

# DESENVOLVIMENTO DE PROGRAMAÇÃO *OFFLINE* E DE NOVOS PROCESSOS ROBOTIZADOS DE LIXAMENTO

Rui Manuel Cardoso da Silva Rodrigues



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2013



Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Rui Manuel Cardoso da Silva Rodrigues, N° 1070304, 1070304@isep.ipp.pt

Orientação científica: Manuel Fernando dos Santos Silva, mss@isep.ipp.pt

Empresa: Grohe Portugal – Componentes Sanitários, LDA.

Supervisão: Sérgio Costa, scosta@grohe.pt



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

25 de Julho de 2013



## *Agradecimentos*

Em primeiro lugar quero agradecer ao meu orientador, o Professor Manuel Silva, pela sua disponibilidade e orientação dada durante a realização deste trabalho. Quero agradecer também pela oportunidade e confiança para a realização do estágio facultado pela Grohe Portugal.

Gostaria ainda de mostrar o meu apreço por todo o apoio, simpatia e disponibilidade de todos com quem tive o prazer de contactar na Grohe, e em especial ao Sérgio Costa, Sérgio Amador e Milton Rino, pelo interesse demonstrado e por todas as explicações cuidadas que contribuíram para o sucesso e enriquecimento deste trabalho.

Desejo também agradecer à minha namorada pela sua companhia e incentivo, determinante para concluir esta etapa.

Por último, dedico um agradecimento especial a toda a minha família e amigos por todo o apoio e força depositada em mim ao longo de todo o percurso académico.



## *Resumo*

Atualmente a vantagem competitiva de uma empresa passa pela sua rápida adaptação às variações de procura do mercado, sendo necessário garantir elevados níveis de produtividade e, simultaneamente, grande flexibilidade, indispensável ao fabrico de pequenos lotes. A necessidade de ajuste do processo e a diminuição da média de vida do produto levam a paragens cada vez mais frequentes da célula de fabrico para programação e afinação, com consequentes perdas de produtividade.

De forma a dar resposta a estes problemas, neste trabalho é testada a viabilidade da utilização da programação e simulação *offline* de tarefas de lixamento na Grohe Portugal, complementando a solução com o desenvolvimento de um novo método de afinação do programa, permitindo uma adaptação às flutuações do processo produtivo.

Para isso foi necessário analisar o estado da arte dos robôs industriais na área de acabamento superficial e respetivos métodos de programação.

Em seguida, após um trabalho prévio rigoroso de preparação e modelação da célula de trabalho, é possível fazer a programação *offline* das várias rotinas e trajetórias complexas que compõem um ciclo de lixamento de um produto, contribuindo para o aumento da qualidade do produto final sem comprometer os níveis de produtividade. Nesta dissertação são descritos e detalhados alguns dos procedimentos fulcrais no sucesso da aplicação deste método de programação.

Por último é feita uma nova abordagem ao método de ajuste ponto-a-ponto convencional, desenvolvendo-se para isso um sistema de ajuste automático do programa, dotando o robô da capacidade de se adaptar às variações do processo, assegurando a consistência do mesmo. Foram realizados testes em pequena escala, extrapolando-se os resultados para a aplicação deste novo método no processo produtivo da Grohe Portugal, como forma de complemento ao método convencional de ajuste ponto-a-ponto do programa, reduzindo o tempo de paragem da célula de trabalho.

***Palavras-Chave***

Lixamento, Robótica, ABB, KUKA, Programação *offline*, ABB RobotStudio, KUKA Sim, RAPID, KRL, TCP.

## *Abstract*

Currently the competitive advantage of a company lies in its rapid adaptation to changes in market demand, being necessary to ensure high levels of productivity and, simultaneously, great flexibility, which is essential to the manufacture of small batches. The need to adjust the process and the reducing average life of the product lead to increasingly frequent stoppages of manufacturing cells for programming and tuning, with consequent loss of productivity.

In order to tackle these problems, in this work is tested the feasibility of using the simulation and *offline* programming in grinding tasks at Grohe Portugal, complementing the solution above with the development of a new method of tuning the program, allowing adaptation to fluctuations in the production process.

For this it is necessary to analyze the state-of-the-art of industrial robots in the field of surface finishing and respective programming methods.

Then, after a previous and rigorous preparation and modeling of the work cell, it is possible to *offline* program the various routines and complex paths that comprise a grinding cycle of a product, thus contributing to increasing the quality of the final product without compromising productivity levels. In this thesis some of the key procedures are described and detailed to achieve the successful application of this programming method.

Finally is made a new approach to the point-to-point standard setting, developing for it a system of automatic adjustment of the program, giving the robot ability to adapt to the variations of the process, ensuring consistency of it. Tests were conducted on a small scale, extrapolating the results to the application of this new method in the production process of Grohe Portugal as a way to complement the conventional method of setting point-to-point program, reducing work cell downtime.

### ***Keywords***

Grinding, Robotics, ABB, KUKA, *offline* programming, ABB RobotStudio, KUKA Sim, RAPID, KRL, TCP.



# Índice

<b>AGRADECIMENTOS</b> .....	<b>V</b>
<b>RESUMO</b> .....	<b>VII</b>
<b>ABSTRACT</b> .....	<b>IX</b>
<b>ÍNDICE</b> .....	<b>XI</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>XV</b>
<b>ÍNDICE DE TABELAS</b> .....	<b>XXI</b>
<b>ACRÓNIMOS</b> .....	<b>XXIII</b>
<b>1. INTRODUÇÃO</b> .....	<b>1</b>
1.1. APRESENTAÇÃO DA GROHE PORTUGAL .....	1
1.2. CONTEXTUALIZAÇÃO .....	2
1.3. OBJETIVOS .....	2
1.4. CALENDARIZAÇÃO .....	3
1.5. ORGANIZAÇÃO DO RELATÓRIO .....	4
<b>2. PROGRAMAÇÃO DE ROBÔS INDUSTRIAIS E PROCESSOS DE LIXAMENTO</b> .....	<b>7</b>
2.1. ROBÔS INDUSTRIAIS .....	7
2.2. PROGRAMAÇÃO <i>ONLINE</i> .....	8
2.3. PROGRAMAÇÃO <i>OFFLINE</i> DE ROBÔS INDUSTRIAIS .....	10
2.3.1. <i>Implementação</i> .....	11
2.3.2. <i>Aplicações de Software de programação offline</i> .....	14
2.3.3. <i>Desvantagens e Limitações</i> .....	15
2.4. OPERAÇÕES DE ACABAMENTO SUPERFICIAL .....	16
2.5. AJUSTE DO PONTO DE REFERÊNCIA DA FERRAMENTA DO ROBÔ .....	18
2.5.1. <i>Dynalog AutoCal</i> .....	19
2.5.2. <i>Leoni tool calibration</i> .....	20
2.5.3. <i>Meta Vision Systems - Laser Sight</i> .....	21
2.6. CONTROLO DE FORÇA EM TAREFAS DE LIXAMENTO ROBOTIZADO.....	22
2.6.1. <i>Sistema de controlo de força da ABB</i> .....	22
<b>3. PROCESSO DE FABRICO E DEFINIÇÃO DO PROBLEMA</b> .....	<b>25</b>
3.1. APRESENTAÇÃO DO PROCESSO PRODUTIVO .....	25
3.1.1. <i>Fundição</i> .....	26
3.1.2. <i>Maquinagem</i> .....	28
3.1.3. <i>Acabamento superficial – lixamento/polimento</i> .....	30
3.1.4. <i>Galvânica</i> .....	34

3.1.5.	<i>Montagem</i> .....	36
3.2.	PRINCIPAIS PROBLEMAS NO LIXAMENTO ROBOTIZADO .....	36
3.2.1.	<i>Paragem dos robôs</i> .....	37
3.2.2.	<i>Dificuldades do lixamento robotizado</i> .....	37
3.2.3.	<i>Instabilidade e desvios dos processos de fundição e maquinação</i> .....	38
3.2.4.	<i>Níveis elevados de rework e sucata</i> .....	39
<b>4.</b>	<b>SIMULAÇÃO E PROGRAMAÇÃO OFFLINE</b> .....	<b>41</b>
4.1.	PREPARAÇÃO PARA A PROGRAMAÇÃO OFFLINE .....	41
4.1.1.	<i>A torneira e as ferramentas a utilizar</i> .....	42
4.1.2.	<i>Avaliação da célula de trabalho</i> .....	44
4.1.3.	<i>Modelação da célula de trabalho</i> .....	46
4.2.	CALIBRAÇÃO DA CÉLULA DE TRABALHO .....	59
4.2.1.	<i>Definição dos workobjects das unidades</i> .....	60
4.2.2.	<i>Definição da Tool utilizada</i> .....	62
4.3.	DESENVOLVIMENTO DO PROGRAMA .....	64
4.4.	PROCEDIMENTOS E RESULTADOS .....	71
4.4.1.	<i>Corpo da peça</i> .....	71
4.4.2.	<i>Bica da peça</i> .....	73
<b>5.</b>	<b>SISTEMA DE MEDIÇÃO E AJUSTE ONLINE DA FERRAMENTA DO ROBÔ</b> .....	<b>79</b>
5.1.	IDENTIFICAÇÃO DO PROBLEMA .....	79
5.2.	ABORDAGEM PROPOSTA .....	80
5.3.	PROGRAMAÇÃO E SIMULAÇÃO NO ROBOTSTUDIO .....	83
5.3.1.	<i>Modelação da célula de testes</i> .....	83
5.3.2.	<i>Medição da ferramenta</i> .....	84
5.3.3.	<i>Desvio da ferramenta</i> .....	87
5.3.4.	<i>Correção do TCP</i> .....	89
5.3.5.	<i>Resultados da simulação no RobotStudio</i> .....	90
5.4.	RESULTADOS DO TESTE NO ROBÔ .....	91
5.4.1.	<i>Verificação inicial dos pontos de teste</i> .....	91
5.4.2.	<i>Sensor de medição</i> .....	92
5.4.3.	<i>Medição da ferramenta</i> .....	92
5.4.4.	<i>Desvio da ferramenta</i> .....	93
5.4.5.	<i>Resultados – verificação final dos pontos de teste</i> .....	94
<b>6.</b>	<b>EXPANSÃO DA APLICAÇÃO DA PROGRAMAÇÃO OFFLINE</b> .....	<b>95</b>
6.1.	PROGRAMAÇÃO OFFLINE – CÉLULA COM ROBÔ KUKA .....	96
6.1.1.	<i>Célula de trabalho</i> .....	96
6.1.2.	<i>Modelação da célula de trabalho</i> .....	97
6.1.3.	<i>Desenvolvimento do programa</i> .....	102
6.1.4.	<i>Simulação offline e resultados</i> .....	106
6.2.	SIMULAÇÃO E PROGRAMAÇÃO OFFLINE DA TORNEIRA EURODISC COSMOPOLITAN .....	109
6.2.1.	<i>Estudo de uma nova ferramenta recorrendo à simulação</i> .....	110

6.2.2.	<i>Procedimentos e desenvolvimento do programa para lixamento da torneira.....</i>	<i>113</i>
6.3.	MODELAÇÃO DE NOVA CÉLULA PARA LIXAMENTO DE TORNEIRA ZEDRA.....	116
6.3.1.	<i>Célula de trabalho.....</i>	<i>117</i>
6.3.2.	<i>Modelação da célula.....</i>	<i>117</i>
6.3.3.	<i>Procedimentos e desenvolvimento do programa.....</i>	<i>122</i>
6.4.	ELIMINAÇÃO DE DESPERDÍCIOS – ALTERAÇÃO DA INTERFACE HOMEM-MÁQUINA .....	126
<b>7.</b>	<b>CONCLUSÃO E PERSPETIVAS DE TRABALHOS FUTUROS .....</b>	<b>131</b>
7.1.	SOLUÇÃO ADOTADA .....	131
7.2.	DIFICULDADES E SOLUÇÕES.....	132
7.3.	ANÁLISE DAS VANTAGENS E LIMITAÇÕES .....	132
7.4.	DESENVOLVIMENTOS FUTUROS .....	134
	<b>REFERÊNCIAS DOCUMENTAIS .....</b>	<b>137</b>
	<b>ANEXO A. PONTO DE REFERÊNCIA DA FERRAMENTA - TCP .....</b>	<b>139</b>
	<b>ANEXO B. DIMENSÕES DA PINÇA E TCP .....</b>	<b>141</b>



## Índice de Figuras

Figura 1	Calendarização das atividades do projeto.....	3
Figura 2	Programação <i>online</i> de um robô de lixamento.....	8
Figura 3	Ecrã da consola da ABB (à esquerda) e da KUKA (à direita) .....	9
Figura 4	<i>Lead through programming</i> – ABB (esquerda) e CMA robotics (direita).....	9
Figura 5	Exemplo de aplicação informática para programação <i>offline</i> .....	10
Figura 6	Modelo da célula de lixamento no programa KUKASim e a respetiva célula real .....	11
Figura 7	Opções de modelação disponíveis no RobotStudio.....	12
Figura 8	Caminho gerado automaticamente com base na aresta da peça .....	13
Figura 9	Ambiente de simulação no RobotStudio: controlos de simulação e análise dos sinais	14
Figura 10	Lixamento por cinta rotativa com roda flexível [8][9].....	17
Figura 11	Célula robotizada de lixamento da Mepsa [9].....	17
Figura 12	Célula robotizada de lixamento da SHL [10].....	18
Figura 13	Robô de lixamento com garra-ferramenta da SHL [10].....	18
Figura 14	Sistema de coordenadas da ferramenta [11].....	19
Figura 15	Autocal da Dynalog [12].....	20
Figura 16	Desvio do TCP apresentado no ecrã do sistema [13].....	20
Figura 17	Integração do sistema no processo de soldadura [13] .....	21
Figura 18	Meta Laser Sight - Sistema de medição da tocha [14] .....	21
Figura 19	Transdutor de força/binário [16] .....	23
Figura 20	Modo convencional, FC Pressure e FC SpeedChange [16] .....	23
Figura 21	Ilustração das fases do processo produtivo da Grohe.....	25
Figura 22	Forno de fusão.....	26
Figura 23	Produção dos “machos” .....	26
Figura 24	Vazamento Manual .....	27
Figura 25	Esquema demonstrativo do processo de vazamento sob baixa pressão .....	27
Figura 26	Corpos à saída do departamento de fundição .....	28
Figura 27	Maquinação dos corpos .....	29
Figura 28	Calibre “passa/não passa” .....	29
Figura 29	Corpos maquinados .....	30
Figura 30	Lixamento robotizado .....	30
Figura 31	Corpos com zona a retocar após o lixamento robotizado.....	31
Figura 32	Punhos e pinças para o lixamento robotizado .....	32
Figura 33	Polimento automático.....	33
Figura 34	Controlo visual de defeitos.....	33

Figura 35	Polimento manual – recuperação da peça .....	34
Figura 36	Corpos nas suspensões – processo de galvanização .....	35
Figura 37	Corpos cromados no final do processo da galvânica.....	35
Figura 38	Linha de montagem em “U”.....	36
Figura 39	Teste funcional das torneiras termostáticas .....	36
Figura 40	Movimento limitado ao atingir o limite do eixo 5 (+120°).....	38
Figura 41	Degradação da pinça devido ao contacto com a lixa.....	38
Figura 42	Corpos à saída do robô de lixamento (vista detalhada à direita) .....	39
Figura 43	Torneira de lavatório da coleção EuroSmart – Grohe .....	42
Figura 44	Corpo da torneira (esquerda) e modelo 3D da mesma (direita).....	42
Figura 45	Costas e parte de baixo do corpo da torneira.....	43
Figura 46	Punho a 45° e possibilidade de acesso ao topo da bica.....	44
Figura 47	Pinça expansível para fixação do corpo .....	44
Figura 48	Célula robotizada de lixamento “Mepsa 13” .....	45
Figura 49	Robô ABB IRB 4400/45 1,96 m [17].....	46
Figura 50	Desenho do perfil de metade da roda .....	47
Figura 51	Espelho e revolução do desenho de perfil .....	47
Figura 52	Montagem dos componentes .....	48
Figura 53	Conjunto do braço e roda da unidade de trabalho quatro .....	48
Figura 54	Conjunto do braço e roda da unidade de trabalho quatro no RobotStudio .....	49
Figura 55	Biblioteca de modelos do RobotStudio .....	49
Figura 56	Configuração dos parâmetros do robô no RobotStudio.....	50
Figura 57	Parâmetros dos elos do robô.....	50
Figura 58	Compensação do atrito das juntas .....	51
Figura 59	Valores de calibração do robô utilizado .....	51
Figura 60	Calibração dos eixos do robô [18].....	52
Figura 61	Configuração de um sinal digital de saída.....	52
Figura 62	Sinais digitais de saída para as quatro unidades de trabalho .....	53
Figura 63	Sinais da unidade virtual no simulador de I/O .....	53
Figura 64	Punho com cunha a 45° montado no robô .....	54
Figura 65	Limite do eixo 5 – colisão entre o sistema anti rotação do punho e o eixo 4 .....	54
Figura 66	Corpo da torneira na pinça expansível e localização do TCP .....	55
Figura 67	Resolução gráfica da roda de contacto no RobotStudio .....	56
Figura 68	Propriedades gráficas de uma unidade de lixamento no RobotStudio .....	56
Figura 69	Simplificação do modelo da unidade de lixamento no Inventor .....	57
Figura 70	Resolução gráfica da roda após alterado o nível de detalhe no RobotStudio .....	57
Figura 71	Propriedades gráficas da unidade de lixamento simplificada.....	58
Figura 72	Monitorização de colisões do robô com a estrutura da célula .....	58
Figura 73	Modelo final da célula de lixamento “Mepsa 13” .....	59

Figura 74	Localização dos <i>Workobjects</i> e TCP definidos no modelo .....	59
Figura 75	Dispositivo de definição de <i>workobjects</i> .....	60
Figura 76	Dispositivo de definição dos <i>workobjects</i> montado no braço da unidade.....	60
Figura 77	Localização do <i>workobject</i> face à roda da unidade de lixamento .....	61
Figura 78	Definição dos <i>workobjects</i> da unidade de lixamento nº1 na célula “Mepsa 13”.....	61
Figura 79	Execução do programa de teste de precisão e repetibilidade de posicionamento do robô no RobotStudio .....	62
Figura 80	Resultados da execução do programa de teste de precisão e repetibilidade de posicionamento na célula “Mepsa 13” .....	62
Figura 81	Método de definição do TCP através de seis pontos [11] .....	63
Figura 82	Menus de definição de uma nova <i>tool</i> na consola do controlador S4CPlus.....	63
Figura 83	Processo de definição da nova <i>tool</i> na célula “Mepsa 13”.....	64
Figura 84	Principais faces do corpo a lixar.....	65
Figura 85	Pontos de controlo da remoção de material.....	65
Figura 86	Reorientação da ferramenta do robô (e do corpo da torneira) no RobotStudio.....	66
Figura 87	Criação de um <i>target</i> .....	66
Figura 88	Vistas definidas para a visualização do centro da roda da unidade de lixamento n.º 1	67
Figura 89	Duas vistas distintas para o mesmo <i>target</i> .....	67
Figura 90	Interpolação ao nível das juntas [11].....	68
Figura 91	Interpolação circular [11].....	68
Figura 92	Alternativas para a configuração das instruções de movimento.....	69
Figura 93	Diferentes configurações do robô para um <i>target</i> .....	69
Figura 94	Ilustração esquemática do funcionamento da rotina de “zona”.....	70
Figura 95	Ajuste dos <i>targets</i> no RobotStudio (esquerda) e na consola do robô (direita).....	71
Figura 96	Costas do corpo da peça a lixar .....	72
Figura 97	Rotinas de lixamento do corpo e cantos da peça.....	72
Figura 98	Resultado após o processo de lixamento com a lixa grossa .....	73
Figura 99	Bica da peça a lixar .....	74
Figura 100	Sequência da rotina de lixamento da parte de baixo da bica na unidade de lixamento n.º 1 .....	74
Figura 101	Resultados do lixamento da parte de baixo da bica após a primeira etapa.....	75
Figura 102	Rotina de lixamento por arrasto cruzado na unidade de lixamento n.º 2 .....	75
Figura 103	Resultado do lixamento da parte de baixo da bica após a segunda etapa.....	76
Figura 104	Evolução da superfície da bica da peça ao longo das três etapas de lixamento .....	76
Figura 105	Peça polida .....	77
Figura 106	Desvios dimensionais do corpo da torneira.....	80
Figura 107	Localização do ponto de referência da ferramenta.....	80
Figura 108	Posição do corpo face ao TCP programado (desvio ampliado) .....	81
Figura 109	Ângulos de Euler – <i>Roll</i> , <i>Pitch</i> e <i>Yaw</i> [20].....	81

Figura 110	Medição do ângulo de rotação “ <i>ry</i> ” em relação ao TCP definido .....	82
Figura 111	Sequência de teste do sistema de ajuste do TCP .....	83
Figura 112	Componentes da célula de teste.....	84
Figura 113	Medição da ferramenta e <i>tools</i> definidas ( <i>tool</i> base, meio e ponta) .....	84
Figura 114	Fluxograma representativo do funcionamento da rotina de medição e interrupção .	85
Figura 115	Contacto da ferramenta com a ponteira do sensor no ambiente de simulação .....	86
Figura 116	Definição de um novo evento.....	86
Figura 117	Eventos de medição – início e fim de colisão com o sensor .....	87
Figura 118	Simulação da colisão com a palete .....	87
Figura 119	Desvio do ferramenta através de um novo evento de colisão com a palete.....	88
Figura 120	Eventos definidos para a simulação .....	88
Figura 121	Desvio do TCP face à posição real da ferramenta.....	88
Figura 122	Esquema das medidas efetuadas antes e após o desvio (desvio aumentado) .....	89
Figura 123	Estrutura de dados <i>tooldata</i> [21] .....	89
Figura 124	Resultados dos testes realizados em ambiente de simulação .....	90
Figura 125	Robô IRB 140 em posição de calibração (todas as juntas a zero) .....	91
Figura 126	Posição da ferramenta no início do teste .....	91
Figura 127	Sensor do sistema de medição da ferramenta.....	92
Figura 128	Medição do topo e da base da ferramenta .....	93
Figura 129	Rotina de colisão da ponteira com a palete .....	93
Figura 130	Posição da ferramenta no fim do teste.....	94
Figura 131	Célula robotizada de lixamento - “KUKA 4” .....	96
Figura 132	Funcionalidades para criação de um componente no KUKA Sim .....	97
Figura 133	Propriedades do sólido cilíndrico .....	98
Figura 134	Diferenças do modelo da roda – 12 secções (esquerda), 48 secções (direita).....	98
Figura 135	Modelo simplificado da unidade de lixamento n.º 1 no Inventor.....	99
Figura 136	Modelo das unidades 1 e 2 no KUKA Sim (à esquerda) e na célula real (à direita) ....	100
Figura 137	Acoplamento do punho ao robô .....	100
Figura 138	Introdução numérica dos dados da ferramenta no robô [22] .....	101
Figura 139	Localização das referências das unidades (base) e da peça ( <i>tool</i> ) .....	101
Figura 140	Modelo final da célula de lixamento “KUKA 4” .....	102
Figura 141	Conceção do programa na aba “ <i>teach</i> ” do KUKA Sim.....	102
Figura 142	Movimento PTP com paragem exata [23].....	103
Figura 143	Movimento LIN com paragem exata [23] .....	103
Figura 144	Configurações do robô para o mesmo <i>target</i> .....	104
Figura 145	Exemplo de comentário apresentado na janela de <i>debug</i> durante a simulação .....	104
Figura 146	Sequência reordenada.....	105
Figura 147	Comando “ <i>Touch-Up</i> ” .....	105

Figura 148	Lixamento da parte de cima da bica .....	106
Figura 149	Lixamento do contorno entre o corpo e a bica .....	107
Figura 150	Lixamento da parte de baixo da bica por arrasto.....	107
Figura 151	Deteção de colisão com a peça à esquerda e com o robô à direita .....	108
Figura 152	Exemplo de posicionamento do robô num <i>target</i> para o lixamento do corpo.....	108
Figura 153	Torneira da coleção Eurodisc Cosmopolitan (à esquerda) e modelo 3D da mesma (à direita) .....	109
Figura 154	Face da torneira não contemplada no processo de lixamento robotizado .....	110
Figura 155	Punho e peça a lixar no modelo de simulação.....	110
Figura 156	Lixamento do corpo na unidade n.º 1 (lixa grossa) .....	111
Figura 157	Lixamento do corpo na unidade n.º 4 (lixa fina) .....	111
Figura 158	Resultado obtido (posicionamento final da peça) .....	112
Figura 159	Pontos de referência definidos para a peça.....	112
Figura 160	Sequência do processo de lixamento da torneira Eurodisc Cosmopolitan .....	113
Figura 161	Rotina de lixamento do corpo (lixa grossa).....	114
Figura 162	Rotina de lixamento da parte de baixo da bica.....	115
Figura 163	Espelhamento da rotina de lixamento da parte de baixo da bica.....	115
Figura 164	Rotina de acabamento do contorno .....	116
Figura 165	Torneira de cozinha da coleção Zedra – Grohe (esquerda) e modelo 3D da mesma (direita) .....	116
Figura 166	Célula robotizada de lixamento – “Mepsa 12” .....	117
Figura 167	Parte superior da unidade de trabalho n.º 1 .....	118
Figura 168	Configuração do mecanismo da unidade de trabalho – junta prismática .....	119
Figura 169	Definição das posições e tempos de transição das unidades de lixamento .....	119
Figura 170	Configuração do mecanismo do punho – junta rotacional .....	120
Figura 171	Definição das posições e tempos de transição do mecanismo do punho.....	121
Figura 172	Localização do TCP – junta a 0º .....	121
Figura 173	Localização do TCP – junta a 90º .....	122
Figura 174	Modelo final da célula de lixamento “Mepsa 12” .....	122
Figura 175	Sequência do processo de lixamento da torneira Zedra – Grohe .....	123
Figura 176	<i>Target</i> da rotina de lixamento do corpo (junta do punho a 0º).....	124
Figura 177	<i>Target</i> da rotina de lixamento da zona do cartucho (junta do punho a 90º).....	125
Figura 178	Colisão com a unidade de trabalho n.º2 .....	125
Figura 179	Paragem dos robôs por falta de abastecimento ou para troca de lixas .....	126
Figura 180	Disposição dos postos de lixamento manual face às células robotizadas.....	127
Figura 181	Ecrã da consola existente utilizado como base.....	128
Figura 182	Ecrã da consola desenvolvido .....	129
Figura 183	Implementação do sistema de ajuste automático do programa .....	134



## *Índice de Tabelas*

Tabela 1	Exemplos de aplicações de <i>software</i> proprietário de programação <i>offline</i> .....	15
Tabela 2	Comparativo do custo total entre os dois métodos de programação .....	133
Tabela 3	Comparativo da produção de 1000 peças.....	133



## *Acrónimos*

3D	–	Três dimensões
ABB	–	Asea Brown Boveri
CAD	–	<i>Computer Aided Design</i>
CAM	–	<i>Computer Aided Manufacturing</i>
CNC	–	<i>Computer Numerical Control</i>
FANUC	–	<i>Fujitsu Automated Numerical Control</i>
FC	–	<i>Force Control</i>
HMI	–	<i>Human-Machine Interface</i>
ISEP	–	Instituto Superior de Engenharia do Porto
KRL	–	<i>KUKA Robot Language</i>
LASER	–	<i>Light Amplification by Stimulated Emission of Radiation</i>
PTP	–	<i>Point-to-Point</i>
STEP	–	<i>Standard for the Exchange of Product Data</i>
STL	–	<i>STereoLithography</i>
TCP	–	<i>Tool Center Point</i>



# 1. INTRODUÇÃO

O presente trabalho, realizado no âmbito da unidade curricular de Tese/Dissertação, foi realizado no departamento de lixamento/polimento da empresa Grohe Portugal, Componentes Sanitários Lda. Este documento descreve o desenvolvimento e implementação da programação *offline* dos robôs do departamento e de um sistema de medição e ajuste da ferramenta do robô.

Este capítulo inclui uma breve apresentação da empresa Grohe Portugal, seguindo-se uma contextualização do trabalho, apontando-se as motivações e necessidades existentes que deram origem ao mesmo. Apresentam-se também os objetivos estipulados, bem como a calendarização do trabalho e, por fim, um resumo da organização do presente documento.

## 1.1. APRESENTAÇÃO DA GROHE PORTUGAL

A Grohe AG é uma empresa multinacional, fundada em 1936 por Friedrich Grohe na Alemanha. É um dos mais prestigiados grupos europeus do sector metalúrgico e é líder mundial na produção de produtos e sistemas técnicos sanitários. A Grohe Portugal, Componentes Sanitários Lda é uma das fábricas do grupo, localizada em Albergaria-a-Velha, tendo sido inaugurada em 1998 e ampliada em 2004 para uma capacidade de produção de 5 milhões de unidades por ano. Os elevados níveis de produtividade, a competitividade de custos de produção, aliados à experiência entretanto adquirida, foram responsáveis pela expansão da empresa. A Grohe tem como objetivo assegurar a sua liderança no mercado, através de uma constante otimização dos seus custos de produção.

Com base nesta filosofia de melhoria contínua, surgiu a necessidade de otimizar o processo de acabamento superficial – desbaste e lixamento do corpo da torneira.

## **1.2. CONTEXTUALIZAÇÃO**

Este trabalho surge da necessidade do departamento de lixamento/polimento da Grohe Portugal aumentar a sua flexibilidade e produtividade de forma a responder às novas exigências do mercado. A produção de pequenos lotes e a entrada em produção de novos produtos conduz a longas paragens das células de produção, havendo a oportunidade de diminuir o tempo de paragem das células para programação de novos produtos pela introdução de novos métodos de programação de robôs (*offline*) e, pelo desenvolvimento de um sistema que permitirá o ajuste automático do programa do robô às variações do processo, permitindo uma melhor qualidade de acabamento e diminuindo também o tempo de paragem para ajustes/afinações. Com isto espera-se contribuir para o aumento da capacidade de resposta do respetivo departamento.

## **1.3. OBJETIVOS**

A necessidade de ultrapassar os principais problemas do departamento de lixamento/polimento para reduzir os seus tempos de paragem e aumentar a qualidade do seu processo, motivam a realização do presente trabalho que estabelece os seguintes objetivos:

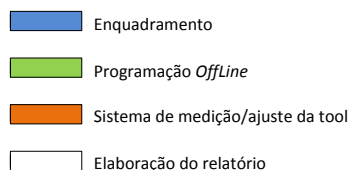
- Conhecer o processo de produção da Grohe de forma a analisar os pontos de melhoria;
- Analisar em detalhe o processo de lixamento (robotizado e manual), devido à complexidade que este processo exige para a obtenção de um produto final com os melhores níveis de qualidade;
- Desenvolver programas de lixamento *offline* de novos produtos;
- Otimização dos programas existentes, procurando diminuir os tempos de ciclo e, simultaneamente, aumentar a qualidade do acabamento superficial;

- Desenvolvimento de um sistema de medição e correção do *offset* da ferramenta do robô (*Tool Center Point – TCP*) diminuindo os tempos de paragem e aumentando a qualidade do produto final.

## 1.4. CALENDARIZAÇÃO

A distribuição das várias tarefas necessárias à realização do presente trabalho pode ser observada no seguinte diagrama de Gantt. É de salientar neste diagrama as três principais etapas: estudo do processo de produção e análise dos programas de lixamento já existentes, programação *offline* e o desenvolvimento do sistema de medição e ajuste da ferramenta do robô.

ID	Tarefa	Inicio	Fim	Duração	2012		2013					
					Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun
1	Estudo do processo de lixamento	22-10-2012	21-12-2012	45d	█							
2	Análise dos programas existentes	05-11-2012	17-12-2012	31d	█							
3	Otimização de um programa - offline	12-11-2012	21-12-2012	30d	█							
4	Ajuste/Afinação de programas online	26-11-2012	30-01-2013	48d	█							
5	Desenvolvimento de um programa offline	26-11-2012	11-01-2013	35d	█							
6	Estudo de melhorias no processo de lixamento	03-12-2012	21-01-2013	36d	█							
7	Desenvolvimento de um programa offline para um novo produto	14-01-2013	08-03-2013	40d			█					
8	Desenvolvimento de um sistema de medição da ferramenta e corpo	04-02-2013	25-03-2013	36d			█					
9	<b>Ponto Situação- Programa e S.Medição</b>	<b>25-03-2013</b>	<b>25-03-2013</b>	0d			◆					
10	Teste do programa e ajustes	04-03-2013	29-03-2013	20d			█					
11	Otimizações do código e tempos de ciclo	25-03-2013	22-04-2013	21d			█					
12	<b>Conclusão preliminar do novo programa</b>	<b>22-04-2013</b>	<b>22-04-2013</b>	0d			◆					
13	Desenvolvimento das rotinas necessárias para sistema de medição	22-04-2013	17-05-2013	20d			█					
14	Implementação do sistema de medição	15-05-2013	24-05-2013	8d			█					
15	Testes e Ajustes do sistema de medição	20-05-2013	31-05-2013	10d			█					
16	<b>Conclusão do sistema de medição</b>	<b>31-05-2013</b>	<b>31-05-2013</b>	0d			◆					
17	Realização da documentação necessária, Ajustes e Otimizações	03-06-2013	28-06-2013	20d			█					
18	Elaboração do relatório final	04-03-2013	28-06-2013	85d			█					



**Figura 1 Calendarização das atividades do projeto**

## 1.5. ORGANIZAÇÃO DO RELATÓRIO

Este relatório é composto por sete capítulos cujo conteúdo se descreve de forma sucinta de seguida.

Neste primeiro capítulo é feita uma breve introdução ao presente trabalho e à empresa Grohe, apresentando-se a contextualização, os objetivos, a calendarização e a organização deste documento.

No capítulo seguinte, o segundo, é apresentado o estado de desenvolvimento dos processos de simulação e programação *offline* de robôs industriais, das operações de acabamento superficial, das soluções de deteção e ajuste dos dados da ferramenta do robô sendo, por fim, abordada a temática do controlo de força em tarefas de lixamento robotizado.

O Capítulo 3 consiste na definição do problema, apresentando-se o processo produtivo da Grohe Portugal, identificando-se os principais problemas nas tarefas de lixamento robotizado, relacionando-os com a qualidade do produto final e de produtividade do departamento.

No Capítulo 4, “Programação e Simulação *Offline*”, apresenta-se o desenvolvimento de um programa *offline* para um produto, detalhando-se as etapas necessárias para a sua implementação e obtenção de um bom acabamento da peça, contribuindo para uma melhoria contínua do departamento.

No Capítulo 5 descreve-se a conceção do sistema de medição e ajuste da ferramenta do robô, iniciando-se pela identificação do problema, passando à formulação de uma abordagem de teste, seguindo-se a programação e simulação *offline* do sistema, com a programação em ambiente gráfico e em RAPID e, por fim, os resultados da correção na simulação e na célula real.

O Capítulo 6 pretende dar continuidade ao trabalho da aplicação da programação *offline*, sendo explorado um novo software de programação e simulação *offline* para robôs KUKA, desenvolvidos novos programas utilizando as potencialidades do método ainda não exploradas e, por fim, é alterado o interface com o utilizador de forma a reduzir os desperdícios de produção, um dos conceitos de *lean manufacturing*.

No último capítulo, o sétimo, são reunidas as principais conclusões e perspectivas possíveis futuros desenvolvimentos na área do lixamento robotizado para o respetivo departamento.



## 2. PROGRAMAÇÃO DE ROBÔS INDUSTRIAIS E PROCESSOS DE LIXAMENTO

Ao longo deste capítulo será feita uma breve introdução ao desenvolvimento e estado atual dos robôs industriais afetos às operações de acabamento superficial e aos métodos de programação utilizados, sendo por fim abordadas as soluções de detecção e ajuste da ferramenta do robô.

### 2.1. ROBÔS INDUSTRIAIS

O aumento da concorrência e a necessidade de melhorar a produtividade fizeram com que os robôs industriais se tornassem parte importante na produção e montagem, na medida em que são, por definição, máquinas versáteis e eficientes, tendo-se assistido a um crescente interesse nesta área. Pode definir-se, genericamente, um robô manipulador como um mecanismo automatizado, mecanicamente concebido para mover peças ou ferramentas através de uma sequência de movimentos ou operações pretendidas.

A flexibilidade do robô é, em grande parte, dada pelos tipos de movimentos e operações que podem ser programados no controlador e pela facilidade com que o programa pode ser

criado e alterado. A programação é feita recorrendo a linguagens de programação proprietárias de alto nível, podendo a mesma ser feita em modo *online* ou *offline*.

## 2.2. PROGRAMAÇÃO *ONLINE*

A programação *online* refere-se a ensinar uma trajetória ao robô através da sua movimentação pelo *joystick* da consola ou dispositivo semelhante. O robô é movimentado manualmente para as posições desejadas, memorizando-se os vários pontos pretendidos na memória do controlador, como se mostra na Figura 2.



**Figura 2** Programação *online* de um robô de lixamento

O programador pode manobrar o robô selecionando diferentes sistemas de coordenadas, como o da base do robô, objeto de trabalho, ferramenta, ou outros ainda, permitindo uma movimentação de forma intuitiva consoante o posicionamento a efetuar [1]. Durante este processo podem também ser adicionadas instruções de interação com outros equipamentos da célula. A programação é feita recorrendo a linguagens proprietárias, sendo este método o mais utilizado atualmente. O facto de serem usadas linguagens proprietárias, específicas de cada construtor, e a falta de universalidade entre as mesmas, dificulta o desenvolvimento pela necessidade de *know-how* específico e dificulta também a portabilidade dos programas entre equipamentos de diferentes fabricantes. Na Figura 3 apresenta-se, como exemplo, um extrato de um programa escrito em linguagem RAPID, proprietária da Asea Brown Boveri (ABB) e outro extrato em KUKA Robot Language (KRL), proprietária da KUKA, sendo estes programas de robôs de lixamento do departamento.

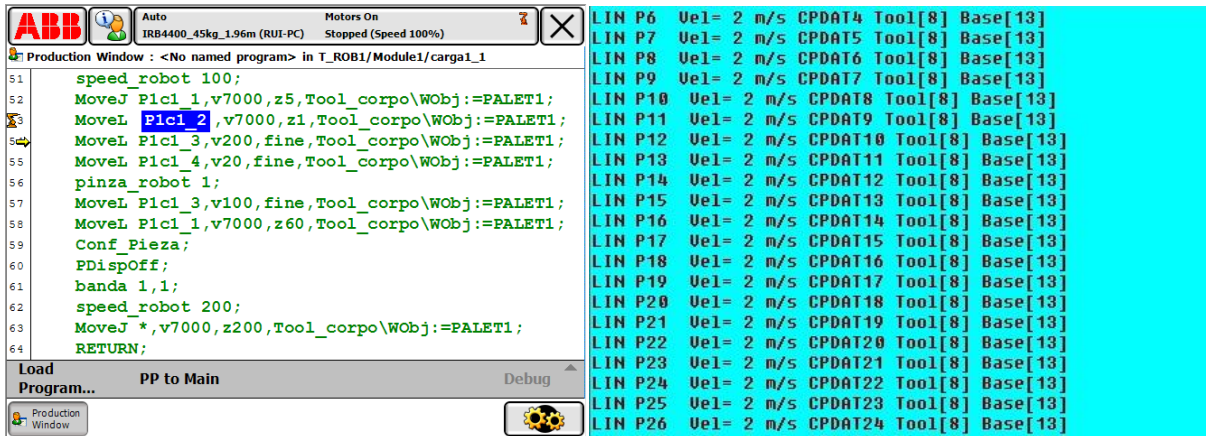


Figura 3 Ecrã da consola da ABB (à esquerda) e da KUKA (à direita)

Este método de programação pode consumir muito tempo, dependendo da complexidade da aplicação final, podendo assim levar à paragem produtiva do robô por longos períodos. Para a produção de lotes pequenos e médios, estes tempos de paragem para *setup* e reprogramação podem ser substanciais, tendo assim um impacto negativo nos níveis de produtividade e consequentemente no custo do produto.

Como alternativa ao método convencional, de forma a reduzir o tempo de programação no caso de trajetórias complexas, pode-se recorrer à cooperação entre o operador e o robô, sendo este capaz de mover o robô manualmente, aplicando força na sua extremidade ou ferramenta, guiando-a ao longo da trajetória pretendida, como se demonstra na Figura 4. Esta técnica designa-se por *lead through programming*, sendo geralmente aplicada em processos de maquinação, soldadura e pintura [2][3].

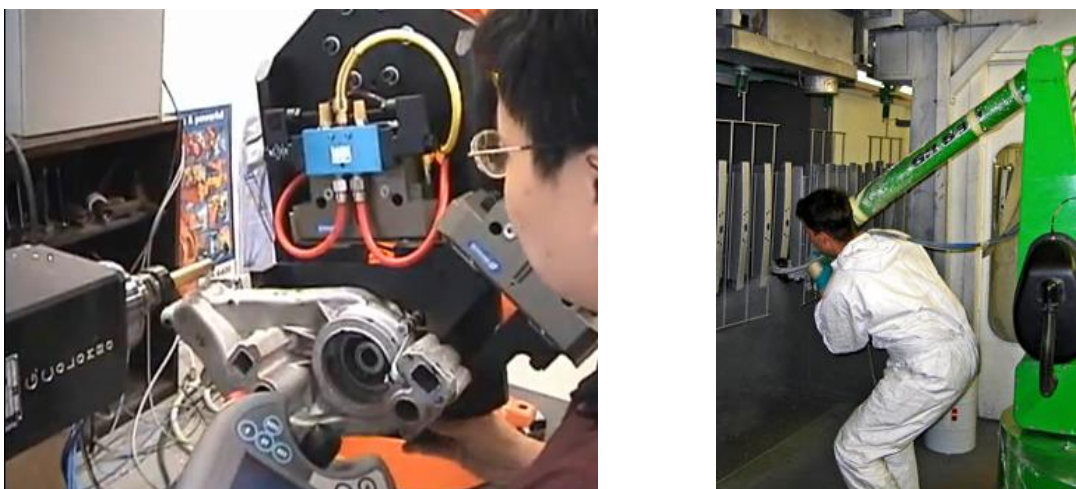


Figura 4 *Lead through programming* – ABB (esquerda) e CMA robotics (direita)

### 2.3. PROGRAMAÇÃO *OFFLINE* DE ROBÔS INDUSTRIAIS

Conforme já foi abordado, a indústria tem que responder aos mercados com produtos variados e em lotes cada vez mais pequenos, tendo para isso que conseguir mudar rapidamente o seu processo produtivo de acordo com as exigências do mercado. Assim, a utilização do robô vem trazer a flexibilidade necessária, podendo o mesmo atingir elevados níveis de produtividade necessários para diminuir os custos de fabrico. De forma a atingir elevados níveis de produtividade deve-se restringir o tempo de paragem do robô ao mínimo necessário. Uma das tarefas mais demoradas é a própria programação *online* do robô, onde mesmo um programador experiente pode ocupar o mesmo durante dezenas de horas. Isto, aliado à constante mudança de *setup*, traduz-se em tempos de utilização muito abaixo do esperado, comprometendo quer o preço por peça, quer a resposta do departamento às variações da procura. A programação *offline* surge assim como forma de minimizar estas desvantagens no contexto do mercado atual.

A programação *offline* pode ser definida como o processo pelo qual os programas do robô são desenvolvidos sem exigir a utilização do próprio robô [4]. Os programas são desenvolvidos numa plataforma que permite a simulação da célula de trabalho (Figura 5), com procedimentos similares aos utilizados na programação *online*.

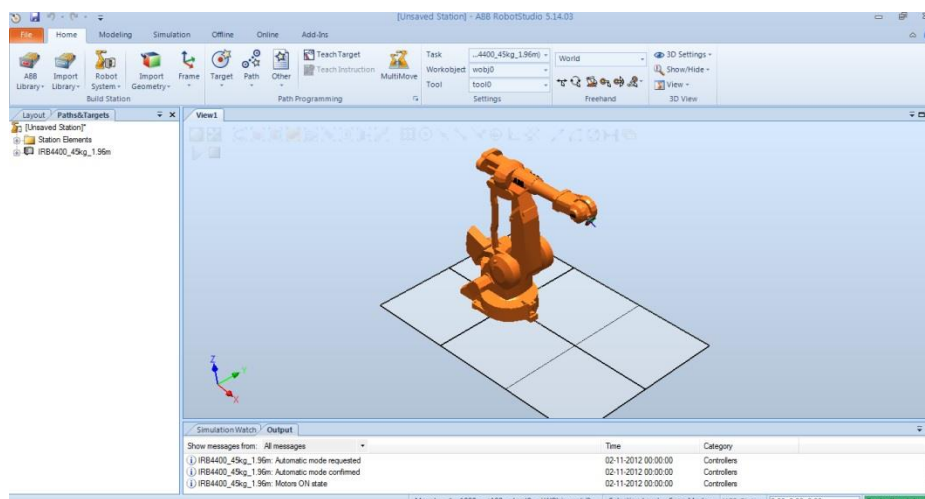


Figura 5 Exemplo de aplicação informática para programação *offline*

Desta forma os tempos de paragem para programação são reduzidos, sendo necessária uma curta paragem para calibração do modelo simulado e para testes do programa. Em comparação com a programação *online*, esta permite também uma maior segurança do

operador, quer pela diminuição do tempo de exposição num ambiente potencialmente perigoso, quer pela deteção de colisões, evitando desta forma acidentes.

