipamentos presentes. Notou-se que no tutorial havia a presença de um controlador virtual em execução, que teria sido preparado anteriormente, não fazendo parte da explicação.

Havendo inúmeras possibilidades para a instalação do controlador, inclusive antes da criação de uma estação (*Create Solution with Station and Robot Controller*), foi realizada pesquisa pela verificação da melhor forma de instalar um controlador em um sistema já existente. Não houve resposta satisfatória por intermédio dos tutoriais audiovisuais, portanto recorreu-se ao manual de instruções oficial da ABB™, no qual, para uma situação semelhante ao do trabalho realizado, havia a opção de gerar um sistema robótico baseado no *Layout* (figura 94).

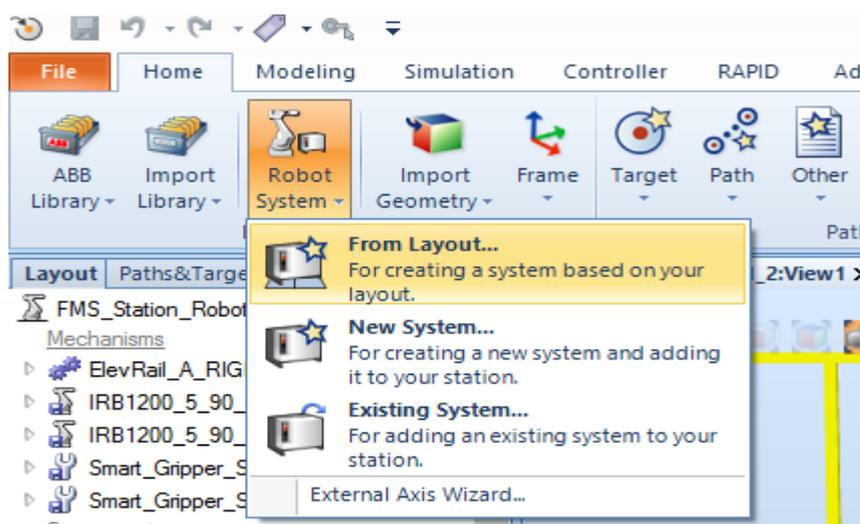


Figura 94. Instalação de um Controle Virtual com base no *Layout*.
Fonte: O autor.

Escolheu-se o *Robotware* 6.02 (software interno de execução na última versão do RobotStudio® sendo utilizado), conforme mostra a figura 95.

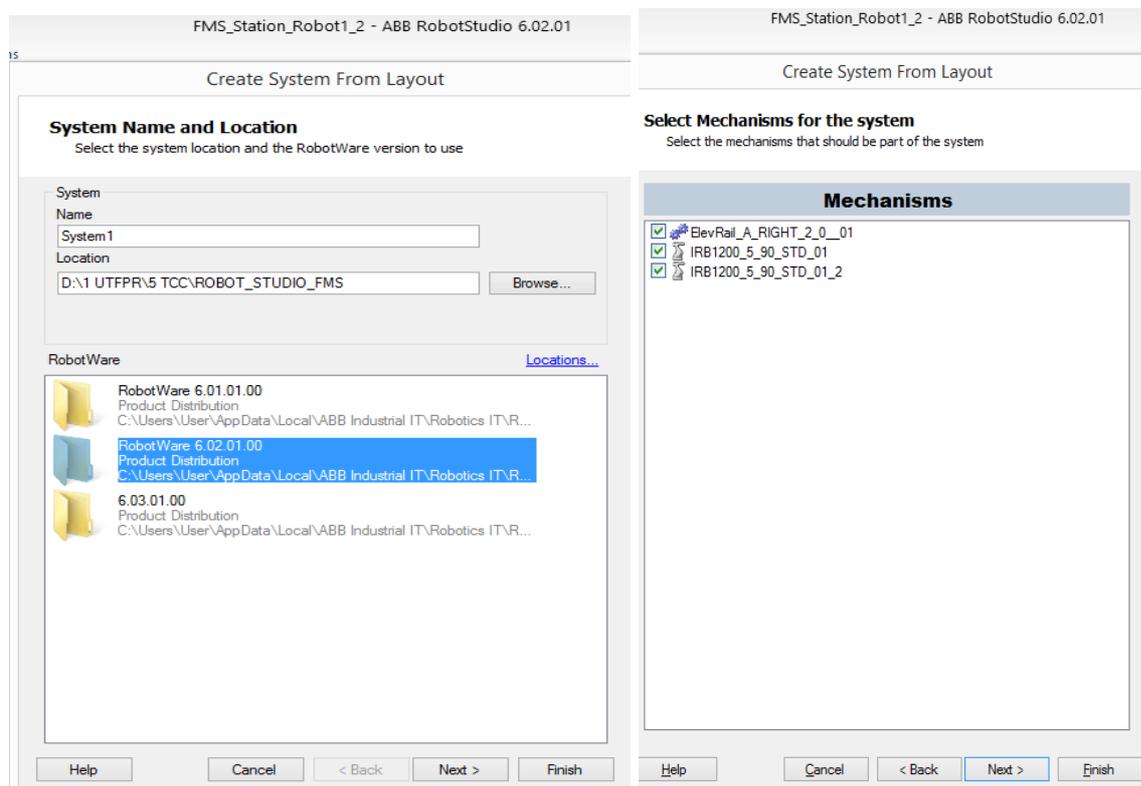


Figura 95. Instalação do VC – Robotware e mecanismos.
Fonte: O autor.

Clica-se em **Finish**. Aguarda-se o tempo necessário para o processo, e a partir do momento em que o controlador é instalado, ele se mostra em execução na janela de status.

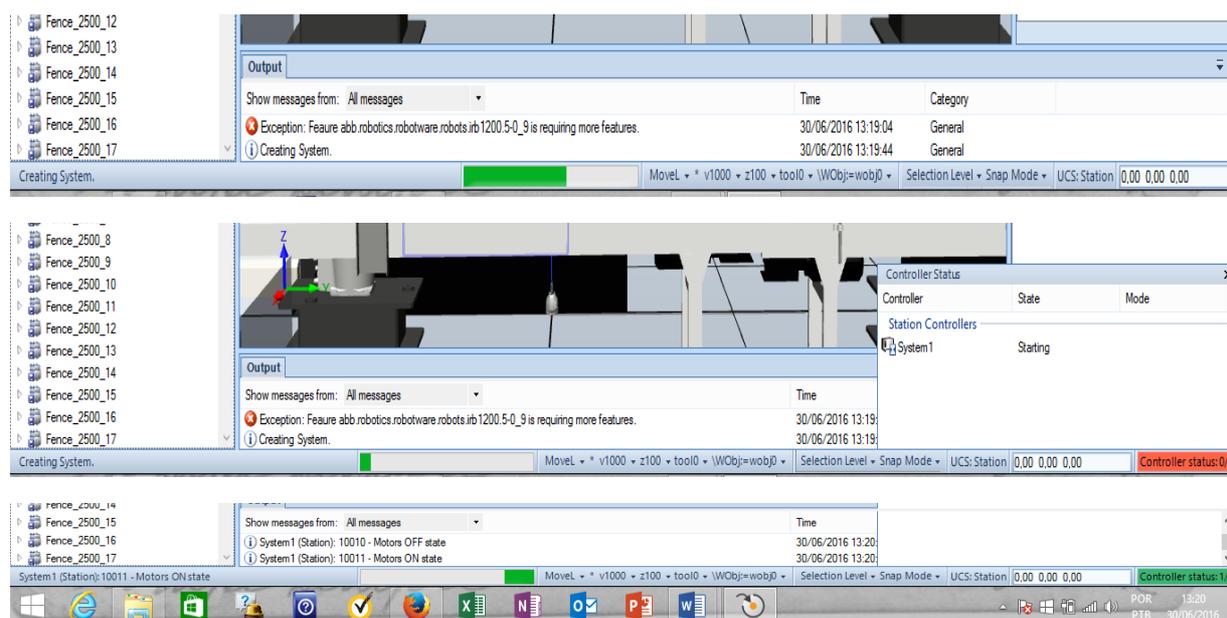


Figura 96. Inicialização do Controle Virtual. Fonte: O autor.

A diferença com a instalação do controlador virtual notou-se logo que o mesmo se estabilizou, pois algumas funções que antes não eram possíveis de serem executadas, tornaram-se disponíveis, como por exemplo as funções **Jog Linear** (movimenta todas as juntas do robô utilizando-se apenas um dos eixos do TCP), **Jog Reorient** (habilita a rotação do TCP completo), e **Multi Robot Jog** (habilita a movimentação de vários mecanismos ao mesmo tempo, como 2 robôs, por exemplo).

A disponibilização dessas funcionalidades pelo controlador virtual facilitou a manutenção do posicionamento do robô, mas **não solucionou** o problema de acessibilidade à peça.

5.2.2 ERROS DE ALCANCE (*Reachability*)

Com base em investigação realizada em diversos tutoriais, todos apresentaram a mesma resolução do problema de acessibilidade: reposicionar o robô ou o objeto de forma que o sistema reconhecesse a posição no sistema cartesiano, como no exemplo da figura 97. No entanto, apesar de serem aceitas as tarefas realizadas nos comandos, todas as tentativas falharam em disponibilizar o ponto de alcance do robô.

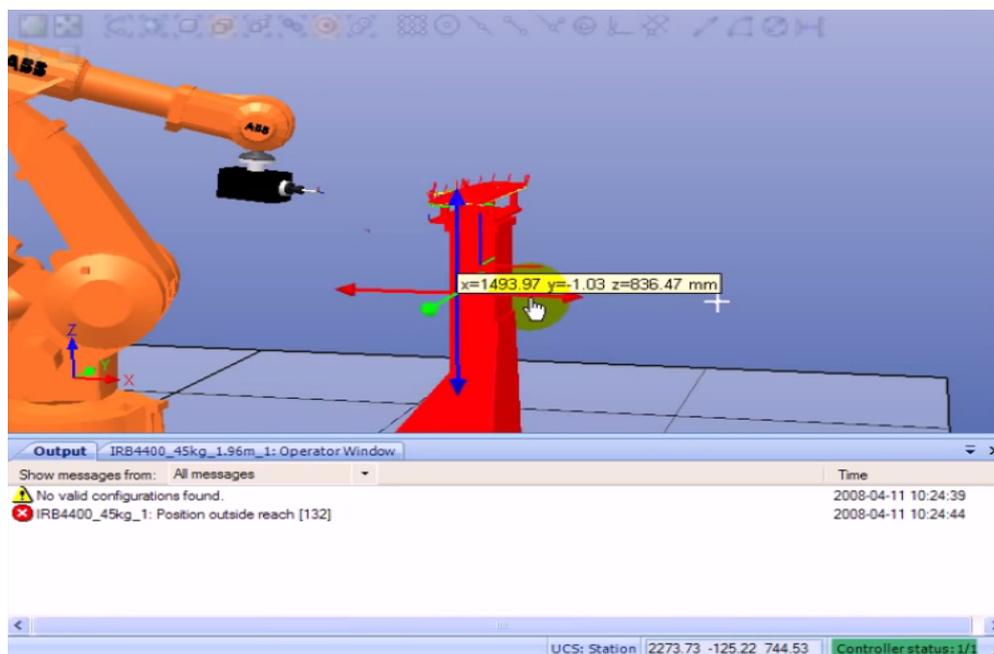


Figura 97. Exemplo de solução para problemas de acessibilidade por aproximação.
Fonte: Reachability – RobotStudio®, 2009.

5.2.3 BUSCA POR SOLUÇÕES

Alternativamente ao manual de operações e tutoriais audiovisuais, buscou-se auxílio de informações em comunidades virtuais de usuário do *software* da ABB™. A maior dificuldade encontrada durante esse processo, apesar de controverso, é o fato de o *software* RobotStudio® ser amplamente flexível, tanto para a realização de tarefas quaisquer, quanto para a solução de problemas. Essa flexibilidade dificultou a localização de uma informação que fosse específica para o presente caso, havendo sempre procedimentos diferentes, incompletos, ou – em muitos casos – com pré-requisitos aplicados, sem a existência de explicação prévia. Por mais que se seguisse as recomendações, outros obstáculos manifestavam-se, configurando-se a necessidade de uma nova solução. O problema permaneceu insolúvel e sem resposta satisfatória, já que o próprio *software* não gerava diagnóstico especificando a razão do erro.

Houve ainda a ocorrência de um fato inconveniente durante a utilização do *software*, pois este teve seu período limitado de teste de 30 dias expirado, limitando as funcionalidades necessárias para a execução das tarefas. Este agravante exigiu a ação inevitável de uma nova instalação em outro computador, para que fosse dada continuidade no trabalho e sua finalização.

5.3 ANÁLISE DO POTENCIAL DE UTILIZAÇÃO

A procura pela utilização do RobotStudio® ocorre principalmente pelo fato de ser um *software* extremamente preciso e confiável, com o qual se pode munir-se de informações valiosas para a implantação de qualquer sistema que exija mecanismos e tarefas automatizadas. Pode-se citar as principais possibilidades para a implantação de um Sistema Flexível de Manufatura, tais como a simulação virtual da movimentação de esteiras e mecanismos gerados pelo próprio usuário, a integração de uma câmera de inspeção, além ser possível realizar a configuração OPC com os robôs, câmeras integradas, dentre outras possibilidades, ainda sob fase de discussão e testes entre os usuários.

5.3.1 SIMULAÇÃO VIRTUAL

Além da simulação padrão realizada no RobotStudio®, o mesmo disponibiliza recursos para o monitoramento virtual das ocorrências de eventos discretos, sendo possível demarcar o traço da trajetória realizada pelo TCP da ferramenta do robô (TCP *trace*, conforme figura 98), assim como o comportamento das saídas e entradas do robô.

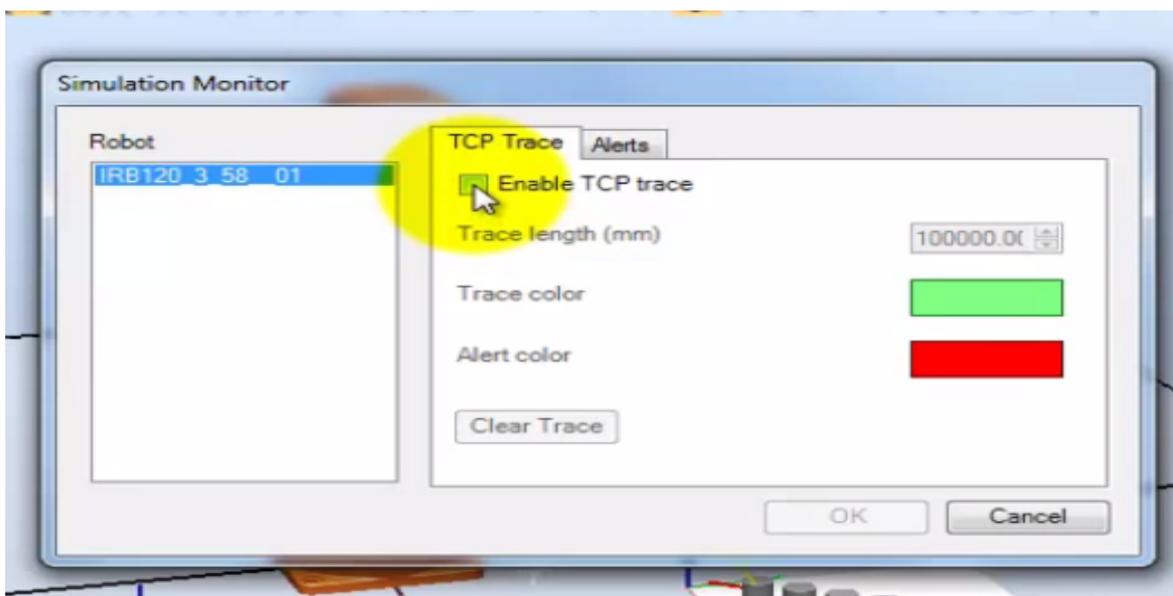


Figura 98. Monitoramento da trajetória de movimentação.
Fonte: O autor.

Todos os recursos (figura 99), incluindo o de análise de sinais *Signal Analyzer*, estão localizados na aba **Simulation**.

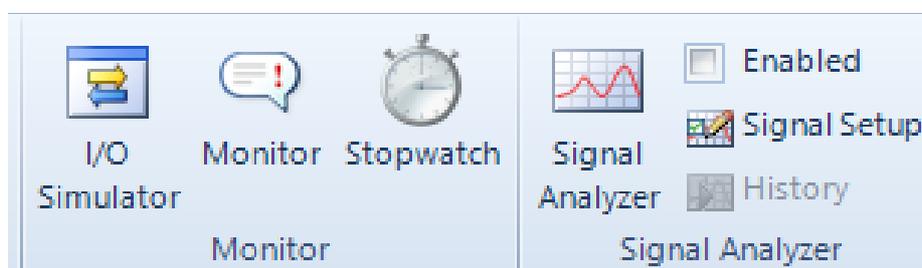


Figura 99. Recursos de monitoramento.
Fonte: O autor.

Para a realização da análise de sinais, uma simulação deve estar em execução para que os sinais sejam detectados pelo sistema, realizando o tratamento dos dados para a geração de gráficos, como o da figura 100, apropriados para a criação de relatórios.