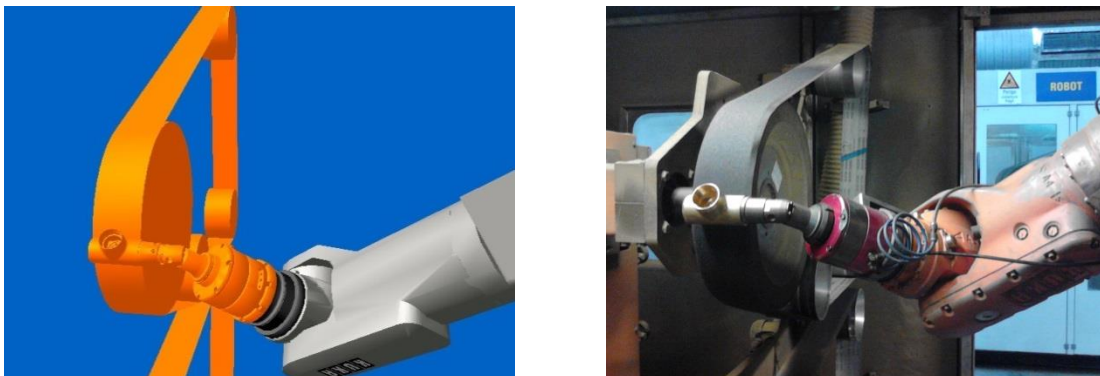
Este método de programação tem também vantagens ao nível da otimização dos programas pela facilidade de simulação dos vários cenários, podendo explorar-se diferentes trajetórias, *layouts* da célula e sequências das tarefas.

### 2.3.1. IMPLEMENTAÇÃO

A implementação da programação *offline* pode ser dividida em três etapas principais: modelação da célula de trabalho, programação do robô e simulação do programa gerado.

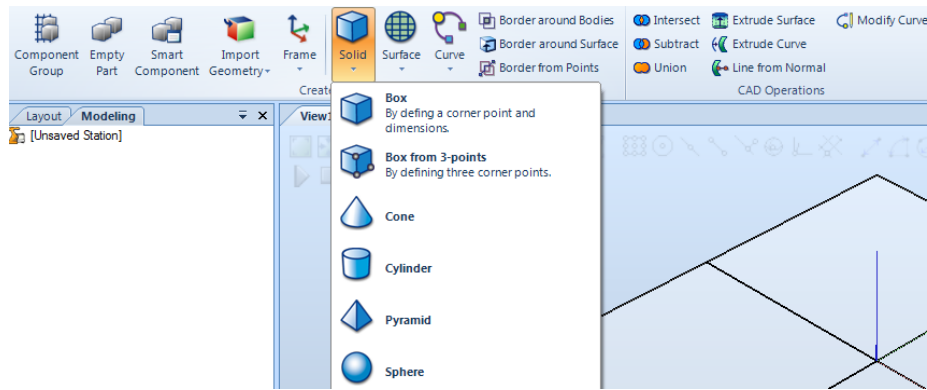
#### 2.3.1.1. Modelo da célula de trabalho

A programação *offline* começa necessariamente pela criação de um modelo 3D da célula robotizada, antes do início da programação. Com base nesta modelação será possível programar e simular todos os movimentos e interações do robô com os elementos da célula de trabalho, devendo a mesma ser precisa e possuir o detalhe que a aplicação a desenvolver exige. Na Figura 6 apresenta-se uma modelação no programa KUKASim e a respetiva célula de trabalho real, podendo observar-se o grau de detalhe conferido ao modelo.



**Figura 6** Modelo da célula de lixamento no programa KUKASim e a respetiva célula real

Este modelo é desenvolvido importando-se os vários modelos geométricos dos equipamentos da célula a partir de sistemas *Computer Aided Design* (CAD), por exemplo através de ficheiros neutros como o Standard for the Exchange of Product Data (STEP). Podem também ser utilizadas as ferramentas CAD existentes nos próprios programas, sendo apresentadas na Figura 7 as opções de modelação do RobotStudio, como a criação rápida de alguns sólidos e as operações de CAD disponíveis.

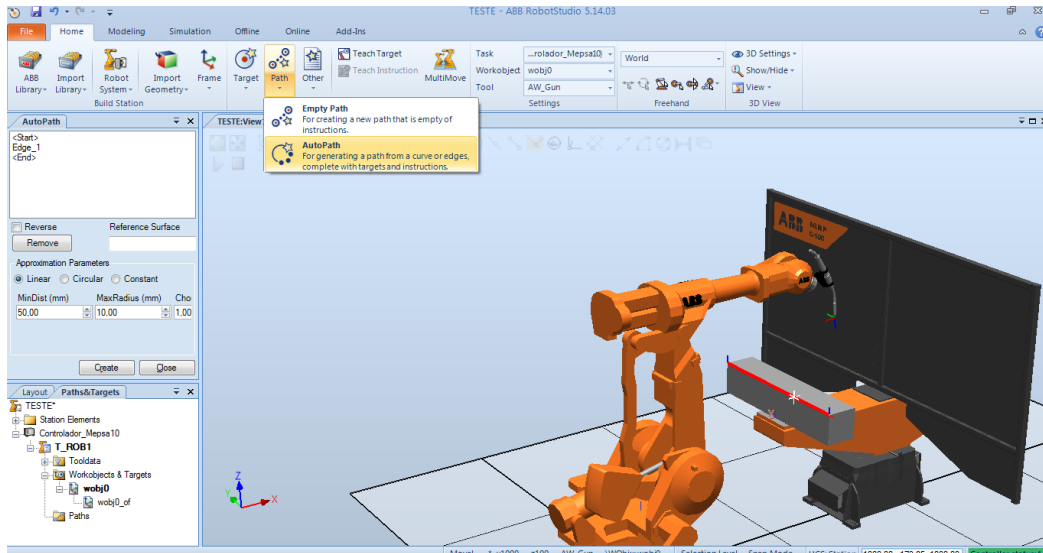


**Figura 7 Opções de modelação disponíveis no RobotStudio**

### 2.3.1.2. Programação

Após criado o modelo da célula robotizada, pode-se dar início à programação, utilizando as funções de alto nível disponibilizadas pelo programa ou utilizando linguagem nativa do controlador. De seguida, apresenta-se uma listagem de algumas características e funções disponibilizadas por grande parte dos programas de simulação e programação *offline* de robôs:

- Utilização de *smart components*, tendo estes um comportamento específico em função de sinais da célula ou outros eventos;
- Os pontos alvo ou *targets* podem ser criados automaticamente, utilizando o modelo 3D, podendo escolher-se os vértices e arestas de um sólido por exemplo;
- Ajuste dos *targets* e dos caminhos, ou *path*, de forma mais intuitiva, através da edição da posição, orientação, realização de grupos e outras funcionalidades;
- Possibilidade de criar os caminhos a percorrer automaticamente, gerando os *targets* e instruções necessárias, com base na geometria do componente, como se apresenta na Figura 8;
- Importação de caminhos gerados por aplicações de *software* de *computer-aided design/computer-aided manufacturing* (CAD/CAM) para processos de maquinação;
- Permite adicionar mecanismos, mesas rotativas, tapetes, entre outros periféricos.



**Figura 8 Caminho gerado automaticamente com base na aresta da peça**

Tal como na programação *online*, o programador deve ter conhecimentos acerca da tarefa, do processo, da célula e das ferramentas que tem à sua disposição, procurando a melhor abordagem para atingir o objetivo.

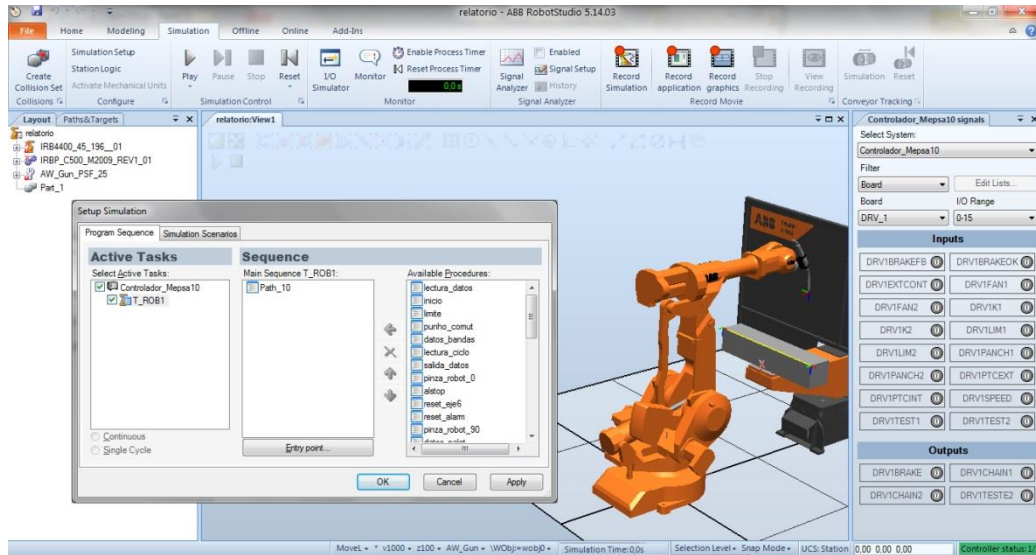
Após a marcação dos pontos e trajetos e realizadas as configurações necessárias, o programa é compilado para a linguagem do robô seguindo-se a sincronização com o controlador do mesmo.

### 2.3.1.3. Simulação

Uma das grandes vantagens da programação *offline* é a possibilidade de simular e validar a programação desenvolvida no modelo 3D da célula, sem que para isso seja necessário parar a célula robotizada. Antes da simulação é feita a verificação do programa, caminhos e instruções. Apresentam-se de seguida algumas das funcionalidades disponíveis na simulação:

- Análise da cinemática do robô, limites das juntas e singularidades;
- Simulação e análise dos sinais da célula (digitais e analógicos), tendo como exemplo o simulador de sinais do RobotStudio, apresentado na Figura 9;
- Cálculo do tempo de ciclo, sendo possível alterar rapidamente a sequência das operações do processo e consequentemente testar várias abordagens, através da interface de escolha de sequência que se pode observar na Figura 9;

- Gestão de eventos, tendo-se por exemplo a possibilidade de efetuar a modificação da pose ou posição de um mecanismo, aquando de uma colisão com um elemento da célula ou ao atingir um determinado tempo de simulação.



**Figura 9 Ambiente de simulação no RobotStudio: controlos de simulação e análise dos sinais**

O RobotStudio, por exemplo, utiliza como base um controlador virtual que é uma cópia exata do controlador do robô real (o RobotWare), permitindo obter simulações realistas, refletindo o comportamento do robô real. É também possível escolher qual das soluções da cinemática inversa é mais vantajosa para alcançar o *target* pretendido, evitando por exemplo uma colisão. Mesmo com estas vantagens que aproximam a realidade e a simulação *offline*, podem ainda assim existir desvios, em grande parte devido aos erros cometidos na modelação do “mundo”. Assim sendo, deve-se proceder à correção dos parâmetros do programa, como os referenciais de trabalho, e dados da ferramenta de acordo com a célula real. Para isso, as ferramentas de programação, juntamente com os robôs, devem dispor de algoritmos e procedimentos de calibração, melhorando o resultado final da simulação.

### **2.3.2. APLICAÇÕES DE SOFTWARE DE PROGRAMAÇÃO OFFLINE**

As aplicações de *software* atualmente disponíveis para programação *offline* são plataformas que permitem a programação e simulação da célula de trabalho, possibilitando o uso de instruções de alto nível para a definição de trajetórias complexas, bem como de procedimentos e linguagem de programação similares aos utilizados na programação *online*. O *software* pode ser classificado como dedicado, quando é vocacionado para uma

tarefa específica, ou classificado como genérico, podendo ainda assim, nestes casos, adicionar-se módulos suplementares para auxílio à programação de tarefas específicas ou mais complexas. O *software* pode ser proprietário ou aberto, podendo também ser compatível com mais do que um fabricante, gerando código na respetiva linguagem do controlador. Na Tabela 1 apresenta-se uma listagem de algumas aplicações de *software* proprietárias, os robôs compatíveis e o software de aplicação / módulos disponíveis para auxiliar a programação de algumas tarefas específicas [5][6][7].

**Tabela 1 Exemplos de aplicações de *software* proprietário de programação *offline***

<i>Software</i>	Robôs	Aplicação/Módulos
<b>RobotStudio</b>	<b>ABB</b>	<ul style="list-style-type: none"> <li>• ArcWelding PowerPac</li> <li>• Bending PowerPac</li> <li>• Machining PowerPac</li> <li>• MachineTending PowerPac</li> <li>• Painting PowerPac</li> <li>• Palletizing PowerPac</li> </ul>
<b>Roboguide</b>	<b>FANUC</b>	<ul style="list-style-type: none"> <li>• HandlingPRO</li> <li>• PaintPRO</li> <li>• PalletPRO</li> <li>• WeldPRO</li> </ul>
<b>KUKA Sim</b>	<b>KUKA</b>	<ul style="list-style-type: none"> <li>• KUKA.ArcTech</li> <li>• KUKA.BendTech</li> <li>• KUKA.ConveyorTech</li> <li>• KUKA.GlueTech</li> <li>• KUKA.LaserTech</li> <li>• KUKA.CAMRob</li> <li>• KUKA.Pallet <i>Layout</i></li> <li>• KUKA.Pallet Pro</li> <li>• KUKA.PlastTech</li> <li>• KUKA.SeamTech</li> <li>• KUKA.ServoGun</li> <li>• KUKA.TouchSense</li> </ul>

### 2.3.3. DESVANTAGENS E LIMITAÇÕES

Conforme visto, a programação *offline* necessita de um modelo da célula robotizada onde é simulado o comportamento real do robô. Deve-se assim considerar o tempo de realização do modelo antes de dar início à programação numa nova célula de trabalho.

A precisão do programa gerado depende do modelo 3D concebido, podendo apresentar erros e imprecisões, devido às tolerâncias, folgas e desgastes mecânicos inerentes à célula

real. Tal pode, em parte, ser ultrapassado pela calibração do modelo existindo, no entanto, dificuldade de repetibilidade deste método.

A falta de universalidade entre linguagens de programação para robôs dificulta também a criação de uma plataforma genérica de desenvolvimento *offline*. Algumas aplicações de *software* mais genéricas têm a capacidade de se adaptar a diferentes linguagens, sendo estas muitas vezes mais limitadas que as aplicações dedicadas a uma certa linguagem.

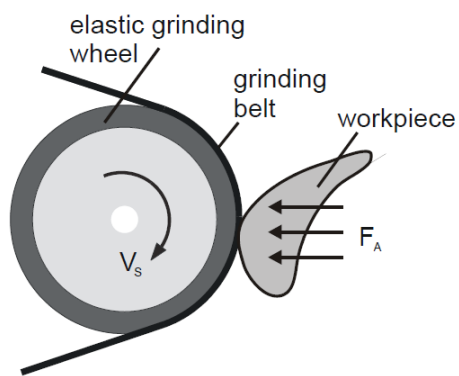
## **2.4. OPERAÇÕES DE ACABAMENTO SUPERFICIAL**

Ao longo desta secção serão abordados os aspetos mais relevantes do acabamento superficial, focando-se essencialmente nos processos abrasivos no contexto da robótica industrial.

Os processos de abrasão envolvem a remoção de material pela passagem de partículas abrasivas duras pelo material a trabalhar. Tradicionalmente, estes processos são executados manualmente, devido à sua complexidade e às elevadas exigências de qualidade. O operador, pela sua experiência, aplica a força e velocidade necessária à peça, de acordo com o processo de abrasão, contorno e resultado pretendido. De referir que estas tarefas são muitas vezes repetitivas e monótonas, levando à desconcentração do operário, o que pode conduzir a acidentes de trabalho ou erros de processamento, não se podendo garantir deste modo uma qualidade constante do acabamento do produto.

O lixamento é o processo de abrasão mais importante, conferindo à peça as dimensões e contornos desejados. O método de lixamento mais comum é a lixamento por cinta rotativa, onde é usada uma cinta abrasiva como ferramenta de corte colocada em tensão entre, pelo menos, duas rodas rotativas.

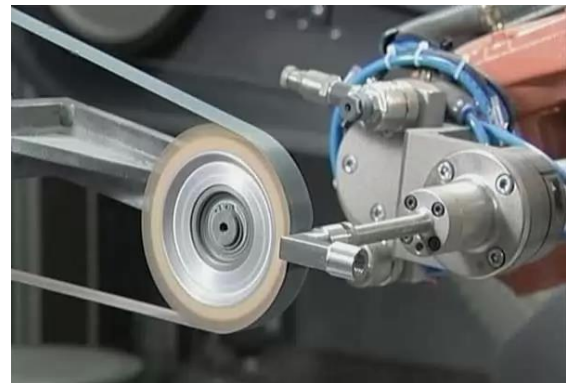
A variante da lixamento por cinta rotativa com uma roda de deformação elástica, apresentada na Figura 10, permite uma maior flexibilidade do processo pela sua adaptação à superfície da peça e por permitir compensar pequenos desvios, produzindo um acabamento geral de melhor qualidade [8].



**Figura 10 Lixamento por cinta rotativa com roda flexível [8][9]**

A quantidade de material removido depende de vários fatores, como sejam as propriedades do material a trabalhar, o tipo de abrasivo, a velocidade da cinta, a força de contacto e os movimentos efetuados (velocidade, caminhos e sequência).

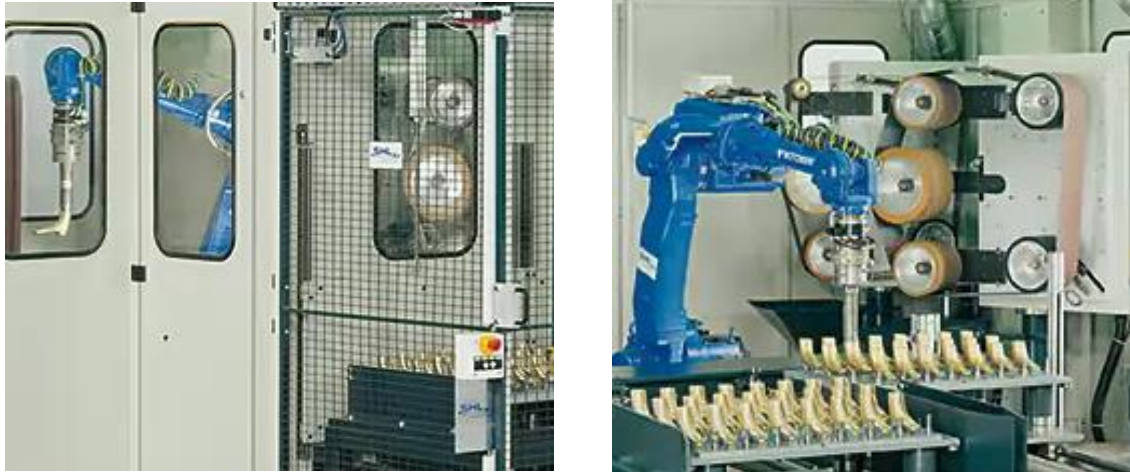
Existem atualmente várias soluções de células robotizadas especializadas para os processos de acabamento superficial, como é o caso das células Mepsa, disponibilizando este fabricante soluções “chave-na-mão” com robôs ABB, apresentando-se na Figura 11 uma célula robotizada de lixamento.



**Figura 11 Célula robotizada de lixamento da Mepsa [9]**

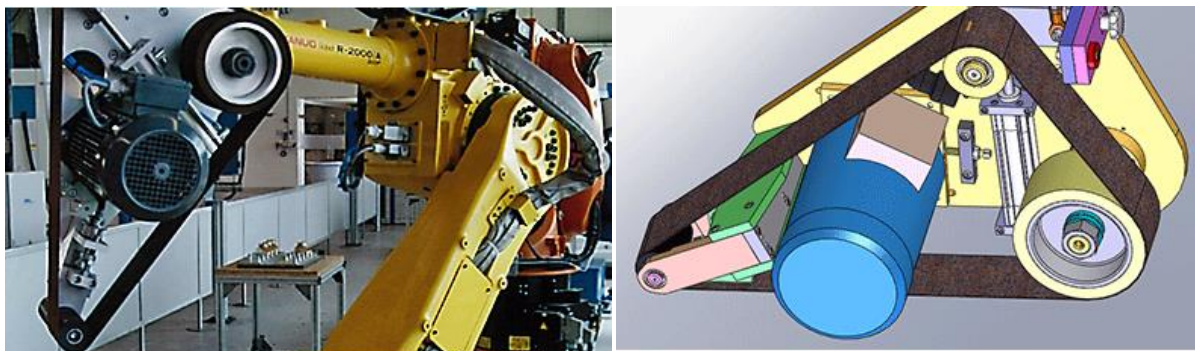
Estas dispõem de um sistema de controlo progressivo e automático, de forma a compensar o desgaste das lixas, obtendo-se um acabamento constante durante a vida útil destas. As unidades têm também um sistema de controlo passivo de força, para uma melhor qualidade de acabamento final. A célula dispõe de um *software* interativo que faz a gestão de múltiplas funções e parâmetros desde o seu exterior [9].

A SHL Automation é uma empresa dedicada ao desenvolvimento de soluções de acabamento superficial, desenvolvendo células robotizadas e linhas de fabrico completamente automatizadas, pela utilização de sistemas de transferência automática dos produtos entre as várias células. Na Figura 12 mostra-se um exemplo de uma célula robotizada e o respetivo sistema de paletização e de transferência.



**Figura 12 Célula robotizada de lixamento da SHL [10]**

A SHL disponibiliza também uma solução que, ao contrário do convencional, leva a ferramenta até a peça, sendo constituída por uma garra-ferramenta para o robô, que se mostra na Figura 13, sendo esta utilizada para aplicações com necessidades especiais [10].

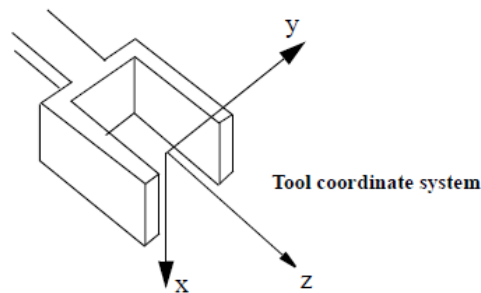


**Figura 13 Robô de lixamento com garra-ferramenta da SHL [10]**

## **2.5. AJUSTE DO PONTO DE REFERÊNCIA DA FERRAMENTA DO ROBÔ**

Sempre que uma ferramenta é acoplada ao robô é necessário definir com precisão a posição e orientação do seu ponto de referência ou TCP (Figura 14). A sua posição é

estrategicamente escolhida de forma a facilitar a movimentação e programação segundo o sistema de coordenadas da ferramenta.



**Figura 14 Sistema de coordenadas da ferramenta [11]**

Assim, caso surjam alterações ou desvios na ferramenta é necessário proceder ao ajuste do TCP, garantindo a repetibilidade e consistência do processo. Atualmente o TCP é definido numericamente no início da conceção do programa pelas coordenadas obtidas através do desenho CAD da ferramenta, apresentando-se um exemplo no Anexo A deste documento. É ainda possível comparar estes valores com a definição efetuada manualmente no robô, sendo esta tarefa demorada e dependente do método de cada operário. Caso existam variações dimensionais da peça a manipular ou desvios de funcionamento da própria ferramenta, torna-se vantajoso fazer o ajuste automático de forma periódica ou a cada *setup*.

De seguida são exploradas algumas das soluções atualmente disponíveis para a calibração automática da ferramenta do robô. Estes sistemas permitem fazer a medição e ajuste da ferramenta autonomamente, após um determinado número de ciclos de trabalho, ajustando o programa em função dos desvios que a ferramenta possa apresentar, minimizando desta forma os tempos de paragem para ajustes e afinações do programa.

### **2.5.1. DYNALOG AUTO CAL**

O sistema AutoCal da Dynalog (apresentado na Figura 15) permite a calibração da ferramenta de forma automática, garantindo o desempenho e precisão do robô. Este sistema é compatível com grande parte dos robôs e aplicações, podendo detetar os desvios do robô (*offsets* das juntas) ou do TCP da ferramenta face à localização real desta [12].



**Figura 15 Autocal da Dynalog [12]**

É possível fazer o controlo da posição da ferramenta, numa questão de segundos, corrigindo eventuais desvios entre ciclos de produção. Assim é reduzido o tempo de paragem para ajustes e assegurada uma qualidade consistente do processo.

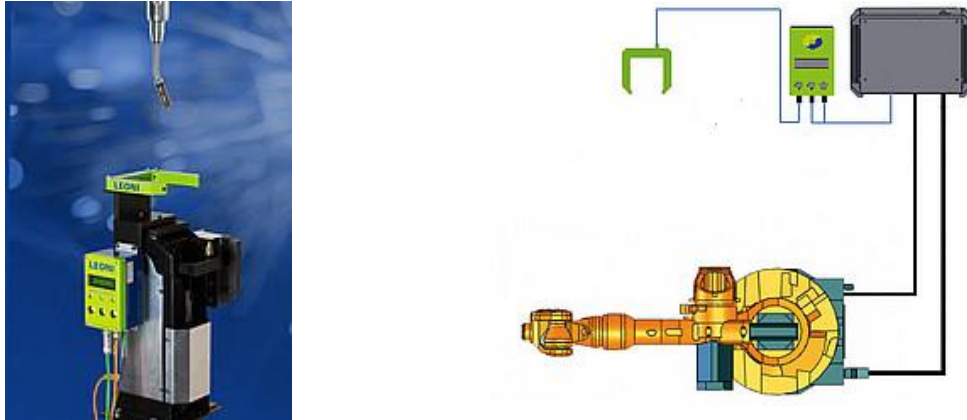
### **2.5.2. LEONI TOOL CALIBRATION**

Esta solução permite também assegurar a correta orientação e posição da ferramenta do robô, como sejam tochas de soldadura. O programa do robô é automaticamente corrigido consoante as variações medidas pelo sistema, assegurando assim a correta orientação da extremidade da ferramenta. Na Figura 16 pode ver-se a indicação do desvio no ecrã do sistema [13].



**Figura 16 Desvio do TCP apresentado no ecrã do sistema [13]**

Estão disponíveis três variantes deste sistema, diferindo no número de dimensões a serem calibradas, sendo a mais completa capaz de calibrar a ferramenta em seis dimensões (três rotacionais e três de translação) de forma precisa e automática [13]. Na Figura 17 apresenta-se a calibração de uma tocha de soldadura e o diagrama do sistema, onde se pode ver a ligação da unidade de medição do sistema ao controlador do robô.



**Figura 17 Integração do sistema no processo de soldadura [13]**

### **2.5.3. META VISION SYSTEMS - LASER SIGHT**

Tal como as soluções anteriores, o sistema da Meta Vision (mostrado na Figura 18) permite fazer o ajuste automático da posição da ferramenta, sendo este dedicado a processos de soldadura [14]. Após a limpeza da tocha, é feita a medição das coordenadas  $x$  e  $y$  da ponta da tocha pelo sensor de feixe Light Amplification by Stimulated Emission of Radiation (LASER). A coordenada  $z$  é obtida aquando da interrupção do feixe LASER, determinando com precisão o seu valor.



**Figura 18 Meta Laser Sight - Sistema de medição da tocha [14]**

Posteriormente é feita a correção do programa dependendo do desvio medido face ao valor inicial de referência.

## **2.6. CONTROLO DE FORÇA EM TAREFAS DE LIXAMENTO ROBOTIZADO**

Nas tarefas de acabamento superficial, o controlo da força de contacto da peça com a ferramenta de corte é um dos fatores cruciais para obter o resultado desejado. No lixamento robotizado, a integração de um sistema de controlo de força conferiria ao robô a capacidade de se adaptar às variações dimensionais da peça, corrigindo a trajetória e velocidade programada, num controlo em malha fechada. Utilizando um controlo de força passivo ou ativo é possível diminuir os defeitos derivados das flutuações dos processos a montante e aumentar a qualidade geral da superfície.

No controlo de força passivo existe uma complacência mecânica do robô ou da ferramenta de corte, permitindo a sua adaptação em função da força aplicada durante o lixamento. É assim possível aumentar a tolerância às flutuações dos processos de fundição e maquinação (erros dimensionais e de posicionamento), como também fazer o amortecimento de alguns eventuais impactos de aproximação ou entrada na lixa. Este tipo de sistema é mais rápido e simples que o controlo de força ativo, não sendo necessário nenhum sensor ou algoritmo de controlo, sendo adequado quando os desvios ou imperfeições se repetem peça após peça.

O controlo de força ativo, por sua vez, é realizado por um controlo em malha fechada das forças de contacto, alterando os parâmetros de trajetória e velocidade segundo uma rotina de correção. Este método tem a desvantagem de ser mais lento, caro e sofisticado que o sistema de controlo de força passivo. Para obter uma velocidade de execução aceitável deve ser usado, em combinação com alguma complacência passiva, de forma a manter as forças de reação abaixo do limite aceitável [15].

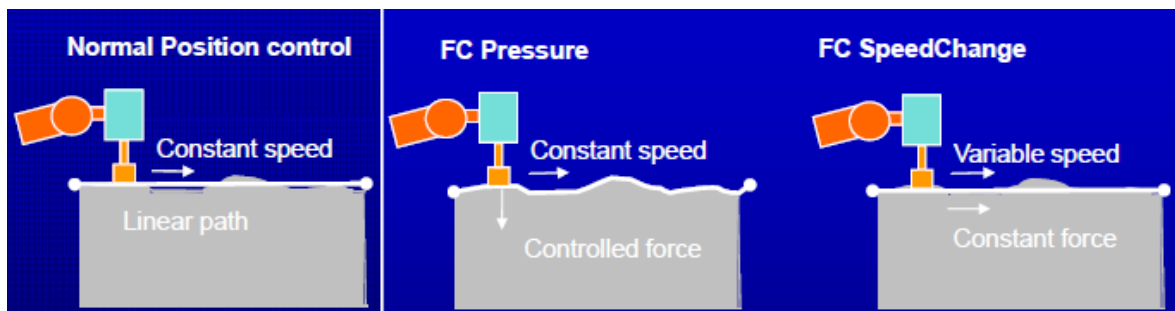
### **2.6.1. SISTEMA DE CONTROLO DE FORÇA DA ABB**

O sistema de controlo de força ativo da ABB é composto pelo *software* do controlador, módulo de aquisição e um transdutor de força/binário que permite fazer a aquisição simultânea da força aplicada verticalmente, lateralmente e longitudinalmente, assim como o valor de binário de cada eixo, como se mostra na Figura 19.



**Figura 19 Transdutor de força/binário [16]**

O sistema é completamente integrado no robô, através do *software* do controlador RobotWare Machining Force Control (FC). O *software* do controlador permite usar a informação do transdutor de duas formas distintas: modo FC Pressure ou modo FC SpeedChange [16]. O modo FC Pressure mantém a velocidade constante, adaptando o percurso entre pontos para que a força se mantenha constante. No modo FC SpeedChange o percurso não é alterado, sendo a velocidade ajustada de forma a manter a força constante no sentido do movimento. Na Figura 20 é possível ver uma representação gráfica destes dois modos de funcionamento.



**Figura 20 Modo convencional, FC Pressure e FC SpeedChange [16]**

Em suma, os novos métodos de programação e de trabalho trazem mais-valias para as aplicações robotizadas, permitindo uma redução dos tempos de programação, melhoria dos tempos de ciclo e uma melhor qualidade do processo ao dotar o robô do *feedback* das forças envolvidas. No próximo capítulo será apresentado o processo produtivo da Grohe Portugal e identificados os principais problemas do lixamento robotizado.



# 3. PROCESSO DE FABRICO E DEFINIÇÃO DO PROBLEMA

Neste capítulo é apresentado o processo de fabrico da Grohe Portugal, identificando-se os principais problemas nas tarefas de lixamento robotizado, relacionando-os com a qualidade do produto final e com a produtividade do departamento.

## 3.1. APRESENTAÇÃO DO PROCESSO PRODUTIVO

A Grohe Portugal tem o seu processo de fabrico organizado em cinco departamentos (Figura 21): a fundição, a maquinagem, o lixamento/polimento, a galvânica e a montagem.

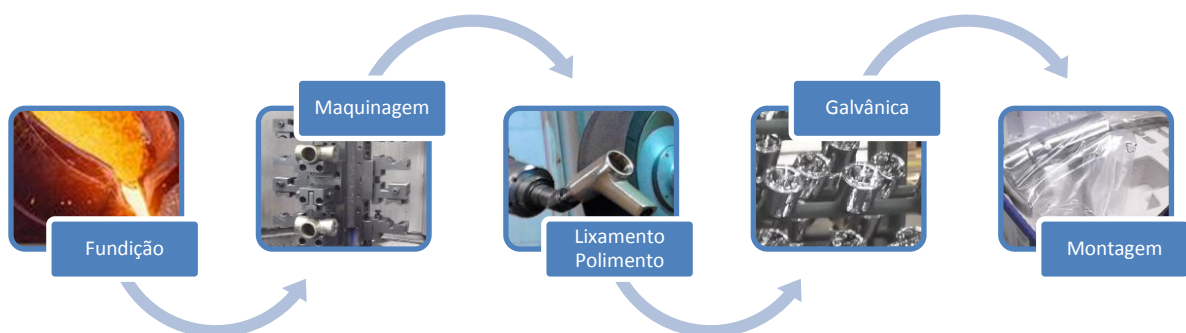


Figura 21 Ilustração das fases do processo produtivo da Grohe

### 3.1.1. FUNDIÇÃO

O processo produtivo tem início na fundição (ver Figura 22), que consiste na fusão das matérias-primas necessárias à obtenção da liga metálica de cobre-zinco que é usada na produção de torneiras.



**Figura 22 Forno de fusão**

Aqui são também produzidos os “machos” (Figura 23), que servem como molde interior, dando estes a configuração interna da torneira. Estes são constituídos por areia e endurecedores sendo moldados a partir da caixa de “machos”. Em seguida os machos devem ser corretamente rebarbados e acondicionados. Uma vez que o metal líquido irá ficar em contacto com o macho, este deve resistir à alta temperatura e ainda tornar-se quebradiço após o arrefecimento, permitindo assim uma fácil remoção da peça fundida.



**Figura 23 Produção dos “machos”**

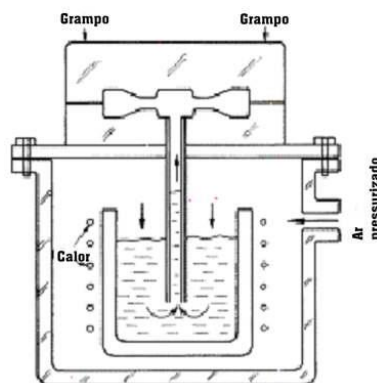
Posteriormente, o metal fundido é vazado num molde metálico permanente designado por coquilha, tendo esta sido preparada com o “macho” correspondente. Este processo de moldação permanente é caracterizado pela utilização de um único molde para a obtenção de dezenas ou centenas de milhares de peças, permitindo obter um bom acabamento superficial do fundido. Nesta etapa podem ser usados dois tipos de vazamento:

- Manual: o metal fundido é derramado para o interior do molde, sendo o operador que, com o auxílio de uma colher, recolhe o material do forno e o verte por ação da gravidade, como se apresenta na Figura 24;



**Figura 24 Vazamento Manual**

- Baixa pressão: o metal líquido é injetado diretamente no molde, através de equipamento pneumático ou hidráulico, como mostrado esquematicamente na Figura 25. Este método apresenta boa estabilidade produtiva, sendo possível um melhor controlo do vazamento.



**Figura 25 Esquema demonstrativo do processo de vazamento sob baixa pressão**

Após a solidificação é possível retirar a peça fundida do interior da coquilha sem a danificar. Segue-se o corte dos gitos (canais de vazamento, drenos ou gitos) e remoção do “macho” de areia recorrendo a máquinas denominadas granalhadoras (remoção da areia por projeção de esferas metálicas). Nesta fase do processo produtivo os corpos das torneiras estão prontos a ser maquinados, conforme se apresenta, a título de exemplo, na Figura 26.



**Figura 26** Corpos à saída do departamento de fundição

### **3.1.2. MAQUINAGEM**

Esta fase do processo produtivo é composta pelo processo de maquinação, lavagem/desengorduramento e teste funcional de estanquicidade do corpo da torneira.

As peças em bruto recebidas da fundição são maquinadas, removendo-se o material desnecessário e efetuando-se furações e roscas necessárias ao funcionamento da torneira. São para isso usados centros de maquinação *Computer Numerical Control* (CNC). Na Figura 27 pode-se observar os corpos das torneiras no fim do processo de maquinação e o pormenor da sua fixação à máquina.



**Figura 27 Maquinação dos corpos**

Após serem maquinadas, as torneiras são sujeitas a uma inspeção visual e é testada a qualidade das furações e roscas do corpo da torneira, através de calibres “passa/não passa” (Figura 28), resultando a aceitação ou rejeição do corpo ou do lote da torneira em questão.



**Figura 28 Calibre “passa/não passa”**

De seguida as torneiras são lavadas e desgorduradas sendo, posteriormente, submetidas a um teste funcional de estanquidade. Caso não passem no teste funcional, as peças são impregnadas e testadas novamente. Após este teste, as peças que forem rejeitadas são consideradas “sucata”, sendo enviadas para a fundição, onde serão reaproveitadas. As peças aprovadas seguem para o lixamento, acondicionadas conforme se mostra na Figura 29.



**Figura 29** Corpos maquinados

### **3.1.3. ACABAMENTO SUPERFICIAL – LIXAMENTO/POLIMENTO**

O acabamento superficial da torneira envolve o lixamento e polimento da sua superfície. Estes processos diferem entre si na taxa de remoção de material e qualidade de acabamento, permitindo obter as dimensões e contornos desejados.

#### **3.1.3.1. Lixamento**

O lixamento poderá ser feito de forma automática ou de forma manual. Dentro do método automático a tarefa é realizada por robôs ABB e KUKA, em células que recorrem ao método de lixamento por cinta rotativa com roda flexível, como se mostra na Figura 30.



**Figura 30** Lixamento robotizado

De forma a obter um melhor acabamento, o lixamento é feito numa primeira fase com lixa grossa de grão 80, sendo removida a película de fundição, rebarbas e outros materiais excedentes, provenientes da fundição da peça. Após esta primeira fase é dado o acabamento final com lixa fina de grão 280, baixando-se a rugosidade superficial de forma a preparar a peça para o processo de polimento.

Na Figura 31 mostra-se o estado dos corpos das torneiras no final do processo de lixamento automático, vendo-se em pormenor uma zona problemática apresentando ainda alguns defeitos de fundição. Por isso, na maioria dos casos, após o lixamento automático é necessário um acabamento manual, quer para fazer o acabamento das zonas de difícil acesso para os robôs, como também para retocar zonas da peça que apresentem defeitos (como por exemplo poros) sendo por isso necessário remover mais material nessas zonas específicas.



**Figura 31** Corpos com zona a retocar após o lixamento robotizado.

Para efetuar o lixamento dos vários corpos é necessário um conjunto de punho, pinça e palete desenvolvido especificamente para uma gama de produtos, de forma a ser possível ao robô fazer a carga do corpo e permitir uma maior liberdade de movimentos. Na Figura 32 podem ver-se alguns dos punhos de aperto rápido e algumas das pinças utilizadas existentes na ferramentaria da Grohe Portugal.



**Figura 32 Punhos e pinças para o lixamento robotizado**

Consoante o produto, pode também ser necessário fazer a mudança das rodas e da posição dos braços da unidade durante o *setup* do corpo da torneira a produzir. Em seguida, o programa é carregado e afinado manualmente ponto a ponto por um afinador experiente, verificando-se a pressão de contacto e a forma de contacto com a lixa. Este procedimento poderá demorar, em peças mais complexas, um turno completo (oito horas de trabalho).

#### 3.1.3.2. Polimento

Tal como sucede no lixamento, o polimento pode ser feito de forma automática, em células dedicadas, ou de forma manual, em máquinas de polimento manual ou combis (máquina que permite lixar e/ou polir, permitindo ao operador corrigir pequenas imperfeições).

No polimento automático são utilizadas máquinas de polimento CNC, onde as peças são polidas através da passagem de escovas rotativas de tecido, embebidas em pasta abrasiva, conforme se pode ver na Figura 33.

Estas são máquinas flexíveis, dando resposta a toda a produção do departamento, sendo o *setup* feito através da troca do porta-peças e da afinação do programa que define a posição, velocidade e outros parâmetros da escova e do porta-peças.



**Figura 33 Polimento automático**

No fim do ciclo de polimento automático os corpos polidos são inspecionados, designando-se esta fase por “controlo de amarelo”. Aqui é controlada a qualidade da superfície da peça e do acabamento final, sendo esta tarefa feita por uma controladora que examina toda a superfície da torneira, procurando defeitos como poros, riscos, vincos, desvios de contorno e outros defeitos. Estes são assinalados a vermelho na peça, como se pode ver na Figura 34, recorrendo a uma simbologia própria consoante o tipo de defeito.



**Figura 34 Controlo visual de defeitos**

Em seguida estas peças são recuperadas no polimento manual, que se apresenta na Figura 35, por um operário experiente. Caso não seja possível recuperar a peça, esta é considerada “sucata”, sendo enviada para a fundição.



**Figura 35 Polimento manual – recuperação da peça**

Após o polimento as peças aprovadas são lavadas e desengorduradas antes de seguirem para o próximo passo do processo produtivo – a galvânica.

#### **3.1.4. GALVÂNICA**

Os corpos das torneiras chegam à galvânica já colocados nas suspensões que serão usadas nos banhos e tratamentos, passando numa linha automatizada onde os corpos são lavados, niquelados e cromados, como se pode observar na Figura 36, podendo assim o processo ser subdividido nestas três fases:

- Lavagem/desengorduramento: as peças são lavadas e desengorduradas para evitar a contaminação dos banhos e tratamentos;
- Niquelagem: nesta fase é utilizado o processo de eletrodeposição de níquel. Este processo tem como objetivo dar resistência à corrosão às peças;
- Cromagem: é utilizado o processo de eletrodeposição de crómio. Este processo tem como objetivo dar um aspeto brilhante e cromado às peças, tal como se pode verificar nos corpos concluídos que se apresentam na Figura 37.



**Figura 36** Corpos nas suspensões – processo de galvanização



**Figura 37** Corpos cromados no final do processo da galvânica

No final as peças são novamente inspecionadas. Nesta inspeção detetam-se ainda peças defeituosas pois alguns dos defeitos apenas são visíveis após a niquelagem e cromagem da peça, realçando-os. No caso de serem aprovadas as peças passam para a fase da montagem. No caso de não corresponderem aos critérios de qualidade as peças são, por questões ambientais e económicas, descromadas, desniqueladas através de um banho ácido, sendo secas em seguida para poderem ser recuperadas (caso seja possível), passando novamente para a fase do lixamento/polimento.

### 3.1.5. MONTAGEM

Nesta fase é efetuada a montagem de todos os componentes da torneira e realizada uma inspeção e teste antes do embalamento, armazenamento e transporte.

Conforme se pode verificar na Figura 38, as linhas de montagem são em “U”, proporcionando assim uma maior aproximação e entreaajuda entre os operadores durante a fase de verificação, montagem dos componentes, teste funcional (realizado no caso das torneiras termostáticas, como se vê na Figura 39), limpeza e embalamento.

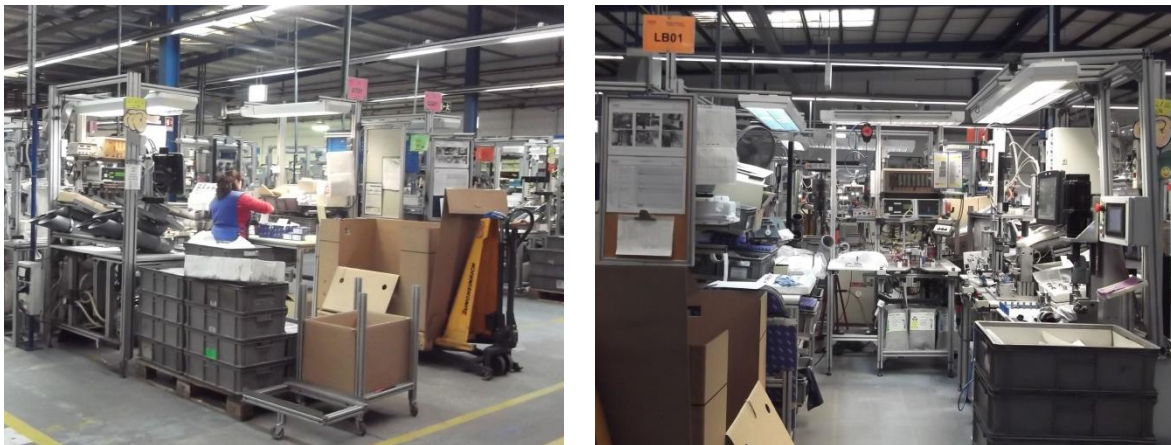


Figura 38 Linha de montagem em “U”



Figura 39 Teste funcional das torneiras termostáticas

### 3.2. PRINCIPAIS PROBLEMAS NO LIXAMENTO ROBOTIZADO

Após a descrição de todo o processo produtivo da Grohe Portugal, passa-se a analisar em detalhe o processo de lixamento robotizado, identificando-se os principais problemas e

dificuldades do método utilizado que, de algum modo, podem comprometer a produtividade e os custos de produção do departamento. De seguida são enumerados alguns dos problemas que motivaram a realização deste trabalho.

### **3.2.1. PARAGEM DOS ROBÔS**

As longas paragens das células para efetuar a programação *online* de um novo produto no robô, aliado à troca frequente de produtos, com tempos de *setup* que podem chegar a demorar um turno completo, traduz-se numa redução da taxa de utilização dos robôs.

A programação *offline* dos novos produtos e o ajuste automático do programa pela análise dimensional do corpo aquando da troca de produto, foram as formas encontradas para se tentar minimizar os tempos de paragem, adequando o processo às exigências da procura e, ao mesmo tempo, melhorando a qualidade do acabamento da superfície.

### **3.2.2. DIFICULDADES DO LIXAMENTO ROBOTIZADO**

Devido à complexidade da geometria dos corpos e à necessidade de efetuar movimentos coordenados, mantendo o paralelismo com a roda de lixar, surgem problemas associados à limitação de movimento dos eixos do robô e ao volume de trabalho do mesmo. De forma a não deixar marcas nem vincos na peça, as entradas (encostar da peça) e saídas (desencostar da peça) da lixa devem ser feitas de forma suave e no sentido do movimento, seguindo-se muitas vezes rotações da peça, paralelamente à roda, devendo estes movimentos ser realizados de forma contínua. Muitas vezes estes movimentos acabam por não ser possíveis por se atingir o limite de movimento de alguns dos eixos do robô. Frequentemente o robô atinge o limite do eixo cinco, limitando o movimento de rotação, situação que pode ser observada na Figura 40.