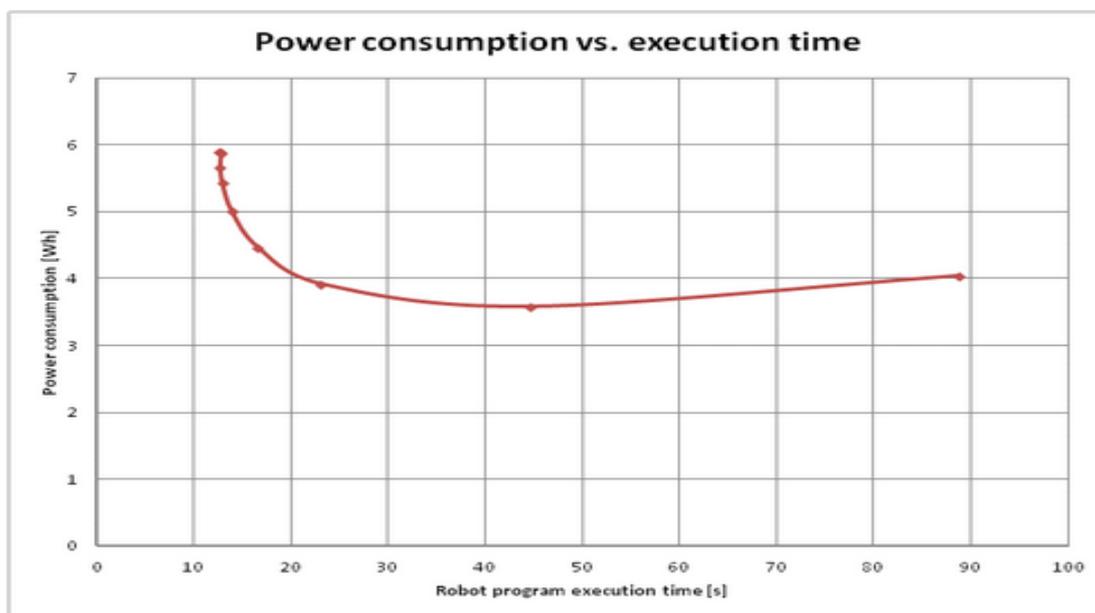


Figura 100. Recurso para geração de gráficos com dados para relatórios.
Fonte: Blog – ABB Robotics.

O *software* disponibiliza ainda recursos (figura 101) para a gravação das simulações em formato de vídeo, assim como a gravação da tela inteira do aplicativo para criar um *clip*, gravação em vídeo dos gráficos ativos, além da possibilidade de visualização de todos os vídeos criados. Todos esses recursos são gerados no próprio *software*, com a possibilidade de visualização imediata, e salvos automaticamente. É necessário ter um programa de visualização de vídeos instalado no PC.

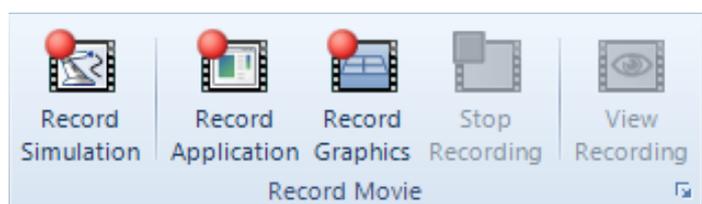


Figura 101. Recurso de gravações de aplicativos, simulação, e gráficos.
Fonte: O autor.

5.3.2 MOVIMENTAÇÃO DE ESTEIRAS

É notória a diversidade dos métodos com os quais se realiza a automatização de esteiras, ou *conveyors*. Um dos métodos aplicados durante pesquisa, seguindo um vídeo tutorial, e aplicado na prática, é descrito a seguir e mostrado na figura 102.

- 1) Criação de um **Empty Station** (Estação Vazia).
- 2) Clicar em **Import Library**, e escolher uma das esteiras (*conveyors*)
- 3) Clicar com o botão direito no nome da esteira listada, selecionar **Modify** e **Set Position**.
- 4) Modificar os valores desejados na janela com as coordenadas do TCP da esteira.

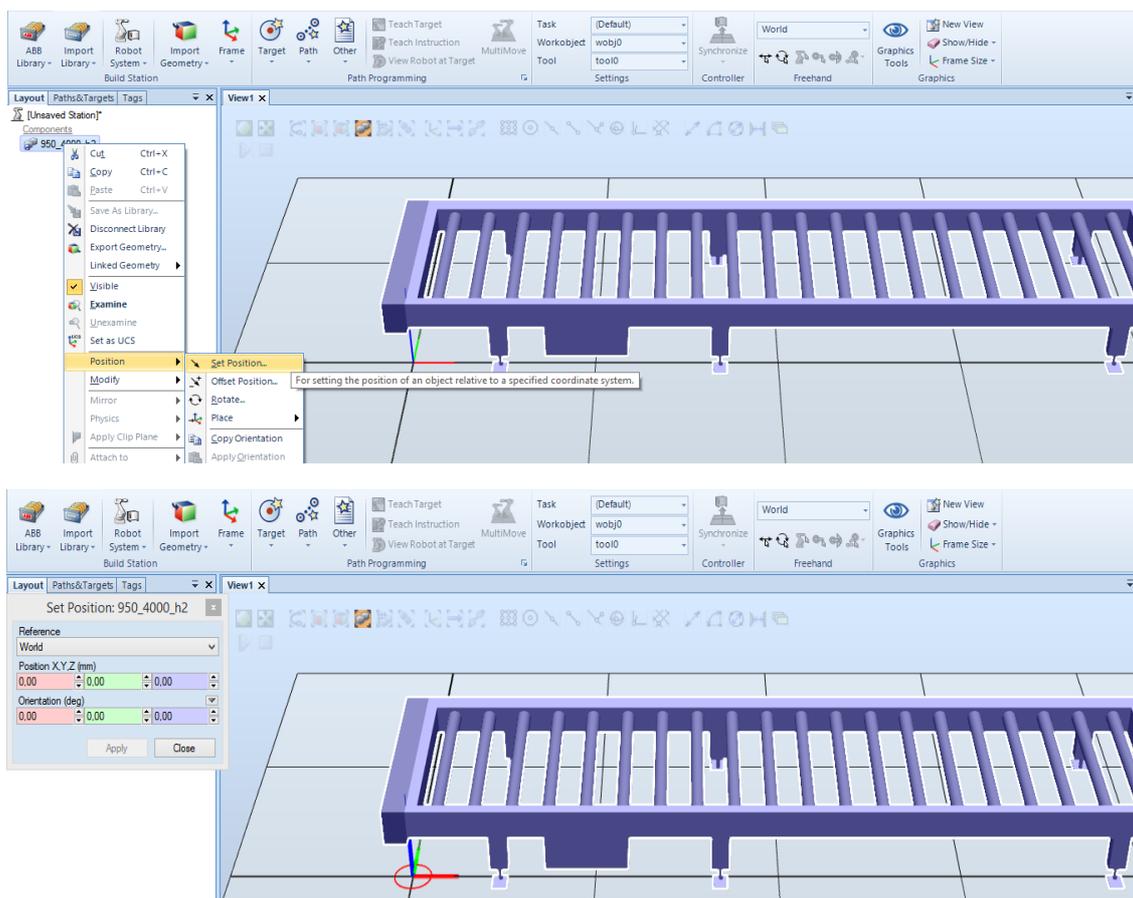


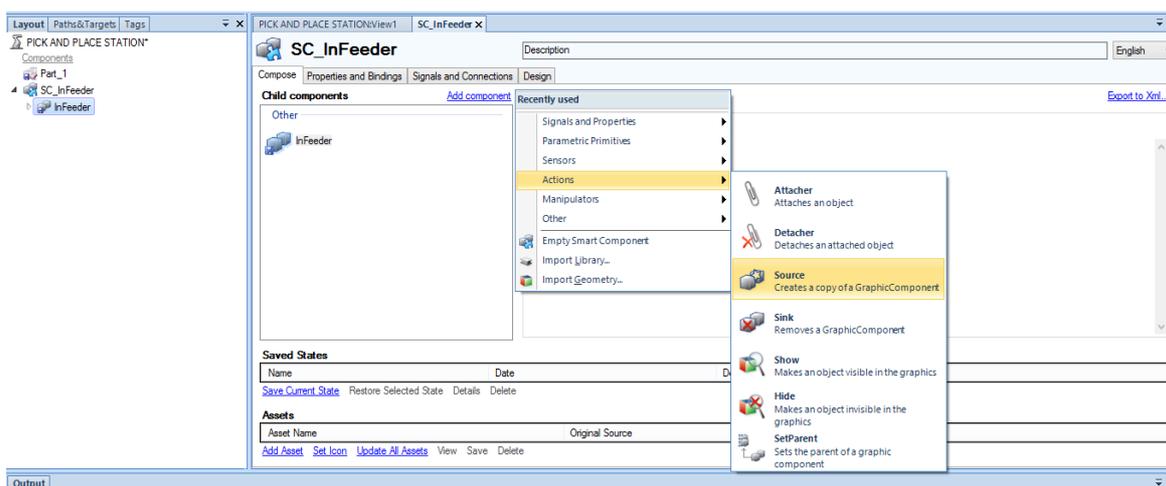
Figura 102. Escolha de um *conveyor* e edição do TCP.

Fonte: O autor.

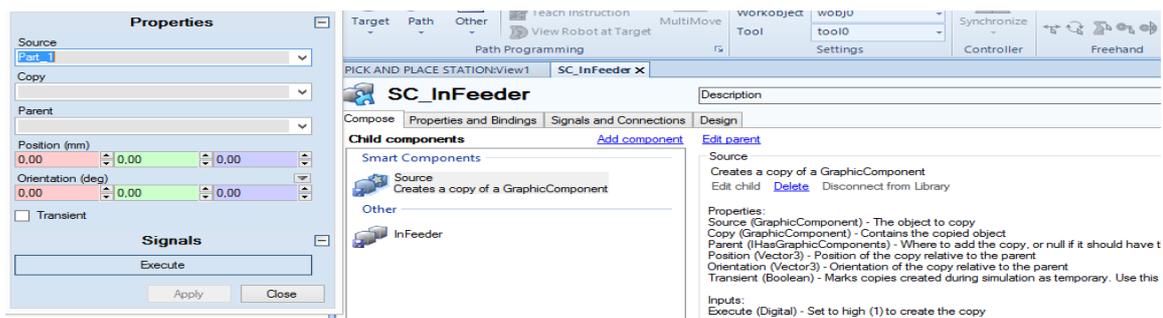
- 5) Criar um objeto 3D que represente o material a ser transportado, ou importar um objeto pronto da biblioteca. No exemplo, clicar em **Import Library**, selecionar **User Library** e clicar no objeto desejado.

A partir desde ponto os resultados, mais uma vez, passaram a apresentar diferenças, gerando um resultado indesejável adiante. A caixa selecionada foi gerada em uma posição que, mesmo fazendo alterações em seu TCP, não modificou a posição de origem de entrada

- 6) Modificar a posição do objeto ou da esteira, se necessário, utilizando o TCP, e renomeá-los como desejar. (Para a esteira, chamamos de *InFeeder*, pois esta alimentará a estação com as caixas).
- 7) Clicar com o botão direito em cima do nome do último objeto importado, e desmarcar a opção **Visible**. Isto tornará o objeto presente, porém momentaneamente invisível.
- 7) Na *tab Modeling*, clicar em **Smart Component**. Renomeá-lo de *SC_InFeeder*.
- 8) Lista à esquerda da tela, puxar o nome do *InFeeder* (esteira) para “dentro” do nome *SC_InFeeder*. Isso tornará a esteira um elemento do componente inteligente.
- 9) Fazer clique duplo no *SC_InFeeder*. Nesta janela, clicar sobre **Add Component, Actions, e Source** (a). Nas propriedades de *Source*, selecionar o objeto a ser transportado (caixa).



(a)



(b)

Figura 103. Configuração inteligente da esteira: (a) configurar componente, (b) propriedades do componente. Fonte: O autor.

- 10) Repetindo o procedimento 9, clicar em **Add Component**, **Other**, e **Queue**.
- 11) Clique duplo em **Queue**, e proceder conforme recomenda Gindiceanu (2011) em seu tutorial “*RobotStudio Tutorials – Create a Pick and Place Station #part 1 – Create InFeeder*”.

Trata-se de um procedimento extenso e meticuloso, envolvendo configuração detalhada dos mecanismos e configurações I/O, velocidade de deslocamento dos objetos, posicionamento de planos sensores de detecção dos objetos, dentre outros e, portanto, não é abordado completamente neste trabalho. Observou-se, no entanto, que apesar de o mecanismo de *conveyors* funcionar na prática, houve erro de localização do objeto, mesmo tendo sido seguido rigorosamente todos os passos recomendados pelo tutorial, conforme exposto na figura 104.

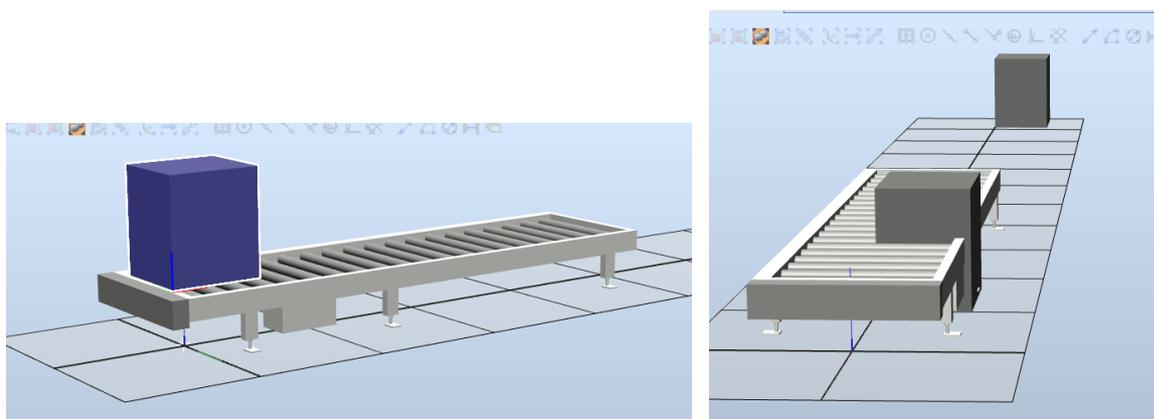


Figura 104. Diferenças dos resultados no mecanismo *conveyor*.
Fonte: O autor.

Outro erro também se mostrou presente na implantação de um plano sensor responsável pela detecção das caixas ao final do percurso da esteira, induzindo-as a parar o seu trajeto (figura 105). Este pode ser considerado um dispositivo “eletrônico” e, como tal, exigiu entradas e configurações discretas. No entanto, essa função não funcionou na prática, pois as caixas continuavam seu trajeto ao passar pelo plano, conforme figura 106.

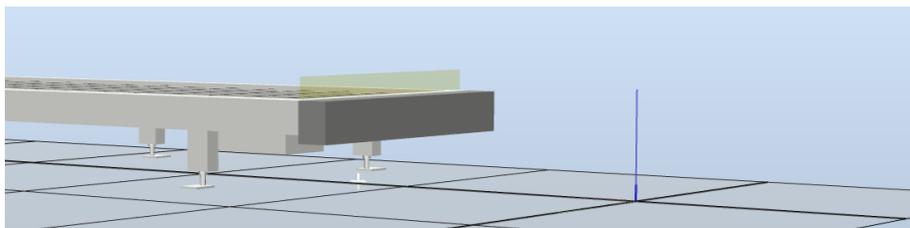


Figura 105. Configuração de um plano sensor de detecção.
Fonte: O autor.

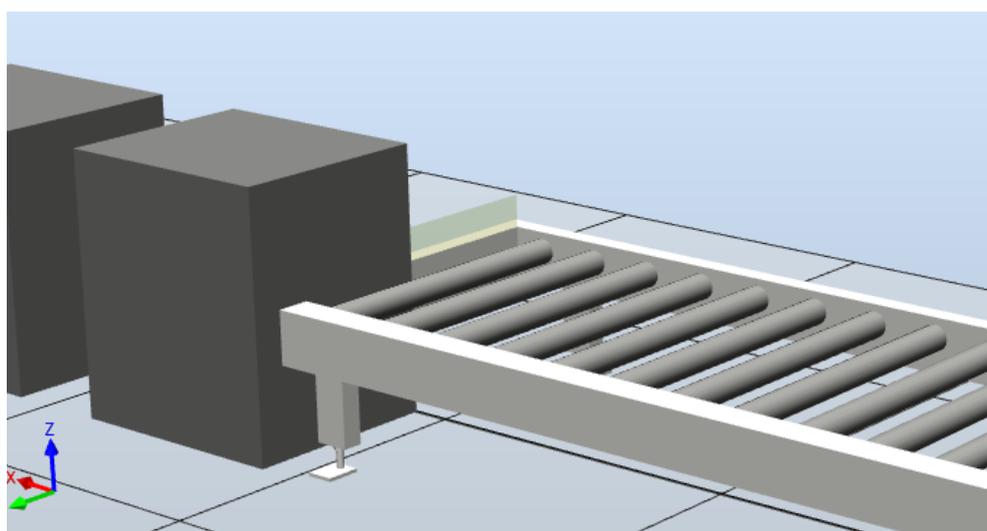


Figura 106. Plano sensor e caixas transportadas.
Fonte: O autor.

Devido a estas falhas apresentadas na prática, interrompeu-se a continuidade do tutorial, uma vez que, principalmente com essa falha no sensor, não haveria como fazer o robô trabalhar com uma peça em movimento, e ainda de acordo com o manual “*Operating Manual: RobotStudio*” da ABB™ (2008 - 2010): “*Creating a mechanism is dependent upon skillful construction of the main nodes of the tree structure*”, ou, em tradução livre – “A criação de um mecanismo depende diretamente da total habilidade de criação dos principais nós da estrutura da árvore”, sugerindo que erros podem acontecer caso a estrutura e ordem da árvore não esteja bem estruturada, o que se caracteriza como mais um obstáculo para o aprendizado intuitivo de um principiante.

Buscou-se ainda - como último recurso para encontrar uma solução para o problema, modificar as configurações de pontuação do próprio sistema operacional do computador em uso, PC Windows 10, para checar a hipótese de que as discrepâncias entre as orientações dos tutoriais e os resultados obtidos na prática não se tratassem apenas de divergência de configurações entre os computadores (comparando-se os utilizados nos tutoriais e o doméstico). No entanto, a mudança nas configurações de pontuação no sistema operacional não aparentou influenciar significativamente o resultado dos testes, que permaneceram apresentando falhas.