Também o local de fixação e a estrutura da ferramenta (pinça, garra ou punho) podem limitar o acesso a algumas áreas a lixar, levando muitas vezes à reestruturação de alguns movimentos, podendo comprometer a qualidade do acabamento. De forma a completar o movimento ou rotação, em alguns casos a pinça chega a entrar em contato com a lixa, degradando a própria ferramenta, conforme se pode observar na Figura 41.

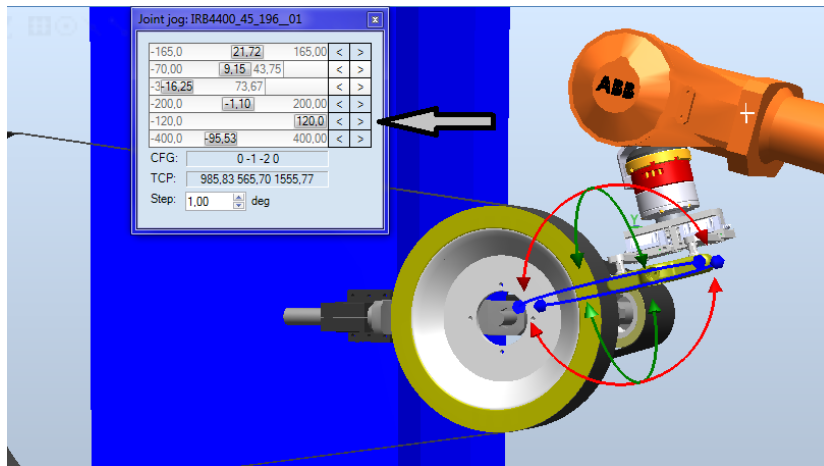


Figura 40 Movimento limitado ao atingir o limite do eixo 5 (+120°)

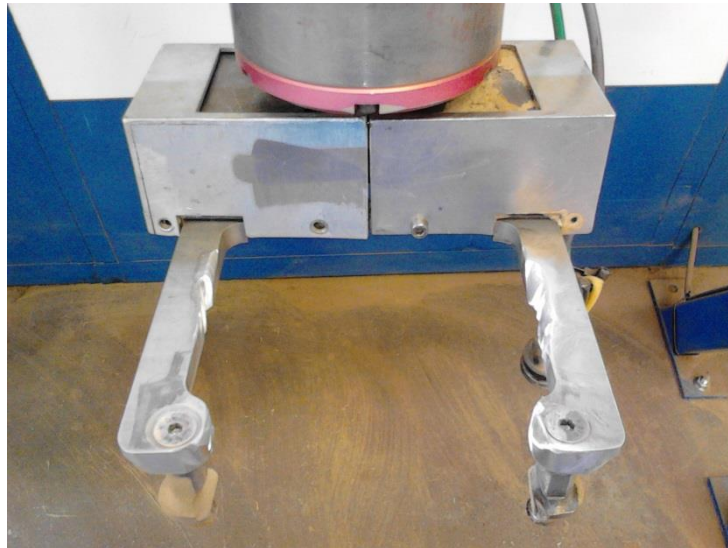


Figura 41 Degradação da pinça devido ao contacto com a lixa

### 3.2.3. INSTABILIDADE E DESVIOS DOS PROCESSOS DE FUNDIÇÃO E MAQUINAÇÃO

Conforme foi referido durante a descrição do processo produtivo, o sistema de controlo a montante do lixamento, apesar de eficaz, apenas controla as furações e roscas do corpo da torneira, ou seja, a parte funcional da torneira. As constantes variações dimensionais resultantes do processo de fundição originam desvios nas dimensões dos corpos, causando estes, por sua vez, desvios na maquinação do corpo (em especial na centragem das furações e roscas). Como o robô realiza sempre os mesmos movimentos, não corrige a falta de concentricidade da peça, comprometendo a qualidade do lixamento. Na Figura 42 é possível verificar que duas das peças (à direita na imagem) apresentam ainda película de

fundição na parte de baixo da bica (ver vista detalhada), o que revela que existem diferenças dimensionais entre os dois corpos da direita e os da esquerda.



**Figura 42** Corpos à saída do robô de lixamento (vista detalhada à direita)

### **3.2.4. NÍVEIS ELEVADOS DE *REWORK* E SUCATA**

Após o lixamento automático grande parte dos corpos são submetidos a um processo de recuperação/acabamento manual, devido a defeitos de lixamento causados pelas variações constantes dos processos precedentes (como já foi mencionado), assim como pela presença de defeitos que surgem apenas após o lixamento, como poros e fissuras, que têm origem no processo de fundição. A recuperação pode não ser possível se a remoção de mais material, para remover o poro, vinco ou fissura, causar desvio de contorno. Neste caso o corpo é considerado “sucata” e é enviado para a fundição, perdendo-se o valor que fora acrescentado ao longo das várias fases do processo até então. A adaptação do programa do robô em função da análise dimensional do corpo permitiria corrigir a pressão exercida pela lixa neste, permitindo evitar alguns dos defeitos de lixamento e contribuindo para a redução dos valores de *rework* e sucata.

Em suma, os principais problemas do departamento de lixamento/polimento levam a perdas de produtividade e problemas de qualidade, abordando-se nos próximos capítulos novos métodos de programação e de afinação do ciclo de lixamento, de forma a dar resposta a estes problemas.



## 4. SIMULAÇÃO E PROGRAMAÇÃO *OFFLINE*

Neste capítulo são descritos os procedimentos de preparação para a programação *offline* de um produto, seguindo-se a simulação e programação *offline* de uma célula robotizada de lixamento da Grohe, utilizando para isso o *software* da ABB, o RobotStudio.

A aquisição de uma nova célula da ABB, de *layout* distinto das restantes do departamento, criou a necessidade de proceder à programação de novos ciclos de lixamento. Assim sendo, a abordagem proposta consiste em realizar a simulação e programação *offline* destas células, preparando-as desta forma para a produção de um leque mais vasto de produtos, contribuindo para o aumento da sua flexibilidade e, simultaneamente, minimizando o impacto na produção por motivos de paragem para a programação *online*.

Por fim serão apresentados os resultados obtidos com este método de programação, verificando-se a sua aplicabilidade em contexto real.

### **4.1. PREPARAÇÃO PARA A PROGRAMAÇÃO *OFFLINE***

Antes de se dar início à programação do robô para a produção de um novo produto é necessário proceder ao levantamento de todos os aspetos inerentes às operações de

lixamento e à célula robotizada para a qual o programa a desenvolver se destina, de forma a conseguir retirar o máximo partido da mesma.

#### **4.1.1. A TORNEIRA E AS FERRAMENTAS A UTILIZAR**

Pretendendo-se efetuar o estudo e conceção de um novo ciclo de lixamento, recorrendo à simulação e programação *offline* do robô e de forma a alcançar uma melhor qualidade e estabilidade de acabamento superficial, foi escolhida uma torneira monocomando de lavatório da colecção EuroSmart, apresentada na Figura 43.



**Figura 43 Torneira de lavatório da colecção EuroSmart – Grohe**

Apresenta-se na Figura 44 o corpo da torneira e respetivo modelo 3D, que será utilizado na célula virtual.



**Figura 44 Corpo da torneira (esquerda) e modelo 3D da mesma (direita)**

O corpo desta torneira é produzido por fundição em coquilha a baixa pressão, sendo posteriormente rebarbado o gito de alimentação. Na Figura 45 mostra-se o acabamento do corpo à chegada ao departamento de lixamento, onde se pode verificar a existência de rebarba

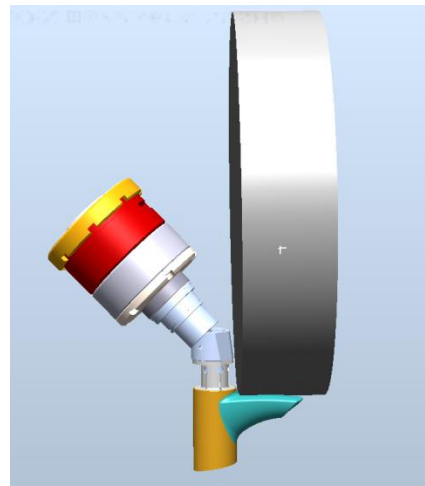
nas costas do corpo proveniente do gito (imagem da esquerda) e também alguma rebarba ao longo de todo o corpo devido à união das duas partes da matriz ou coquilha (imagem da direita).



**Figura 45** Costas e parte de baixo do corpo da torneira

Conhecido o corpo da peça a trabalhar e as suas características mais relevantes para a operação de lixamento, passa-se agora a analisar a fixação da peça ao robô e as ferramentas a utilizar.

Um dos aspetos fundamentais para o lixamento robotizado passa pela manipulação da peça a trabalhar, de forma a garantir o acesso a toda a superfície desta e, ao mesmo tempo, permitir suportar as forças envolvidas durante o lixamento. Conforme já foi referido no capítulo anterior, existem dificuldades no posicionamento e realização de alguns movimentos, apesar dos 6 graus de liberdade do robô. Assim, a escolha do punho e da pinça a utilizar é feita consoante a geometria da peça e das áreas a lixar, tendo também em consideração, sempre que possível, a padronização das ferramentas, permitindo uma melhoria do *setup* ao reduzir os tempos improdutivos para a sua troca. Recorrendo ao RobotStudio foi possível testar a aplicabilidade do punho com cunha a 45°, verificando-se o acesso ao topo da bica conforme se mostra na Figura 46.



**Figura 46 Punho a 45° e possibilidade de acesso ao topo da bica**

Relativamente à pinça, é utilizada para este tipo de torneira uma pinça pneumática expansível que se apresenta na Figura 47.



**Figura 47 Pinça expansível para fixação do corpo**

Em seguida são descritas as fases que constituíram a avaliação e modelação da célula de lixamento.

#### **4.1.2. AVALIAÇÃO DA CÉLULA DE TRABALHO**

Como já referido, o processo de programação *offline* começa necessariamente pela criação de um modelo 3D da célula de trabalho antes de se dar início à programação propriamente dita. Este modelo da célula real é essencial para a correta programação do ciclo de lixamento do corpo da torneira. Para a sua conceção torna-se necessário avaliar a célula de

trabalho a modelar, de forma a determinar os componentes necessários para uma representação fiel da célula real no RobotStudio.

A célula de trabalho em questão é constituída por quatro unidades de lixamento – duas para o lixamento com lixa grossa e as restantes para o acabamento com lixa fina (Figura 48). Estas unidades são habitualmente equipadas com duas rodas de 450 mm de diâmetro nas unidades superiores e duas rodas de 150 mm de diâmetro nas unidades inferiores, ambas com 100 mm de largura, permitindo obter um bom acesso às superfícies a lixar de grande parte dos corpos produzidos no departamento.



**Figura 48 Célula robotizada de lixamento “Mepsa 13”**

Sendo o objetivo principal a correta programação de todo o ciclo de lixamento do corpo, torna-se imprescindível fazer a representação gráfica das rodas e da lixa, sendo esta a zona mais importante onde ocorre o contacto da ferramenta com a peça, devendo a mesma ser precisa e realista, possuindo todo detalhe que uma aplicação de contacto exige. Decidiu-se não modelar os restantes componentes da célula visto que não eram essenciais para a programação dos movimentos ao não constituírem perigo de colisão, devido à sua disposição, e simultaneamente, como forma de melhorar o desempenho gráfico da simulação da célula virtual.

O robô da célula de trabalho é um ABB IRB 4400 com o controlador S4CPlus, mostrando-se na Figura 49 o robô e a sua área de trabalho. Trata-se de um robô compacto para cargas médias, tendo uma capacidade de carga máxima de 45 kg e um alcance máximo de 1,96 m, sendo assim adequado para as forças envolvidas no processo de lixamento.

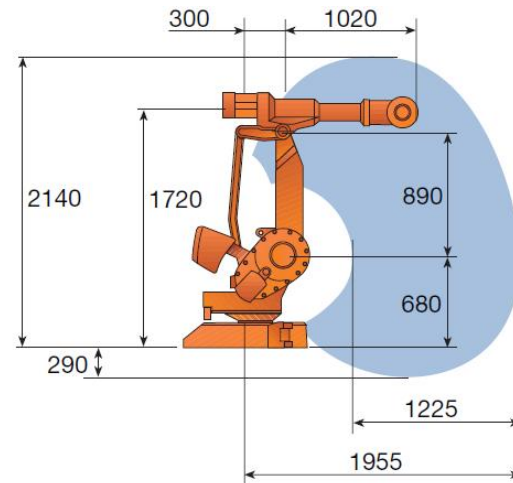


Figura 49 Robô ABB IRB 4400/45 1,96 m [17]

#### 4.1.3. MODELAÇÃO DA CÉLULA DE TRABALHO

O desenvolvimento de uma aplicação começa necessariamente pela concepção do modelo do mundo apresentando-se, nas subsecções seguintes, a modelação geométrica, cinemática e funcional da célula de trabalho, seguindo-se os testes e otimizações ao modelo inicialmente concebido.

##### 4.1.3.1. Modelação geométrica dos componentes

A primeira etapa da concepção do modelo do mundo passa pela definição do modelo geométrico, sendo este essencial à programação *offline*. O RobotStudio permite conceber o modelo da célula robotizada real, sendo a modelação dos vários elementos da célula feita através das funções CAD incluídas ou importando os modelos de outros programas de CAD, sendo suportados diversos formatos neutros. Além disto, o RobotStudio disponibiliza ainda bibliotecas com os modelos dos robôs, ferramentas, tapetes rolantes (*conveyors*) e outros equipamentos que habitualmente compõem a célula de trabalho.

Para se efetuar a modelação das unidades de lixamento recorreu-se à utilização de um programa de CAD 3D, o Inventor da AutoDesk, a partir do qual foi possível desenhar os componentes necessários e efetuar a sua montagem. Destes componentes alguns existiam já em CAD, como foi o caso da chumaceira da roda, por exemplo, tendo sido importados e montados.

A título exemplificativo, demonstra-se de seguida a construção do modelo 3D da roda. Concebeu-se para isso o desenho do perfil de metade da roda, conforme se mostra na Figura 50, sendo este em seguida espelhado dando origem ao desenho completo do perfil da roda. Por fim, para a criação do corpo, fez-se a revolução do desenho do perfil como se mostra na Figura 51.

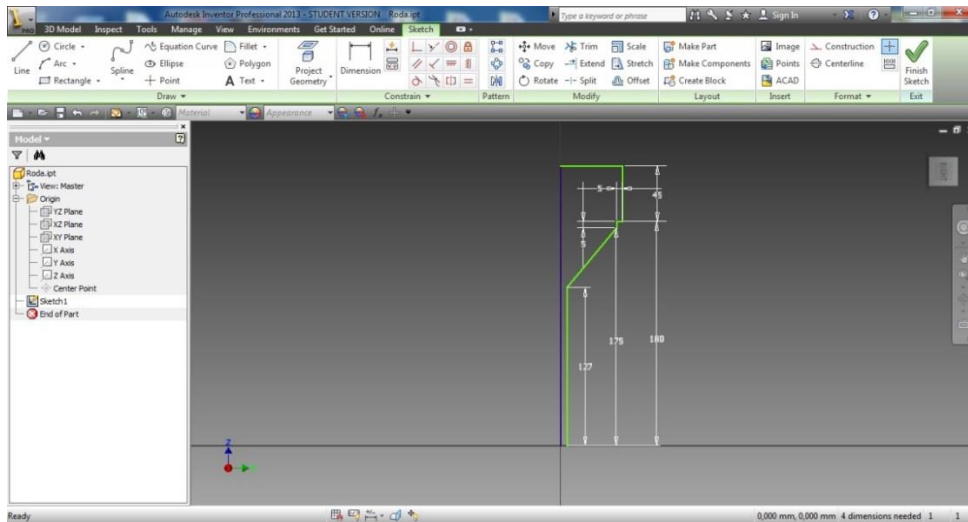


Figura 50 Desenho do perfil de metade da roda

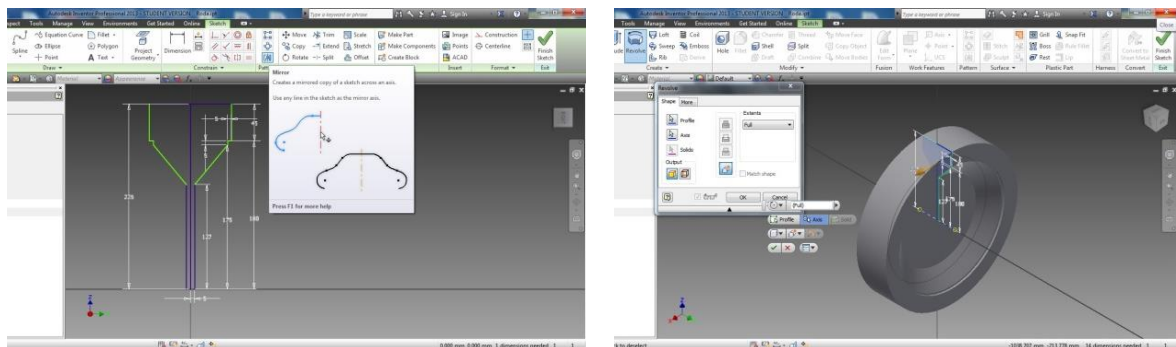
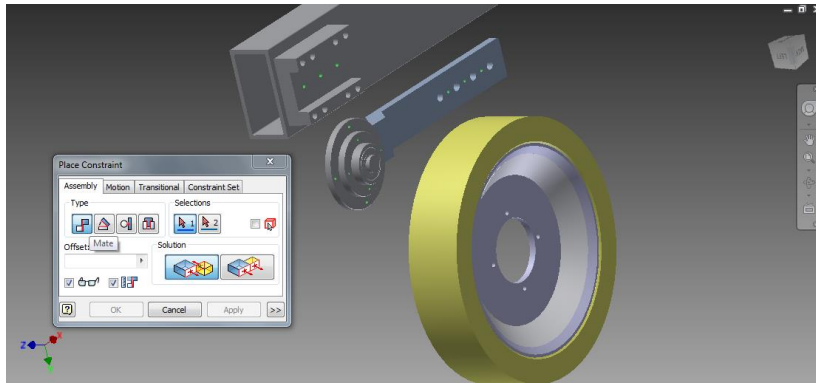


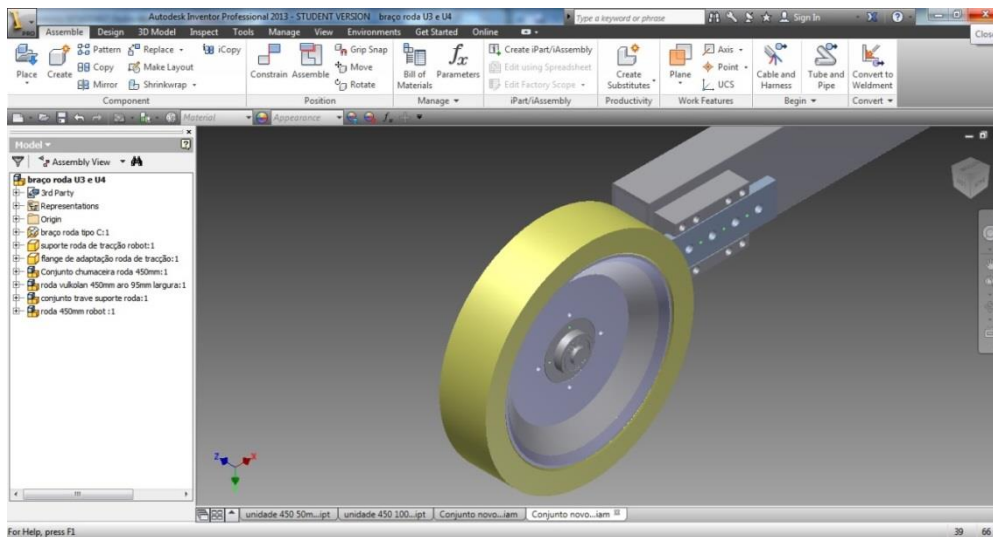
Figura 51 Espelho e revolução do desenho de perfil

Após se conceber e reunir todos os componentes, montou-se o conjunto da unidade criando um *assembly* dos vários corpos que a constituem. O correto posicionamento dos componentes é assegurado pelas restrições dadas por funcionalidades como, por exemplo, o “*Mate*”, apresentado na Figura 52.



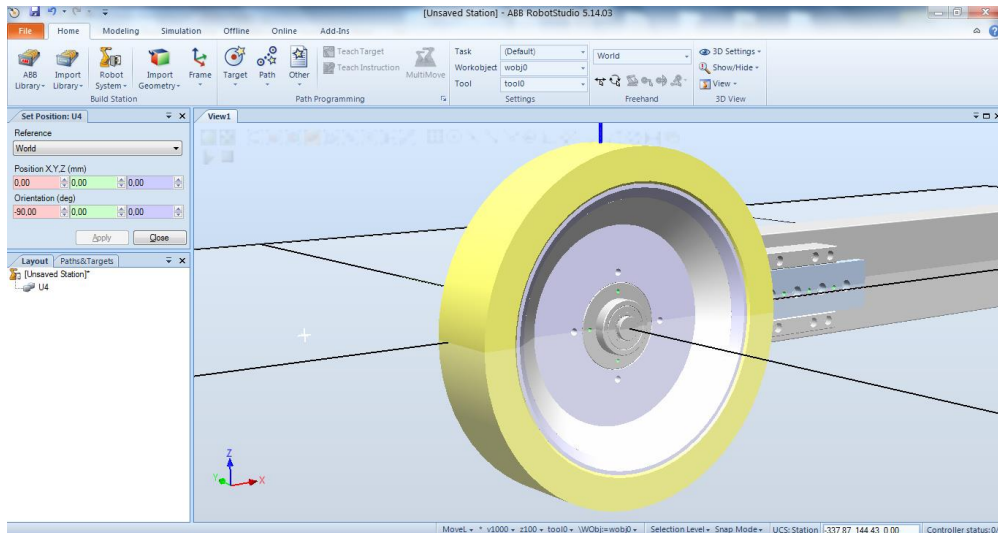
**Figura 52 Montagem dos componentes**

Pode ver-se na Figura 53 o conjunto do braço e da roda da unidade de lixamento quatro já montado. Este conjunto foi exportado num dos formatos neutros disponíveis no Inventor. Neste caso utilizou-se o formato STEP, um dos formatos compatíveis com o RobotStudio.



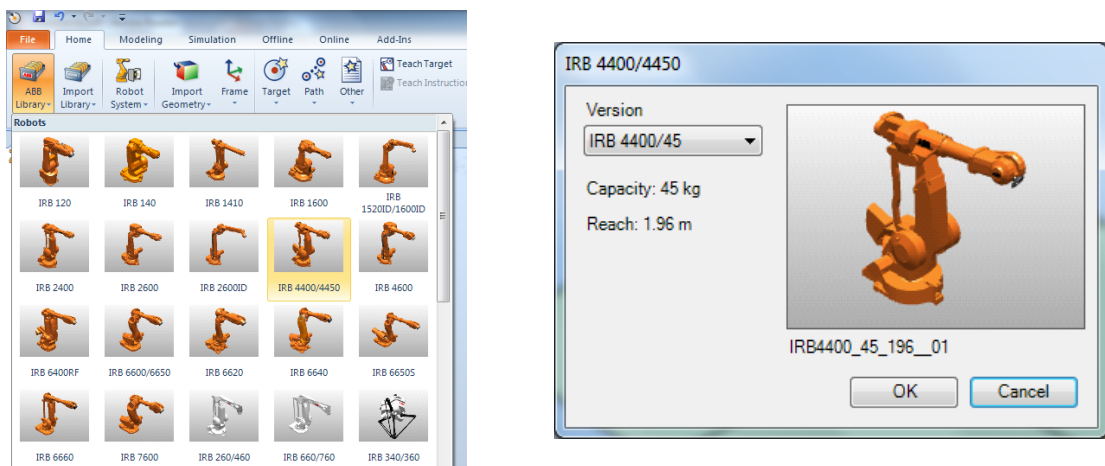
**Figura 53 Conjunto do braço e roda da unidade de trabalho quatro**

Gerados os ficheiros das quatro unidades de trabalho, procedeu-se à sua importação para uma nova célula virtual utilizando a funcionalidade “*Import Geometry*” do RobotStudio, posicionando-se de seguida as quatro unidades de acordo com a sua disposição na célula real, como se mostra na Figura 54.



**Figura 54** Conjunto do braço e roda da unidade de trabalho quatro no RobotStudio

Após a importação das unidades, foi adicionado o robô à célula virtual a partir da biblioteca de modelos do RobotStudio, escolhendo-se o robô ABB IRB4400 de 45 kg e alcance 1,96 m, visível na Figura 55.



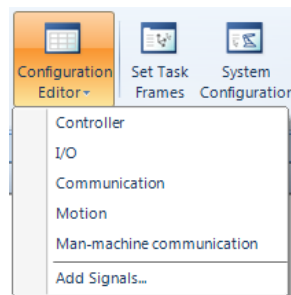
**Figura 55** Biblioteca de modelos do RobotStudio

Desta forma é adicionada à célula o modelo geométrico do robô sendo necessário adicionar o controlador (modelação cinemática) permitindo o movimento dos mecanismos da célula.

#### 4.1.3.2. Modelação cinemática e funcional do controlador do robô

O RobotStudio incorpora modelos dos controladores (ou controladores virtuais) para cada robô, contendo estes uma cópia do *software* do controlador real, permitindo assim que os cálculos da cinemática inversa produzam os mesmos resultados, quer na simulação, quer na célula real. De forma a garantir uma modelação perfeita do controlador utilizado na

célula real, é necessário carregar os parâmetros do controlador do robô real, sendo estes carregados no RobotStudio através de um *backup* do robô da célula de trabalho ou copiando os ficheiros de configuração do robô existentes na pasta “SYSPAR” do *backup* - “EIO.cfg”, “MMC.cfg”, “MOC.cfg”, “SIO.cfg” e “SYS.cfg”, sendo também possível criar um sistema de raiz e copiar cada parâmetro individualmente, como apresentado na Figura 56. Contudo, o RobotStudio não é inteiramente compatível com os ficheiros de configuração do controlador da célula em estudo (S4CPlus), não permitindo importar o *backup* completo. Por este motivo decidiu-se criar um sistema de raiz e detalhar alguns dos passos mais relevantes.



**Figura 56 Configuração dos parâmetros do robô no RobotStudio**

Os parâmetros de especial interesse para a simulação encontram-se no ficheiro de configuração de movimento – “MOC.cfg”. Dentro deste podem-se localizar os parâmetros de cada eixo do robô, parâmetros de calibração e supervisão do movimento, sendo necessário passar estes valores para o controlador virtual de forma a obter uma simulação consistente com a realidade. Na Figura 57 apresentam-se os parâmetros relativos aos elos do robô, estando aqui parametrizados os limites de cada junta (em radianos), quer para o lado positivo do eixo – *Upper Joint Bound*, quer para o lado negativo do eixo – *Lower Joint Bound*, definindo estes parâmetros o volume de trabalho do robô, dentro da sua limitação mecânica.

Type name	Name	Use Check Point	Use Arm Load	Independent Joint	Upper Joint Bound	Lower Joint Bound
Acceleration Data	rob1_1			Off	2,87979	-2,87979
Arm	rob1_2			Off	1,65806	-1,22173
Arm Check Point	rob1_3 rob1_3			Off	2,70526	-0,523599
Arm Load	rob1_4			Off	1000	-1000
Brake	rob1_5			Off	2,0944	-2,0944
Control Parameters	rob1_6			On	10000	-10000
CSS						

**Figura 57 Parâmetros dos elos do robô**

Pela análise da figura pode verificar-se que os movimentos dos eixos quatro e seis são ilimitados, sendo os únicos eixos que podem ser, por opção, utilizados desta forma. No

entanto, o eixo quatro encontra-se fisicamente limitado a apenas algumas voltas devido ao abastecimento de ar comprimido para o punho.

Voltando a analisar o ficheiro de configuração, passa-se agora aos valores de compensação do atrito das juntas que se apresentam na Figura 58. O atrito existe em todos os mecanismos, sendo definido como a força tangencial de reação entre duas superfícies de contacto. É assim necessário modelar o atrito que surge, por exemplo, nos rolamentos e engrenagens das juntas do robô para efeitos de controlo, melhorando o desempenho e precisão do robô.

Type name	Name	Friction FFW On	Friction FFW Level	Friction FFW Ramp
Friction Compensation	r1_CFric_1	No	0,1	0,04
Jog Parameters	r1_CFric_2	No	0,36	0,04
Joint	r1_CFric_3	No	0,29	0,04
Lag Control Master 0	r1_CFric_4	No	0,09	0,04
Linked M Process	r1_CFric_5	No	0,37	0,04
Mains	r1_CFric_6	No	0,42	0,04
Measurement Channel				

**Figura 58 Compensação do atrito das juntas**

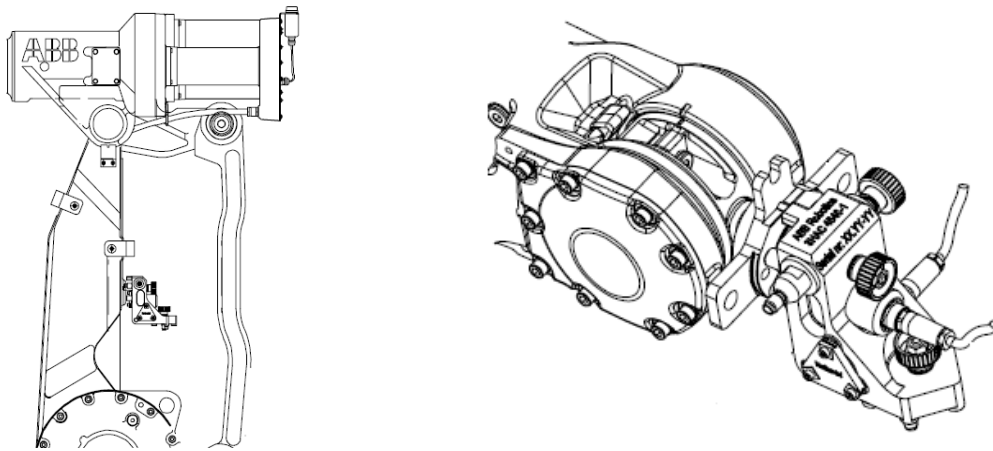
Encontram-se ainda neste ficheiro os valores de calibração dos motores e comutadores, como representado na Figura 59. Esta compensação dos erros de cada junta permite corrigir os desvios de posição do elemento terminal, conferindo precisão ao robô. O robô é entregue já calibrado, e fornecido com os dados de calibração para cada motor, sendo apenas necessário proceder à calibração fina do robô no caso de alguma intervenção mecânica ou colisão, sendo para isso necessário recorrer ao equipamento de calibração para redefinir os valores de calibração.

Type name	Name	Commutator Offset	Commutator Offset Valid	Calibration Offset	Calibration Offset Valid
Motor	LOCKED	1,5708	Yes	0	Yes
Motor Calibration	rob1_1	1,5708	Yes	5,77496	Yes
Motor Type	rob1_2	1,5708	Yes	1,55201	Yes
Path Sensor Synchronization	rob1_3	1,5708	Yes	2,44929	Yes
PMC sensor	rob1_4	1,5708	Yes	1,5195	Yes
PMC Sensor Setup	rob1_5	1,5708	Yes	3,44503	Yes
Process	rob1_6	1,5708	Yes	3,13948	Yes
Relay					

**Figura 59 Valores de calibração do robô utilizado**

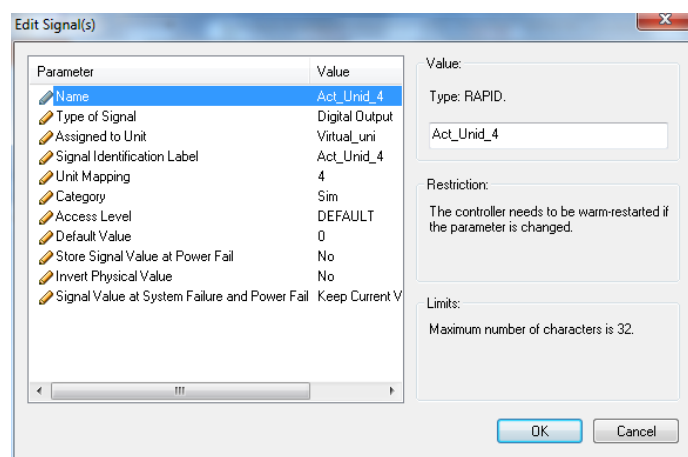
Este equipamento consiste num nível eletrónico de precisão, sendo medido o nível dos eixos em função do nível da base do robô [18]. Apresenta-se na Figura 60 a montagem

para a calibração do eixo 2 (à esquerda) e a montagem para a calibração dos eixos 3, 4, 5 e 6 (à direita).



**Figura 60 Calibração dos eixos do robô [18]**

Analisados alguns dos tópicos do ficheiro de configuração do movimento, passa-se a analisar o ficheiro de sinais de entrada/saída – “EIO.cfg”. Neste ficheiro encontram-se especificadas as unidades de I/O (“EIO\_UNIT”), nome e características de sinais de entrada e saída (“EIO\_SIGNAL”), grupos e ligações cruzadas entre sinais (“EIO\_CROSS”), entre outras informações relativas aos sinais do sistema. De forma a fazer a programação e simulação dos sinais da célula, foram adicionados os sinais da célula de trabalho ao controlador virtual, especificando-se uma unidade virtual para o efeito, mostrando-se na Figura 61 o processo de configuração de um novo sinal digital de saída.



**Figura 61 Configuração de um sinal digital de saída**

Na Figura 62 apresenta-se uma lista dos sinais criados e que são utilizados durante a programação para a ativação das unidades de trabalho, ficando estes também disponíveis no simulador de I/O, como se mostra na Figura 63.

Type name	Name	Type of Signal	Assigned to Unit	Signal Identification Label	Unit Mapping
Signal	Act_Unid_1	Digital Output	Virtual_uni	Act_Unid_1	1
System Input	Act_Unid_2	Digital Output	Virtual_uni	Act_Unid_2	2
System Output	Act_Unid_3	Digital Output	Virtual_uni	Act_Unid_3	3
Unit	Act_Unid_4	Digital Output	Virtual_uni	Act_Unid_4	4

**Figura 62 Sinais digitais de saída para as quatro unidades de trabalho**

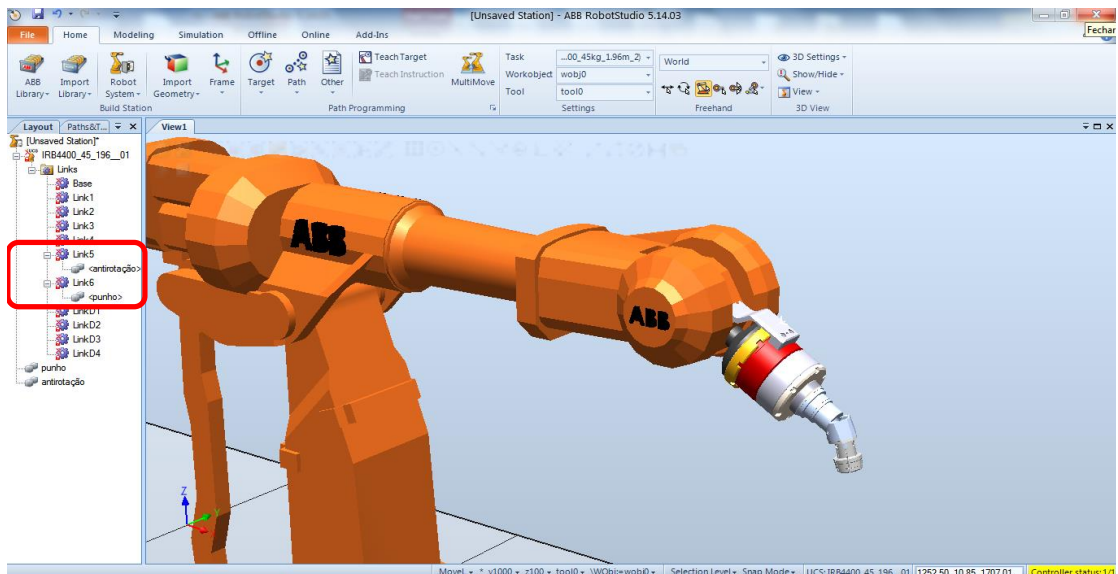


**Figura 63 Sinais da unidade virtual no simulador de I/O**

Para concluir este processo, é necessário combinar o modelo do controlador com o modelo geométrico de forma a ser possível “animar” o modelo geométrico do robô ou de outro mecanismo adicionado à célula de trabalho.

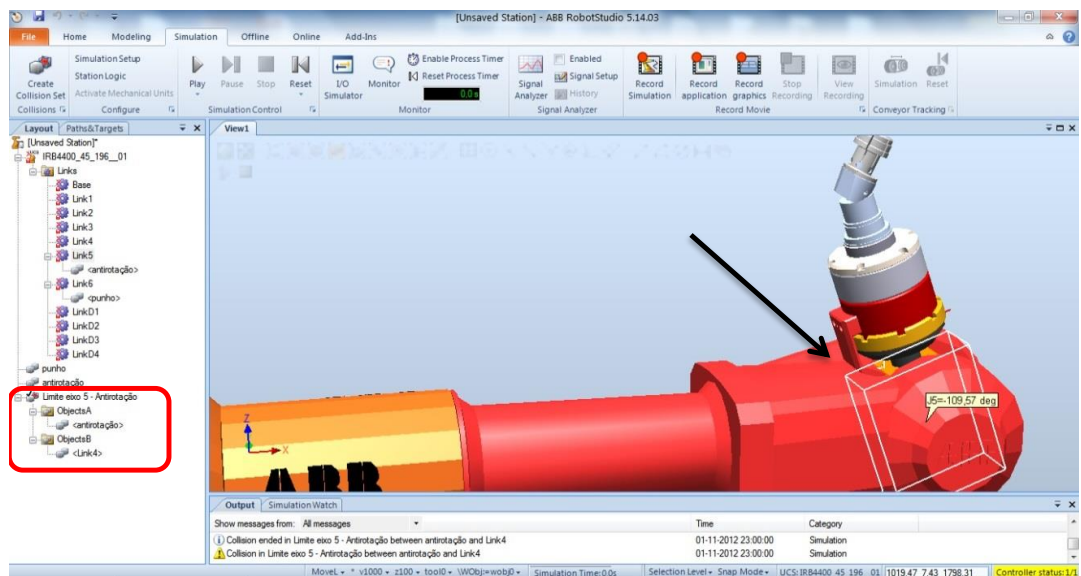
#### 4.1.3.3. Modelação geométrica e funcional da ferramenta do robô

Após concluída a modelação do controlador do robô, passa-se à modelação geométrica e funcional da ferramenta do robô. Para o efeito, importou-se para a célula virtual o punho de fixação com cunha a 45°. Foi também modelada a peça que impede que parte do punho rode juntamente com o eixo 6, devido ao sistema de ar comprimido que alimenta a pinça, sendo assim esta peça designada de anti rotação do punho. Na Figura 64 pode ver-se o robô com o punho e as ligações estabelecidas com os elos do robô, ficando a peça de anti rotação ligada ao eixo 5 e a parte restante do punho ligada à extremidade do robô – eixo 6.



**Figura 64 Punho com cunha a 45° montado no robô**

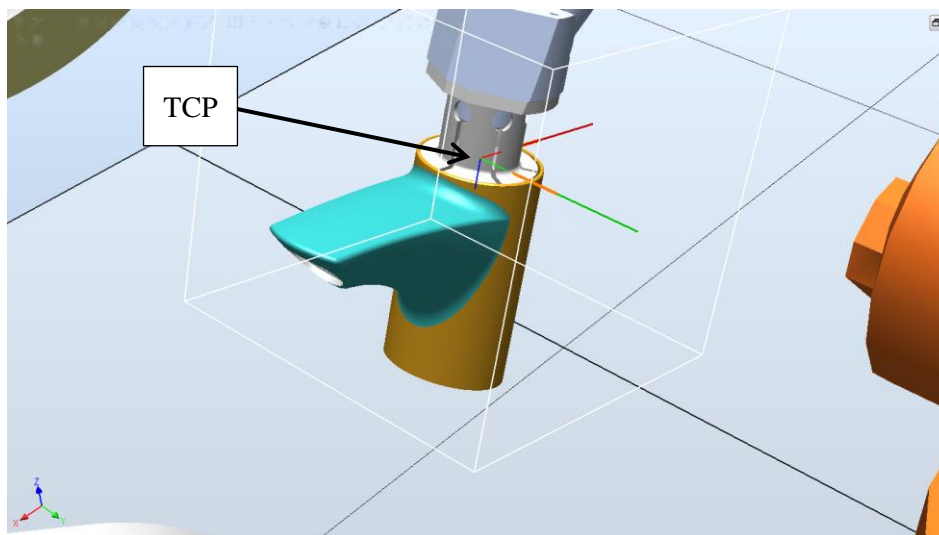
Conforme se pode verificar na Figura 65, a peça que impede a rotação de parte do punho pode colidir com o elo quatro, impedindo a junta cinco do robô de se mover até ao seu fim de curso num dos sentidos, devendo ter-se em consideração esta limitação durante a programação e simulação dos movimentos. Para isso, definiu-se uma monitorização de colisão entre o elo 4 e o corpo do sistema de anti rotação do punho, sendo indicada a ocorrência de uma colisão ou iminência desta (“*near miss*”).



**Figura 65 Limite do eixo 5 – colisão entre o sistema anti rotação do punho e o eixo 4**

Em seguida é importado o modelo do corpo da torneira, fazendo-se a ligação desta ao punho e eixo 6 do robô, ficando a corpo da peça fixo na pinça expansível, como mostrado

na Figura 66. De forma a facilitar a movimentação e a programação, foi definido um novo ponto de referência para este conjunto, localizado ao centro do corpo – ver Figura 66. O desenho da montagem com as coordenadas do TCP encontra-se no Anexo A deste documento.



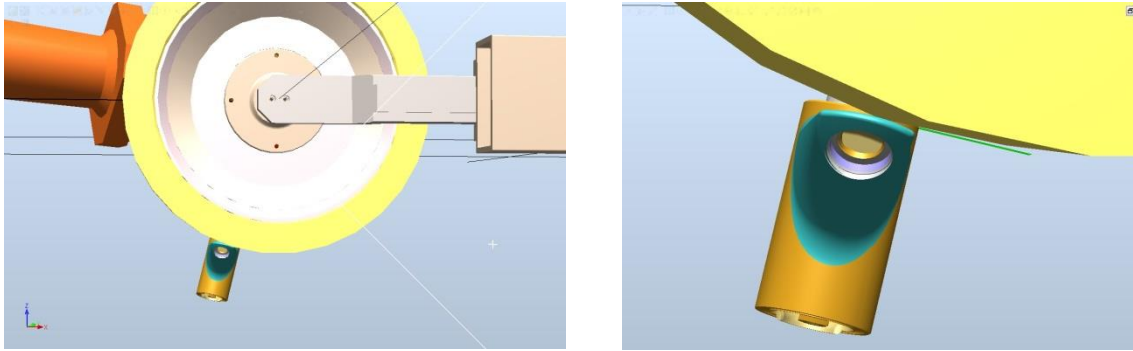
**Figura 66** Corpo da torneira na pinça expansível e localização do TCP

Com isto conclui-se a modelação da célula de trabalho, apresentando-se de seguida os testes do modelo concebido.

#### 4.1.3.4. Testes do modelo da célula

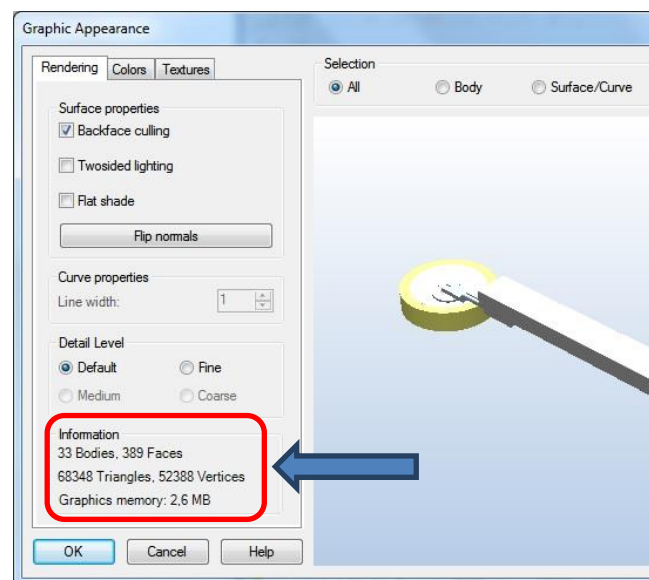
Concluído este passo foi possível testar a célula modelada, particularmente a visualização das duas superfícies de contacto. Na Figura 67 apresenta-se o teste de acesso à parte superior da bica, onde se verifica que existe falta de definição gráfica da roda para efeitos da simulação do processo de lixamento, pois este exige uma boa visualização e definição das duas superfícies de forma a ser possível seguir a geometria e contornos da peça a lixar. Também se considerou relevante fazer a modelação da cinta da lixa, uma vez que poderia ser necessário recorrer às zonas onde o corpo entra e sai da roda, criando aí uma zona de banda livre.

De forma a testar a simulação importaram-se alguns programas de outros produtos, concebidos pelo método de programação *online*, sendo possível fazer uma avaliação preliminar ao desempenho da simulação *offline* e restante modelação da célula de trabalho.