5.3.3 CÂMERA DE INSPEÇÃO INTEGRADA

Um dispositivo virtual como uma câmera, conectada a um robô via *Ethernet* ou porta serial, pode ser emulado utilizando-se um programa terminal, como o **Hercules SETUP utility** (HW Group, 2016) utilizado no tutorial “*ROBT3341 Using a Virtual Camera with RobotStudio*” (BCIT Mechatronics and Robotics, 2014) . Segundo o instrutor, esse terminal suporta variadas interfaces. Um *link* de acesso a esse programa terminal – para *downloads*, informações, bem como todas as instruções e orientações de utilização do mesmo – está relacionado na seção de referências ao final deste trabalho.

Nesse programa, clica-se em *TCP Server* , e em *Server Status* define-se a porta a ser utilizada, e clica-se em *Listen*, para inicializar o servidor.

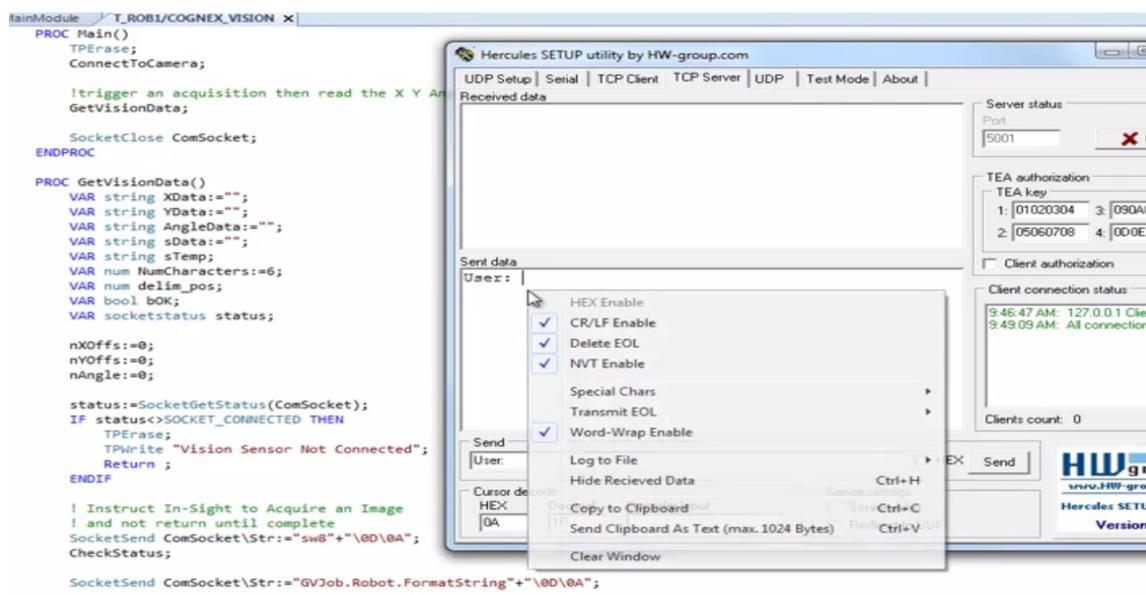


Figura 107. Integração com o servidor de uma câmera virtual.

Fonte: BCIT Mechatronics and Robotics. ROBT3341 Using a Virtual Camera with RobotStudio.

Os procedimentos seguintes são realizados no programa *RAPID* no RobotStudio®, realizando-se verificações quanto ao endereço de servidor e porta serial (figura 108).



Figura 108. Programação de Comunicação de uma Câmera Virtual.

Fonte: BCIT Mechatronics and Robotics. ROBT3341 Using a Virtual Camera with RobotStudio®.

Este procedimento exige familiaridade aprofundada com a programação *RAPID* e, tendo em vista que o tutorial não ensina a programação preparada anteriormente ao início do vídeo, bem como não fornece o código fonte utilizado no

tutorial, a prática desta parte não foi efetuada, sendo necessário um posterior estudo detalhado. Um *link* de acesso ao manual de referência técnica – **Technical reference manual: RAPID Instructions** (ABB Robotics, 2004-2010), com instruções e funções da programação RAPID (para *Robotware* versão 5.13) está também referenciado ao fim deste trabalho, para consulta e estudo.

5.3.4 CONFIGURAÇÃO OPC

De acordo com a *OPC® Foundation – The interoperability Standard For Industrial Automation™*, o padrão OPC trata-se de um procedimento de interoperabilidade padrão para um seguro e confiável tráfego de dados no espaço da automação industrial e outras indústrias. É uma plataforma independente que assegura o fluxo contínuo de informações entre dispositivos provenientes de fornecedores diferentes.

No RobotStudio®, essa comunicação é feita principalmente pelo **IRC5 OPC Server**, cuja aplicação – segundo a ABB™ – é utilizado para criar e gerenciar *Aliases* para os controladores robóticos IRC5 da ABB™. “*Alias*” é um desencriptador que representa a interface de comunicação para um controlador robótico IRC5 da ABB™, e devem se criados para cada controlador robótico para que sejam acessados pelo servidor *ABB IRC5 Server*. Um *link* de acesso ao manual de utilização do *IRC5 OPC Server – Application manual: IRC5 OPC Server help* (ABB, 2007-2015) está disponível nas referências ao fim deste trabalho.

5.3.4.1 CLP – ROBÔS

Há controvérsias nas discussões de fórum – como o *Robot-Forum: Industrial Robots Community*, cujo *link* de acesso é trazido ao final do trabalho nas referências – sobre a comunicação de uma célula virtual com manipuladores robóticos a um CLP externo real, no qual é discutido que a comunicação com um CLP externo pode ser realizado com o protocolo de comunicação OPC, de forma que, comunicando um servidor OPC ao CLP verdadeiro, seja possível simular determinados cenários comuns, com referências a uma informação fornecida pela própria ABB™.

A informação, no entanto, foi divulgada no ano de 2012, junto com o lançamento dos *Smart Components*, uma vez que um “*OPC smartcomponent*” foi

lançado, gratuitamente, no site da comunidade da ABB™. No entanto, este componente foi retirado alguns meses depois, por não ser “estável o suficiente”. Desde então, conforme o comentário do usuário, ninguém mais tratou a respeito desse *smart component*, silenciando a discussão sobre essa forma de comunicação.

Há, no entanto, tutorial de comunicação de um *software* desenvolvido pela Siemens®, o **SIMATIC WinCC Flexible** – cujo *link* para download se encontra disponível nas referências ao fim do trabalho – no qual se mostra como conectar o RobotStudio® com esse *software* baseado no conceito SCADA. O *link* de acesso ao tutorial - **RobotStudio®: Comunicación con WinCC Flexible mediante OPC Server** (CASTILLO, 2015) encontra-se também referenciado para pesquisa e estudo.

Trata-se portanto, de um procedimento que aspira ser possível de ser realizado, mas ainda passando por um período de testes e pesquisa pela comunidade.

5.3.4.2 CÂMERA INTEGRADA AO FMS

Similarmente, discute-se ser possível a conexão entre o RobotStudio® e uma câmera real, através do protocolo TCP/IP (protocolo de envio e recebimento de dados na comunicação entre computadores em rede), cujo endereço no Controlador Virtual (VC) é 127.0.0.1, de acordo com um dos membros do fórum O *link* de acesso a essas informações constam nas referências ao fim do trabalho, intitulado **TCP/IP connection between RobotStudio and Camera** (ABB, 2010). Há relatos de funcionalidade de conexão entre dois controladores virtuais e uma câmera ligados ao PC, no fórum **Connect to VC via OPC Server** (ABB, 2012). Recomenda-se o estudo de instruções de programação RAPID, como *SocketCreate*, *SocketSend* e *SocketReceive*, contidos no manual de programação disponível nas referências ao fim deste trabalho, intitulado **Technical reference manual: RAPID Instructions, Functions, and Data Type** (ABB Robots, 2004-2010).

6 CONSIDERAÇÕES FINAIS

O *software* RobotStudio® é um produto largamente cobiçado na comunidade tecnológica existente ao redor do mundo, desde profissionais quanto entusiastas da área, devido ao seu potencial e flexibilidade de utilização.

Trata-se de uma ferramenta de uso profissional, com alta precisão e confiabilidade, uma vez que sua utilização exige entrada de dados precisos e bem definidos para simular, da forma mais próxima da realidade possível, o comportamento do sistema.

Foi notório, no entanto, o grau de habilidade exigido para utilizar essa ferramenta eficientemente, para a realização do modelamento de um Sistema Flexível de Manufatura. A alta versatilidade deste *software* exige pesquisa meticulosa e detalhada sobre as melhores formas de aplicação dos mecanismos necessários à construção do projeto. Não se pode afirmar tratar-se de um *software* intuitivo e de fácil acesso, pelo fato de que a eficiência e funcionalidade do mesmo exige conhecimentos aprofundados na área, assim como grande familiaridade com o *software*, em um nível somente alcançado após longos períodos de prática e treinamento especializado.

Do projetista, é exigido que tenha senso investigativo e lógica bem desenvolvida para gerenciar a manifestação dos obstáculos de forma independente, pois nem todas as informações sobre a manipulação do *software* estão facilmente acessíveis, caso existam. Os manuais, disponíveis apenas em inglês, desatualizam-se rapidamente, já que a cada lançamento de uma versão nova do RobotStudio®, muitos dos procedimentos, bem como configuração visual, mudam.

Ainda com relação aos manuais de utilização, apesar de bem organizados, não seguem uma ordem específica de ações a serem tomadas para determinado fim, o que faz sentido pelo fato de cada projeto ter características e necessidades funcionais distintas, exigindo tomadas de ações diferentes – porém possíveis, devido à flexibilidade do *software*.

Os problemas encontrados permaneceram sem solução definida até o momento, devido à falta de diagnósticos específicos do RobotStudio®. As razões de suas ocorrências podem estar relacionadas à falta de habilidade do autor que, não dispondo de tempo hábil para desenvolver familiaridade suficiente com o *software*, não obteve êxito para a finalização do funcionamento de todos os mecanismos,

tampouco obteve treinamento para a realização do mesmo. É recomendado um trabalho com maior tempo de dedicação e constante prática diária, para se obter os resultados desejados.

Nesta mentalidade, recomenda-se o desenvolvimento de futuros trabalhos e pesquisas que envolvam, além da construção de *layout* de um sistema fabril, a efetiva comunicação real entre equipamentos físicos e o *software* RobotStudio®, utilizando os recursos destinados para tal. Esses equipamentos podem ser CLPs, câmeras de inspeção integradas, controladores, manipuladores robóticos (desde que compatíveis com a linguagem de programação RAPID, desenvolvido pelo RobotStudio®), e outros mecanismos que envolvam comunicação via OPC e/ou *Ethernet*.

Indica-se a utilização deste *software* para a capacitação prática de alto nível de estudantes e profissionais que aspiram lidar com a programação direta de manipuladores robóticos de alto desempenho, como os modelos robóticos da ABB™, proporcionando grande desenvoltura em todos os conceitos envolvendo o funcionamento do chão de fábrica.

REFERÊNCIAS

ABB. **Application manual: IRC5 OPC Server help**. 2007-2015. Disponível em: <http://developercenter.blob.core.windows.net/devcenter/Downloads/IRC5_OPC/AM-IRC5_OPC_Server.pdf>. Acesso em: 20 de junho de 2016.

ABB. **Connect to VC via OPC Server – Fórum de discussões da RobotStudio®**. Novembro de 2012. Disponível em: <https://forums.robotstudio.com/discussion/7317/connect-to-vc-via-opc-server>. Acesso em: 20 de junho de 2016.

ABB. **Operating Manual: RobotStudio® 5.13**. 2008 – 2010. Disponível em: http://www.iu.hio.no/~georgm/robot/doc/RobotStudio%20Operating%20Manual/3HAC032104-001_revC_en.pdf. Acesso em: 12 de junho de 2016.

ABB. **TCP/IP connection between RobotStudio® and Camera**. Janeiro de 2010. Disponível em: <<https://forums.robotstudio.com/discussion/3757/tcp-ip-connection-between-robotstudio-and-camera>>. Acesso em: 20 de junho de 2016.

ABB Robotics. **Technical reference manual: RAPID Instructions, Functions, and Data Type**. 2004 – 2010. Disponível em: <https://library.e.abb.com/public/688894b98123f87bc1257cc50044e809/Technical%20reference%20manual_RAPID_3HAC16581-1_revJ_en.pdf>. Acesso em: 20 de junho de 2016.

ARMADA, M. A. **Controle de Robots**. *XV Curso de Automática em la Industria*. Aguadulce, 1995.

BAISE, Luiza Zamarian. **Robótica – Arquitetura de Robôs e Sensores 1**. Disponível em: <<http://slideplayer.com.br/slide/1251175/>>. Acesso em: 13 de abril 2016.

BCIT Mechatronics and Robotics. **RobotStudio® Tutorial Getting Started 1**, 21 de agosto de 2015. Disponível em: <<https://www.youtube.com/watch?v=bm-Tyz4Gmzw&list=PLE6CB81A3320750CB>>. Acesso em: 04 de abril de 2016.

BCIT Mechatronics and Robotics. **RobotStudio® Tutorial Getting Started 2**, 21 de agosto de 2015. Disponível em: <https://www.youtube.com/watch?v=zYS_OluiXw&list=PLE6CB81A3320750CB&index=2>. Acesso em: 04 de abril de 2016.

BCIT Mechatronics and Robotics. **RobotStudio® How To Lab 3**, 23 de outubro de 2014. Disponível em: <<https://www.youtube.com/watch?v=npXi7RWVwRA>>. Acesso em: 18 de junho de 2016.

BCIT Mechatronics and Robotics. **ROBT3341 Using a Virtual Camera with RobotStudio®**. 23 de outubro de 2014. Disponível em: <<https://www.youtube.com/watch?v=ECtJW3pXOps>>. Acesso em: 19 de junho de 2016.

Blog: ABB™ Robotics. Disponível em: <<http://www.abb.com/cawp/gad00540/5ffd0dbdc68fd6ffc12579bb002e1c43.aspx>>. Acesso em: 20 de junho de 2016.

CASTILLO, Juan Carlos Martin. **RobotStudio®: Comunicación com WinCC Flexible mediante OPC Server**. 25 de julho de 2015. Disponível em: <<https://www.youtube.com/watch?v=rYYVDdOm2QE>>. Acesso em: 20 de junho de 2016.

ENGELBERGER, J. F. **Robotics in the 21th century**. *Scientific American*, 1995.

GINDICEANU, Cristian. **RobotStudio Tutorials - Create a Pick and Place Station #part 1 - Create InFeeder**. 7 de maio de 2011. Disponível em: <<https://www.youtube.com/watch?v=4XgH5N3gjuo&t=59s>>. Acesso em: 20 de junho de 2016.

HW Group. **Hercules SETUP utility**. Disponível em: <http://www.hw-group.com/products/hercules/index_en.html>. Acesso em: 20 de junho de 2016.

OPC® Foundation. **WHAT IS OPC?**. 2016. Disponível em: <<https://opcfoundation.org/about/what-is-opc/>>. Acesso em: 19 de junho de 2016.

Reachability - RobotStudio®. 13 de novembro de 2009. Disponível em: <<https://www.youtube.com/watch?v=fjvrcgWVYMA>>. Acesso em: 18 de junho de 2016.

Robot-Forum: Industrial Robots Community, Topic: RobotStudio: Real PLC with Virtual Controller. Disponível em: <http://www.robot-forum.com/robotforum/abb-robot-forum/robotstudio-real-plc-with-virtual-controller/>. Acesso em: 20 de junho de 2016.

RobotStudio® da ABB™ Site Oficial. Disponível em: <<http://new.abb.com/products/robotics/robotstudio>>. Acesso em: 6 de Abril de 2016.