**Figura 67** Resolução gráfica da roda de contacto no RobotStudio

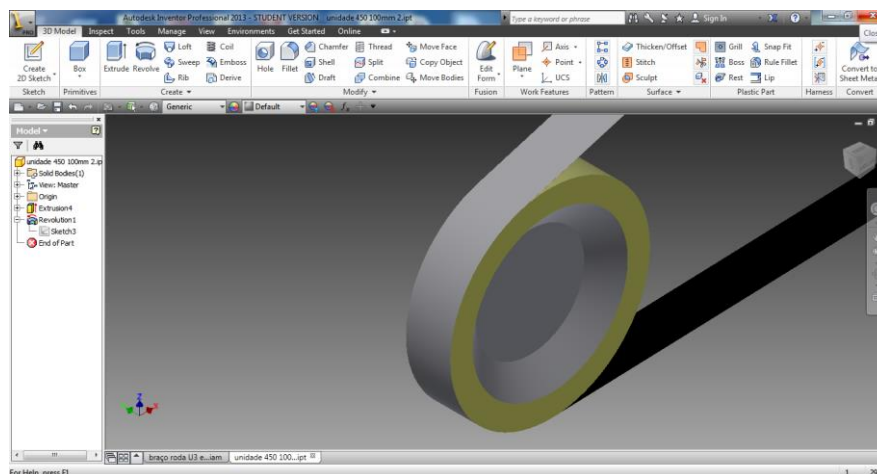
Foi notória uma falta de fluidez dos movimentos do robô, sendo esta característica fundamental para a aplicação em questão, de forma a verificar-se a trajetória do movimento de ligação entre os *targets* programados, de forma realista e fluida. Assim, passaram-se a analisar os componentes da célula que, devido à sua complexidade e tamanho da geometria, poderiam ser a causa da falta de desempenho da simulação. Através da funcionalidade “*Graphic Appearance*”, que se mostra na Figura 68, é possível aceder às informações dos componentes importados para a modelação, podendo-se visualizar alguns aspetos importantes relacionados com a sua renderização, como tamanho do ficheiro em memória gráfica e constituição da geometria.



**Figura 68** Propriedades gráficas de uma unidade de lixamento no RobotStudio

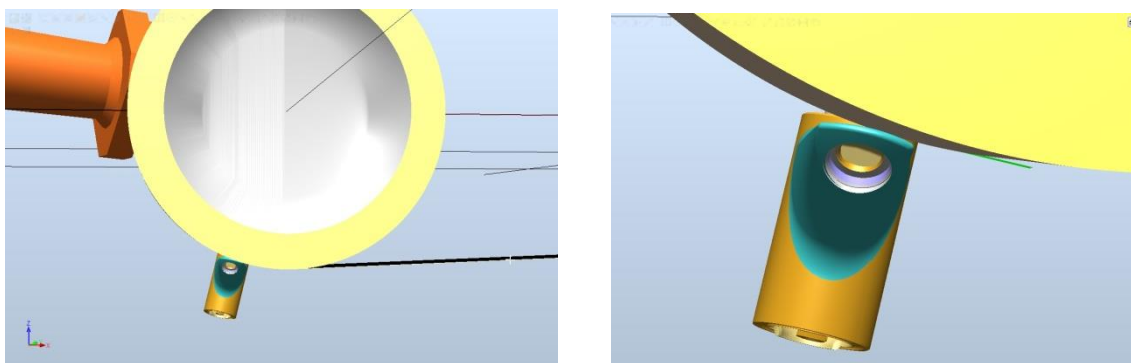
#### 4.1.3.5. Otimização do modelo da célula

Os detalhes do modelo gráfico, como furos, chanfres e outras funções aplicadas aos componentes modelados, aumentam a complexidade gráfica e diminuem o desempenho da simulação [21]. Por este motivo, decidiu-se desenhar uma versão simplificada das unidades de lixamento, removendo os detalhes que não são necessários para a simulação e programação. Foi também modelada a cinta da lixa, conforme se mostra na Figura 69.



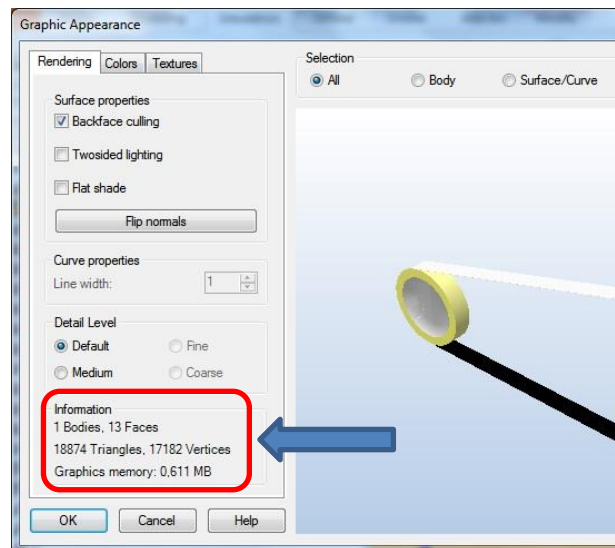
**Figura 69 Simplificação do modelo da unidade de lixamento no Inventor**

De forma a obter uma melhor resolução gráfica da unidade de lixamento no RobotStudio, foram alteradas as configurações de importação das geometrias. É possível aumentar a precisão da geometria importada para a célula virtual à custa de uma representação com um maior número de polígonos. Desta forma reduziu-se o erro máximo permitido para representar uma superfície esférica, alcançando-se uma melhor representação da unidade de lixamento na célula virtual, conforme se pode constatar na Figura 70, em contraste com a Figura 67.



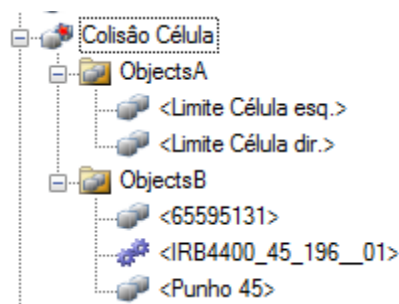
**Figura 70 Resolução gráfica da roda após alterado o nível de detalhe no RobotStudio**

Após a importação foi possível comparar as características gráficas da unidade simplificada, face ao primeiro conjunto. Como se pode ver na Figura 71 (e através da comparação com os dados apresentados na Figura 68), mesmo com maior definição das unidades, este conjunto tem uma complexidade e tamanho gráfico menor, permitindo obter um melhor desempenho durante a simulação.



**Figura 71 Propriedades gráficas da unidade de lixamento simplificada**

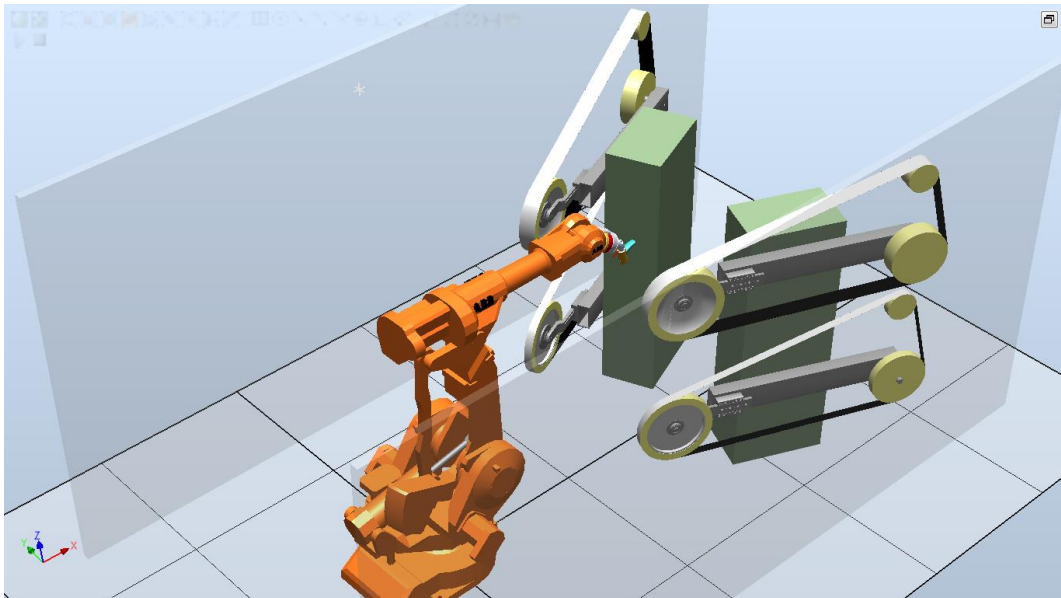
Para concluir a modelação da célula foram adicionados os painéis laterais, modelados diretamente no RobotStudio, que definem os limites da célula de trabalho. Definiu-se ainda uma monitorização de colisão entre o robô com ferramenta e peça e a célula de trabalho, como se mostra na Figura 72.



**Figura 72 Monitorização de colisões do robô com a estrutura da célula**

Por último, de forma a facilitar a visualização e programação, foram criadas diferentes vistas das unidades de lixamento, sendo possível comutar facilmente entre estas durante a

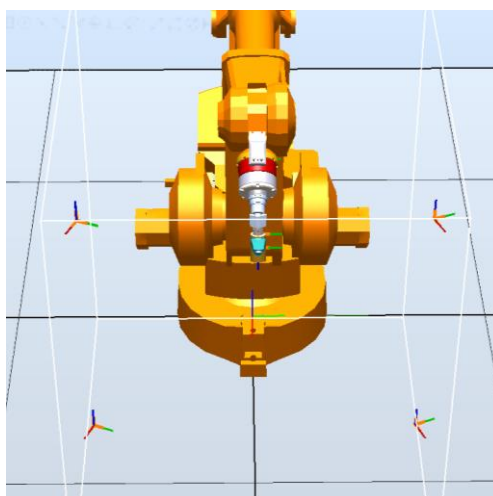
programação dos pontos. Apresenta-se na Figura 73 o modelo completo da célula de lixamento “Mepsa 13” utilizada para os testes finais dos programas.



**Figura 73** Modelo final da célula de lixamento “Mepsa 13”

#### **4.2. CALIBRAÇÃO DA CÉLULA DE TRABALHO**

Os resultados “práticos” da programação *offline* dependem da aproximação à realidade do modelo da célula efetuada no RobotStudio. Este não é totalmente exato, existindo desvios de modelação e posicionamento dos componentes entre o modelo e os valores reais na célula de trabalho. Assim sendo, é necessário proceder à calibração dos *workobjects* das unidades de trabalho e do TCP definido, apresentados na Figura 74.



**Figura 74** Localização dos *Workobjects* e TCP definidos no modelo

#### 4.2.1. DEFINIÇÃO DOS *WORKOBJECTS* DAS UNIDADES

Para a definição dos *workobjects* das unidades de lixamento foi utilizado o dispositivo apresentado na Figura 75.

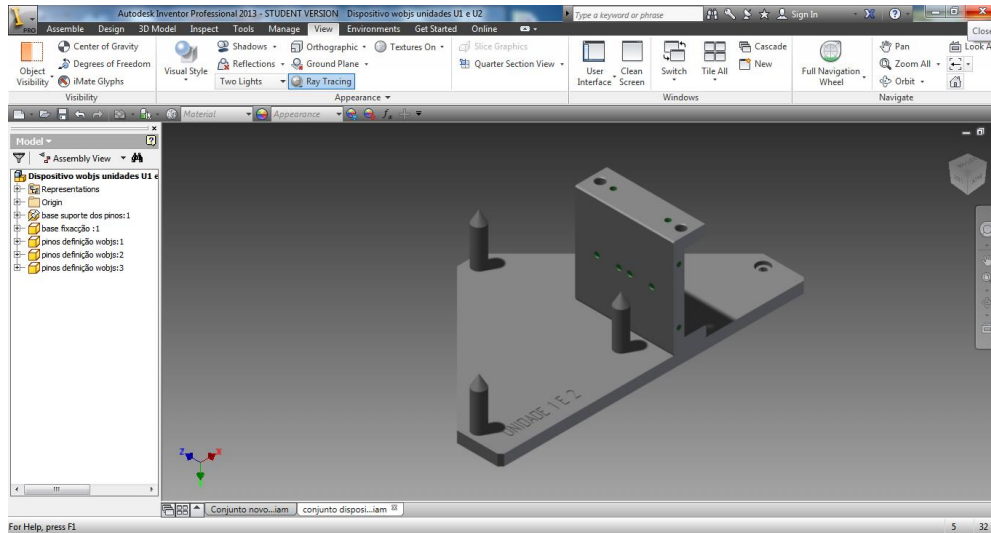


Figura 75 Dispositivo de definição de *workobjects*

Este é acoplado diretamente ao braço da unidade de trabalho, permitindo, com um único dispositivo, definir os *workobjects* das quatro unidades de trabalho e mantendo o sentido do sistema de coordenadas para as quatro unidades, facilitando a concepção e afinação do programa. A montagem do dispositivo no braço da unidade é apresentada na Figura 76.

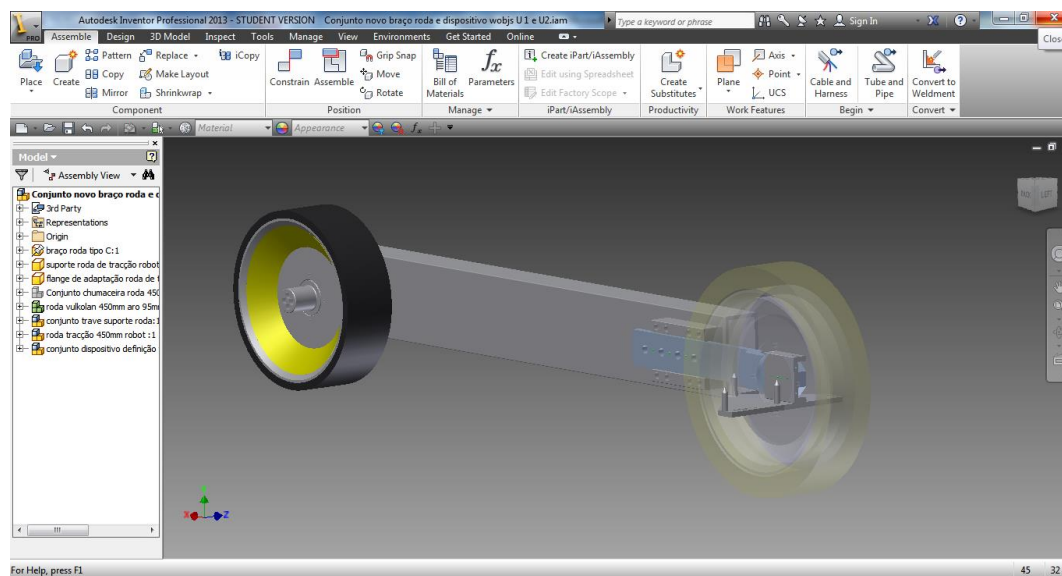
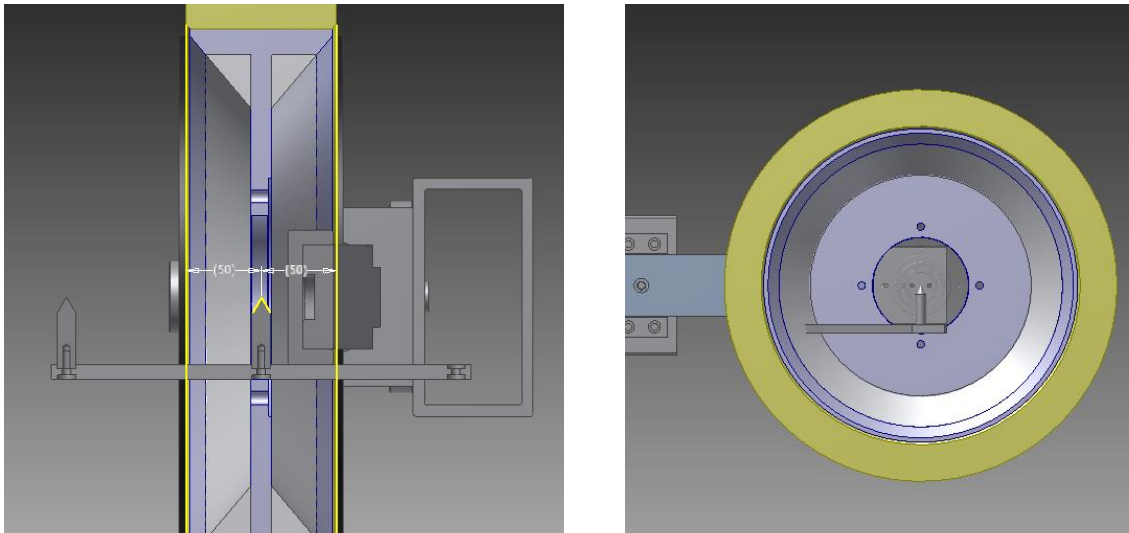


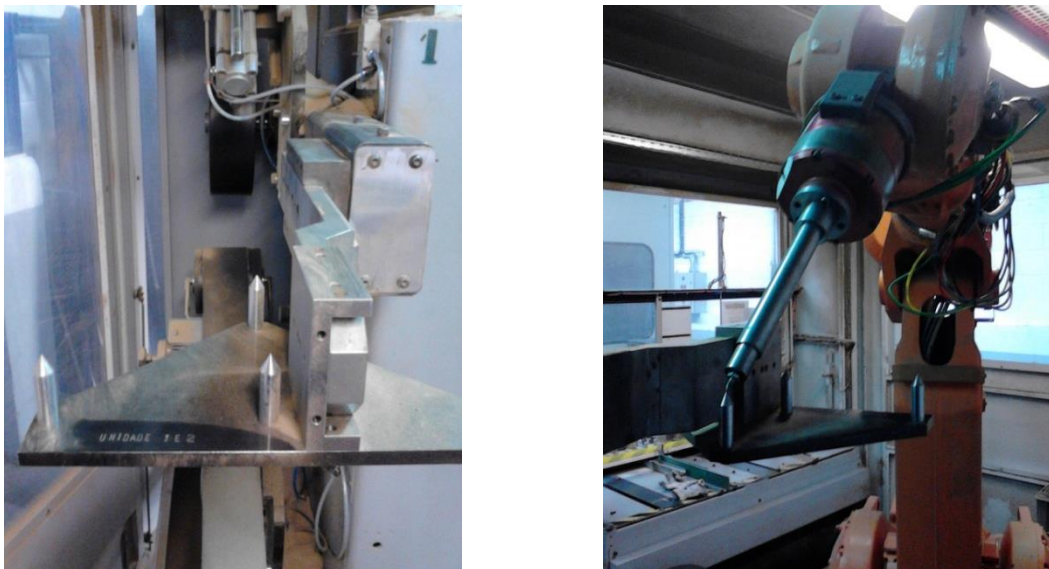
Figura 76 Dispositivo de definição dos *workobjects* montado no braço da unidade

Na Figura 77 demonstra-se o posicionamento deste dispositivo face à roda de lixamento, estando a origem do sistema de coordenadas – ponto “x1”, exatamente no centro da roda.



**Figura 77** Localização do *workobject* face à roda da unidade de lixamento

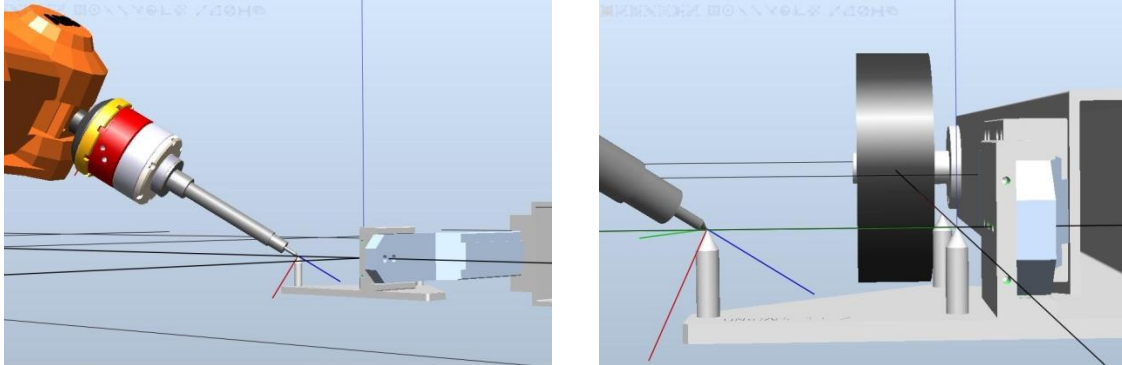
Este dispositivo permite a calibração manual dos *workobjects* baseada em três pontos, realizando-se o alinhamento entre a ponteira de teste do robô e os três pontos de referência do dispositivo (“x1”, “x2” e “y1”). A Figura 78 apresenta o desenrolar deste processo.



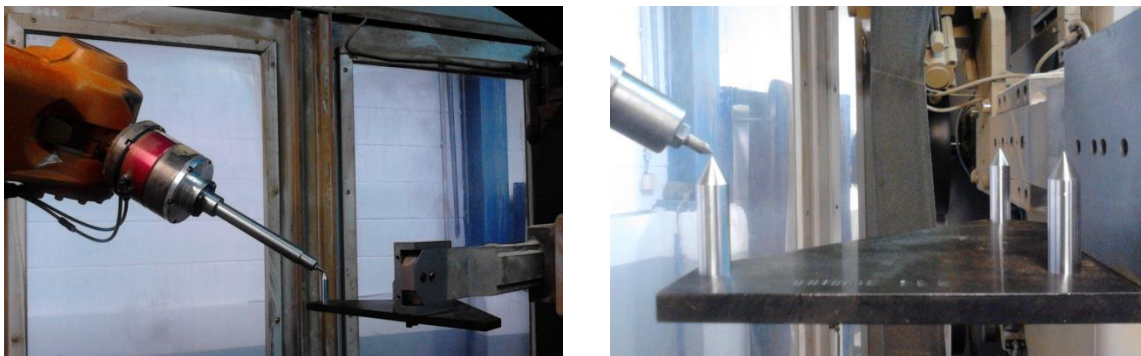
**Figura 78** Definição dos *workobjects* da unidade de lixamento nº1 na célula “Mepsa 13”

Após a definição de todos os *workobjects* desenvolveu-se um programa no RobotStudio para testar o posicionamento dos *workobjects* definidos e, simultaneamente, a precisão e repetibilidade de posicionamento do robô. Foram assim criados *targets* nos pontos de

referência do dispositivo e nas extremidades das rodas, verificando-se o movimento e posicionamento entre estas e comparando-se também o posicionamento no mesmo *target* com diferentes configurações das juntas do robô. Na Figura 79 apresenta-se o teste realizado no RobotStudio e na Figura 80 o resultado obtido na célula real.



**Figura 79** Execução do programa de teste de precisão e repetibilidade de posicionamento do robô no RobotStudio



**Figura 80** Resultados da execução do programa de teste de precisão e repetibilidade de posicionamento na célula “Mepsa 13”

#### **4.2.2. DEFINIÇÃO DA *TOOL* UTILIZADA**

Conforme referido anteriormente, a programação *offline* depende da calibração dos referenciais com os da célula real, sendo por isso necessário fazer a calibração do ponto de referência da ferramenta face ao valor definido numericamente.

Para a calibração do TCP no robô optou-se pelo método de seis pontos, quatro de aproximação a uma ponteira fixa de referência no “mundo” com orientações distintas e dois outros que definem a direção de “xx” e de “zz” (ver Figura 81), ficando assim a ferramenta totalmente definida no espaço.

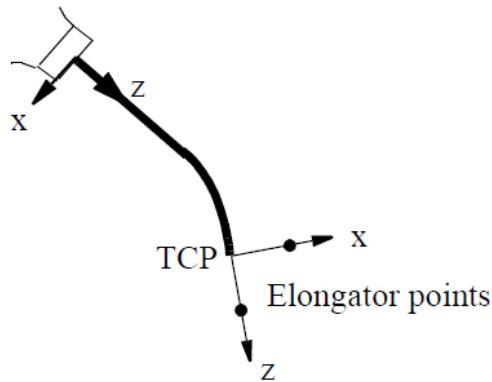


Figura 81 Método de definição do TCP através de seis pontos [11]

Na Figura 82 apresentam-se os menus de definição da *tool* apresentados na consola do controlador S4CPlus.

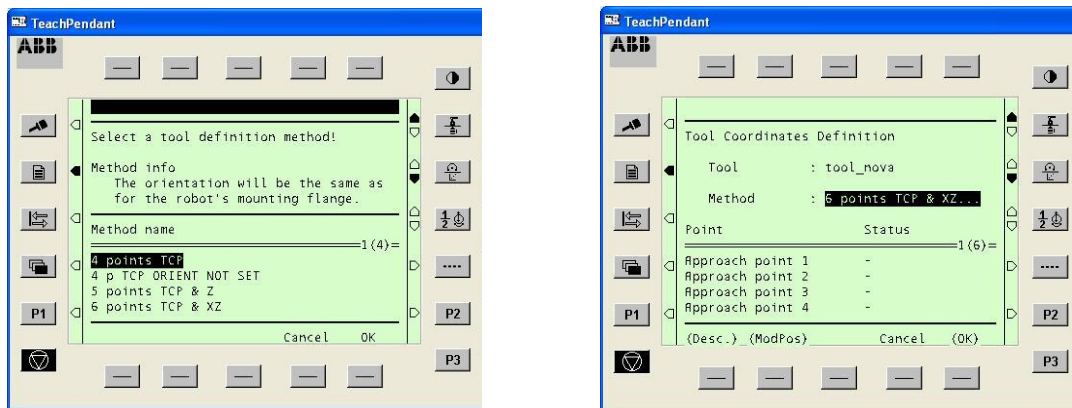
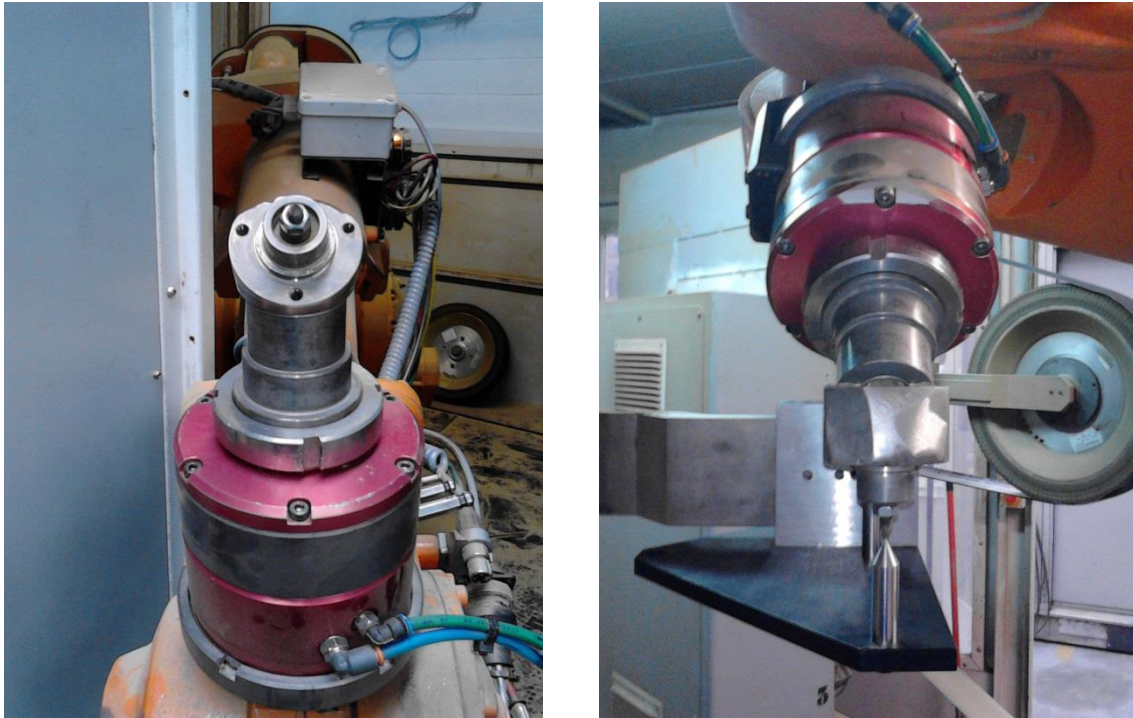


Figura 82 Menus de definição de uma nova *tool* na consola do controlador S4CPlus

Como é possível ver na Figura 83, a definição do TCP foi feita utilizando uma ponteira acoplada na cunha do punho a 45° e com o auxílio do dispositivo de definição de *workobjects*, utilizando uma das ponteiras deste (y1) como ponto fixo de referência no “mundo”.



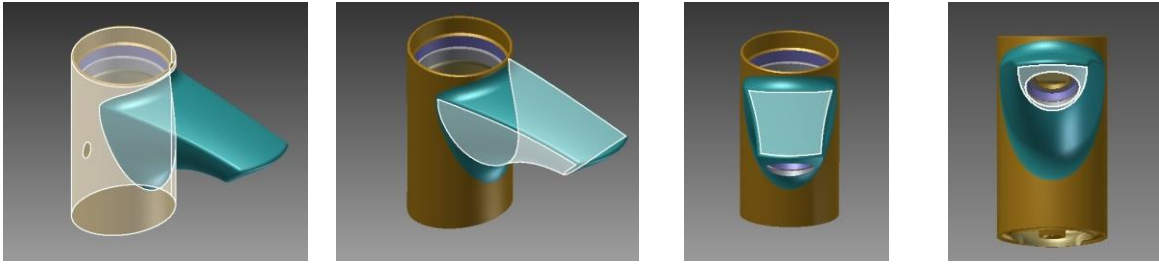
**Figura 83** Processo de definição da nova *tool* na célula “Mepsa 13”

Após a definição dos seis pontos, são obtidos os resultados do cálculo do TCP com o erro associado, tendo-se obtido um erro médio de 0,1 mm (média das distâncias dos pontos de aproximação ao TCP calculado).

### **4.3. DESENVOLVIMENTO DO PROGRAMA**

Antes de dar início à programação deve-se fazer um planeamento prévio do trabalho a executar, passando este pela decomposição do processo global em operações de trabalho bem definidas, facilitando assim a conceção e posteriores ajustes do programa.

Na Figura 84 mostram-se as principais superfícies do corpo da torneira que necessitam de ser lixadas (da esquerda para a direita na figura: corpo, parte inferior, parte superior e a saída da bica), tendo sido feita a decomposição do processo global em rotinas com as trajetórias de lixamento para estas superfícies. Para o desenvolvimento de cada rotina avaliou-se a peça em conjunto com os afinadores e lixadores experientes e, em conjunto, foram delineadas algumas estratégias e movimentos necessários ao sucesso do lixamento em algumas zonas e superfícies mais problemáticas.



**Figura 84 Principais faces do corpo a lixar**

Para conseguir obter um bom acabamento da superfície da torneira é necessário desbastar algum material de forma a remover as impurezas e imperfeições provenientes do processo de fundição. De forma a controlar a taxa de remoção de material ao longo da superfície da torneira, é efetuado um controlo de picagem a uma amostra de torneiras no início da produção, permitindo garantir que o processo de lixamento desenvolvido está a remover a taxa de material necessária, sendo para isso controlados os pontos assinalados na Figura 85.



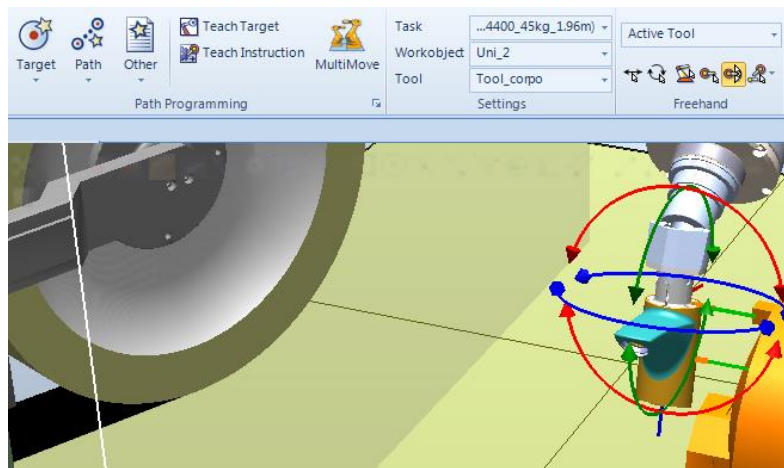
**Figura 85 Pontos de controlo da remoção de material**

O processo desenvolvido deve permitir remover e desbastar a película superficial de material, de forma homogénea, ao longo de toda a superfície da torneira, removendo as picagens feitas nos pontos de controlo, isto sem remover demasiado material que poderia causar imperfeições de acabamento, tais como desvios de contorno, levando à rejeição da peça.

Analisados os pontos essenciais a ter em consideração na conceção do método de lixamento apropriado, passa-se agora a explorar as ferramentas disponibilizadas pela programação *offline* que auxiliam e aceleram o seu desenvolvimento.

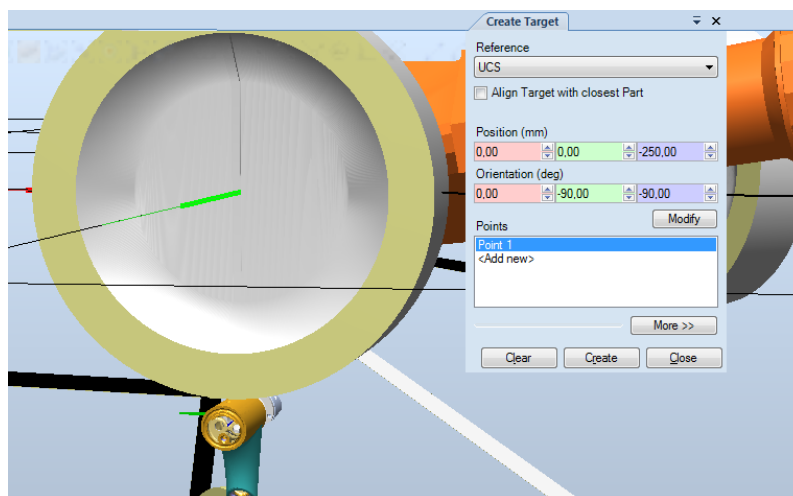
O *software* da ABB, o RobotStudio, permite fazer a programação em ambiente gráfico, com comandos idênticos aos encontrados no robô real. Disponibiliza ainda outras

funcionalidades avançadas que auxiliam a programação permitindo, por exemplo, utilizar a geometria da peça ou da ferramenta para auxiliar o posicionamento do robô durante a sua movimentação manual (*freehand*), como se mostra na Figura 86, podendo escolher-se os vértices, arestas ou outro ponto de interesse da geometria para posicionar o robô. É também possível criar os *targets* e caminhos automaticamente nas arestas e contornos dos sólidos, podendo-se definir um ângulo de aproximação e um *offset* ao ponto escolhido.



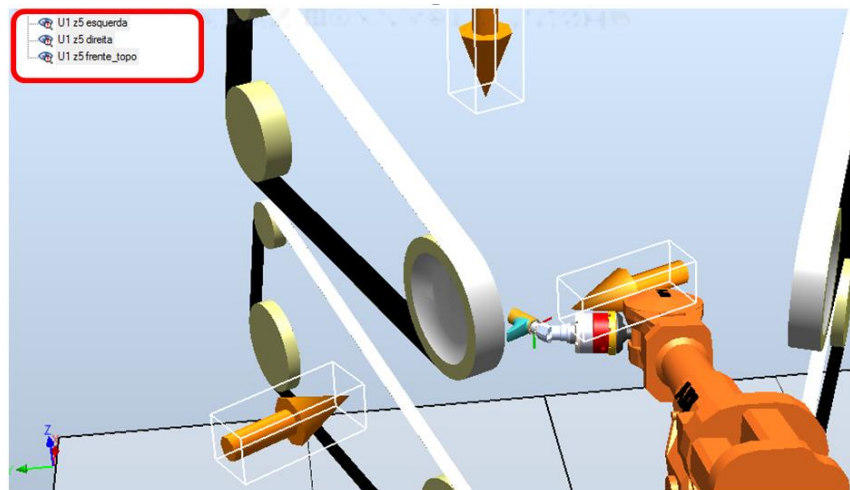
**Figura 86** Reorientação da ferramenta do robô (e do corpo da torneira) no RobotStudio

Contudo, é também possível definir ou alterar um *target* numericamente, sendo esta opção interessante quando é possível medir facilmente a distância do contorno a programar ao *workobject*. Na Figura 87 mostra-se, a título de exemplo, a definição de um *target* recorrendo-se às dimensões do raio do corpo e do raio da roda para se encontrarem as coordenadas do *target*.



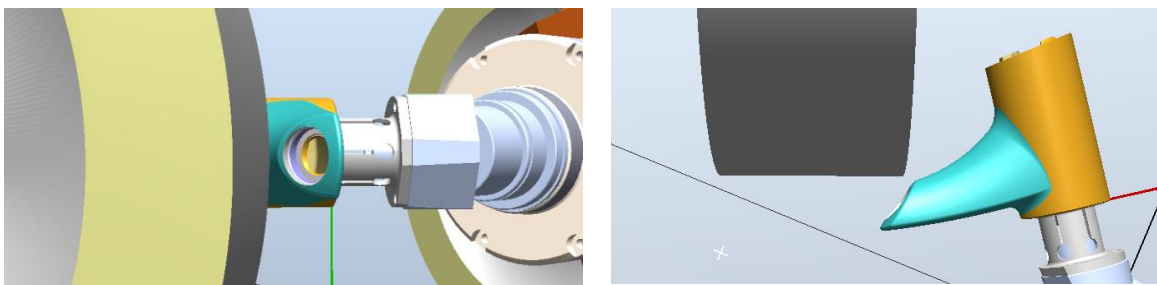
**Figura 87** Criação de um *target*

Durante o posicionamento e definição dos *targets* sentiu-se alguma dificuldade em obter as vistas necessárias à visualização da zona de contacto pelos constantes ajustes necessários à janela de visualização do RobotStudio. Uma forma de minimizar o problema passou pela definição de várias vistas fixas para as zonas mais utilizadas das unidades de trabalho, podendo assim verificar-se rapidamente a posição do corpo face à roda alternando entre vistas. Na Figura 88 podem ver-se as três vistas definidas para a zona de trabalho frontal da unidade de lixamento n.º 1.



**Figura 88 Vistas definidas para a visualização do centro da roda da unidade de lixamento n.º 1**

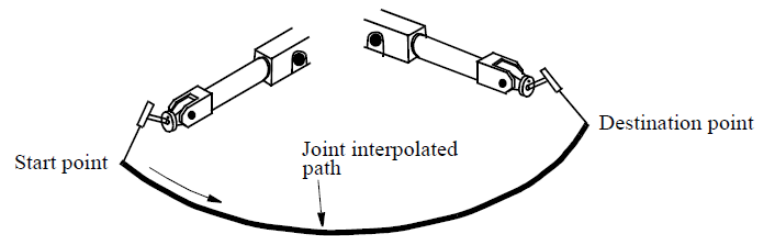
A Figura 89 apresenta, por sua vez, o *target* definido visto de duas vistas distintas (lateral esquerda e superior), onde se constata a sua necessidade para a correta definição deste *target* em concreto.



**Figura 89 Duas vistas distintas para o mesmo *target***

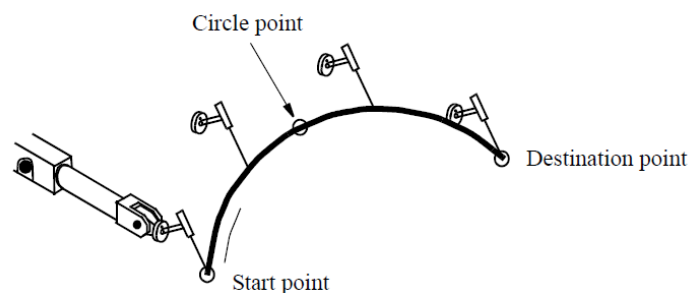
Após definidos os *targets* necessários passa-se à programação do caminho (*path*) que o robô irá percorrer. No robô ABB os principais movimentos que podem ser programados são:

- Movimento das juntas ou interpolação das juntas (MoveJ) – permite mover rapidamente o robô, numa trajetória não linear, como se pode ver na Figura 90. Este tipo de interpolação é usado para movimentos afastados da roda de trabalho, por exemplo para movimentar o robô rapidamente entre as unidades de trabalho;



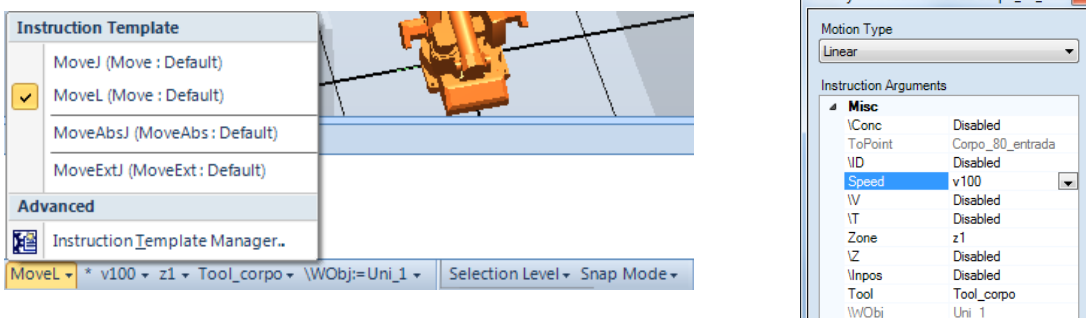
**Figura 90** Interpolação ao nível das juntas [11]

- Movimento linear ou interpolação linear (MoveL) – é utilizado para mover o TCP linearmente para o ponto de destino, sendo este tipo de interpolação usado mais frequentemente em zonas de contacto com a lixa;
- Movimento circular ou interpolação circular (MoveC) – é usado quando se pretende mover o TCP sob a forma de arcos de circunferência para o ponto de destino, como apresentado na Figura 91. Pode ser usado em alguns casos mais específicos.



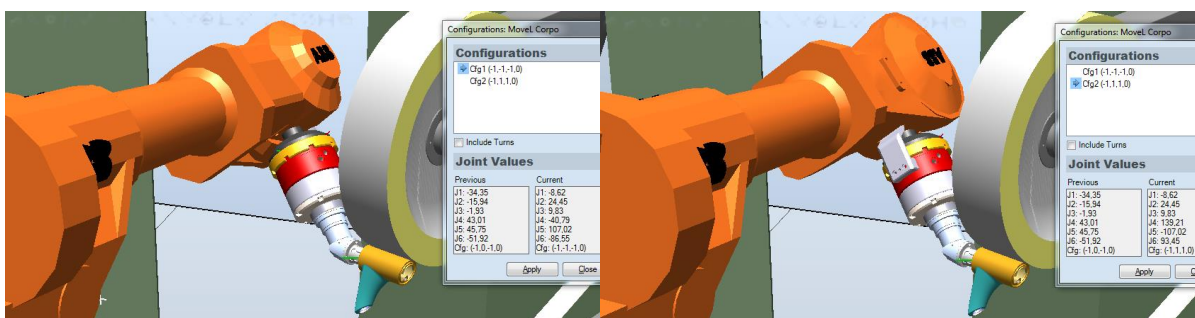
**Figura 91** Interpolação circular [11]

São também utilizadas diferentes zonas de aproximação aos pontos (“precisão” de aproximação) e velocidades de movimento de forma a produzir o nível de acabamento desejado. Os parâmetros do movimento a gerar podem ser predefinidos na barra inferior do RobotStudio, podendo ser posteriormente alterados através da funcionalidade “*Modify Instruction*”, como mostra a Figura 92.



**Figura 92** Alternativas para a configuração das instruções de movimento

Gerados os caminhos a percorrer, passa-se à configuração das posições do robô, escolhendo-se das diferentes soluções da cinemática inversa qual deverá ser adotada pelo robô. Na Figura 93 podem ver-se as duas soluções possíveis da cinemática inversa para alcançar um *target*.



**Figura 93** Diferentes configurações do robô para um *target*

O RobotStudio permite também a configuração automática dos *targets* de um *path* através da funcionalidade “*Auto Configuration*” [19]. A utilização desta requer alguns cuidados, pois o seu resultado depende do seguinte:

- Configuração atribuída ao primeiro *target* do caminho, sendo as configurações seguintes dadas em função desta;
- Os *targets* que já tiverem uma configuração atribuída não serão alterados;
- Não existe a ligação da autoconfiguração entre caminhos, sendo muitas vezes necessário manter o mesmo número de voltas dadas, por exemplo ao eixo 4 (eixo ilimitado neste robô), entre caminhos, sendo estas voltas revertidas nos restantes movimentos contrários do caminho seguinte.

Concluída a configuração do robô o caminho está pronto a ser simulado, podendo utilizar-se para testes a função “*Move Along Path*”. Após isto pode-se carregar o caminho no

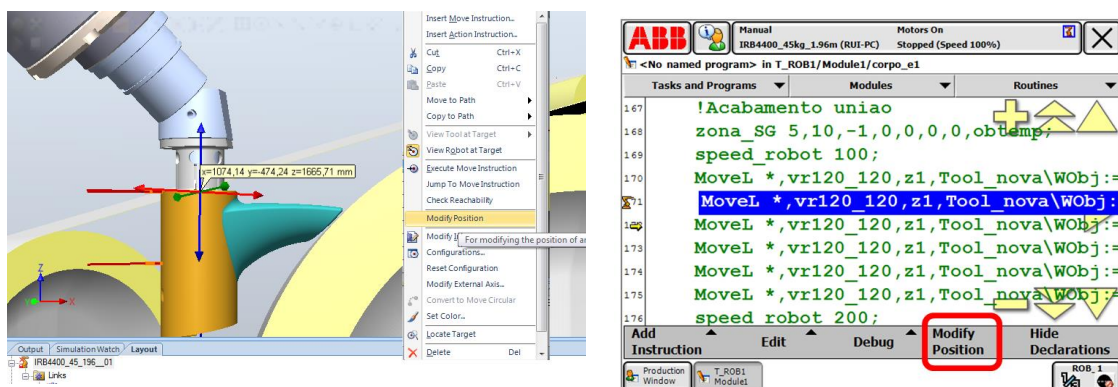
controlador virtual, sincronizando-o com a estação de trabalho, sendo gerado o código RAPID das instruções definidas no interface gráfico do RobotStudio.

É necessário repetir todo o procedimento descrito para cada rotina de lixamento definida, sendo necessário fazer a interligação das várias rotinas, quer em termos de movimento, quer em termos do controlo de equipamentos auxiliares na célula. Torna-se necessário programar em RAPID algumas instruções que controlam, por exemplo, as unidades de trabalho, sendo este controlo feito em rotinas de outros módulos do programa do robô. Assim, foi necessário fazer o estudo das funções e dos parâmetros envolvidos de forma a obterem-se os resultados pretendidos. Uma das rotinas mais utilizadas é a rotina de “zona” que permite fazer a compensação do diâmetro das rodas e pequenas afinações, definindo a posição do *workobject* ativo em função do diâmetro da roda, da unidade de trabalho, da zona da roda e dos *offsets* dados por questão de afinação. Apresentam-se na Figura 94 os parâmetros que intervêm no cálculo e o resultado desta rotina.



**Figura 94 Ilustração esquemática do funcionamento da rotina de “zona”**

Depois de gerado o programa ou caminho é possível, tal como no robô, ajustar os *targets* que sejam necessários recorrendo à funcionalidade “*Modify Position*” (como mostrado na Figura 95). Este é um processo utilizado para corrigir alguns movimentos do robô, afinando o processo de lixamento. Numa primeira fase procedeu-se à afinação *offline* de alguns movimentos do programa desenvolvido, por forma a conseguir reduzir o tempo de afinação *online* no robô.



**Figura 95 Ajuste dos *targets* no RobotStudio (esquerda) e na consola do robô (direita)**

No final de todo este processo é exportado o módulo do programa gerado pelo RobotStudio, sendo posteriormente carregado no robô. O programa do robô passa a conter o módulo com o processo de lixamento completo desenvolvido no RobotStudio e dois módulos com as funções auxiliares de controlo da célula de trabalho.

#### **4.4. PROCEDIMENTOS E RESULTADOS**

Nesta secção apresentam-se alguns dos resultados obtidos, verificando-se a aplicabilidade do processo realizado. Para permitir uma melhor associação e compreensão entre alguns dos resultados apresentados e os procedimentos que lhes deram origem, esta secção encontra-se organizada em subsecções, cada uma delas abordando os resultados do processo de lixamento de uma das superfícies do corpo. Em cada subsecção apresenta-se, em primeiro lugar, a identificação da superfície da peça a lixar, seguindo-se a exposição dos procedimentos e rotinas de lixamento envolvidas e, por fim, são apresentados os resultados obtidos.

##### **4.4.1. CORPO DA PEÇA**

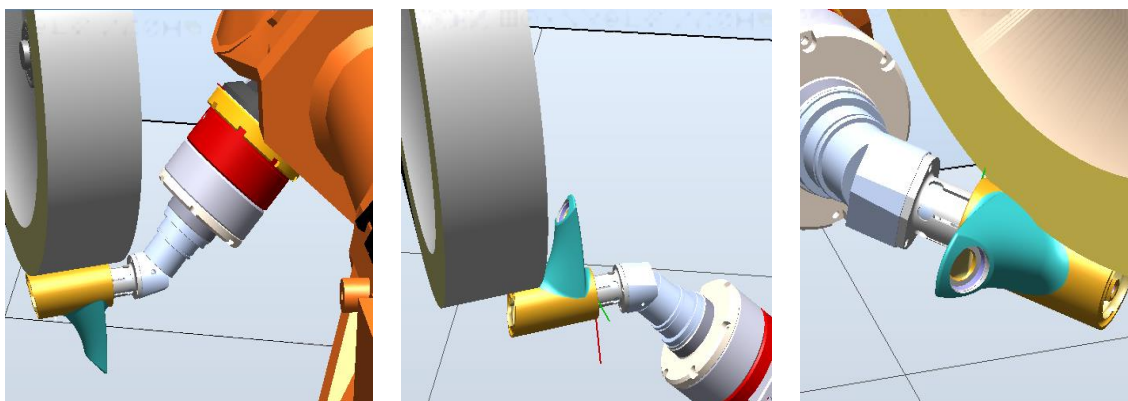
O corpo representa a maior área da superfície a lixar na peça, sendo necessário remover nesta zona as imperfeições da fundição, parte do gito localizado nas costas e, simultaneamente, manter o topo do corpo já maquinado para a cota final. Na Figura 96 pode ver-se a superfície do corpo da torneira antes do lixamento, onde se salientam as imperfeições e a saliência do gito.



**Figura 96 Costas do corpo da peça a lixar**

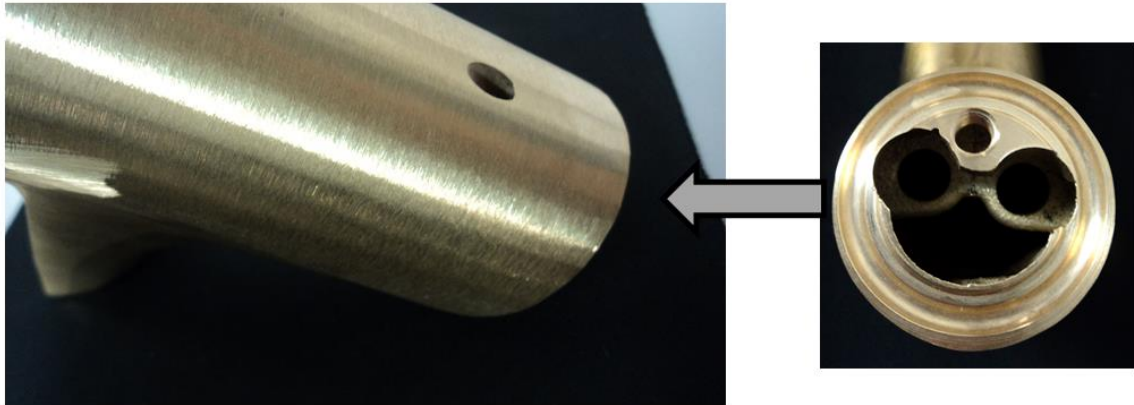
O programa para o lixamento do corpo foi feito de forma a conseguir obter um nível de desbaste uniforme e ao mesmo tempo permitir manter a zona maquinada no topo do corpo. Com o auxílio do RobotStudio foi possível espaçar uniformemente as passagens e realizar uma simetria das passagens da parte direita e parte esquerda do corpo da peça, contribuindo para um bom acabamento da superfície com a lixa grossa. As faces criadas na peça pela remoção de material na lixa grossa serão eliminadas pelo movimento realizado na lixa fina, deixando a superfície sem vincos, nem sulcos.

Na Figura 97 são mostrados três instantes do processo de lixamento do corpo, sendo visível na imagem da esquerda a zona de trabalho na roda escolhida devido à limitação do eixo 5 que, como se pode ver, se encontra próximo do valor máximo. Na imagem do meio mostra-se o lixamento da parte de baixo do corpo, por baixo da bica. Aqui foi necessário ter atenção ao movimento da bica próxima à roda. Na imagem da direita mostra-se a execução de um movimento de maior complexidade para definir o contorno entre a bica e o corpo.



**Figura 97 Rotinas de lixamento do corpo e cantos da peça**

Na Figura 98 apresenta-se a superfície do corpo após o lixamento. É possível ver os vincos das várias passagens com espaçamento uniforme, sendo estes removidos na passagem cruzada na lixa fina. A imagem da direita revela alguns problemas de concentricidade da peça, estando a ser removido mais material na parte da frente do corpo e menos na parte das costas, devido quer à localização e altura do gito, quer a algum desvio do próprio material.



**Figura 98 Resultado após o processo de lixamento com a lixa grossa**

#### **4.4.2. BICA DA PEÇA**

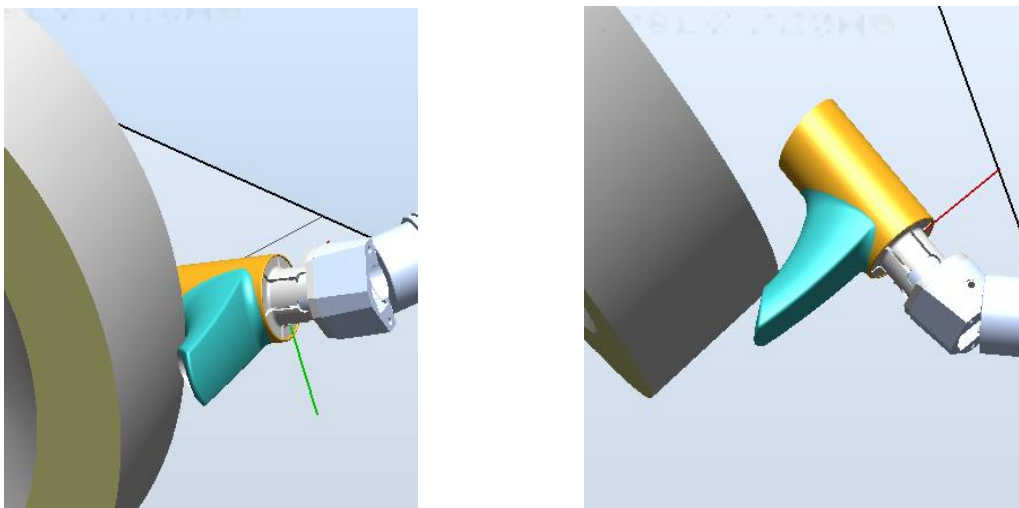
A bica é a parte da peça de contorno mais delicado, pelo seu raio de curvatura e contorno de ligação ao corpo, sendo necessário desbastar uniformemente ao longo desta curvatura, sem causar desvios de contorno. Ao mesmo tempo deve manter-se a simetria dos dois lados da peça, permitindo a mesma taxa de remoção de ambos os lados da torneira. Qualquer desvio do corpo ou no processo de carga pode comprometer o lixamento desta zona, aumentando o desbaste de um lado da peça em detrimento do lado oposto. A parte superior da bica é também a zona da torneira mais exposta sendo assim necessária a melhor qualidade de acabamento nesta zona.

A título exemplificativo é exposto o processo de lixamento da parte inferior da bica, podendo ver-se na Figura 99 a superfície a lixar, onde se observa a presença de alguma rebarba na parte da frente da bica e a meio do corpo.



**Figura 99 Bica da peça a lixar**

Devido à dificuldade do contorno, foi necessário fazer o lixamento em duas etapas distintas. A primeira etapa permitiu desbastar algum material ao seguir o contorno da bica na parte da frente da roda (conforme se mostra na Figura 100) criando pequenas faces de passagem. Recorrendo à programação *offline*, foi possível escolher o melhor espaçamento entre passagens de forma a adaptá-lo à variação do ângulo de curvatura do corpo, permitindo uma remoção de material homogênea ao longo de toda a superfície. Foi também tido em consideração o número de pontos gerados, de forma a permitir um rápido ajuste desta rotina por parte dos afinadores ao longo do ciclo de vida do produto.



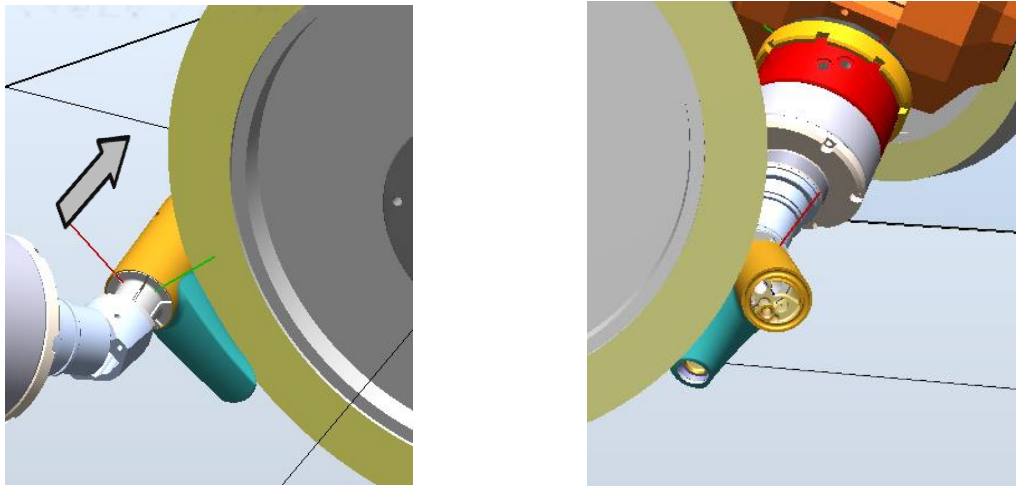
**Figura 100 Sequência da rotina de lixamento da parte de baixo da bica na unidade de lixamento n.º 1**

Na Figura 101 é possível ver as marcas bem definidas das várias passagens efetuadas, sendo estas marcas e vincos removidos na segunda etapa do processo de lixamento, ainda na lixa grossa.



**Figura 101** Resultados do lixamento da parte de baixo da bica após a primeira etapa

Na segunda etapa do processo foi feita uma passagem a cruzar o sentido da anterior, abrangendo-se toda a bica por arrasto com o auxílio de uma roda de menor dureza e de menor largura (50 mm, ao invés de 100 mm) instalada na unidade de lixamento número 2, removendo-se desta forma os vincos deixados pelo processo anterior. Na Figura 102 é assinalado o sentido do movimento anteriormente descrito, tendo sido dado mais pressão contra a roda a estes *targets*, o que na realidade será absorvido pela baixa dureza e flexibilidade da roda, adaptando-se aos contornos da peça.



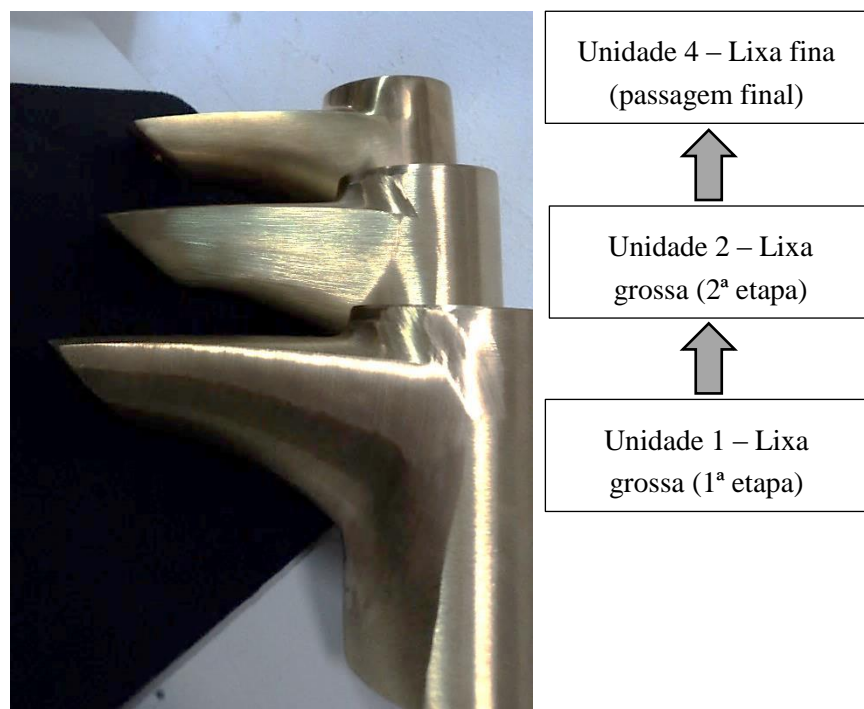
**Figura 102** Rotina de lixamento por arrasto cruzado na unidade de lixamento n.º 2

Apresenta-se na Figura 103 o resultado do lixamento da bica por arrasto, removendo esta passagem todas as marcas e vincos, mantendo bem definida a geometria da peça.



**Figura 103** Resultado do lixamento da parte de baixo da bica após a segunda etapa

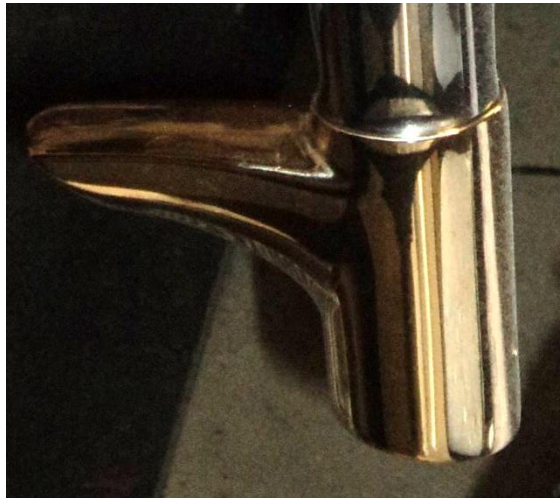
Nota-se que a superfície apresenta ainda alguns sulcos da passagem da lixa grossa, acabando estes por desaparecer durante as restantes passagens do processo (lixa fina e polimento). Na Figura 104 apresenta-se a evolução da superfície da bica ao longo do processo de lixamento.



**Figura 104** Evolução da superfície da bica da peça ao longo das três etapas de lixamento

Após o desenvolvimento de todo o processo de lixamento, com algumas das rotinas aqui apresentadas, a peça está pronta a ser polida. Neste processo de abrasão são removidas

algumas marcas e sulcos deixados pela lixa, conferindo à peça um aspeto brilhante, conforme mostrado na Figura 105.



**Figura 105**      **Peça polida**

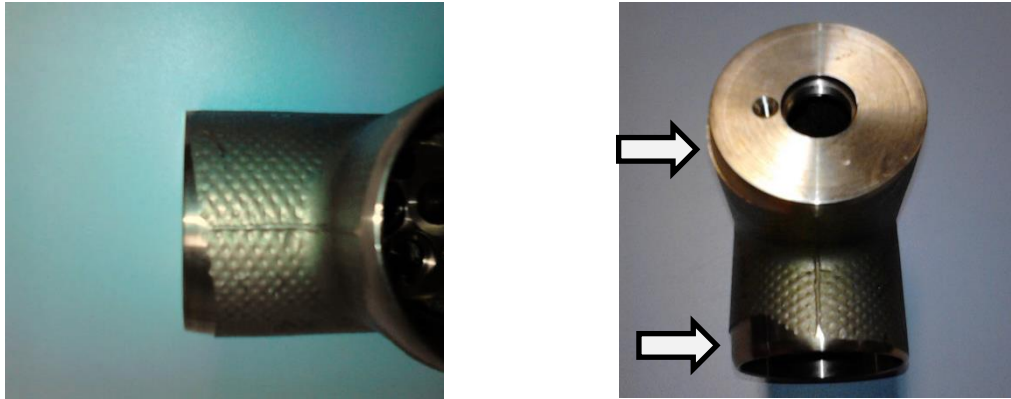


# 5. SISTEMA DE MEDIÇÃO E AJUSTE *ONLINE* DA FERRAMENTA DO ROBÔ

Ao longo deste capítulo é apresentado o desenvolvimento do sistema de medição e ajuste *online* da ferramenta do robô sendo, em primeiro lugar, identificado o problema em questão, seguindo-se a abordagem proposta para o sistema a desenvolver e a simulação e programação *offline* de um sistema de teste. Por fim são apresentados os resultados obtidos avaliando-se a sua aplicabilidade em contexto real, no robô disponível nos laboratórios do Instituto Superior de Engenharia do Porto (ISEP).

## 5.1. IDENTIFICAÇÃO DO PROBLEMA

As oscilações dimensionais das peças a lixar comprometem muitas vezes a qualidade global do lixamento, levando a afinações dos programas ou trabalhos de recuperação adicional e consequentes atrasos de produção. Estes desvios provêm das flutuações dos processos de fundição e de maquinação, apresentando-se na Figura 106 um exemplo dos desvios encontrados.

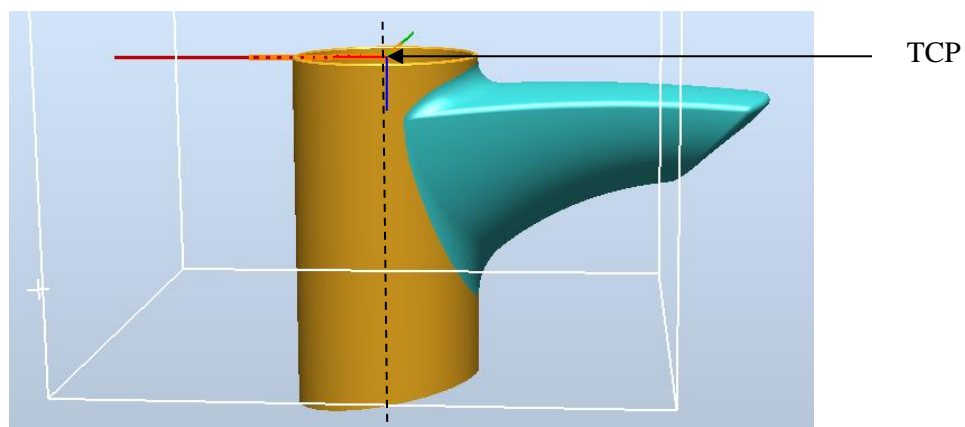


**Figura 106** Desvios dimensionais do corpo da torneira

De forma a permitir contornar este problema, é necessário desenvolver um sistema de medição e análise dimensional do corpo, permitindo ao robô adaptar-se de forma automática às dimensões das peças, corrigindo o processo de lixamento em função dos desvios apresentados por estas.

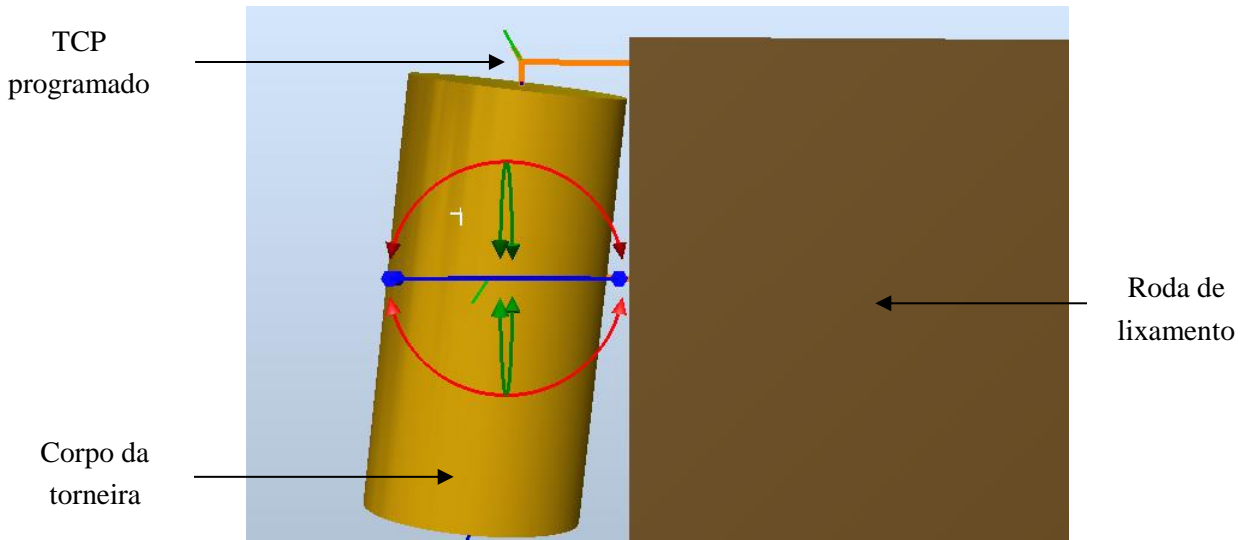
## 5.2. ABORDAGEM PROPOSTA

Para corrigir de forma eficaz a rotina de lixamento em função dos desvios que o corpo possa apresentar é necessário proceder à correção do TCP que o programa utiliza para o lixamento da peça, modificando-o de acordo com as dimensões e orientações medidas. Como o robô percorre todos os *targets* com o TCP definido no programa é possível, através da alteração da posição e orientação deste referencial, corrigir o posicionamento da peça em todos os pontos do programa. Na Figura 107 apresenta-se a posição e orientação do TCP do corpo da torneira.



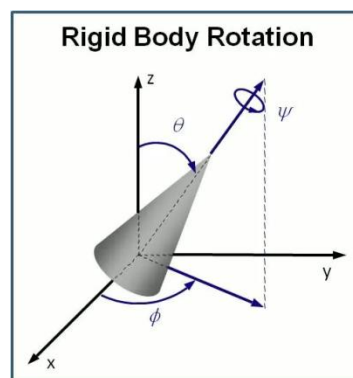
**Figura 107** Localização do ponto de referência da ferramenta

No caso de existirem desvios na peça, o TCP definido está deslocado face ao centro desta, comprometendo a qualidade de lixamento. Um exemplo desta situação pode ser observado no esquema da Figura 108, onde a peça apresenta uma componente de rotação em torno do eixo “y” face ao TCP programado, estando esta rotação ampliada para facilitar a interpretação da figura.



**Figura 108** Posição do corpo face ao TCP programado (desvio ampliado)

O TCP é definido pela sua posição e orientação em relação ao sistema de coordenadas da flange do robô (ou “*tool0*”), podendo a orientação ser descrita segundo os ângulos de Euler (ver Figura 109) ou através de quatérniões unitários.

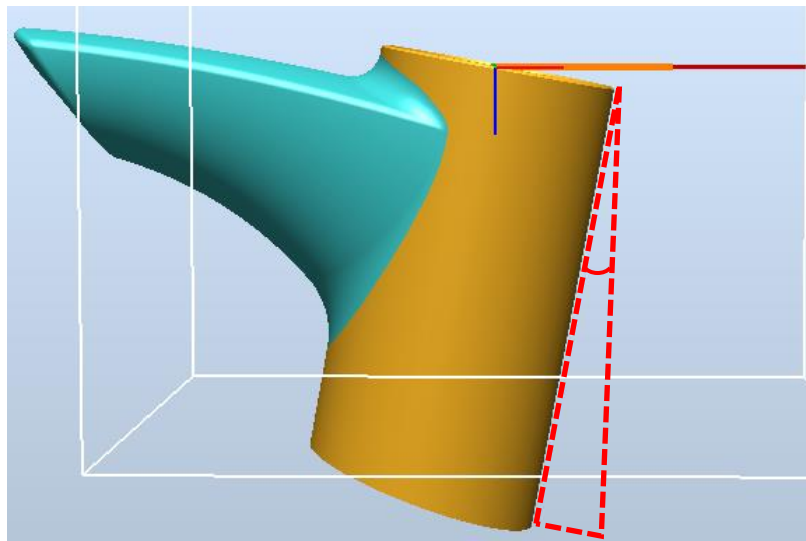


**Figura 109** Ângulos de Euler – *Roll, Pitch e Yaw* [20]

A utilização dos ângulos de Euler é a mais intuitiva, pois existe uma similaridade entre as coordenadas lineares e as coordenadas angulares. No entanto, tem como desvantagem o problema das singularidades ou “*gimbal lock*”, onde é perdido um grau de liberdade.

Por uma questão de simplificação, decidiu-se proceder à correção de apenas uma posição e uma orientação, sendo possível fazer posteriormente a extrapolação dos resultados obtidos para um ajuste completo da ferramenta – três posições e três orientações.

De forma a determinar a variação da orientação e da posição da peça face ao TCP pode proceder-se à medição do corpo em dois pontos da peça (topo e base), obtendo-se assim o ângulo de rotação de uma das componentes face ao valor definido no TCP, conforme se mostra no esquema da Figura 110. Nesta figura é possível observar o desvio na orientação da peça em relação ao TCP definido para esta.



**Figura 110** Medição do ângulo de rotação “ $r_y$ ” em relação ao TCP definido

Para a medição destes dois pontos decidiu-se utilizar o robô como sistema de medição, deslocando a peça até um sensor de proximidade/contacto, sendo assim possível obter a distância ou as coordenadas do ponto de contato. Conhecendo a dimensão da peça e considerando as restantes orientações corretas, é possível determinar matematicamente a nova posição/orientação da *tool*. Depois de calculada a nova posição/orientação é feita a correção do TCP usado ao longo do programa, sendo este reorientado e/ou reposicionado conforme o desvio apresentado.

Uma forma de testar esta abordagem passa pela conceção de um programa de testes que permita testar a resposta do sistema ao desvio da peça. Para isso é necessário fazer uma verificação e medição inicial da posição e orientação da ferramenta, provocar em seguida um desvio intencional da sua posição e orientação e, em seguida, fazer uma nova medição que permitirá fazer o ajuste do TCP de acordo com o desvio observado face à situação

inicial. Assim, é possível obter na verificação final o alinhamento e posição inicialmente verificados. A sequência descrita encontra-se esquematizada na Figura 111.



**Figura 111** Sequência de teste do sistema de ajuste do TCP

Em seguida são descritos os passos principais da conceção do programa e da simulação que permitiram testar e validar a abordagem proposta.

### **5.3. PROGRAMAÇÃO E SIMULAÇÃO NO ROBOTSTUDIO**

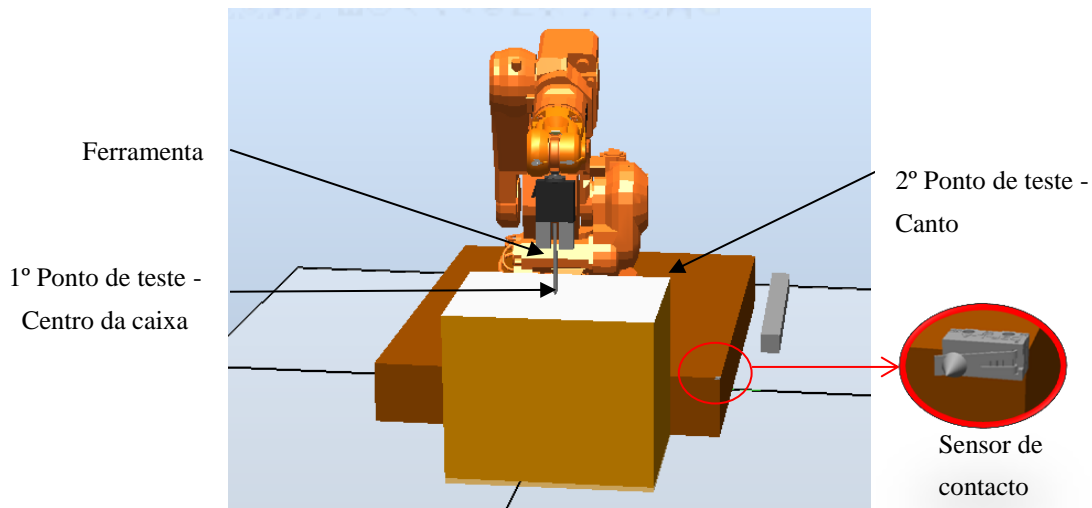
De forma a testar a abordagem proposta utilizou-se o robô ABB IRB 140 do ISEP. Recorreu-se ainda assim à programação *offline* aproveitando todas as suas potencialidades para:

- Acelerar o desenvolvimento do sistema, pela programação quer em ambiente gráfico quer diretamente em RAPID;
- Testar múltiplas abordagens, permitindo definir e simular vários tipos de desvios da ferramenta, sendo facilmente mensuráveis;
- Facilitar e acelerar o *debug* do código, visualizando-se todas as variáveis necessárias do programa RAPID desenvolvido.

#### **5.3.1. MODELAÇÃO DA CÉLULA DE TESTES**

A célula de testes é essencialmente constituída por uma garra pneumática, a ferramenta cilíndrica em teste, uma caixa para testes de posicionamento, o sensor de medição e uma paleta de suporte, como se pode observar na Figura 112.

Para a modelação e calibração do mundo do robô recorreu-se aos mesmos procedimentos referidos no Capítulo 4 deste documento, definindo-se os *workobjects* (caixa e paleta) e o TCP (ferramenta) no robô.

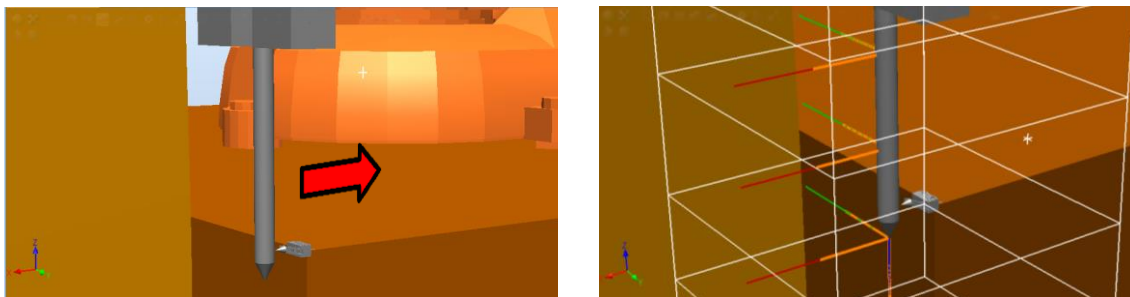


**Figura 112** Componentes da célula de teste

O sensor utilizado é um sensor mecânico de fim de curso com uma pequena ponteira, sendo explicado no tópico seguinte o seu funcionamento.

### 5.3.2. MEDIÇÃO DA FERRAMENTA

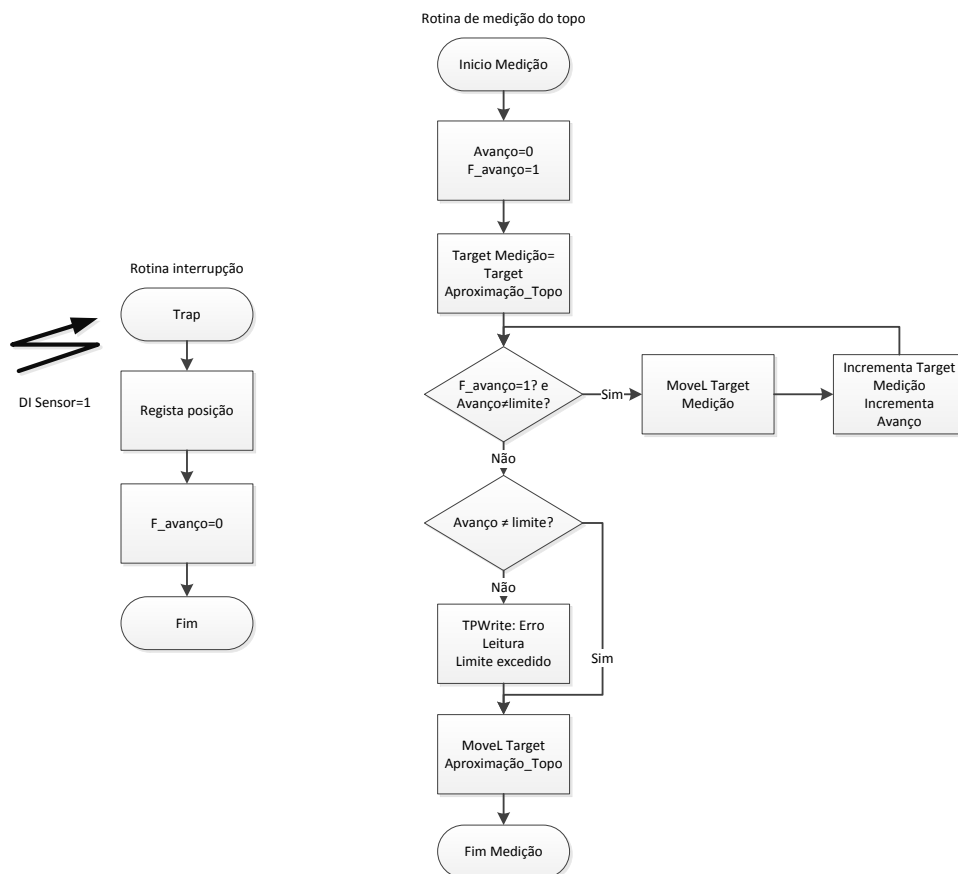
A medição da ferramenta deve ser feita sempre que se pretende obter os novos valores do TCP. Para cada orientação, é necessário realizar duas medidas à peça a medir: uma medida na base do corpo e uma medida no topo. Para isso, é feita uma aproximação inicial à ponteira do sensor (conhecendo a geometria da peça, posicionando-a a uma distância segura), seguindo-se uma aproximação progressiva até ocorrer o contacto, obtendo-se a distância da base e do topo do corpo cilíndrico ao TCP programado. Na Figura 113 pode ver-se o local de medição do topo (sentido de medição indicado pela seta) e a localização das *tools* definidas para a realização dos testes (*tool* base, meio e ponta).



**Figura 113** Medição da ferramenta e *tools* definidas (*tool* base, meio e ponta)

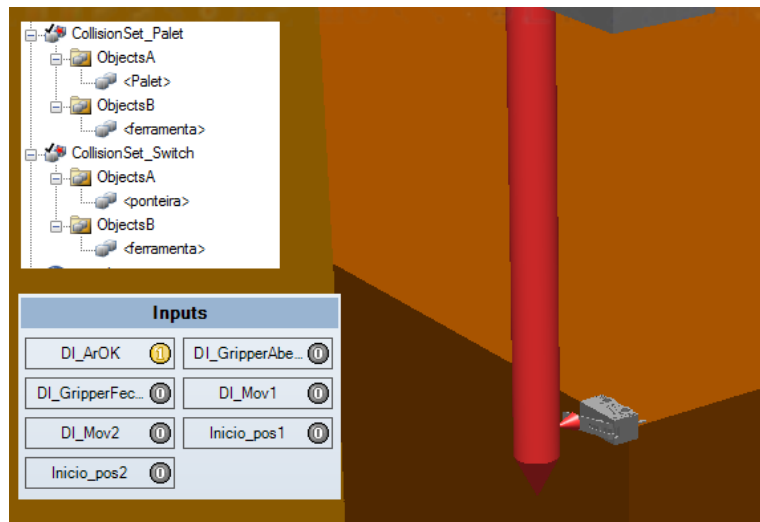
Na Figura 114 é apresentado o fluxograma simplificado da rotina de medição, sendo esta rotina programada em RAPID. Conforme se pode ver no fluxograma, inicialmente são

declaradas as variáveis de controlo do avanço e feita a cópia da posição do *target* de aproximação à ponteira para um *target* de medição. Seguidamente, se for possível fazer o avanço (caso o limite de segurança não estiver excedido e caso não esteja já em contacto com o sensor), o robô move-se linearmente (MoveL) para o *target* de medição, sendo este *target* posteriormente deslocado (incrementando a coordenada respectiva) e repetindo-se a movimentação do robô para o *target* agora deslocado, ficando o robô um incremento mais próximo do contacto com o sensor. Este processo repete-se, aproximando progressivamente a ferramenta ao sensor, até à deteção de contacto ou até ao limite de segurança estabelecido. No momento de deteção é gerada uma interrupção, executando a rotina de “*trap*” onde é suspenso o movimento do robô e registada a posição de contacto com o sensor. No final da rotina o robô regressa ao ponto inicial de aproximação à ponteira. Esta rotina é utilizada para fazer a medição da ponta, seguindo-se a medição da base, sendo possível após estas duas medidas determinar uma orientação e uma posição para a *tool* definida.



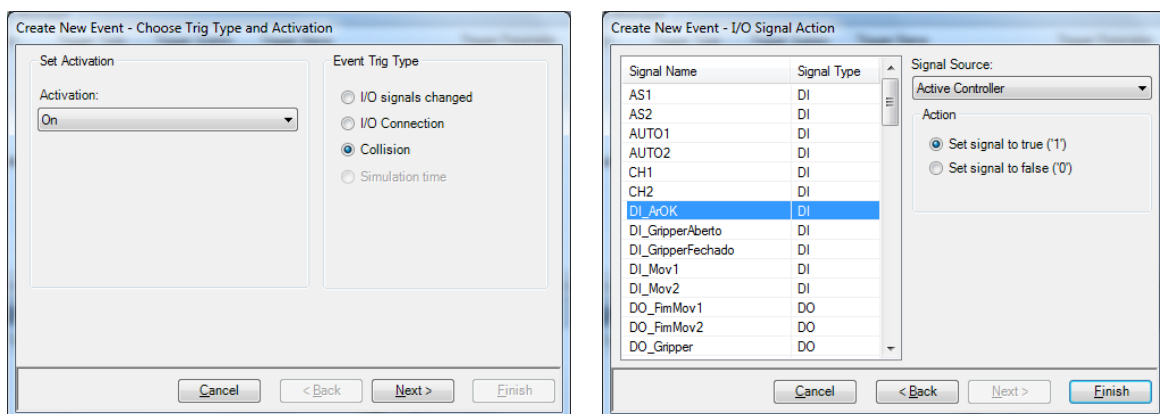
**Figura 114** Fluxograma representativo do funcionamento da rotina de medição e interrupção

Para efetuar a simulação do funcionamento do sensor no RoboStudio foi necessário definir uma deteção de colisão entre a ponteira e a ferramenta e criar um evento que permite trocar um sinal de entrada (*Digital Input – DI*), gerando a interrupção que regista a posição no momento do contacto e suspende o avanço do robô. Na Figura 115 apresenta-se a deteção de colisão definida (“CollisionSet\_Switch”) e o contacto com o sensor na simulação, passando o sinal “DI\_ArOK” ao nível lógico alto “1”.



**Figura 115** Contacto da ferramenta com a ponteira do sensor no ambiente de simulação

A definição do evento requer que se tenha definido previamente a deteção de colisão com o sensor, associando-a à ativação do evento. É também especificada a ação de resposta do evento, como se mostra na Figura 116.



**Figura 116** Definição de um novo evento

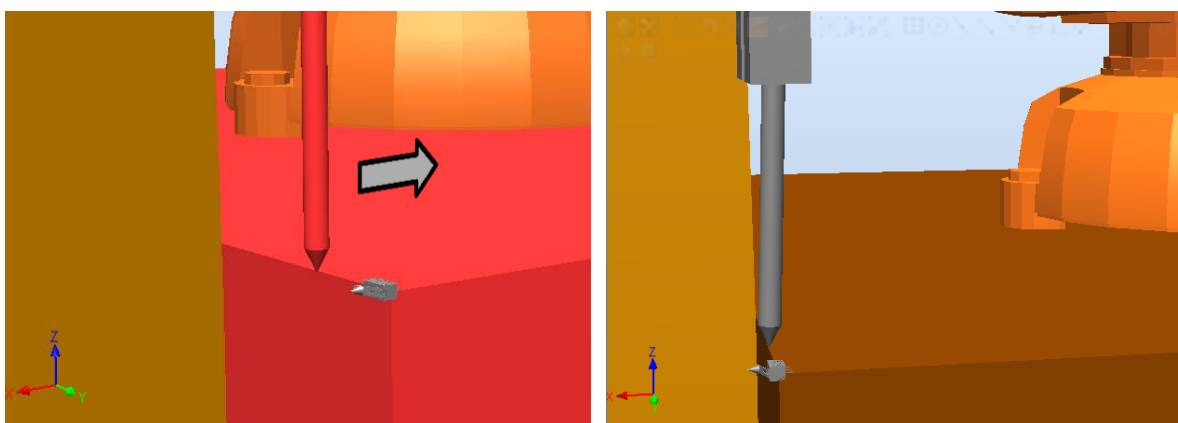
São assim necessários dois eventos para fazer a medição, de forma a efetuar a troca do sinal no início e no fim da colisão (indicados na Figura 117), tal como acontece no sensor real aquando do contato físico.

Events		Trigger Type	Trigger Name	Trigger Par...	Action Type	Action N...	Acti...
Add...	Collision	CollisionSet_Switch	Started	Change I/O	DI_ArOK	1	
	Collision	CollisionSet_Switch	Ended	Change I/O	DI_ArOK	0	

**Figura 117** Eventos de medição – início e fim de colisão com o sensor

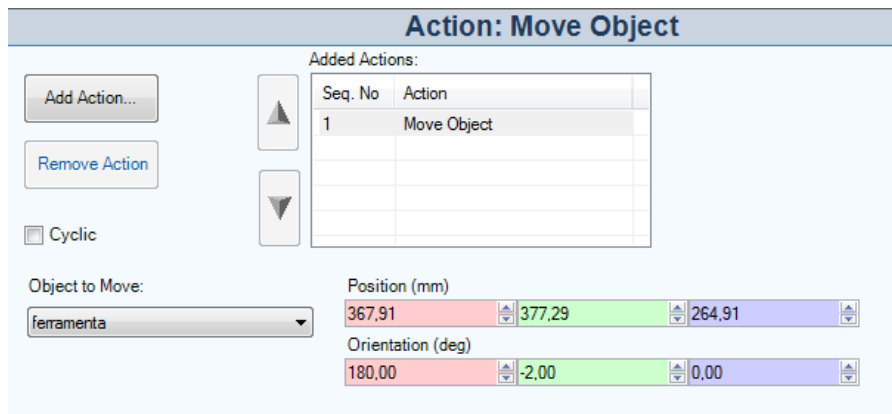
### 5.3.3. DESVIO DA FERRAMENTA

De forma a testar a correção do TCP, é possível no RoboStudio proceder à alteração da posição e da orientação da ferramenta, originando um desvio mensurável face à posição programada do TCP. Este desvio é feito durante a simulação, aquando da colisão com a paleta, através do reposicionamento e reorientação da ferramenta. Na célula real, este desvio é dado pelo avanço do robô face ao ponto de contacto da ponta da ferramenta. A Figura 118 apresenta a deteção da colisão com a paleta e posterior reorientação da peça.



**Figura 118** Simulação da colisão com a paleta

É então necessário definir um novo evento para a deteção de colisão com a paleta, especificando a ação do evento para mover a ferramenta, criando um desvio nesta face ao TCP. Como apresentado na Figura 119, é possível definir a posição e orientação da ferramenta, permitindo testar o sistema através da verificação da resposta de correção obtida.