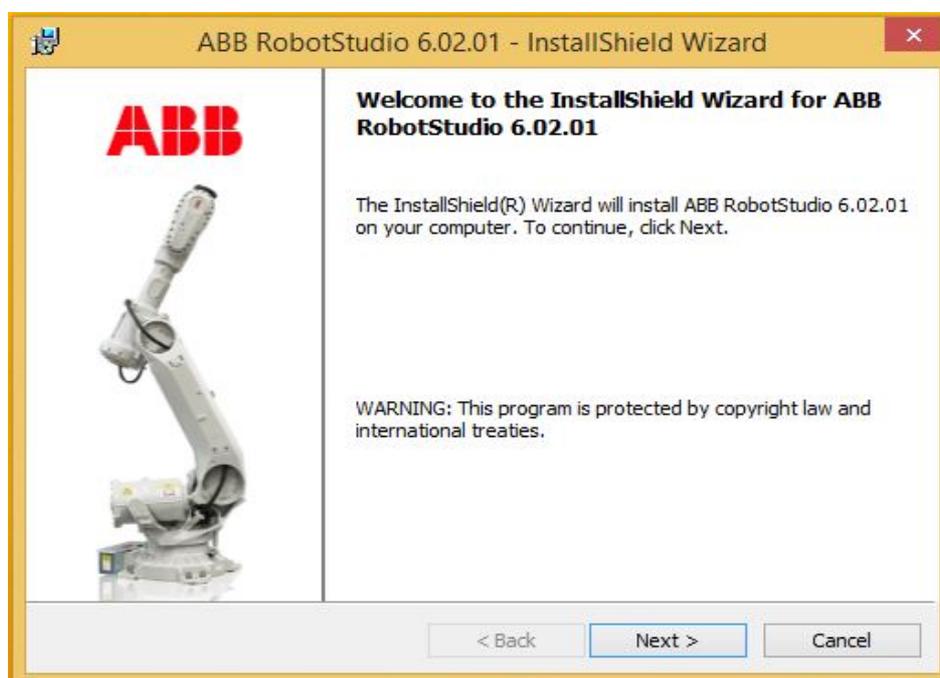
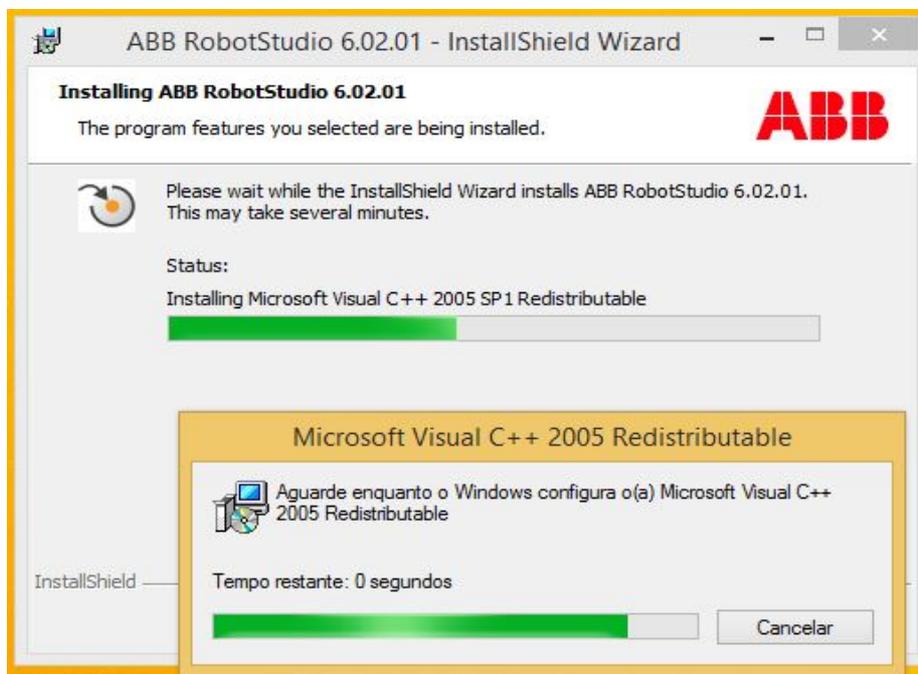
ROSÁRIO, João Maurício. **Princípios de mecatrônica**. São Paulo, SP: Prentice-Hall, 2005.

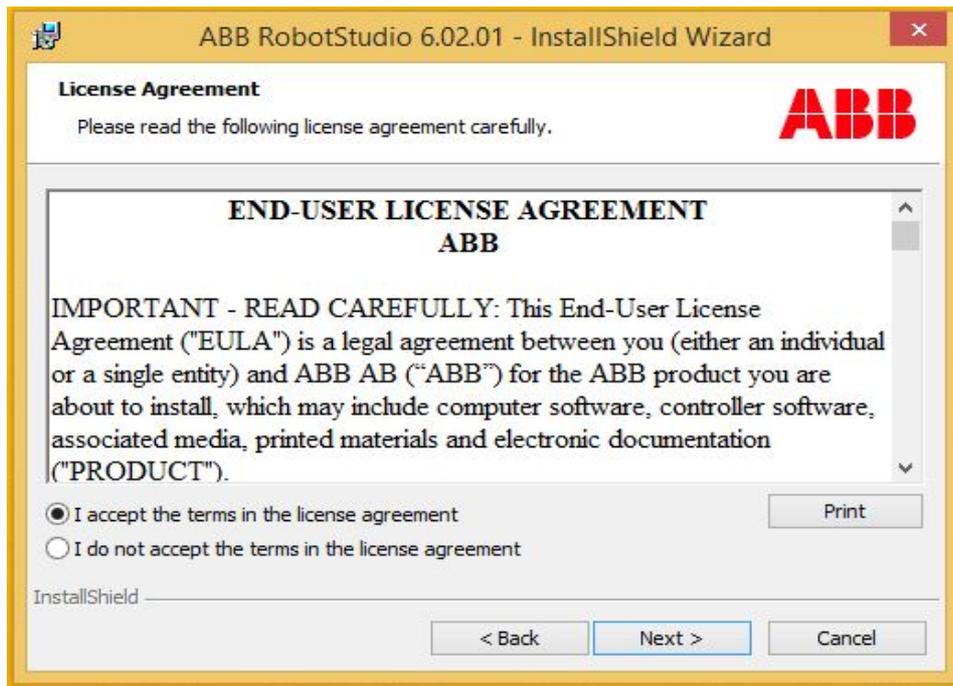
SILVEIRA, Rosemari M. C. F.; BAZZO, Walter A. CIÊNCIA E TECNOLOGIA: Transformando a relação do ser humano com o mundo. **Tecnologia e Civilização**. IX Simpósio Internacional Processo Civilizador, Ponta Grossa, Paraná, Brasil, 2005. Disponível em: <<http://www.uel.br/grupo-estudo/processoscivilizadores/portugues/sites/anais/anais9/artigos/workshop/art19.pdf>>. Acesso em: 14 de junho de 2016.

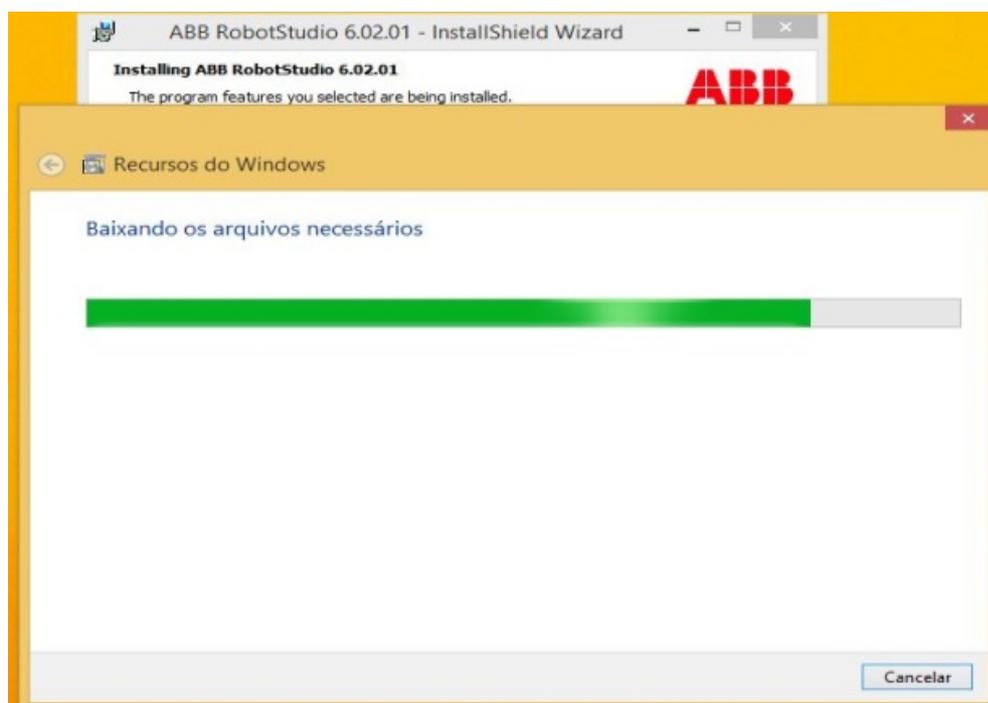
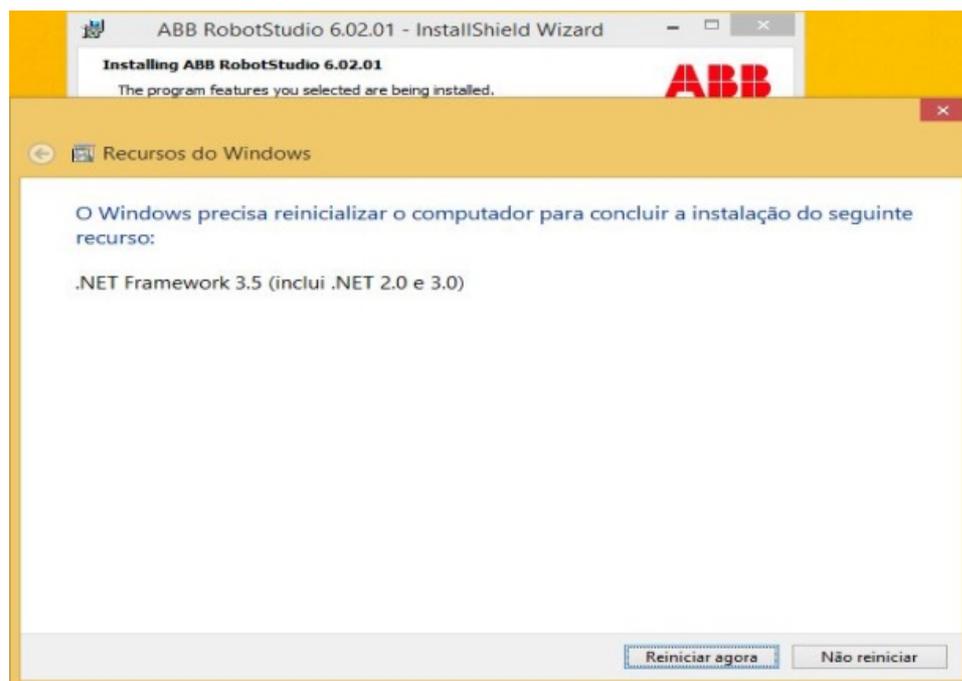
Wikipedia. **R.U.R.** Disponível em: <<https://en.wikipedia.org/wiki/R.U.R.>>. Acesso em: 11 de maio de 2016.