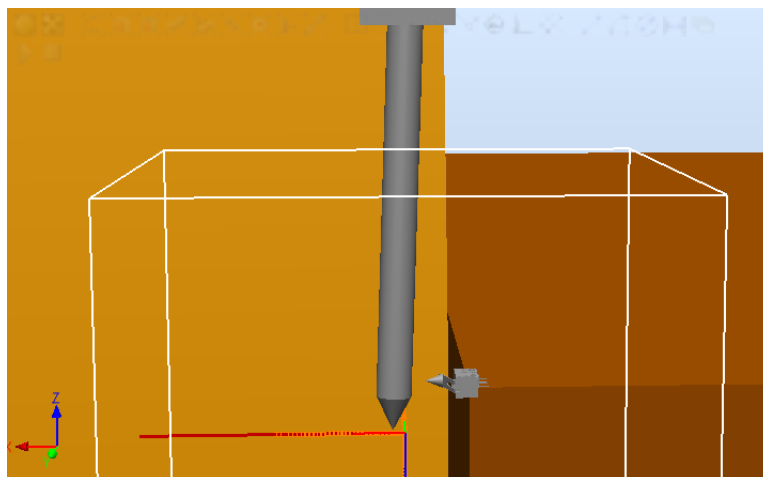
**Figura 119** Desvio do ferramenta através de um novo evento de colisão com a paleta

Apresentam-se na Figura 120 os eventos definidos na simulação para o teste do sistema, permitindo o funcionamento do teste na simulação *offline*.

Events		Activation	Trigger Type	Trigger Name	Trigger Par...	Action Type	Action N...	Action Par...
Add...	On	Collision	CollisionSet_Palet	Started	Move Object	Move Ob...	ferramenta	
Delete	On	Collision	CollisionSet_Switch	Started	Change I/O	DI_ArOK	1	
	On	Collision	CollisionSet_Switch	Ended	Change I/O	DI_ArOK	0	

**Figura 120** Eventos definidos para a simulação

Após o desvio da peça é necessário proceder novamente à medição e correção do TCP, de forma a obter o correto posicionamento da ponteira nos pontos de teste. Na Figura 121 é possível verificar o desvio do TCP face à posição real da ponta da ferramenta.



**Figura 121** Desvio do TCP face à posição real da ferramenta

### 5.3.4. CORREÇÃO DO TCP

Após a obtenção da nova distância do topo e da base da ferramenta ao sensor é possível calcular o desvio face à primeira medição, retirando o desvio da posição e da orientação. Para isso, é calculado o ângulo do corpo face ao TCP, obtido através da razão trigonométrica entre o valor da distância dos catetos conhecidos – diferença entre medições e diferença em “z” das duas medições, conforme se apresenta esquematizado na Figura 122.

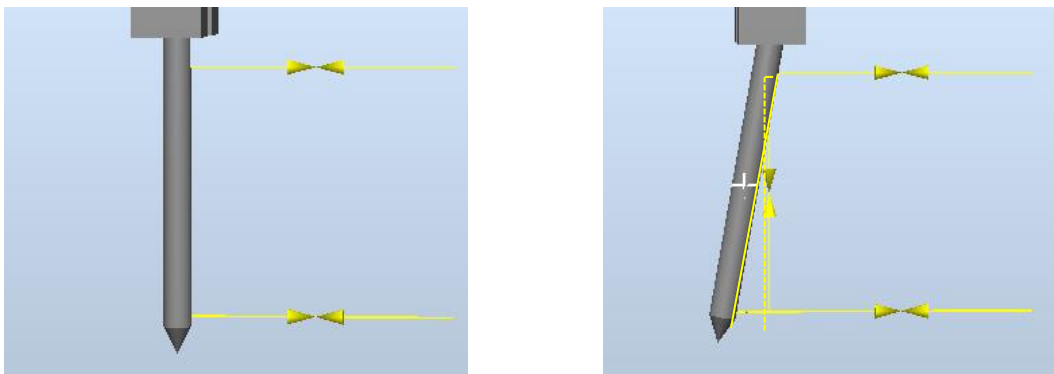


Figura 122 Esquema das medidas efetuadas antes e após o desvio (desvio aumentado)

Calculado o ângulo de desvio, é então possível fazer o ajuste do TCP definido no programa. A ferramenta encontra-se definida no código RAPID numa variável do tipo *tooldata* (cuja estrutura se apresenta na Figura 123) onde se encontram descritas as suas características.

```
< dataobject of tooldata >  
< robhold of bool >  
< tframe of pose >  
  < trans of pos >  
    < x of num >  
    < y of num >  
    < z of num >  
  < rot of orient >  
    < q1 of num >  
    < q2 of num >  
    < q3 of num >  
    < q4 of num >  
< tload of loaddata >  
  < mass of num >  
  < cog of pos >  
    < x of num >  
    < y of num >  
    < z of num >  
  < aom of orient >  
    < q1 of num >  
    < q2 of num >  
    < q3 of num >  
    < q4 of num >  
  < ix of num >  
  < iy of num >  
  < iz of num >
```

TCP

Figura 123 Estrutura de dados *tooldata* [21]

O TCP encontra-se definido no campo *tool frame*, contendo este a posição e orientação da ferramenta. Conforme se pode ver na Figura 123, a orientação da ferramenta face à flange do robô é descrita pelos quatro quaterniões ( $q_1, q_2, q_3, q_4$ ), sendo assim necessário fazer a conversão do ângulo obtido. Para isso, foi usada a função OrientZYX [21], permitindo obter a orientação ( $q_1, q_2, q_3, q_4$ ) através dos ângulos de Euler ZYX. Assim sendo é possível alterar a rotação da *tool* (componente “rot”) da seguinte forma:

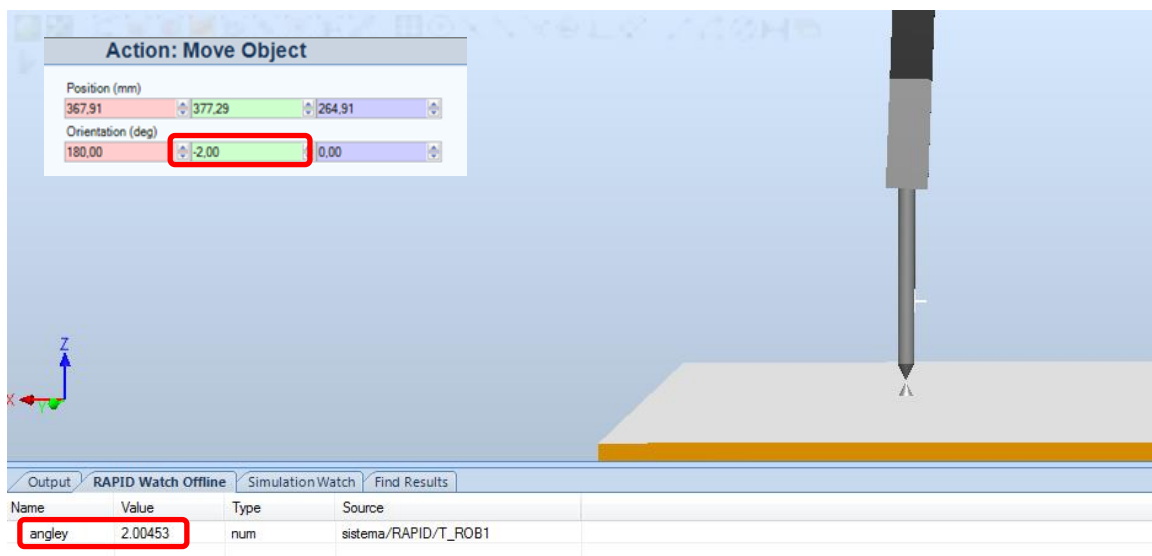
```
tool_teste.tframe.rot:=OrientZYX(anglez,angley,anglex);
```

A posição do TCP pode ser modificada recorrendo à alteração da componente de translação - “trans”:

```
tool_teste.tframe.trans.x:=offset_x;
```

### 5.3.5. RESULTADOS DA SIMULAÇÃO NO ROBOTSTUDIO

A simulação no RobotStudio permitiu agilizar o processo de testes, pela facilidade de obtenção de desvios mensuráveis e possibilidade de posterior comparação com a resposta do sistema real, podendo avaliar-se o comportamento e estabilidade do processo desenvolvido. Na Figura 124 apresenta-se o resultado da simulação, onde se verifica a correta orientação da ferramenta no *target* de teste ao centro da caixa, após a correção do desvio de  $-2^\circ$  em “y” apresentado pela ferramenta.



**Figura 124** Resultados dos testes realizados em ambiente de simulação

## 5.4. RESULTADOS DO TESTE NO ROBÔ

Após validado o funcionamento do sistema na simulação *offline*, prosseguiu-se para a simulação do sistema com o robô ABB IRB 140. Antes de iniciar o teste verificou-se o alinhamento do robô (posição de calibração) e o alinhamento da ferramenta na garra pneumática, como se mostra na Figura 125.

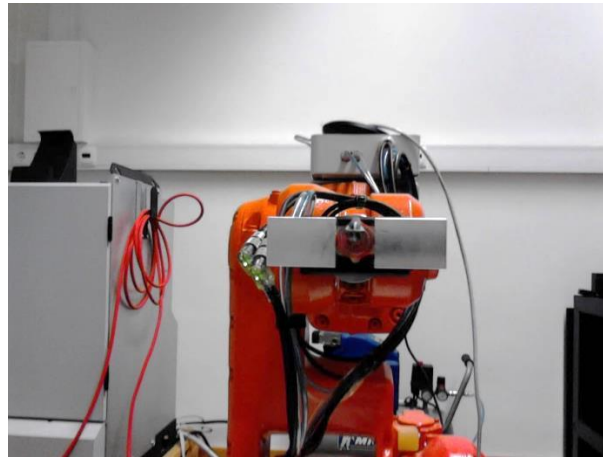


Figura 125 Robô IRB 140 em posição de calibração (todas as juntas a zero)

### 5.4.1. VERIFICAÇÃO INICIAL DOS PONTOS DE TESTE

O programa de teste começa por fazer o posicionamento da ferramenta em dois *targets* de teste (parafuso ao centro da caixa e no canto da caixa) podendo-se verificar o correto alinhamento e posicionamento inicial, conforme se mostra na Figura 126.

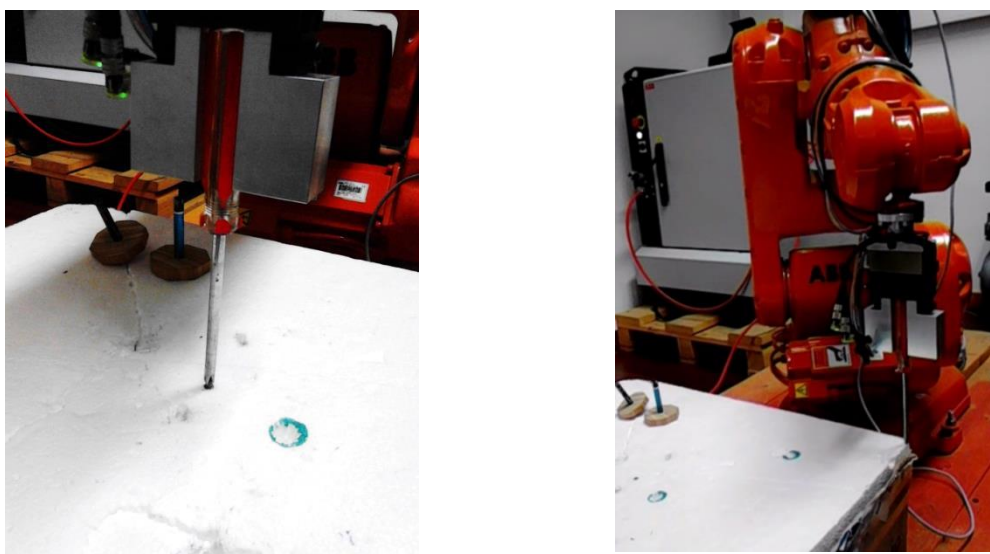


Figura 126 Posição da ferramenta no início do teste

#### 5.4.2. SENSOR DE MEDIÇÃO

Para a medição da ferramenta utilizou-se um sensor de fim de curso (*microswitch*) de acionamento mecânico, sendo este tipo de sensor capaz de ser acionado por uma pequena força de contacto. Este é colocado no percurso de medição, conforme se pode ver na Figura 127. A ligação do sensor é feita diretamente ao controlador, utilizando uma entrada digital disponível, passando esta a nível lógico elevado quando o sensor for acionado (contacto normalmente aberto).

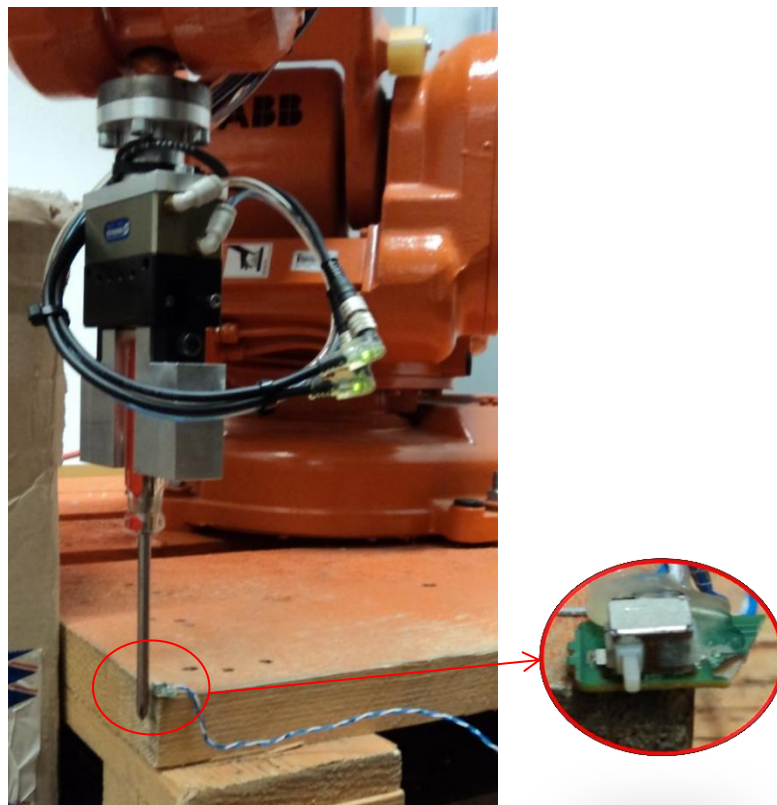
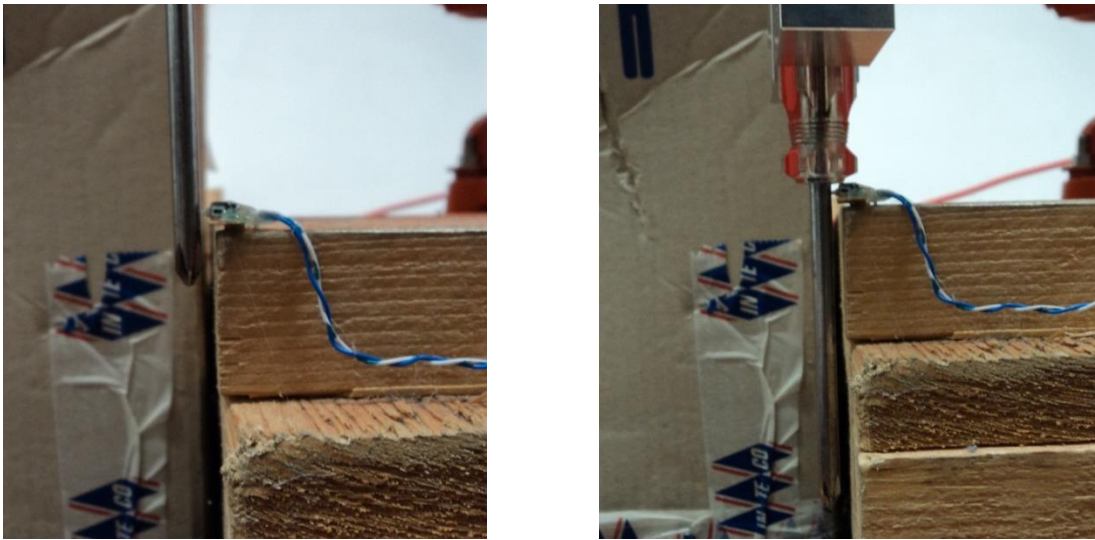


Figura 127 Sensor do sistema de medição da ferramenta

#### 5.4.3. MEDIÇÃO DA FERRAMENTA

A rotina de medição da ferramenta permite obter os valores da distância da base e do topo da peça, relativamente ao *target* inicial de medição (referência), calculando-se posteriormente a sua posição e orientação. Foram realizadas duas medições: uma antes do desvio da ferramenta, obtendo-se as medidas iniciais, e outra após o desvio (colisão) da ferramenta, obtendo-se assim a respetiva correção do TCP, de forma a garantir que mesmo após a rotina de desvio (colisão) era reposta a posição e orientação inicialmente observada na rotina de verificação final. A medição ocorreu de forma análoga à observada na

simulação *offline*, espaçando-se ao máximo a medição do topo e da base (ver Figura 128), obtendo-se um menor erro de cálculo.



**Figura 128** Medição do topo e da base da ferramenta

#### **5.4.4. DESVIO DA FERRAMENTA**

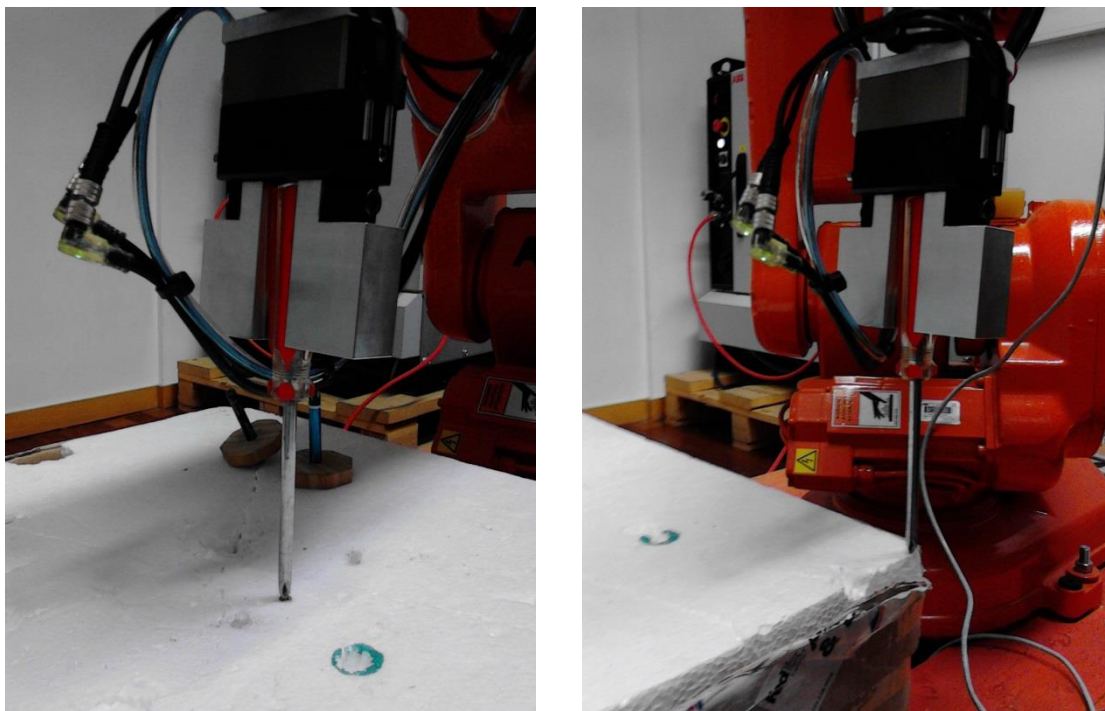
O desvio da ferramenta foi conseguido através do contacto/colisão da ferramenta com a palete, como mostrado na Figura 129, desviando-a segundo o sentido de rotação permitido fisicamente pela garra (um grau de liberdade).



**Figura 129** Rotina de colisão da ponteira com a palete

#### 5.4.5. RESULTADOS – VERIFICAÇÃO FINAL DOS PONTOS DE TESTE

Efetuada o desvio da ferramenta, o programa executa uma nova medição de forma a efetuar o ajuste necessário ao desvio apresentado face ao TCP inicial. Após calculado o ajuste é então corrigido o TCP, realizando-se posteriormente um novo teste de posicionamento para testar e validar a resposta de correção do sistema. Apresentam-se na Figura 130 os resultados obtidos, onde se pode verificar o correto alinhamento da ferramenta.



**Figura 130** Posição da ferramenta no fim do teste

Em suma, pela análise dos resultados, verificou-se que o sistema desenvolvido corrigiu o TCP de forma eficaz, conseguindo-se manter o posicionamento nos pontos de teste mesmo após uma colisão. Assim sendo, resta replicar este sistema de pequena escala numa célula de trabalho da Grohe, de forma a auxiliar o método atual de ajuste ponto-a-ponto, diminuindo ou eliminando algumas paragens para afinação.

No capítulo seguinte é complementado o estudo da programação *offline* com novos desenvolvimentos e também é descrito como se procedeu à alteração da interface homem-máquina, de forma a eliminar algumas pausas da célula robotizada.

## 6. EXPANSÃO DA APLICAÇÃO DA PROGRAMAÇÃO *OFFLINE*

Este capítulo vem dar continuidade ao trabalho até agora desenvolvido, contribuindo para o aumento da produtividade das células robotizadas do departamento. Para isso é necessário expandir a aplicação da programação *offline* a novos produtos e novas células, utilizando as potencialidades deste método de programação para realizar estudos que conduzam ao desenvolvimento de ferramentas necessárias à execução destes novos processos. Isto permite trabalhar o programa em antecipação, permitindo reduzir o tempo necessário até à colocação do produto no mercado.

De forma a complementar o trabalho, é fulcral abranger outras células de trabalho do departamento, explorando-se, para isso, um novo *software* de simulação e programação *offline* da KUKA, o KUKA Sim, e comparando-se os processos desenvolvidos. De seguida apresenta-se a realização de novos programas no RobotStudio, sendo descrito o estudo de uma nova ferramenta recorrendo tanto à simulação como à modelação de uma nova célula de trabalho. Por fim pretende-se reduzir outros desperdícios das células robotizadas, sendo este um dos conceitos *lean manufacturing*.

## 6.1. PROGRAMAÇÃO *OFFLINE* – CÉLULA COM ROBÔ KUKA

Ao longo desta secção apresenta-se a modelação, programação e simulação de uma célula de lixamento com robô KUKA, utilizando-se para isso o *software* do próprio fabricante - o KUKA Sim. De forma a estabelecer-se um termo de comparação entre as duas aplicações de simulação e programação *offline* (RobotStudio e Kuka Sim) decidiu-se escolher o mesmo produto apresentado no Capítulo 4 (torneira de lavatório da coleção EuroSmart), realizando-se também a avaliação das vantagens e desvantagens entre os dois *layouts* para o mesmo produto.

### 6.1.1. CÉLULA DE TRABALHO

A célula de trabalho escolhida para a realização da programação *offline* é uma das mais recentes do departamento e, por isso, existe a necessidade de aumentar a gama de produtos que nesta podem ser produzidos. Esta célula de trabalho é igualmente constituída por quatro unidades de lixamento, mas com um *layout* distinto das restantes, estando as unidades de lixamento frente-a-frente, conforme se pode ver na Figura 131.



**Figura 131** Célula robotizada de lixamento - “KUKA 4”

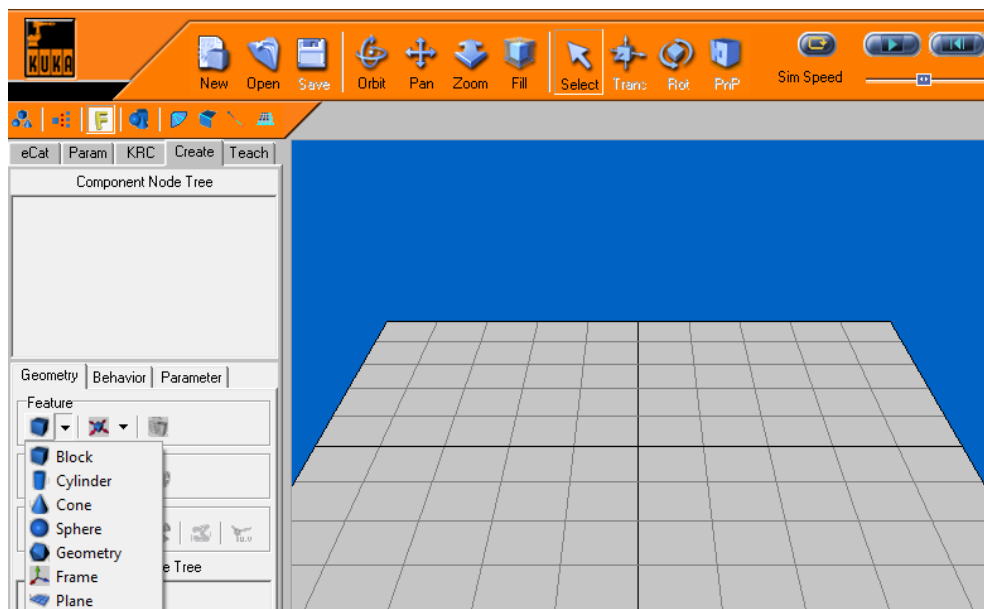
O robô desta célula de trabalho é um KUKA KR30/2 com o controlador KRC1, tratando-se de um robô para cargas médias com capacidade de carga máxima de 30 kg e um alcance máximo de 2,04 m.

## 6.1.2. MODELAÇÃO DA CÉLULA DE TRABALHO

Tal como já foi referido, antes da programação propriamente dita é necessário modelar a célula de trabalho, apresentando-se de seguida alguns passos que se destacam face à modelação já apresentada no Capítulo 4, mostrando-se também algumas funcionalidades do KUKA Sim.

### 6.1.2.1. Generalidades da aplicação KUKA Sim

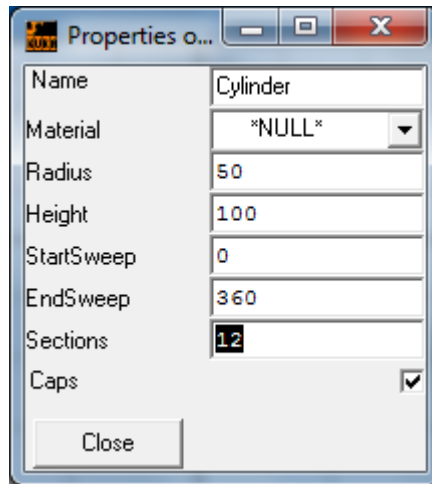
Para conceber o modelo da célula real, o KUKA Sim disponibiliza uma série de funcionalidades CAD que permitem desenvolver um novo *layout*. O modelo do robô e de outros componentes são disponibilizados *online*, podendo ser incluídos no *layout* através do catálogo eletrónico (eCat). Na Figura 132 são apresentadas algumas funcionalidades de criação de componentes (sólidos geométricos).



**Figura 132** Funcionalidades para criação de um componente no KUKA Sim

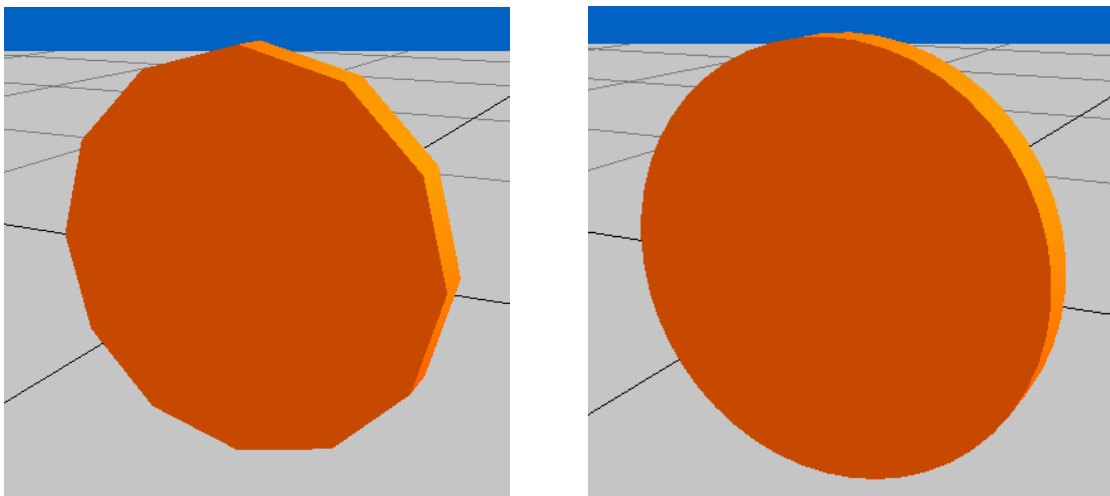
À modelação geométrica de cada componente pode ser associada a modelação cinemática e funcional, podendo utilizar-se diversos comportamentos predefinidos para, por exemplo, animar um tapete rolante (*conveyor*).

Este *software* permite também a alteração das propriedades do componente após a sua criação, sendo possível alterar as suas dimensões, as características do material e o número de secções que compõem o sólido, mostrando-se na Figura 133 as propriedades para um sólido cilíndrico.



**Figura 133** Propriedades do sólido cilíndrico

Ao definir o número de secções da representação 3D do sólido controla-se o nível de detalhe e realismo do modelo. A escolha correta do nível de detalhe para cada componente permite obter um bom desempenho de simulação, aliado a uma representação realista dos componentes que são essenciais ao processo, como é o caso da roda de lixamento que se apresenta na Figura 134.

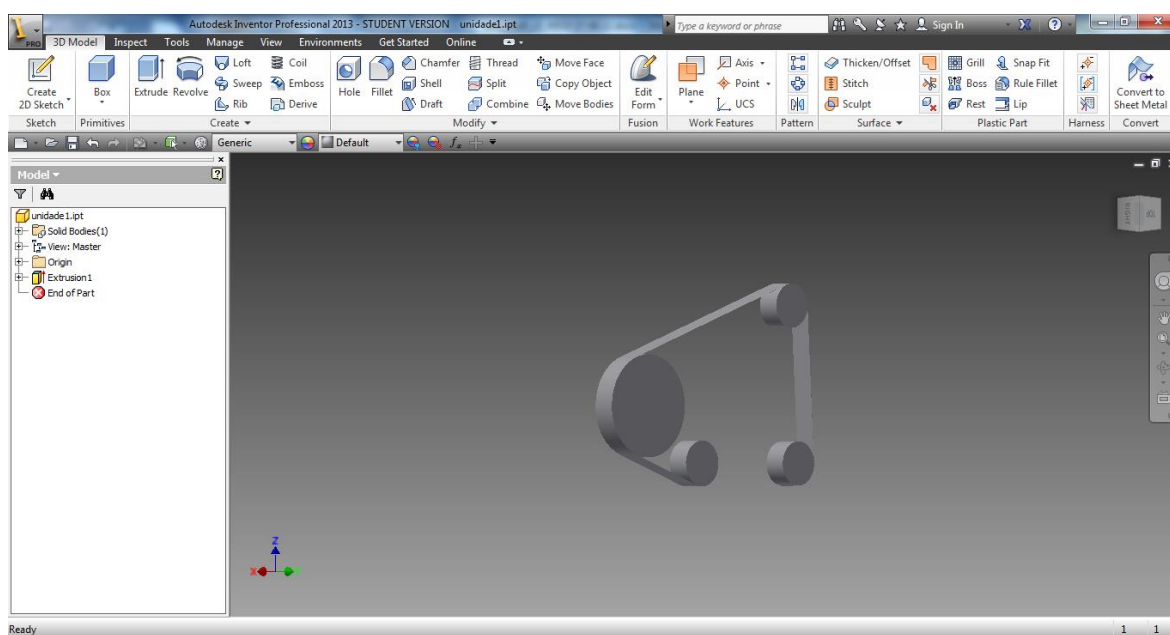


**Figura 134** Diferenças do modelo da roda – 12 secções (esquerda), 48 secções (direita)

O KUKA Sim permite também a importação de componentes desenvolvidos noutros sistemas CAD, não suportando, no entanto, alguns formatos frequentemente utilizados na indústria, como por exemplo o STEP.

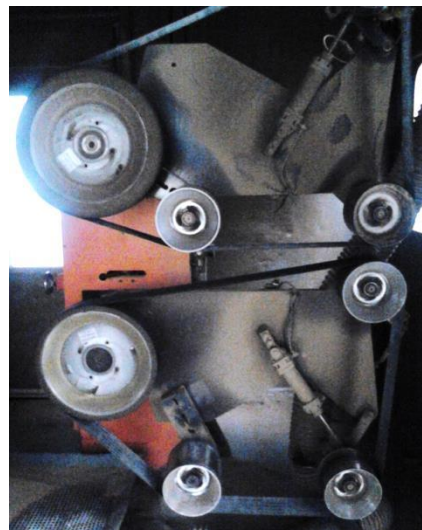
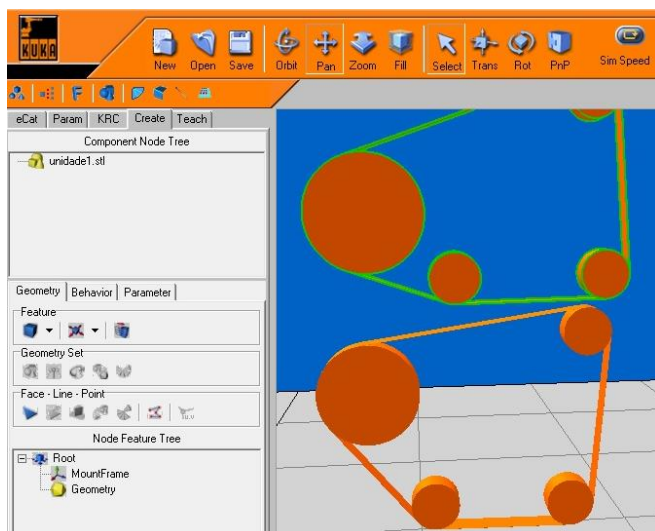
### 6.1.2.2. Modelação geométrica das unidades de trabalho

Para se efetuar a modelação das unidades de lixamento recorreu-se novamente ao Inventor, concebendo-se uma representação simplificada destas, composta pela cinta de lixa, roda de trabalho e rodas auxiliares que guiam e esticam a cinta de lixa. Nestas unidades é possível utilizar algumas zonas entre a roda de trabalho e as rodas auxiliares para fazer o lixamento em banda livre, sendo assim necessário modelar corretamente a disposição das rodas de cada unidade de trabalho. A título exemplificativo, apresenta-se na Figura 135 o modelo da unidade de lixamento n.º 1 no Inventor



**Figura 135** Modelo simplificado da unidade de lixamento n.º 1 no Inventor

Concebidas as unidades da célula de trabalho, procedeu-se à exportação para um formato CAD passível de ser interpretado pelo KUKA Sim, utilizando-se neste caso o formato StereoLithography (STL) da 3D Systems. Aquando da exportação no Inventor é possível definir a resolução que se pretende dar ao modelo, sendo necessário aumentar a precisão do mesmo, de forma a obter uma boa definição da superfície de contacto no KUKA Sim. Em seguida, os ficheiros em formato STL são importados para o *layout*, definindo-se a sua posição e tipo de material de forma a obter-se um modelo realista das unidades de trabalho, conforme se pode ver na Figura 136.

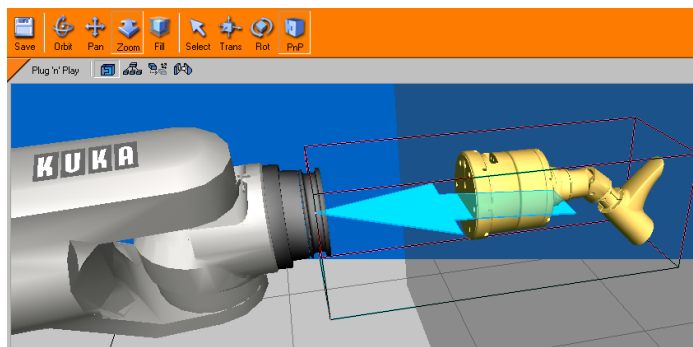


**Figura 136** Modelo das unidades 1 e 2 no KUKA Sim (à esquerda) e na célula real (à direita)

Após este passo é também modelado no KUKA Sim a base de suporte às unidades de trabalho, de forma a detetar na simulação uma possível colisão com estes elementos.

#### 6.1.2.3. Modelação geométrica e funcional da ferramenta do robô

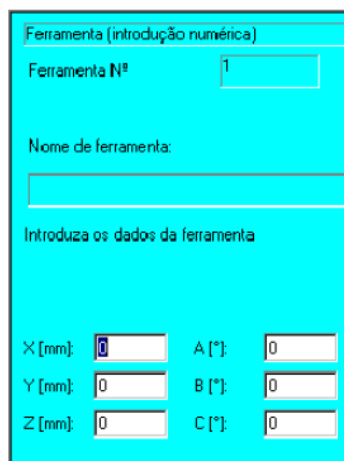
Concluída a modelação das unidades de trabalho passa-se à modelação da ferramenta do robô, importando-se para o efeito o punho com cunha a 45° já utilizado anteriormente. Para este robô, o sistema de acoplamento deste sistema não causa qualquer limitação de movimento dos eixos do robô, não sendo por isso necessário efetuar a sua modelação. Após a importação do ficheiro em formato STL, é necessário estabelecer a ligação do componente ao robô utilizando-se para isso o comando “*Plug and Play*”, seleccionando-se o centro da flange do robô como “*parent node*”, conforme se mostra na Figura 137.



**Figura 137** Acoplamento do punho ao robô

#### 6.1.2.4. Referências do programa

Acoplada a ferramenta ao robô é necessário definir a posição e orientação do ponto de referência da ferramenta no KUKA Sim e também no robô, apresentando-se na Figura 138 a definição numérica da ferramenta na consola do robô.



Ferramenta (introdução numérica)

Ferramenta N°

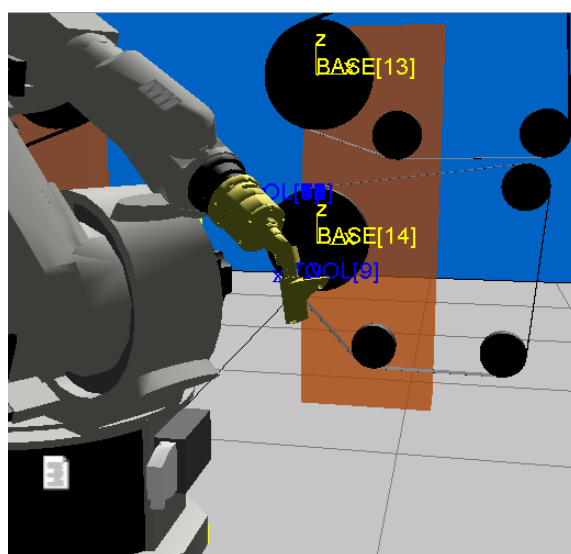
Nome de ferramenta:

Introduza os dados da ferramenta

X [mm]:	<input type="text" value="0"/>	A [°]:	<input type="text" value="0"/>
Y [mm]:	<input type="text" value="0"/>	B [°]:	<input type="text" value="0"/>
Z [mm]:	<input type="text" value="0"/>	C [°]:	<input type="text" value="0"/>

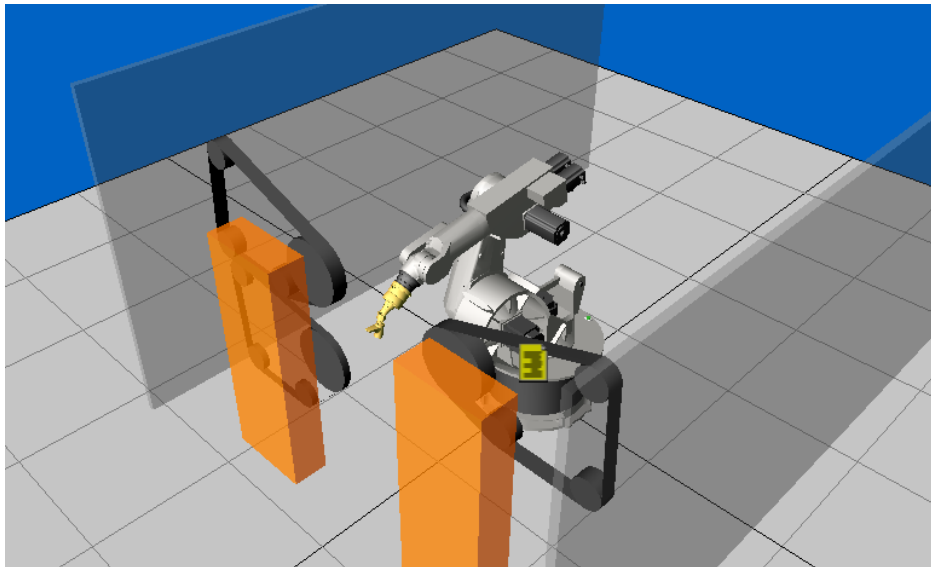
**Figura 138** Introdução numérica dos dados da ferramenta no robô [22]

Nesta versão do controlador podem ser armazenadas no máximo 16 ferramentas e 16 bases (ponto de referência da peça ou *workobject*) para todos os programas do robô. Assim, deve realizar-se o estudo do TCP para que possa ser reutilizado noutros programas. Na Figura 139 pode-se observar o posicionamento do TCP definido - *Tool[9]* – e das bases ao centro da roda de trabalho das unidades de lixamento.



**Figura 139** Localização das referências das unidades (base) e da peça (*tool*)

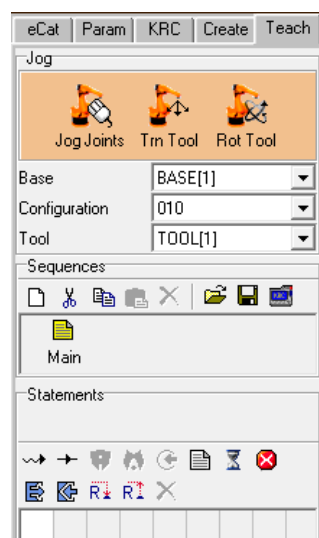
Conclui-se desta forma a modelação da célula de lixamento “KUKA 4”, apresentando-se o modelo final na Figura 140.



**Figura 140** Modelo final da célula de lixamento “KUKA 4”

### 6.1.3. DESENVOLVIMENTO DO PROGRAMA

O desenvolvimento do programa pode ser realizado no KUKA Sim recorrendo à aba “*teach*” (ver Figura 141). Nesta aba encontram-se os comandos de movimentação do robô, as sequências de trabalho e as instruções da sequência seleccionada.

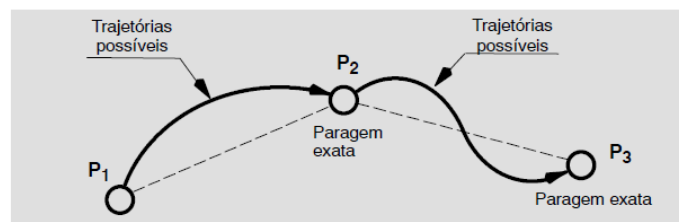


**Figura 141** Conceção do programa na aba “*teach*” do KUKA Sim.

De forma a programar os *targets* graficamente é necessário movimentar o robô até à posição pretendida, gravando-se em seguida as coordenadas do ponto. Para tal, é possível movimentar a ferramenta do robô (movimento de translação ou rotação) ou movimentar as juntas do robô. É também possível fazer a translação e rotação da ferramenta por incrementos, tal como no robô. Ao contrário do RobotStudio, nesta aplicação só é possível memorizar o *target* numa instrução de movimento, estando assim sempre associados. Por isso, para o desenvolvimento do programa, são adicionados os movimentos pretendidos diretamente na rotina ativa (equivalente ao comando “*teach instruction*” no RobotStudio).

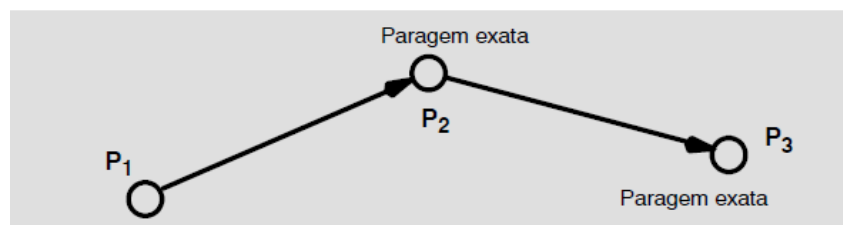
No KUKA Sim os dois tipos de movimento que podem ser programados são:

- Movimento *Point-to-Point* (PTP) ou interpolação das juntas: o movimento é descrito através da evolução angular das juntas, por isso, o trajeto do robô não é previsível com precisão. Nos movimentos PTP com paragem exata é feita a deslocação ao ponto de destino, com paragem na posição exata deste, conforme se mostra na Figura 142.



**Figura 142** Movimento PTP com paragem exata [23]

- Movimento Linear (LIN) ou interpolação linear: o ponto de referência da ferramenta é movimentado em direção ao ponto de destino numa trajetória linear. É utilizado quando é necessário obter um trajeto linear entre dois pontos.



**Figura 143** Movimento LIN com paragem exata [23]

Durante a programação dos movimentos é possível escolher a configuração pretendida do robô para alcançar o ponto desejado, conforme o exemplo apresentado na Figura 144,

sendo esta opção usada apenas para movimentos do tipo PTP, pois para os movimentos lineares é automaticamente escolhida a configuração mais próxima.

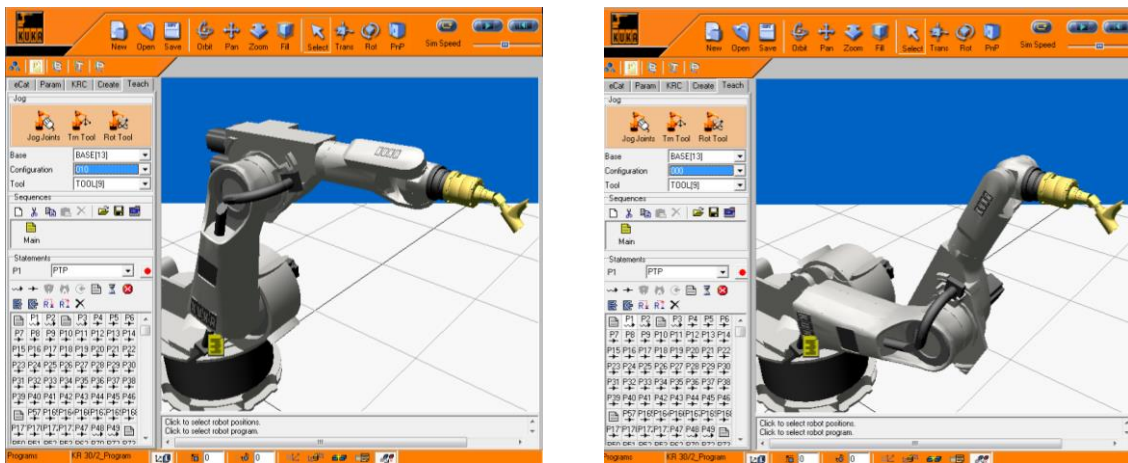


Figura 144 Configurações do robô para o mesmo *target*

Na sequência de trabalho podem também ser adicionadas outras instruções. Pode-se destacar a possibilidade de introdução de comentários no código, sendo estes apresentados no decorrer da simulação. Isto permite comentar o programa de acordo com as superfícies a lixar, facilitando o desenvolvimento e posterior afinação do programa. Na Figura 145 apresenta-se um exemplo, sendo apresentado o comentário “Início” na janela de *debug*.

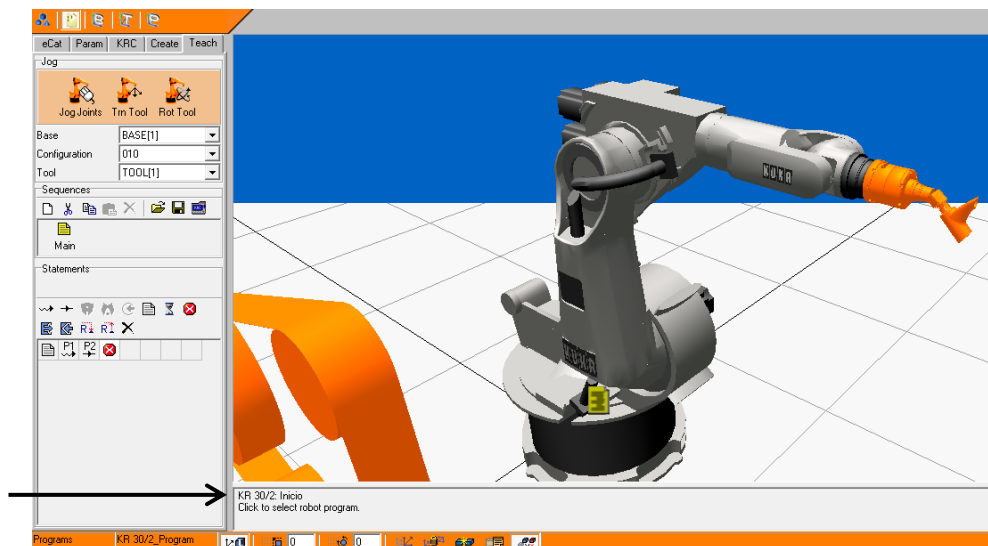


Figura 145 Exemplo de comentário apresentado na janela de *debug* durante a simulação

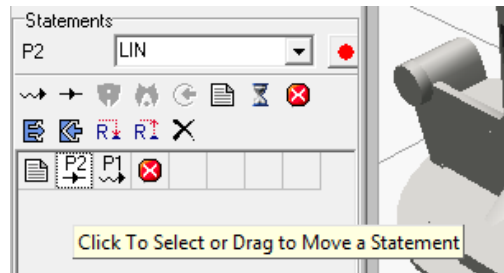
Fazendo a exportação do programa para KRL verifica-se que o comentário foi também incluído no código, conforme pretendido.

```

; Início
PTP P1 Vel= 100 % PDAT1 Tool[1] Base[1]
LIN P2 Vel= 2 m/s CPDAT1 Tool[1] Base[1]
;STOP

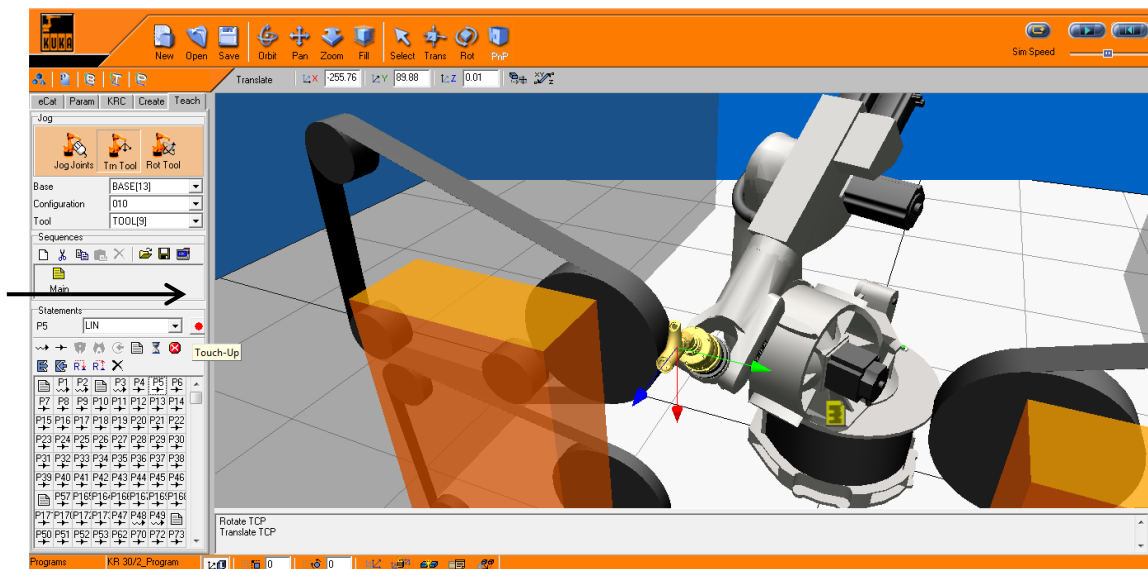
```

A sequência de instruções gerada pode também ser editada, sendo permitido adicionar, apagar ou reordenar as instruções (arrastando a instrução para a posição pretendida). A título exemplificativo, na Figura 146, mostra-se a sequência reordenada de forma a ser executado, em primeiro lugar, o movimento P2.



**Figura 146** Sequência reordenada

É possível alterar o ponto de destino programado depois de gerada a sequência, utilizando para isso o comando “*touch-up*” (visível na Figura 147), sendo transferidas as coordenadas do ponto onde se encontra o robô para a instrução de movimento na qual se encontra o cursor, tal como acontece no robô.

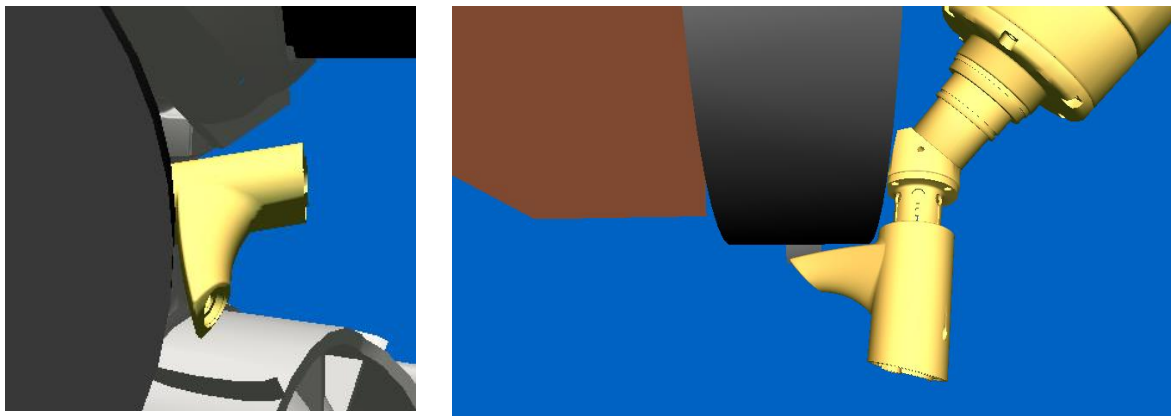


**Figura 147** Comando “*Touch-Up*”

Procedeu-se assim à afinação em modo *offline* de alguns movimentos, de forma a reduzir o tempo de paragem para ajustes na célula de trabalho, à semelhança do que foi feito no programa desenvolvido no RobotStudio.

#### 6.1.4. SIMULAÇÃO *OFFLINE* E RESULTADOS

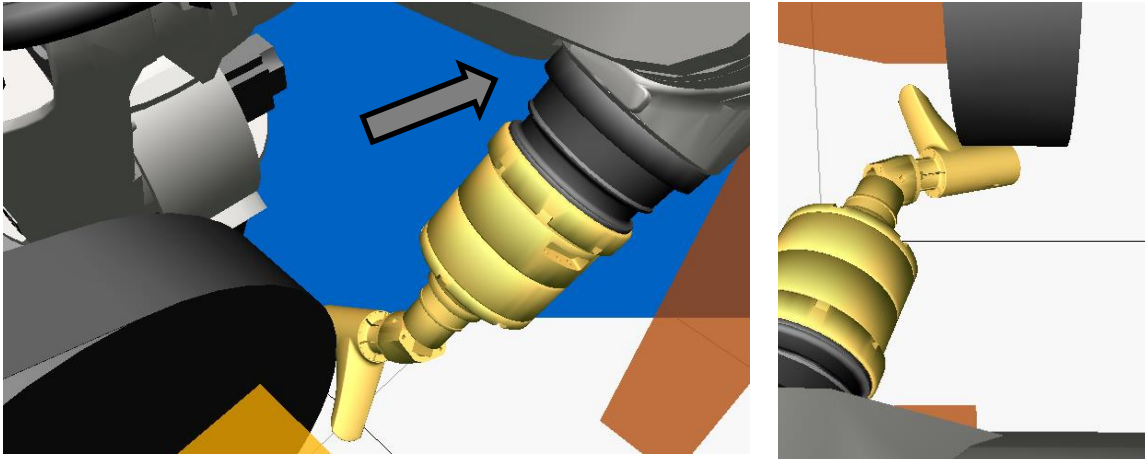
Nesta subsecção apresentam-se alguns procedimentos de lixamento que, de alguma forma, se distinguem dos desenvolvidos no Robotstudio para a célula “Mepsa 13”. Apesar de se tratar da mesma peça, o facto do *layout* da célula ser diferente leva a que tenha de se reformular, quer a abordagem de lixamento, quer os movimentos envolvidos. A disposição das unidades nesta célula facilita, por exemplo, o acesso à parte de cima da bica (como se mostra na Figura 148), permitindo uma maior liberdade de movimentos.



**Figura 148** Lixamento da parte de cima da bica

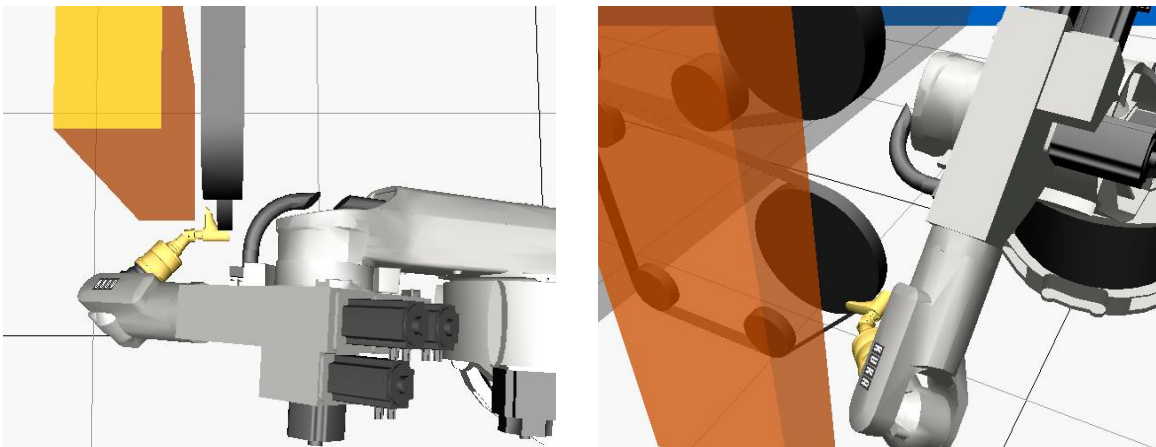
Na célula “Mepsa 13” não foi possível lixar esta superfície de uma só vez pois durante o movimento atingia-se o limite do eixo 5 do robô, tendo sido esta dificuldade ultrapassada repartindo a superfície a lixar em duas metades, fazendo o lixamento do lado esquerdo num lado da roda e o lixamento do lado direito no lado oposto da roda.

Por outro lado, a melhoria de processo de algumas rotinas de lixamento neste *layout* surge em detrimento de outras rotinas onde é necessário aceder ao lado da roda oposto ao robô, como por exemplo para o lixamento do contorno entre o corpo e a bica de um dos lados da peça, como se pode observar na Figura 149.



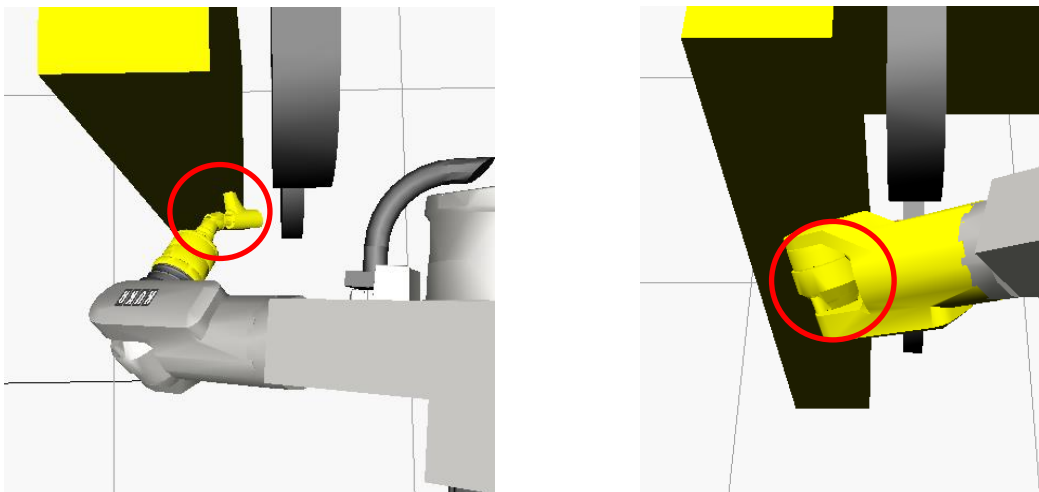
**Figura 149** Lixamento do contorno entre o corpo e a bica

Para o lixamento da parte de baixo da bica recorreu-se ao método de passagem por arrasto na roda de 50 mm de largura, instalada na unidade de lixamento número 2, podendo verificar-se na Figura 150 o acesso e posicionamento da peça no primeiro *target* de contacto com a lixa.



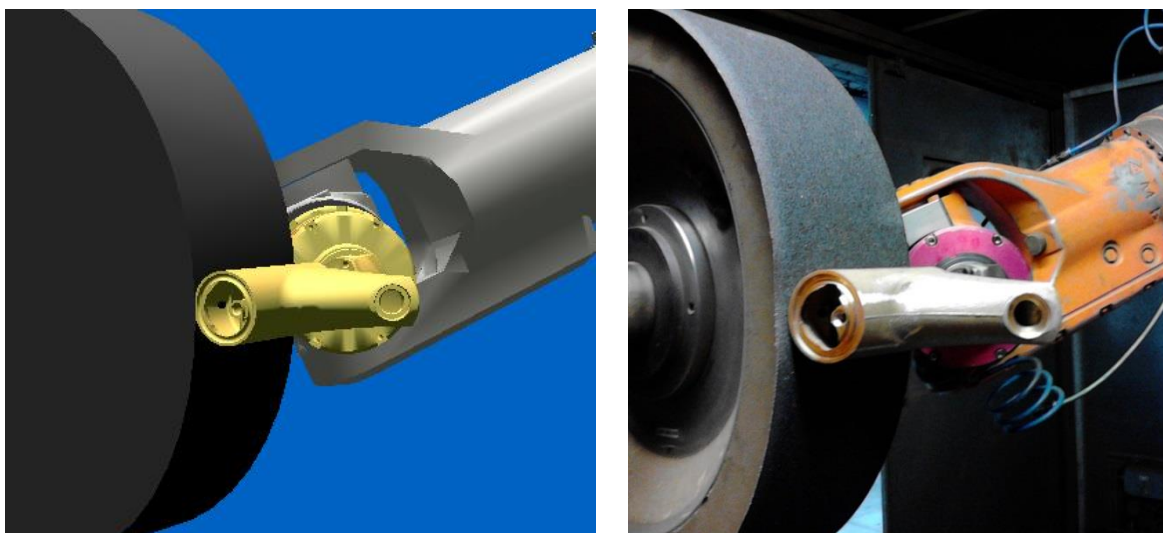
**Figura 150** Lixamento da parte de baixo da bica por arrasto

Como se pode ver na figura anterior, durante esta rotina existe o perigo de colisão com a base de suporte das unidades de lixamento, sendo por isso necessário detetar na simulação qualquer possibilidade de uma colisão na célula real. Para tal, utilizou-se a funcionalidade de deteção de colisões do KUKA Sim, seleccionando-se neste *software* apenas o componente a ser monitorizado, sendo assinalados os elementos intervenientes aquando da ocorrência de uma colisão. Na Figura 151 apresenta-se o resultado da simulação de uma colisão com a base de suporte das unidades de lixamento.



**Figura 151** Detecção de colisão com a peça à esquerda e com o robô à direita

Após a verificação do programa é possível exportar o código do programa em KRL, sendo este composto pelo ficheiro “.SRC”, que contém o código do programa, e pelo ficheiro “.DAT”, que contém os dados do programa. Apresenta-se na Figura 152, a título de exemplo, o resultado do posicionamento do robô num *target* da rotina de lixamento do corpo, no KUKA Sim e na célula real, confirmando-se a correta transposição do programa para a célula real.



**Figura 152** Exemplo de posicionamento do robô num *target* para o lixamento do corpo

Em conclusão, obteve-se uma reprodução fiel da simulação *offline*, sendo apenas necessário efetuar pequenos ajustes ao processo. Os resultados do lixamento não são aqui apresentados, visto produzirem um resultado final semelhante ao mostrado no Capítulo 4.

## 6.2. SIMULAÇÃO E PROGRAMAÇÃO *OFFLINE* DA TORNEIRA EURODISC COSMOPOLITAN

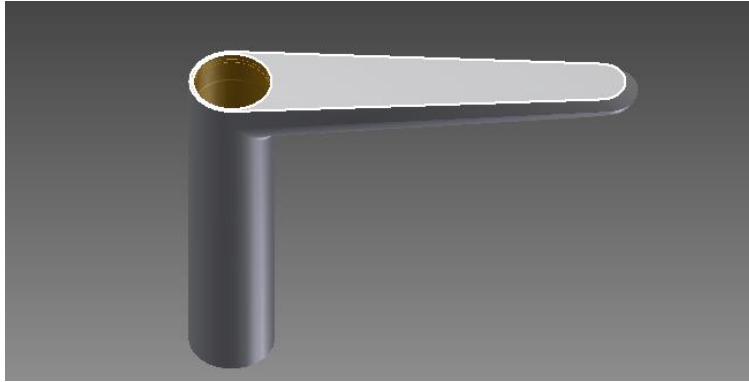
De forma a expandir e consolidar a utilização do método de simulação e programação *offline* no departamento, é apresentado nesta secção o processo de desenvolvimento de um novo ciclo de lixamento de uma torneira de bica longa, fazendo-se também o estudo de uma nova ferramenta necessária à correta execução deste processo. Assim, é apresentada na Figura 153 a torneira de cozinha da coleção Eurodisc Cosmopolitan, à qual se destina o desenvolvimento do processo de lixamento robotizado.



**Figura 153** Torneira da coleção Eurodisc Cosmopolitan (à esquerda) e modelo 3D da mesma (à direita)

Conforme se pode ver, trata-se de uma torneira caracterizada pela sua bica alongada e pelo corpo de grande dimensão. Esta torneira apresenta uma dificuldade acrescida para o lixamento manual, devido às dimensões da peça e à sua superfície extensa de trabalho. Existe assim a necessidade de automatizar o processo de lixamento desta torneira, de forma a garantir a consistência e qualidade do processo, contribuindo para a diminuição da taxa de rejeição do mesmo.

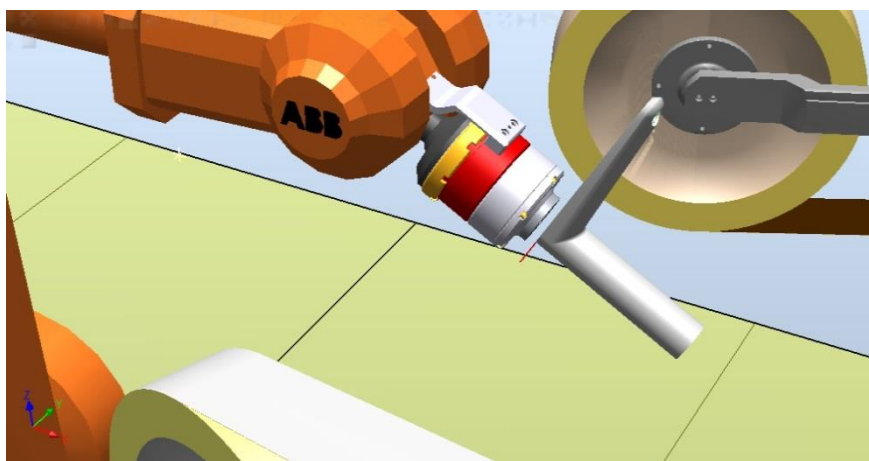
Assim, o processo a desenvolver deverá contemplar o lixamento de toda a torneira excluindo a face plana da parte de cima da bica (ver Figura 154), pois esta superfície é retificada no processo de maquinação.



**Figura 154** Face da torneira não contemplada no processo de lixamento robotizado

### **6.2.1. ESTUDO DE UMA NOVA FERRAMENTA RECORRENDO À SIMULAÇÃO**

Conhecida a peça a trabalhar, apresenta-se o estudo da ferramenta de fixação para o lixamento robotizado desta torneira. Recorreu-se à simulação de forma a realizar o estudo da melhor configuração do conjunto composto pelo punho e pinça para o produto em questão. A ferramenta a desenvolver deve permitir o acesso a todas as superfícies a lixar, facilitando a manipulação da peça a trabalhar, sendo por isso necessário testar alguns movimentos na simulação. Visto que a face plana da parte de cima da bica não será trabalhada pelo robô, a carga pode ser feita através de uma pinça expansível na zona de montagem do cartucho (mono comando) sendo possível, nesta posição, testar o acesso às restantes superfícies. Para isso importou-se a peça e o punho (sem pinça) para o modelo existente na simulação (ver Figura 155).

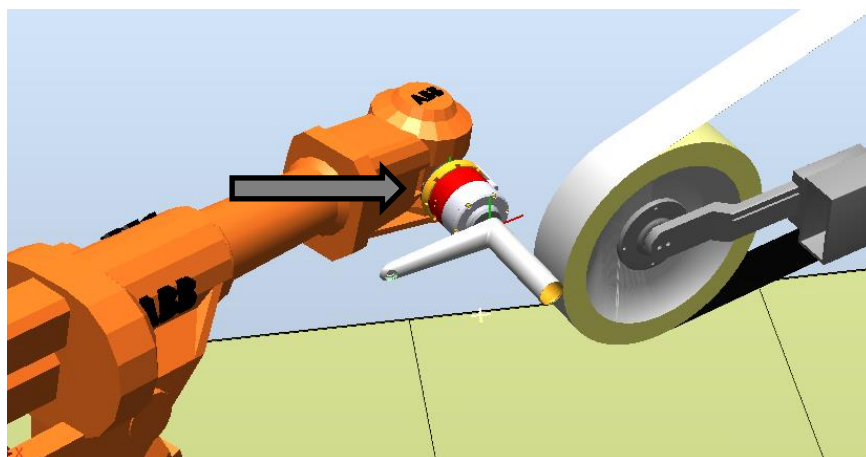


**Figura 155** Punho e peça a lixar no modelo de simulação

Foram então realizados alguns posicionamentos nas várias superfícies a lixar, verificando-se a facilidade de acesso a estas áreas e a não existência de colisões.

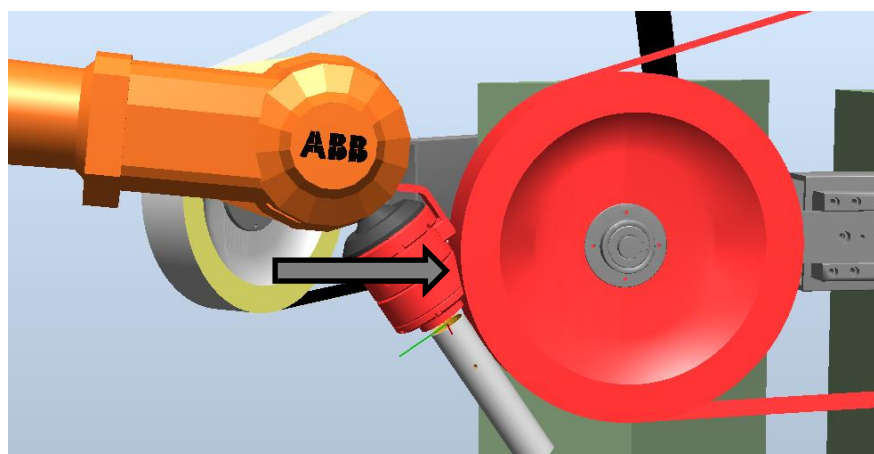
Pela análise do lixamento do corpo foi possível verificar que o comprimento da pinça deve ser escolhido de forma a:

- Assegurar que se pode fazer a rotação do corpo sem que se atinja o limite do eixo 5, aquando da primeira passagem com lixa grossa na unidade n.º 1 (ver Figura 156);



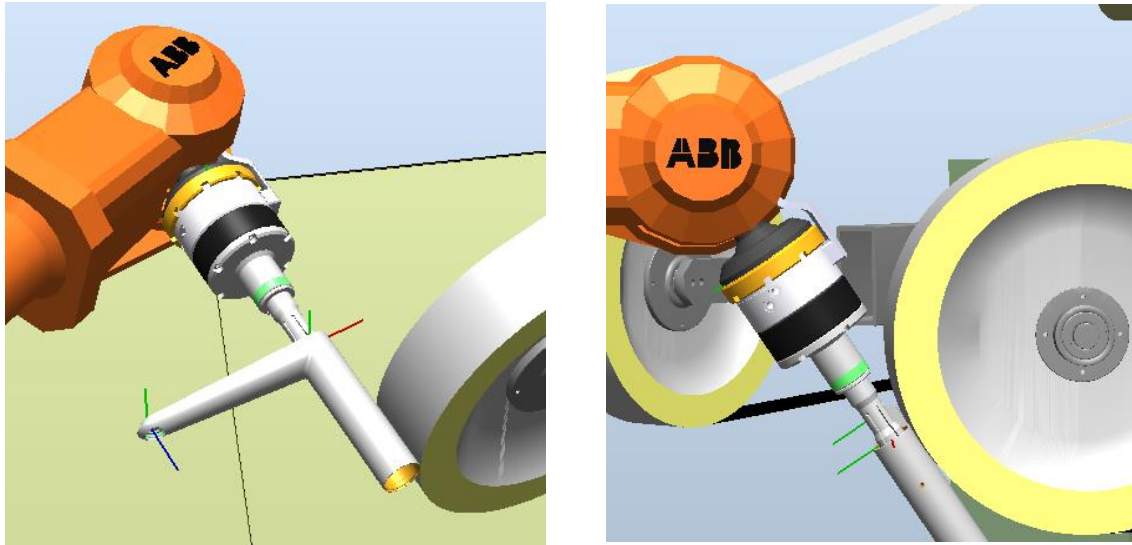
**Figura 156** Lixamento do corpo na unidade n.º 1 (lixa grossa)

- Assegurar que não ocorre nenhuma colisão entre o punho e a roda durante o lixamento do corpo na unidade n.º 4, aquando da segunda passagem cruzada com lixa fina (ver Figura 157).



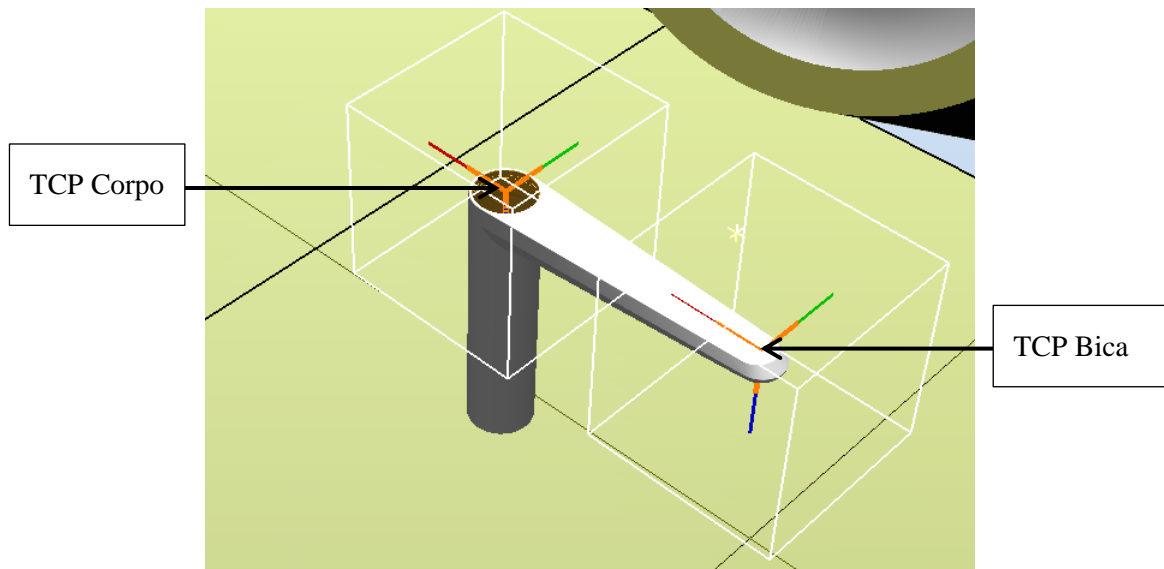
**Figura 157** Lixamento do corpo na unidade n.º 4 (lixa fina)

Assim, foi possível determinar o comprimento necessário da pinça para cumprir estas duas condições, melhorando-se a manipulação da peça e o acesso às superfícies a lixar, contribuindo assim para um nível de acabamento superior. Na Figura 158 é apresentado o resultado obtido a partir do estudo efetuado, para os dois *targets* vistos anteriormente (o desenho e medidas do conjunto encontram-se no Anexo B deste documento).



**Figura 158** Resultado obtido (posicionamento final da peça)

Após o desenvolvimento da ferramenta é necessário definir o TCP de forma a facilitar a programação das várias rotinas. Definiu-se um TCP ao centro do corpo, para auxiliar a rotação do próprio corpo da peça, e outro TCP no centro do raio de curvatura da ponta da bica, para auxiliar à definição deste contorno. Na Figura 159 apresenta-se a localização destes pontos de referência na peça, sendo a localização em detalhe apresentada no Anexo B deste documento.



**Figura 159** Pontos de referência definidos para a peça

Concluída a etapa da preparação para a programação *offline*, passa-se agora à descrição do seu desenvolvimento.

## 6.2.2. PROCEDIMENTOS E DESENVOLVIMENTO DO PROGRAMA PARA LIXAMENTO DA TORNEIRA

Tal como já foi referido anteriormente, antes de dar início à programação de um novo produto deve-se fazer um planeamento prévio do trabalho a executar, decompondo o processo global em operações de trabalho bem definidas, facilitando o desenvolvimento do programa e posteriores ajustes por parte dos afinadores.

Apresenta-se na Figura 160 uma ilustração com as principais rotinas de lixamento a desenvolver e a sequência do processo de lixamento. Esta decomposição do processo global resulta de uma avaliação e análise prévia da peça, com o objetivo de fazer um planeamento do trabalho de acordo com a geometria, o nível de acabamento pretendido e de forma a contornar algumas dificuldades do lixamento robotizado.



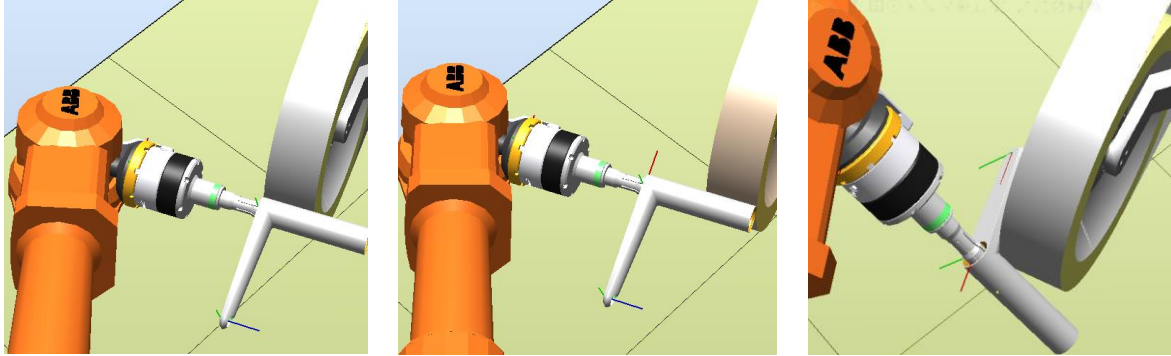
**Figura 160** Sequência do processo de lixamento da torneira Eurodisc Cosmopolitan

De seguida são apresentados os procedimentos desenvolvidos, organizados pelas superfícies a lixar.

### 6.2.2.1. Corpo da peça

Para o lixamento do corpo da peça, foi necessário desenvolver um método de lixamento que abrangesse todo o corpo, deixando a peça com um acabamento uniforme e sem marcas, sendo este de dificuldade acrescida devido à grande extensão da área a lixar. Caso fosse feita apenas a rotação em torno da base e do topo do corpo na roda de lixamento, o acabamento ficaria irregular, com vincos nas zonas de união das passagens, conduzindo à rejeição da peça. Assim, foi necessário abranger todo o corpo, conjugando passagens longitudinais com a rotação do corpo. Este método provoca o aparecimento de faces na

superfície, sendo estas facilmente removidas na segunda passagem do processo. Na Figura 161 é mostrado o robô posicionado em três *targets* do processo de lixamento do corpo, sendo visível a passagem do corpo na roda de extremo a extremo e, por fim, na figura da direita, o acabamento da união com a bica.



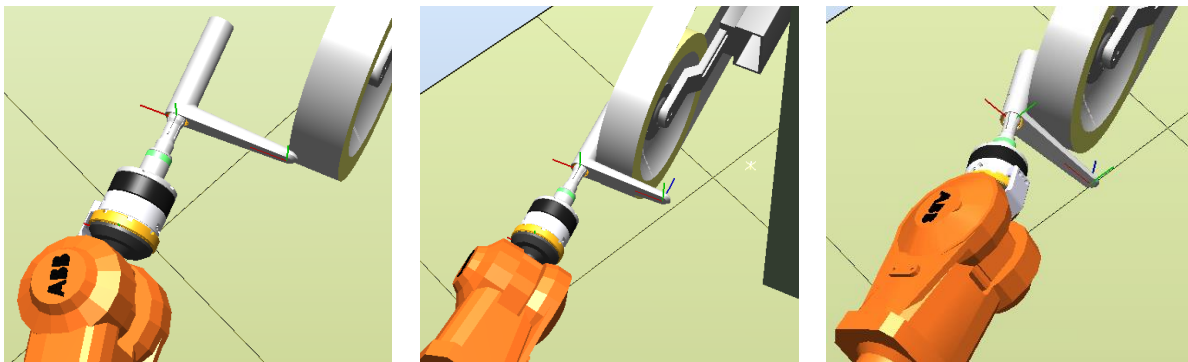
**Figura 161** Rotina de lixamento do corpo (lixa grossa)

O tempo de ciclo desta rotina (constituída por 90 *targets*) determinado no RobotStudio é de cerca de 55 segundos, sendo esta demorada devido às grandes dimensões da peça e ao número de passagens necessárias para garantir um bom acabamento e a taxa de remoção de material necessária.

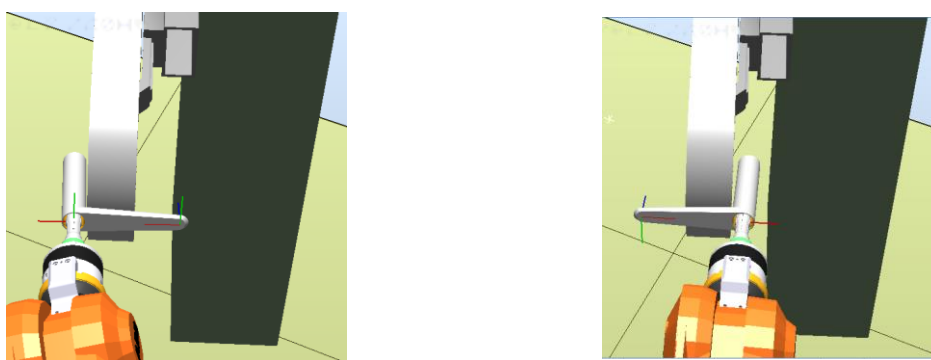
#### 6.2.2.2. Parte de baixo da bica

A parte de baixo da bica da peça é uma das superfícies com contorno mais delicado, devido à curvatura ao longo de toda a extensão da bica, sendo necessário um desbaste uniforme ao longo da bica, sem causar alterações na geometria da peça. Assim, o lixamento da parte de baixo da bica deve garantir o acabamento homogêneo e simétrico da superfície da mesma. Na Figura 162 são mostrados três instantes do programa, com uma das passagens da parte de baixo da bica e definição do contorno.

De forma a garantir a simetria da peça, e simultaneamente acelerar o desenvolvimento do processo, foi concebida apenas metade da rotina (metade da parte de baixo da bica), sendo a mesma posteriormente afinada no robô. Recorrendo novamente ao RobotStudio, procedeu-se ao espelhamento do trajeto previamente afinado segundo o plano do centro da roda, obtendo-se desta forma as duas partes que compõem a rotina completa, conforme se mostra na Figura 163.



**Figura 162** Rotina de lixamento da parte de baixo da bica



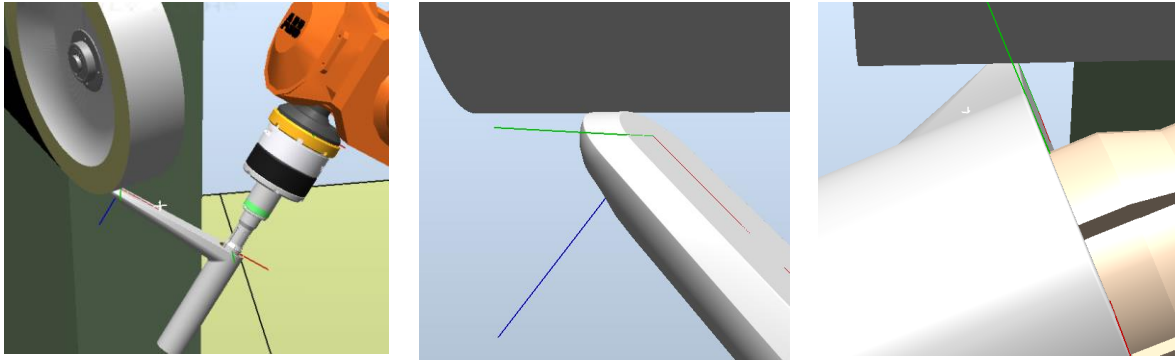
**Figura 163** Espelhamento da rotina de lixamento da parte de baixo da bica

Desta forma é possível reduzir praticamente para metade o tempo de afinação convencional na célula de trabalho.

#### 6.2.2.3. Definição do contorno da bica

Uma das dificuldades do lixamento desta peça é a definição do contorno da bica, sendo necessário manter a linha do contorno ao longo de toda a sua extensão. Na extremidade da bica o contorno é conseguido através da rotação da peça segundo o TCP aí definido (TCP bica) sendo, em seguida, realizada a linha de contorno lateral através da passagem para o TCP definido ao centro do corpo da peça (ver Figura 164).

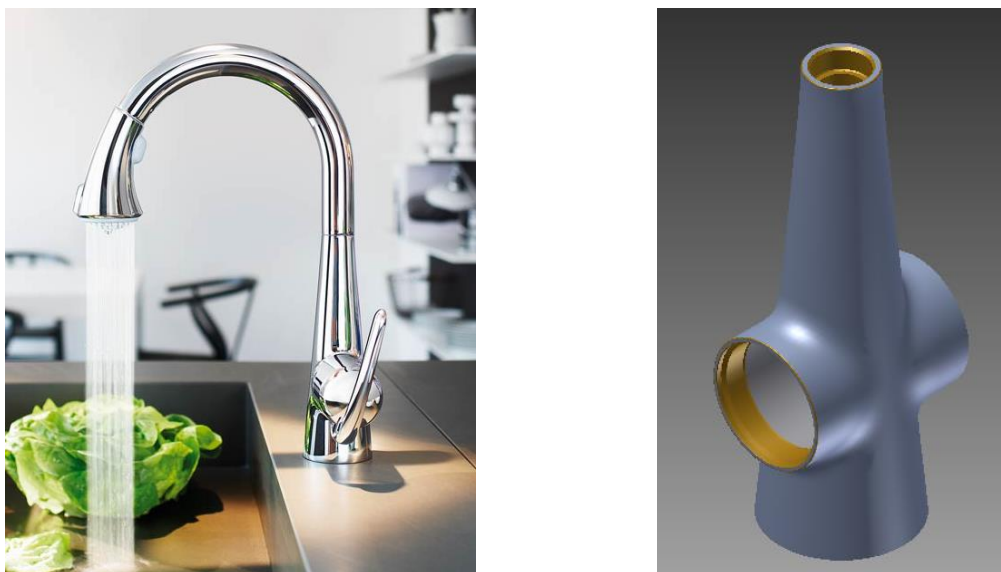
Finaliza-se aqui a apresentação da programação *offline* deste ciclo de lixamento, não sendo possível apresentar os resultados práticos da sua utilização pois, até a data de escrita deste documento, ainda não tinha sido entregue a pinça cujo comprimento foi estudado nesta secção.



**Figura 164** Rotina de acabamento do contorno

### **6.3. MODELAÇÃO DE NOVA CÉLULA PARA LIXAMENTO DE TORNEIRA ZEDRA**

De forma a finalizar o trabalho de expansão do método de programação *offline* e o teste da sua aplicabilidade, apresenta-se por fim a modelação de uma nova célula de trabalho seguida da simulação e programação *offline* de um novo produto – a torneira de cozinha da coleção Zedra, apresentada na Figura 165.



**Figura 165** Torneira de cozinha da coleção Zedra – Grohe (esquerda) e modelo 3D da mesma (direita)

A descrição deste processo será feita de forma dar a conhecer os principais pontos de interesse face às modelações anteriormente apresentadas.

### 6.3.1. CÉLULA DE TRABALHO

A modelação a desenvolver irá permitir abranger um total de 12 células de trabalho, todas com o mesmo *layout*, fabricadas pela Mepsa. Estas células foram as primeiras a ser adquiridas pelo departamento, tendo assim um vasto leque de produtos já programados para as mesmas. Apesar de estas 12 células terem um *layout* igual, na prática existem diferenças que inviabilizam a portabilidade dos programas, sendo necessário um ajuste profundo de todo o programa, chegando a ser necessário um turno completo (8 h) para concluir a transferência de um produto entre células. Estas células de fabrico contêm também quatro unidades de lixamento, equipadas com rodas de 450 mm de diâmetro nas extremidades e 150 mm nas unidades centrais, na configuração mais comum, conforme se apresenta na Figura 166.



**Figura 166** Célula robotizada de lixamento – “Mepsa 12”

As unidades de lixamento desta célula têm duas posições, uma à frente quando é ativada e uma recuada de repouso, sendo ativada unidade por unidade aquando do ciclo de lixamento, deslocando-se esta para a frente, ficando as restantes recuadas para facilitar o acesso à unidade em utilização.

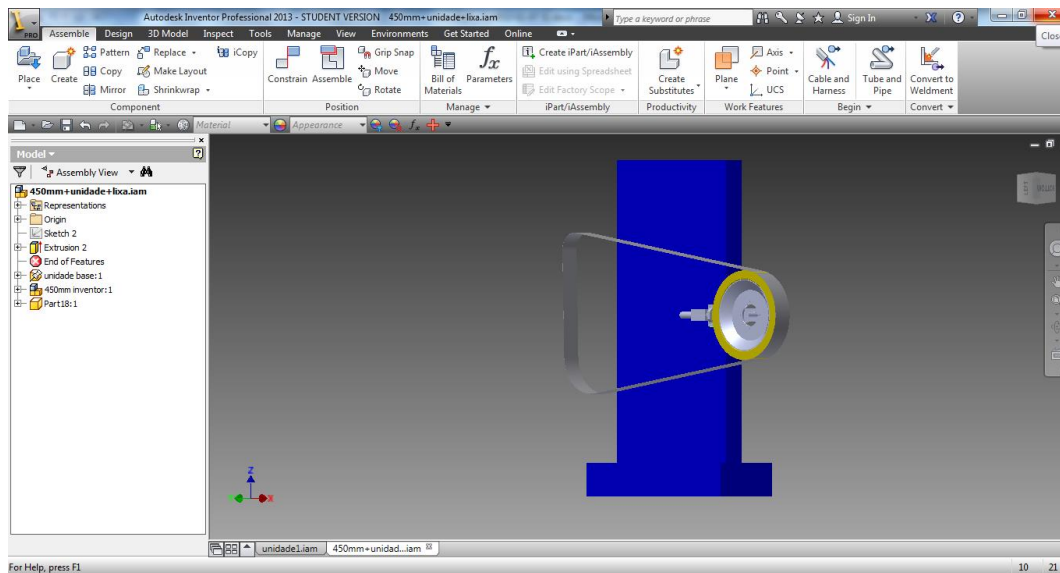
### 6.3.2. MODELAÇÃO DA CÉLULA

Passa-se agora à conceção do modelo da célula de trabalho, diferindo esta modelação das restantes, quer devido ao *layout* da célula, quer devido ao comportamento das unidades de

trabalho, sendo assim necessário conceber o modelo geométrico, cinemático e funcional destas unidades.