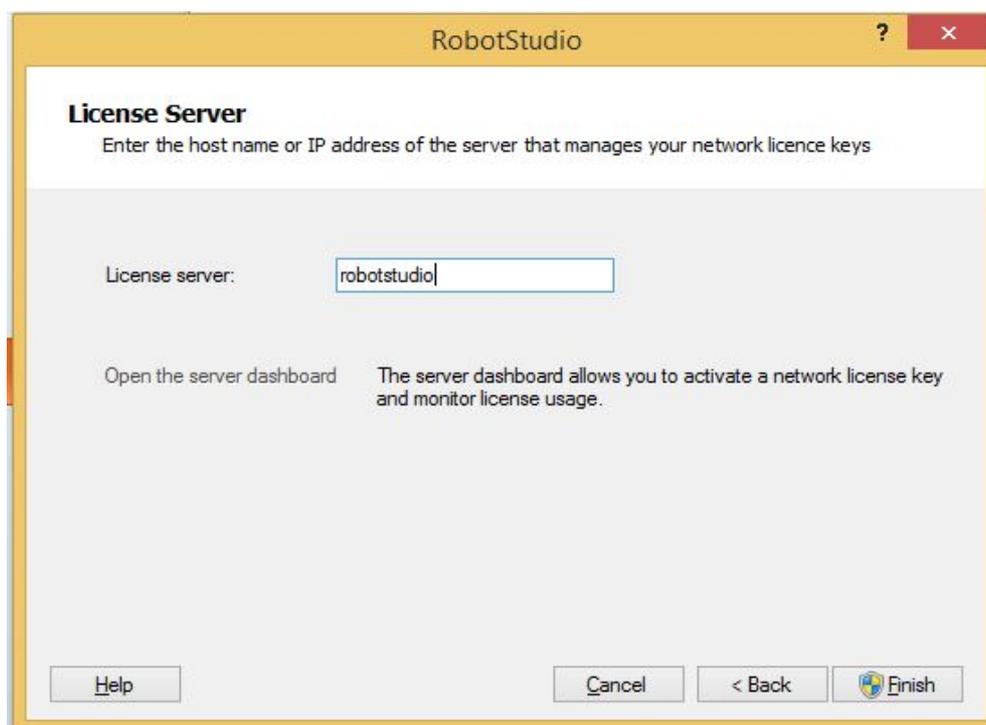
ZALZALA, Ali MS. **Robots from Humans or... Humans from Robots**. Disponível em: <<http://zalzala.org/IKMA/LinkedDocuments/RobotsHumans/sld001.htm>>. Acesso em: 18 de junho de 2016.

ANEXO A – Figuras detalhando o procedimento de instalação do software licenciado.







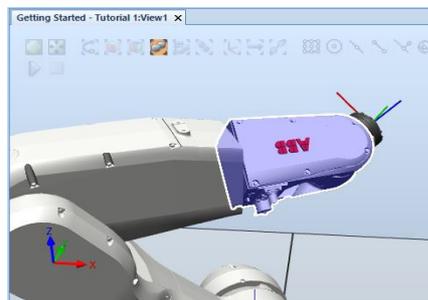


Processo de Instalação do Software.
Fonte: O autor.

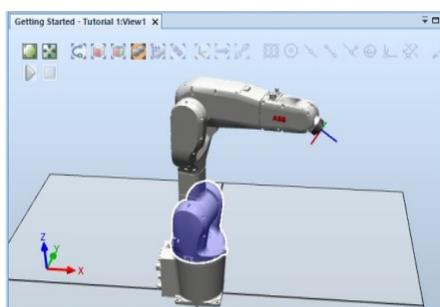
ANEXO B– Figuras representando todas as movimentações de juntas de um robô no ambiente virtual do *software* RobotStudio®, utilizando as ferramentas de manipulação de juntas “jogs”.



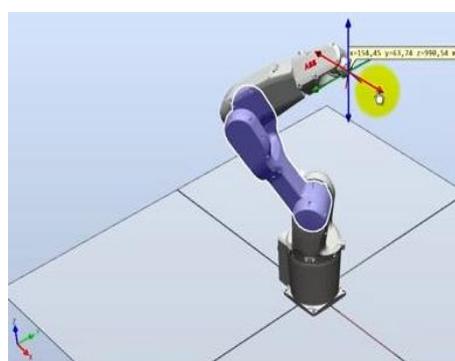
Jogs - ferramentas de manipulação das juntas.



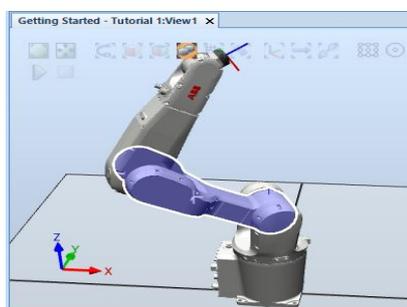
Inclinação do efetuador do robô.



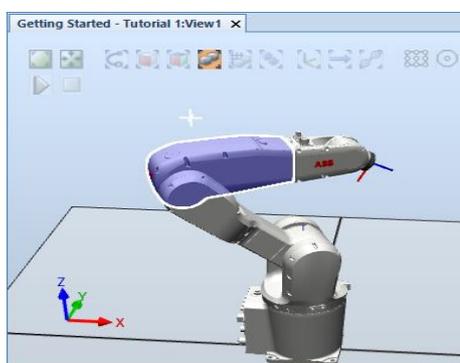
Axis Rotation (rotação do eixo principal).



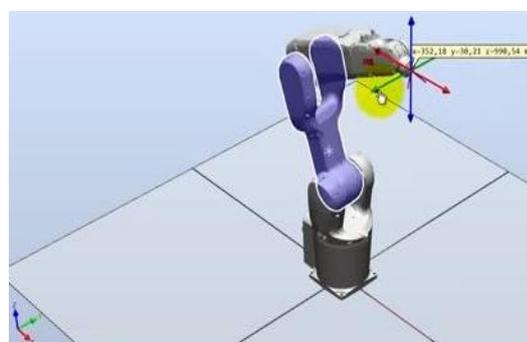
Jog Linear – movimentação do robô inteiro ao longo do eixo 'x'.



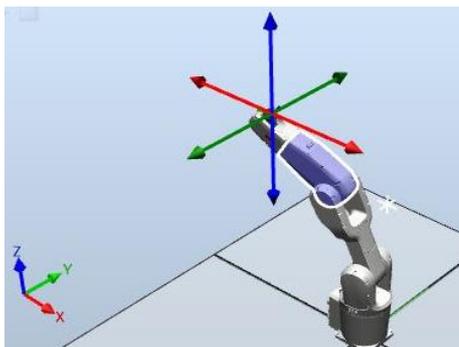
Axis Inclined (inclinação de eixo)



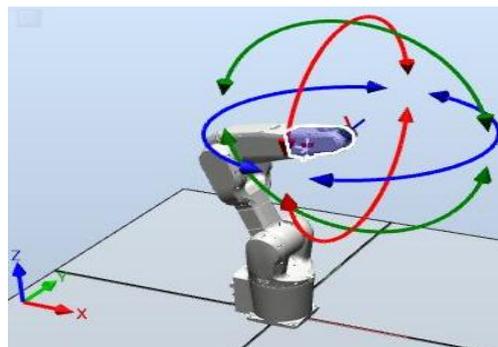
Inclinação da terceira junta.



Jog Linear – movimentação do robô inteiro ao longo do eixo 'y'.



Jog Linear – movimentação do robô inteiro ao longo do eixo 'z'.



Jog Reorient – habilita rotações em torno dos eixos de coordenadas.

Fonte: O autor.

ANEXO C – AVISO DE ISENÇÃO DE RESPONSABILIDADES POR MUDANÇAS DE INFORMAÇÕES, SEM AVISO PRÉVIO, NOS MANUAIS DA ABB.

The information in this manual is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this manual.

Except as may be expressly stated anywhere in this manual, nothing herein shall be construed as any kind of guarantee or warranty by ABB for losses, damages to persons or property, fitness for a specific purpose or the like.

In no event shall ABB be liable for incidental or consequential damages arising from use of this manual and products described herein.

This manual and parts thereof must not be reproduced or copied without ABB's written permission, and contents thereof must not be imparted to a third party nor be used for any unauthorized purpose. Contravention will be prosecuted.

Additional copies of this manual may be obtained from ABB at its then current charge.

© Copyright 2008-2010 ABB All rights reserved.

ABB AB
Robotics Products
SE-721 68 Västerås
Sweden

Fonte: ABB. Operating Manual: RobotStudio®, 2008 – 2010.