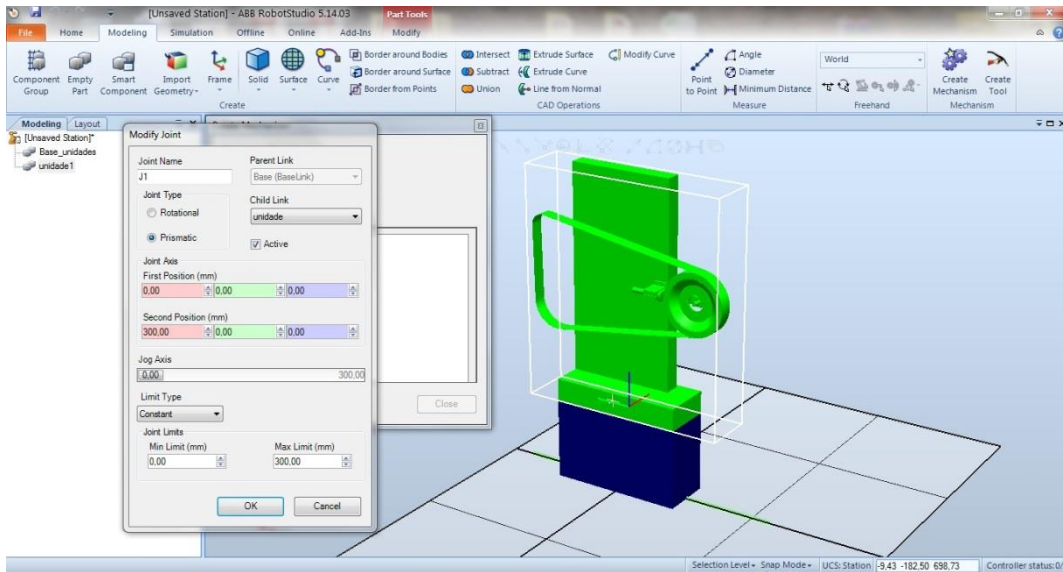
### 6.3.2.1. Modelação das unidades de trabalho

A modelação das unidades de trabalho foi feita dividindo a estrutura em dois elementos, a parte inferior (parte fixa da unidade, de geometria simples) e a parte superior (parte móvel da unidade), que contém a roda de trabalho e a cinta de lixa, sendo necessário nestas peças um maior detalhe e realismo. Por isso, decidiu-se modelar a parte superior da unidade no Inventor, sendo concebido um desenho da roda, braço de suporte desta, cinta de lixa e placa base de suporte. Após o desenho de cada um destes componentes montou-se o conjunto, conforme se mostra na Figura 167.



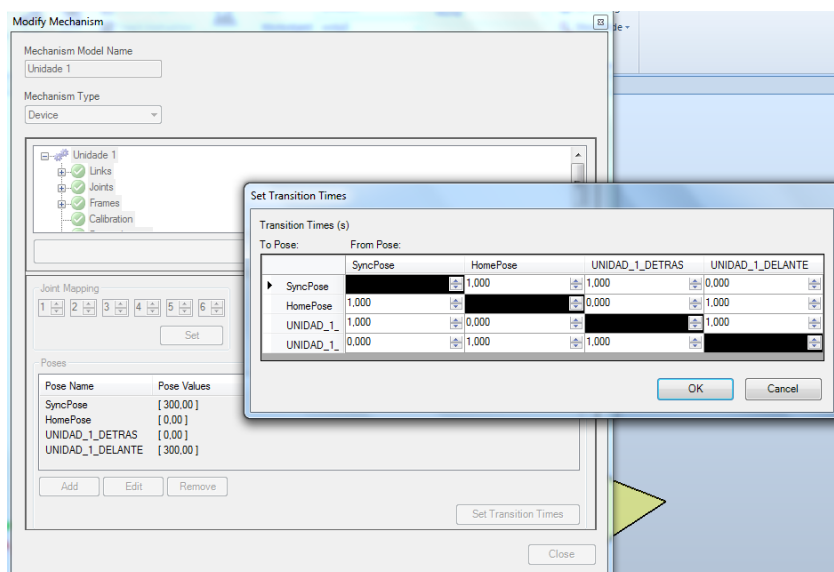
**Figura 167** Parte superior da unidade de trabalho n.º 1

Concluída a modelação da parte superior da unidade no Inventor, passou-se à modelação da parte inferior da unidade diretamente no RobotStudio, configurando-se no final o mecanismo da unidade de lixamento, constituído pelas duas partes supracitadas. Desta forma, obtém-se o mesmo comportamento da célula real no ambiente virtual recriado, tornando a programação e simulação mais realistas, tirando o máximo partido destas células de trabalho. Na Figura 168 é mostrada a configuração da junta prismática do mecanismo, sendo definido um curso total de 300 mm.



**Figura 168** Configuração do mecanismo da unidade de trabalho – junta prismática

Por fim foram criadas as duas posições de trabalho das unidades, uma posição de repouso e uma posição à frente (avanço de 300 mm), sendo também definidos os tempos de transição entre estas posições (ver Figura 169).



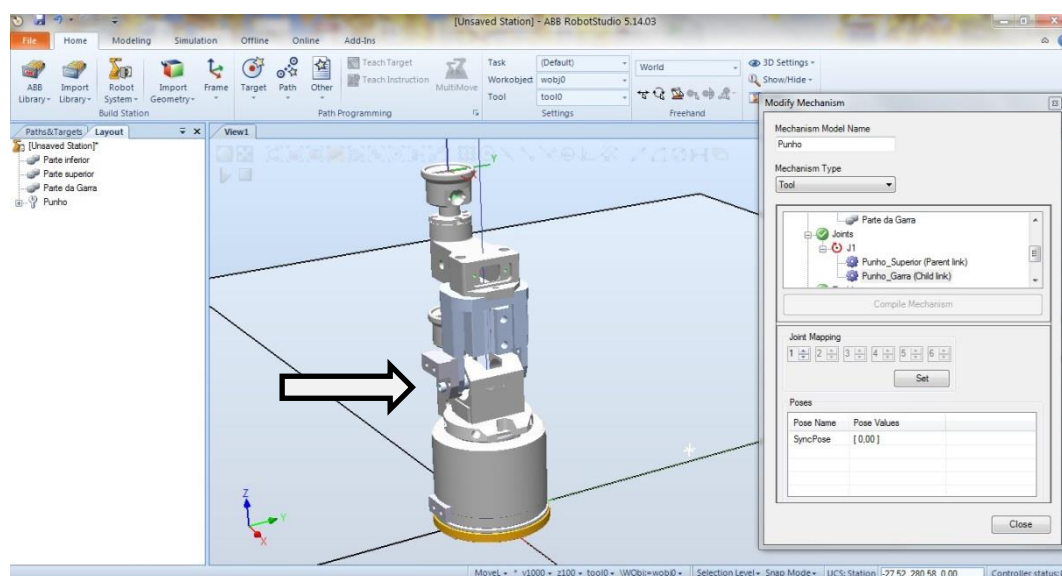
**Figura 169** Definição das posições e tempos de transição das unidades de lixamento

O mecanismo pode então ser gravado como biblioteca para posterior utilização na conceção do modelo final da célula.

### 6.3.2.2. Modelação geométrica e funcional da ferramenta do robô

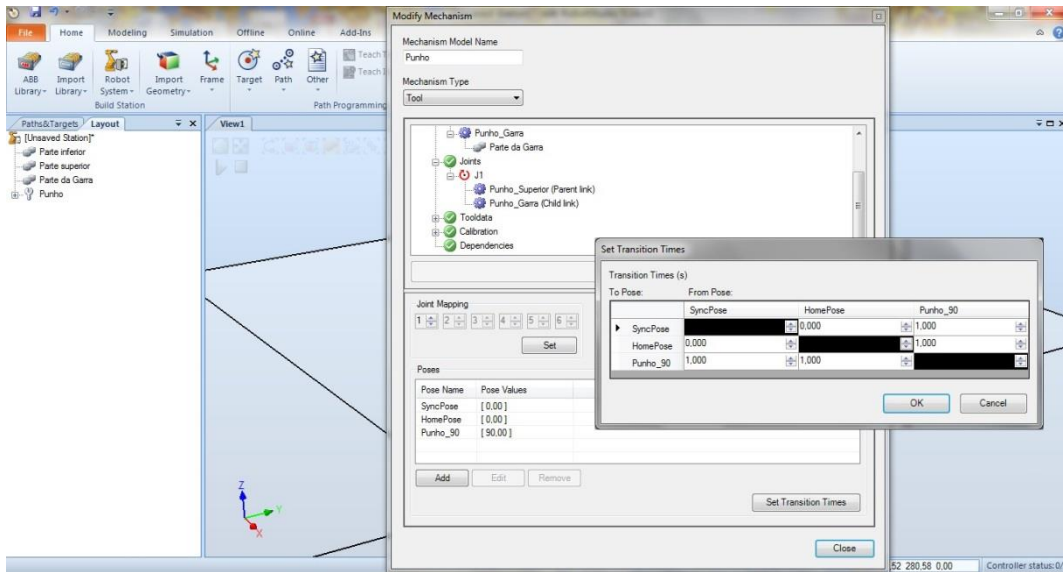
Analisando a torneira a lixar, verifica-se que existe a necessidade de manipular a peça, de forma a permitir o acesso a toda a extensão da superfície do corpo e da superfície do cartucho. Assim, foi necessário utilizar um punho mais complexo, possuindo este uma junta rotativa que permite reorientar a peça face à unidade de lixamento, aumentando desta forma a capacidade de manipulação do robô.

Para a modelação da ferramenta na célula virtual, é necessário definir um novo mecanismo, interligando o modelo geométrico ao modelo cinemático e funcional da ferramenta. O mecanismo a desenvolver deve conter as duas partes do punho, uma parte fixa à flange do robô (*baselink*) e o elo final, ligadas entre si por uma junta rotacional. Na Figura 170 é apresentado o processo de definição do mecanismo do punho encontrando-se a junta rotacional já definida com uma rotação máxima de 90°.



**Figura 170** Configuração do mecanismo do punho – junta rotacional

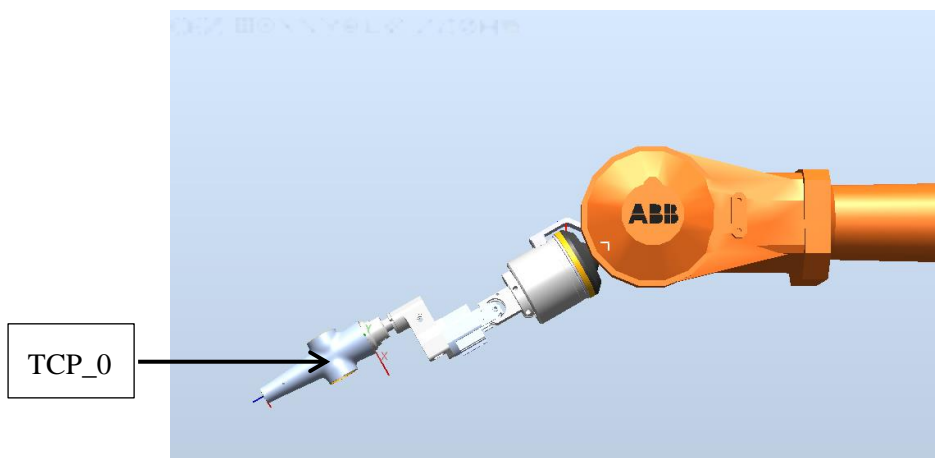
Concluída a configuração da junta é feita a definição das duas posições do mecanismo: uma posição direta com a junta a 0° e uma posição a 90°, definindo-se ainda os respetivos tempos de transição associados (ver Figura 171).



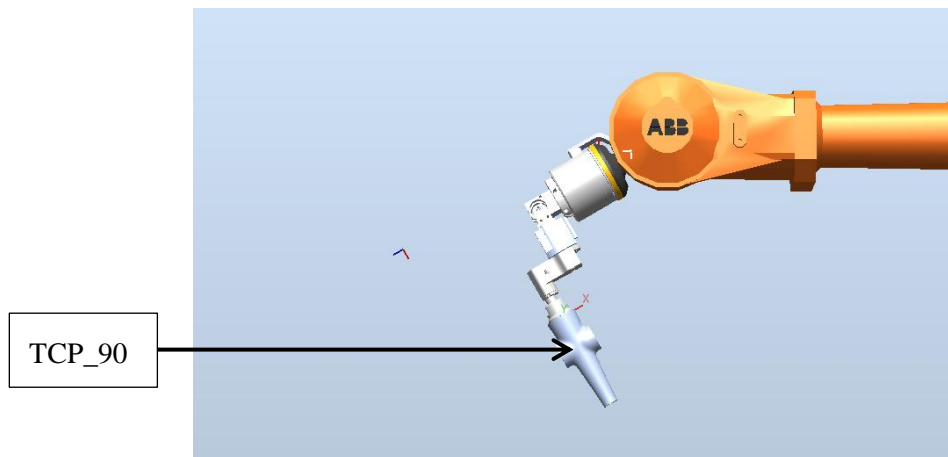
**Figura 171** Definição das posições e tempos de transição do mecanismo do punho

Na célula de trabalho, a comutação entre as posições é despoletada quando necessário durante o programa, através da ativação de uma saída do controlador, sendo necessário criar um evento que realize a troca de posições deste mecanismo de forma automática durante a simulação do programa.

De forma a facilitar a programação e movimentação do robô, são definidos dois pontos de referência da ferramenta para as duas posições do mecanismo do punho, ambas ao centro do corpo e ao centro do cartucho para cada posição (junta a 0° e junta a 90°). Apresenta-se na Figura 172 a posição do TCP para a posição do punho com junta a 0° e na Figura 173 a posição do TCP para a posição do punho a 90°.

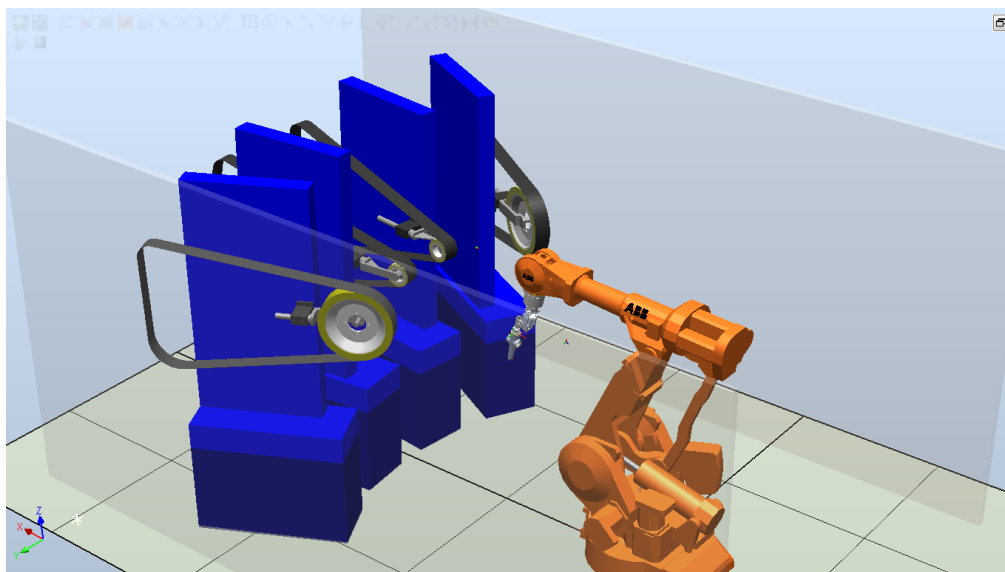


**Figura 172** Localização do TCP – junta a 0°



**Figura 173** Localização do TCP – junta a 90°

Conclui-se assim a apresentação da modelação dos mecanismos que compõem o modelo final desta célula de trabalho. O posicionamento da unidade de trabalho é feito de acordo com a localização do respetivo *workobject*, sendo assim assegurado o posicionamento da roda e da cinta de lixa. O resultado final desta modelação é apresentado na Figura 174.



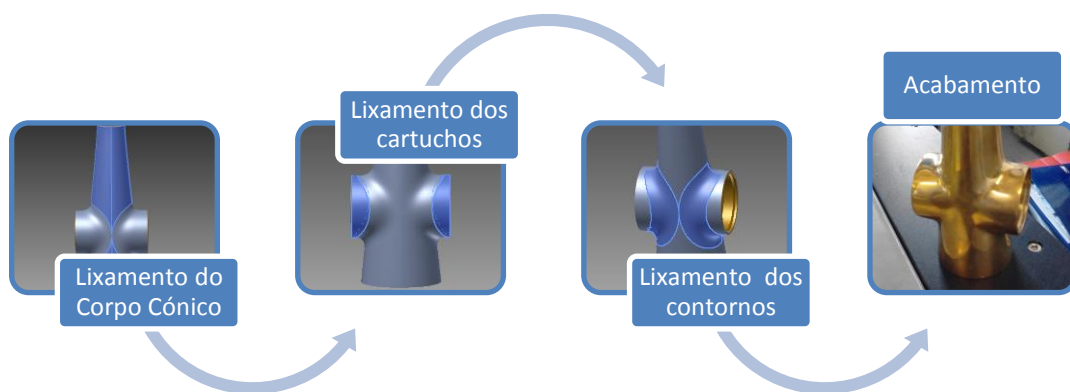
**Figura 174** Modelo final da célula de lixamento “Mepsa 12”

### 6.3.3. PROCEDIMENTOS E DESENVOLVIMENTO DO PROGRAMA

Em contraste com os procedimentos até agora desenvolvidos, o facto de se utilizar uma ferramenta mais complexa de duas posições oferece mais facilidades no manuseamento devendo, no entanto, organizar-se a sequência para que as trocas sejam reduzias ao

mínimo, pois a cada transição perdem-se em média quatro segundos de tempo de ciclo (incluindo os tempos de posicionamento, verificação dos sensores e transição de estado).

Tal como nos programas anteriormente apresentados, foi feita uma análise inicial da peça, decompondo-se o processo global de lixamento em sub-rotinas bem definidas, sendo assim identificadas as principais superfícies a lixar. Destas, salienta-se o lixamento da superfície do corpo, o lixamento da zona dos cartuchos e o lixamento dos contornos dos cartuchos (ver Figura 175).

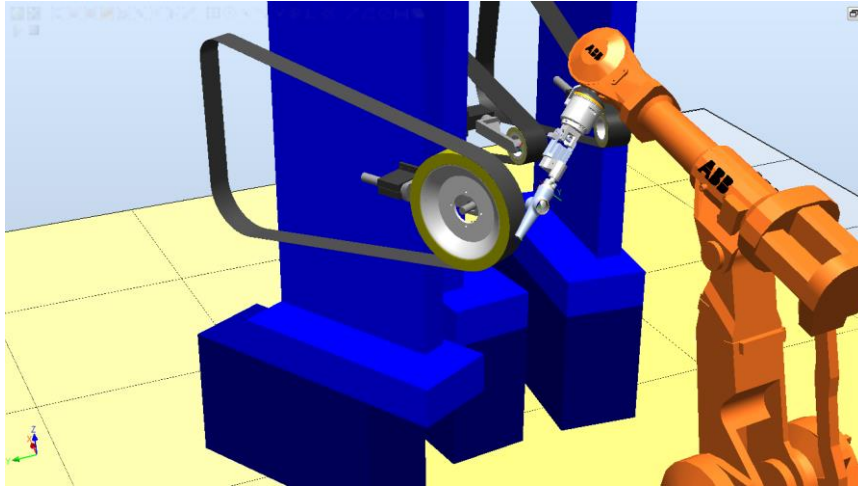


**Figura 175** Sequência do processo de lixamento da torneira Zedra – Grohe

De seguida são apresentados os procedimentos de lixamento do corpo e do cartucho, sendo apresentados alguns dos *targets* a partir dos quais se pode verificar a importância da modelação funcional do punho e das unidades.

#### 6.3.3.1. Corpo da peça

No lixamento desta superfície deve-se garantir a manipulação da peça de forma a fazer a rotação completa sobre o seu centro, sendo para isso utilizada a posição do punho alinhado com o eixo 6. Assim, utilizando o TCP definido para esta posição, é possível fazer a rotação do corpo na roda de trabalho, conforme se mostra na Figura 176.

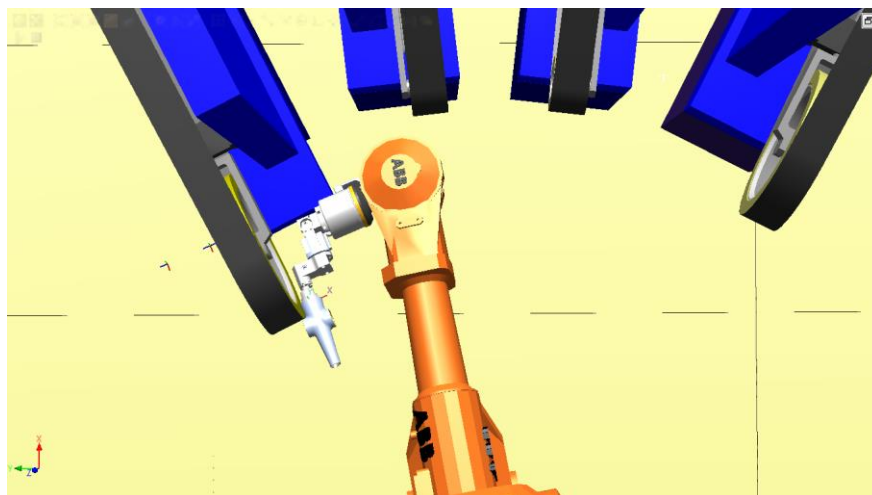


**Figura 176** *Target da rotina de lixamento do corpo (junta do punho a 0°)*

Devido à geometria da peça o procedimento foi repartido em duas metades, sendo feito o lixamento da parte superior do corpo (zona acima do cartucho) e, em seguida, a parte inferior do corpo (zona abaixo do cartucho), sendo a emenda das passagens disfarçada numa segunda etapa do processo e nas rotinas de lixamento do contorno entre o cartucho e o corpo. A passagem na lixa fina também será realizada de forma a cruzar o sentido de passagem da lixa grossa, eliminando os sulcos deixados por esta.

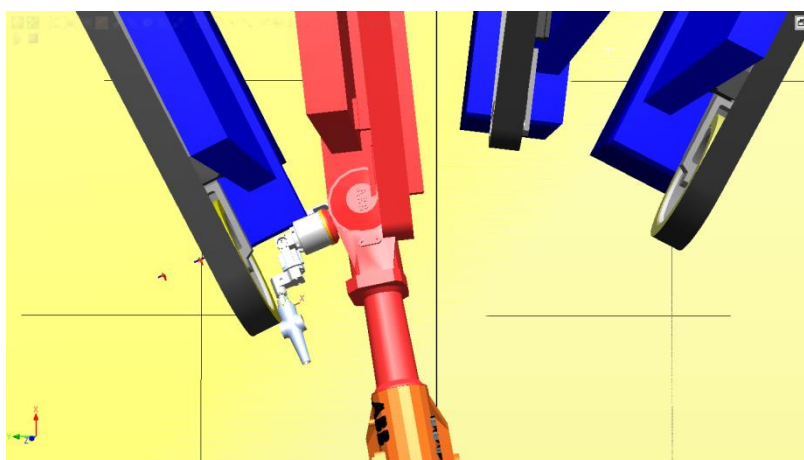
#### 6.3.3.2. Zona do Cartucho

Para o lixamento da zona do cartucho é necessário alterar a posição da ferramenta, de forma a ser possível fazer a rotação do cartucho de uma só vez. Para a programação altera-se a posição do mecanismo, alterando diretamente o valor da junta rotacional com a funcionalidade “*Joint Jog*”, ou utilizando o “*Move To Pose*” no separador “*Motion*”. Desta forma a peça é posicionada a 90° face ao eixo 6 do robô, sendo facilmente conseguida a rotação do cartucho segundo o TCP definido ao centro do mesmo para esta posição. Na Figura 177 apresenta-se a posição do robô num *target* desta rotina, onde se pode ver que a peça pode ser manuseada facilmente sobre o eixo de rotação do centro do cartucho sem se atingir o limite de movimento dos eixos do robô.



**Figura 177** *Target da rotina de lixamento da zona do cartucho (junta do punho a 90°)*

Salienta-se ainda na figura anterior a importância da posição das unidades, estando a unidade n.º 1 deslocada para a frente e as restantes em posição de repouso ou recuadas. Caso não se garantisse esta condição poderiam existir colisões, conforme se mostra na Figura 178 onde o robô colide com a unidade n.º 2, pois neste caso esta não foi desativada durante a programação.

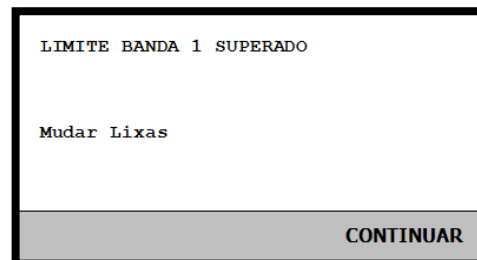


**Figura 178** *Colisão com a unidade de trabalho nº2*

Chegados aqui, finaliza-se esta descrição sem ser possível, no entanto, apresentar os seus resultados práticos, pois até a data de escrita deste documento, esta peça ainda não se encontrava em produção.

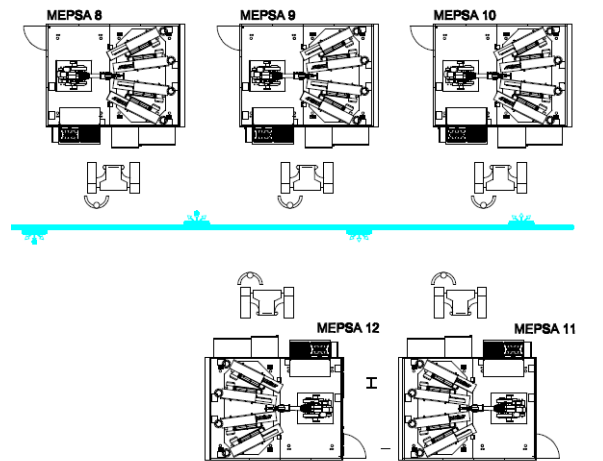
#### 6.4. ELIMINAÇÃO DE DESPERDÍCIOS – ALTERAÇÃO DA INTERFACE HOMEM-MÁQUINA

Após o trabalho de melhoria dos métodos de programação e de afinação convencionalmente utilizados no departamento, contribuindo para uma melhoria da taxa de utilização e aumento de produtividade, descreve-se a intervenção efetuada tendo em vista a redução de outro tipo de desperdícios - os tempos de espera dos robôs. Embora com menor impacto, os tempos de espera causam uma redução das taxas de utilização das células de trabalho, com consequências na produtividade do departamento. Analisando o problema, identificaram-se as principais causas dos tempos de espera, observando-se principalmente a paragem por falta de abastecimento de material e a paragem quando é atingido o limite das lixas, ficando o robô em espera até à intervenção do operador. A Figura 179 mostra um exemplo destas duas situações, apresentando-se na figura da esquerda o robô à espera da intervenção para carga e descarga da palete, e na figura da direita a mensagem da consola aquando da paragem do robô para a troca de lixas.



**Figura 179** Paragem dos robôs por falta de abastecimento ou para troca de lixas

O processo de lixamento robotizado conta com operadores diretos que têm como funções a carga e descarga do robô, a troca das cintas de lixa, o controlo visual dos corpos e, como tarefa principal, a recuperação ou acabamento das peças. Assim, consoante o tempo de acabamento manual da peça, o operário pode ser alocado a um ou mais robôs, podendo desta forma ficar afastado da célula de trabalho a que está afeto. Na Figura 180 é visível a disposição dos postos de lixamento manual face às células de trabalho no departamento, mostrando-se na figura da direita o *layout* de um dos grupos de robôs.



**Figura 180** Disposição dos postos de lixamento manual face às células robotizadas

A abordagem proposta consistiu em dar a conhecer ao operário o tempo necessário até ter de intervir no robô, permitindo assim uma melhor gestão de tempo por parte deste, assim como baixar os desperdícios de capacidade relacionados com os tempos de espera destas células de trabalho. Para isso foi necessário intervir na interface homem-máquina ou, em inglês, *human-machine interface* (HMI), sendo apresentado ao operário o tempo até ser necessária a sua intervenção no robô, consoante o estado das cintas de lixa e tempo de ciclo da máquina.

Para isso foi utilizada a interface da consola do robô para apresentar as informações necessárias. Neste caso, os robôs do departamento com controladores S4C são mais limitados do que a geração IRC5, permitindo este último a utilização de aplicações e interfaces gráficas mais poderosas, programados em C#. Visto grande parte dos robôs ABB do departamento possuir o controlador S4C, e de forma a padronizar a interface, foram utilizadas as funcionalidades de comunicação existentes para as consolas deste controlador, dando assim continuidade à atual interface existente e em utilização nos robôs, adicionando-se novas funcionalidades. Na Figura 181 é apresentada a interface existente, através do simulador da consola do RobotStudio (*Virtual FlexPendant*), onde se pode salientar a nível de informações apresentadas o tempo de ciclo da peça, número de peças produzidas por turno e tempo de paragem.

```

PROG_402027 Bica 402027
Versão: Teste
T. SAP: 180 T.Ciclo Robô:200s * Desac
A trabalhar peça 1 da Palete 1
Total paragens este turno: 0 minutos
*****PRODUÇÃO DAS ULTIMAS 24 HORAS*****
Turno da Noite      8
Turno da Tarde     31
Turno da Manhã     16
Este Turno         15
*****2013-06-26 17:48:08*****

```

**Figura 181** Ecrã da consola existente utilizado como base

No ecrã existente, é dado a conhecer ao operador o tempo de ciclo da máquina mas não o tempo até à próxima intervenção, sendo por isso necessário adicionar algumas informações na interface apresentada. Assim, foi necessário programar algumas rotinas em RAPID para apresentar o tempo até à próxima intervenção no robô.

O tempo até ao próximo abastecimento das peças pode ser calculado através do número total de peças por palete, da peça atual que o robô está a trabalhar e do tempo de ciclo real do robô. O tempo de ciclo do produto é obtido com a função “ClkRead” permitindo obter o valor do contador, sendo posteriormente multiplicado pelo número de peças restantes até à saída da paleta. Por fim, este valor é arredondado, obtendo-se o valor inteiro de minutos restantes até ao próximo abastecimento (apresenta-se de seguida um excerto do código):

```

!tempo até ao fim da paleta
tmp_palet:=Round(((Pieza_tl-
Pieza_act)+1)*(ClkRead(clock1)))/60);

!escrita para o Teach Pendant
TPWrite "Tempo para Sair Paleta1:
"+ValToStr(tmp_palet)+"min";

```

Relativamente ao cálculo do tempo até à troca das cintas de lixa, é necessário determinar a cinta de lixa que se encontra mais próxima do limite máximo de utilização, multiplicando-se o número de ciclos restantes pelo tempo de ciclo de produto. De seguida apresenta-se um excerto do código resumido apenas à unidade 2:

```

!nº de ciclos até ao limite da unidade2
restante2:=(limbanda2-Cont_2t);

!determina se é esta é a unidade com menor numero
de ciclos de vida
IF minimo>=restante2 AND restante2>0 THEN
    minimo:=restante2;
ENDIF

```

```

!tempo restante até a próxima troca
minimo:=Round(minimo*(ClkRead(clock1))/60);

!escrita para o Teach Pendant
TPWrite "Proxima Troca de lixa resta:
"+ValToStr(minimo)+string4;

```

O aspeto final do ecrã da consola do robô após a introdução destes campos é apresentado na Figura 182.

```

PROG_402027 Bica 402027
Versão: Teste
T.SAP:189s T.Ciclo Robô:200s * Desac
A trabalhar peça 1 da Palete 1
Tempo para Sair Paletel: 3min
Este turno tem: Perdas 0min PECAS 11
LIXA:U1:29 U2:29 U3:29 U4:29 Int:1
Proxima Troca de lixa resta: 20min
*****PRODUÇÃO DAS ULTIMAS 24 HORAS*****
Noite:8 Tarde:31 Manha:16
*****2013-06-26 17:33:04*****

```

**Figura 182** Ecrã da consola desenvolvido

Foi também feita uma proposta para se proceder à simplificação deste ecrã, devido ao excesso de informação que não é utilizada, permitindo melhorar a sua visualização. Esta proposta encontra-se, de momento, em fase de estudo.



# 7. CONCLUSÃO E PERSPETIVAS DE TRABALHOS FUTUROS

Com a realização deste trabalho pretendeu-se dar resposta aos problemas de produtividade do departamento de lixamento/polimento da Grohe Portugal relacionados com as frequentes paragens das células robotizadas para programação e ajustes de programas.

## 7.1. SOLUÇÃO ADOTADA

Assim sendo, testou-se a viabilidade da programação *offline* como alternativa ao método convencional, recorrendo-se à utilização do *software* da ABB, RobotStudio, e da KUKA, KUKA Sim, para a programação e simulação do ciclo de lixamento de novos produtos, sendo apresentadas as vantagens, dificuldades e limitações deste método de programação para esta operação de contacto. Ainda para resolver os problemas de produtividade, complementou-se este trabalho com o desenvolvimento de um novo método de ajuste/afinação *online* do programa face ao método convencional (e muito demorado) de ajuste ponto-a-ponto. Procedeu-se também à alteração da interface homem-máquina,

procurando reduzir-se os principais desperdícios encontrados nas células de trabalho robotizadas, aplicando assim um dos princípios do pensamento *lean*.

## **7.2. DIFICULDADES E SOLUÇÕES**

Para o sucesso da implementação da programação *offline* foi necessário a aquisição de conhecimentos sobre as operações de acabamento superficial e um trabalho rigoroso de preparação antes da programação propriamente dita. Este método de programação exige inicialmente um grande esforço para a correta modelação da célula de trabalho, dependendo o sucesso deste método do nível de realismo conferido às duas superfícies de contacto, aproximando o ambiente virtual à célula real.

Foram assim desenvolvidos esforços para ultrapassar algumas limitações e restrições encontradas ao nível das aplicações de *software* utilizadas, de forma a adaptar e melhorar a sua utilização às necessidades deste tipo de aplicações de contato. Isto conduziu também à necessidade de aprofundar os conhecimentos sobre o funcionamento da aplicação e do próprio robô.

De forma a enriquecer os conhecimentos na área, foram também acompanhadas e realizadas algumas intervenções de manutenção periódica e de reparação dos robôs que contribuíram para um melhor conhecimento de todo o sistema do manipulador, desde o seu sistema de controlo até ao movimento do seu elo terminal. Isto permitiu, juntamente com o estudo da documentação do robô e da literatura na área, conhecer os principais problemas que podem estar na origem da falta de precisão e repetibilidade do robô, podendo comprometer a aplicação da programação *offline* ou a portabilidade dos programas entre células do departamento com o mesmo *layout*.

## **7.3. ANÁLISE DAS VANTAGENS E LIMITAÇÕES**

Após terem sido ultrapassadas as principais dificuldades da modelação, analisou-se o resultado da aplicação ao nível da qualidade do processo, tempo de ciclo do produto e, essencialmente, tempo de programação *online* e ajuste na célula de trabalho. Relembrando que o objetivo deste trabalho é a redução de desperdícios de produção devido a pausas e paragens das células produtivas, a aplicação do método de programação *offline* e das novas técnicas de programação revelaram ser uma alternativa viável, obtendo-se uma redução do tempo de paragem da célula e, simultaneamente, bons resultados ao nível do acabamento

do produto, sendo desta forma uma mais-valia para o departamento, ao nível da produtividade e qualidade geral.

Analisando as vantagens ao nível de eficiência, foi possível reduzir drasticamente o tempo de paragem da célula. Enquanto a programação inteiramente *online* de um produto envolve a paragem da célula durante 5 dias em média (40 horas de trabalho), com a programação *offline* reduziu-se esse tempo para apenas 6 horas, sendo este tempo necessário para a realização de testes e pequenos ajustes do programa, o que representa uma redução de 85%. Na prática, o tempo necessário ao desenvolvimento do primeiro programa *offline* de uma célula requer a paragem da mesma para a sua avaliação, calibração dos referenciais do programa e possíveis testes, aumentando o tempo de paragem da célula para sensivelmente o dobro, ou seja, 12 horas (ainda assim obtém-se redução média de 70%).

Considerando-se, como referência, um custo médio por hora do robô de 13,58 €, o custo da programação de um novo ciclo de lixamento através dos dois métodos de programação pode ser calculado conforme se apresenta na Tabela 2.

**Tabela 2 Comparativo do custo total entre os dois métodos de programação**

	Tempo de paragem (h)	Custo por hora (€)	Custo total (€)
<b>Programação <i>online</i></b>	40	13,58	543,20
<b>Programação <i>offline</i></b>	6	13,58	81,48
<b>Ganhos por produto</b>	34	13,58	461,72

Recorrendo à simulação e programação *offline* foi também possível otimizar o tempo de ciclo do produto, obtendo-se uma redução média de 10% face ao processo convencional, permitindo assim um aumento significativo de produtividade. Apresenta-se na Tabela 3 um comparativo relativamente ao custo e tempo necessário à produção de 1000 peças.

**Tabela 3 Comparativo da produção de 1000 peças**

	Tempo de ciclo médio por peça (s)	Tempo 1000 peças (h)	Custo robô 1000 peças (€)
<b>Programa <i>online</i></b>	150	41,67	565,88
<b>Programa <i>offline</i> otimizado</b>	135	37,50	509,25
<b>Ganhos obtidos</b>	15	4,17	56,63

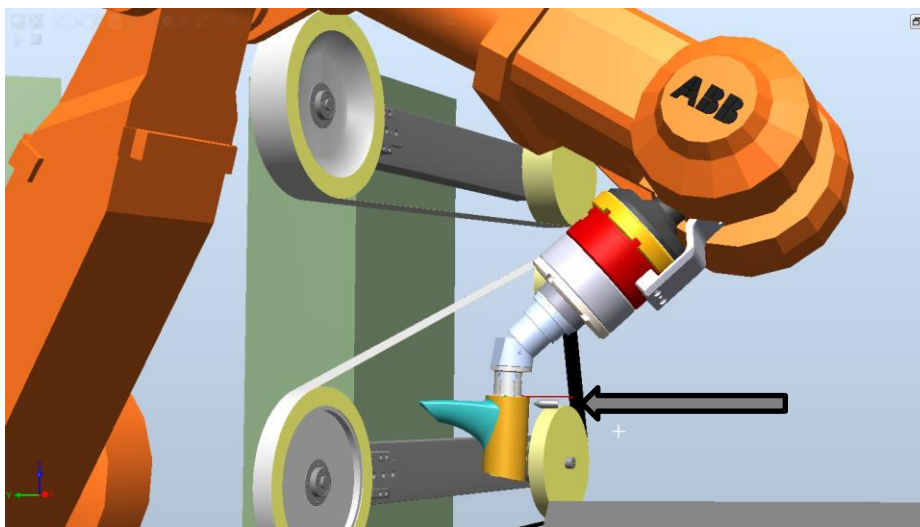
Num cenário de inovação constante, a aplicação deste método de programação provou ser uma mais-valia, permitindo diminuir o impacto da programação *online* e diminuir o tempo de ciclo do processo.

Quanto às limitações, o principal problema deste método de programação reside na falta de realismo da flexibilidade da roda e da falta de sensibilidade da força de contacto com a mesma. Tal situação pode ser ultrapassada após alguma experiência do processo e após algumas transferências da simulação para a realidade, de forma a se ter a percepção na simulação da aproximação que é necessária para se obter a força pretendida na célula de trabalho.

#### **7.4. DESENVOLVIMENTOS FUTUROS**

Como possíveis desenvolvimentos futuros nesta área podem-se identificar alguns trabalhos que favoreceriam, quer a produtividade do departamento, quer a qualidade final do produto. Assim, estes trabalhos são numerados de seguida:

- Implementação do sistema de ajuste do programa: verificado o correto funcionamento do sistema na simulação no robô é possível passar à implementação do sistema numa célula robotizada de lixamento. O processo de implementação encontra-se atualmente em fase de estudo, sendo apresentada na Figura 183 a rotina de medição já desenvolvida no RobotStudio, procedendo-se ao ajuste automático do programa entre ciclos de produção ou após um *setup*;



**Figura 183** Implementação do sistema de ajuste automático do programa

- Desenvolvimento de um sistema de controlo de força, utilizando para isso um extensómetro na pinça ou punho do robô, para permitir alguma complacência às variações apresentadas pelas peças.

Tendo em conta todo o conhecimento e experiência adquirida e a possibilidade de pôr em prática muitos conhecimentos debatidos ao longo do meu percurso académico, fez desta uma oportunidade de excelência para entrar em contacto com o mundo do trabalho, proporcionando uma experiência real na área da robótica industrial.

Concluindo, esta foi uma experiência muito enriquecedora, tendo o privilégio de desenvolver este trabalho numa empresa como a Grohe, num departamento que procura uma constante melhoria dos seus processos.



## *Referências Documentais*

- [1] ABB, Manual IRC5 com Flex Pendant, IRC5 M2004, ID do documento: 3HAC 16590-6, Revisão L.
- [2] ABB, Force Control for Machining Applications, Acedido em 05 de março de 2013, <http://www.abb.pt/product/seitp327/cbadd5a0fe0ad677c125735d00496705.aspx?productLanguage=us&country=PT>.
- [3] CMA Robotics, Self-Learning Painting Robot, Acedido em 06 de maio de 2013 <http://www.cmarobot.it/eng/robot.php#>.
- [4] Nof, Shimon Y. - *Handbook of Industrial Robotics*; John Wiley & Sons, Inc; 1999. ISBN 0-471-17783-0.
- [5] ABB, *RobotStudio*, Acedido em 05 de março de 2013 <http://www.abb.pt/product/seitp327/78fb236cae7e605dc1256f1e002a892c.aspx>.
- [6] KUKA, *Application Software*, Acedido em 05 de março de 2013 [http://www.kuka-robotics.com/en/products/software/application\\_software/start.htm](http://www.kuka-robotics.com/en/products/software/application_software/start.htm).
- [7] FANUC, *RobotGuide*, Acedido em 05 de março de 2013 <http://www.fanucrobotics.com/Products/vision-software/ROBOGUIDE-simulation-software.aspx>.
- [8] Bernd Kuhlenkoetter; Xiang Zhang. *A Robot System for High Quality Belt Grinding and Polishing Processes*, Cutting Edge Robotics, Vedran Kordic, Aleksandar Lazinica and Munir Merdan (Ed.), InTech, 2005. ISBN: 3-86611-038-3.
- [9] MEPSA, *Grinding robotic cell*, Acedido em 05 de março de 2013 [http://www.mepesa.es/acabado\\_de\\_superficies.php?idioma=12&aplicacion=5&proceso=6](http://www.mepesa.es/acabado_de_superficies.php?idioma=12&aplicacion=5&proceso=6).
- [10] SHL, *Grinding robots*, Acedido em 05 de março de 2013 [http://www.shl-automatisierung.de/cms/fr/poncage\\_automatise/grinding\\_robots/](http://www.shl-automatisierung.de/cms/fr/poncage_automatise/grinding_robots/).
- [11] ABB, Robot User's Guide, 1998, ID do documento: 3HAC 0966-21.
- [12] Dynalog, *AutoCal*, Acedido em 05 de março de 2013 <http://www.dynalog-us.com/solutions/?CategoryID=2&cID=5>.
- [13] Leoni, *Tool Calibration*, Acedido em 05 de março de 2013 <http://www.leoni-industrial-solutions.com/Werkzeugkalibrierung.10038.0.html?&L=1>.
- [14] Meta, Robot welding systems Acedido em 05 de março de 2013 [http://www.meta-mvs.com/lp\\_lasersight.html](http://www.meta-mvs.com/lp_lasersight.html).

- [15] Siciliano, Bruno; Khatib, Oussama - Springer Handbook of Robotics; Springer+Business Media; 2008. ISBN 978-3-540-23957-4, e-ISBN 978-3-540-30301-5.
- [16] WERTZ, Alan; Robot grinding Using ABB Force Control Technology, Benton Foundry, INC. 2011.
- [17] ABB, IRB4400 Datasheet, Acedido em 06 de maio de 2013  
[http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/0dc8b0df9cc39049c125772e0057e806/\\$file/irb%204400%20pr10035en%20r8.pdf](http://www05.abb.com/global/scot/scot241.nsf/veritydisplay/0dc8b0df9cc39049c125772e0057e806/$file/irb%204400%20pr10035en%20r8.pdf).
- [18] ABB, Calibration Pendulum Instruction, 2007, ID do documento: 3HAC 16578-1, Revisão E.
- [19] ABB, Operating Manual RobotStudio 5.15, 2013, ID do documento: 3HAC 032104-001, Revisão K
- [20] Mathworks, Matlab Central – quaternion.m, Acedido em 27 de junho de 2013  
<http://www.mathworks.com/matlabcentral/fileexchange/33341-quaternion-m>
- [21] ABB, RAPID Reference Manual, IRC5, 2009, ID do documento: 3HAC16581-1, Revisão B, 2009.
- [22] KUKA, Entrada em serviço, Release 4.1, 2002
- [23] KUKA, User Programming, KR C2 /KR C3, Release 5.2, 2003





