

Desenvolvimento da componente de controlo e de montagem de um sistema de montagem de pinças kocher

FILIPE DANIEL FERREIRA

setembro de 2018

Desenvolvimento da componente de controlo e de montagem de um sistema de montagem de Pinças Kocher

Filipe Daniel Ferreira



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2018

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

Candidato: Filipe Daniel Ferreira, Nº 1130425, 1130425@isep.ipp.pt
Orientação científica: Manuel Fernando dos Santos Silva, mss@isep.ipp.pt
Empresa: RobotSol - Engenharia Industrial
Supervisão: Ricardo Guedes Pimenta, ricardo.pimenta@robotsol.com



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

30 de julho de 2018

Agradecimentos

Em primeiro lugar quero agradecer aos meus orientadores, Ricardo Pimenta e Manuel Silva, pelo auxílio prestado e pela troca de conhecimentos.

De seguida agradeço à empresa RobotSol pela oportunidade única concedida para a elaboração deste projeto e pela confiança demonstrada ao longo do mesmo. Quero agradecer a todos os departamentos que sempre se mostraram prontos a colaborar para a conclusão com sucesso do projeto referido.

Agradeço também aos meus colegas Pedro Miguel, Ricardo Martins e José Nunes que em ambiente de estágio se mostraram sempre prestáveis e ajudaram à elaboração deste projeto. Ao Pedro Miguel agradeço todo o apoio para a conclusão deste projeto que no seu global pertence a ambos.

Por fim quero agradecer a todos aqueles que direta ou indiretamente permitiram a conclusão deste projeto.

Resumo

Atualmente, com a evolução tecnológica e a competitividade dos mercados, surge a necessidade de se aumentar a produção e a especificidade dos produtos por forma a responder à necessidade dos mercados. Também com o avanço tecnológico surge a necessidade de se retirarem os colaboradores de operações de possível perigo e repetibilidade que não ajudam no desenvolvimento do mesmo.

Nesta perspetiva, face a um pedido externo a RobotSol, sentiu a necessidade de criar um sistema automático para uma aplicação de montagem por forma a evitar que colaboradores humanos executassem tal tarefa repetitiva sem qualquer evolução. O sistema visava também a perspetiva de melhoria na produção e redução do tempo de ciclo da produção.

O objetivo desta dissertação reside na simulação, programação e controlo de um sistema totalmente automatizado para montagem de pinças Kocher com recurso a robôs industriais. O trabalho realizado incidiu na simulação de um ambiente industrial, programação do centro de controlo da célula e estudo e programação de um robô do tipo paralelo para uma tarefa de *pick and place* e montagem.

Com base na automatização da tarefa de montagem e controlo do restante ambiente da célula procedeu-se ao desenvolvimento de um sistema autónomo que permitisse alcançar as produções definidas com reduzidos tempos de ciclo.

Para a implementação deste sistema foi necessário proceder a uma análise de soluções de *pick and place* e montagem executadas por sistemas robóticos semelhantes. O objetivo do estudo dos componentes da solução foi criar bases de raciocínio para a programação do sistema e conhecimentos técnicos para a implementação da célula. Após este estudo teórico procedeu-se à implementação da programação e controlo do sistema. Após a montagem do sistema e da sua implementação executaram-se testes para a validação da solução final. Durante esta fase foi testado o sistema de controlo, o de montagem, as comunicações e a capacidade de rejeição de peças por parte do sistema. Ainda durante esta fase foi analisado o tempo de ciclo do sistema. Estes testes foram efetuados em am-

biente de simulação e em contexto real. Os resultados alcançaram o esperado, o sistema conseguindo montar 100% das peças em menos de 12 s e caso fosse necessário rejeitar 100% das peças a rejeitar.

Com esta dissertação foi possível a aquisição de conhecimentos acerca do mercado associado a este tipo de aplicação, nomeadamente as áreas de cliente mais propícias à aplicação de soluções de *pick and place* e montagem. Foram também adquiridas valências na configuração e programação de um robô do tipo paralelo, bem como a sua inserção numa solução final. Associadas as estas valências foi possível adquirir conhecimentos e melhorar bases de conhecimento na área da automação industrial, nomeadamente no controlo através de autómatos programáveis e segurança através de relés. Por fim, foi possível ganhar experiência em contexto de cliente através da implementação da solução num cliente final, adquirindo experiência única em ambiente industrial.

Palavras-Chave

Robótica industrial, automação industrial, controlo, controlador lógico programável, robô paralelo, segurança, Fanuc, Omron, sensorização, montagem, *pick and place*.

Abstract

Nowadays, with the technological evolution and the competitiveness of the markets, the need arises to increase the production and specificity of the products in order to respond to the need of the markets. With the technological advance arises the need to withdraw the employees from operations of possible danger and repeatability that do not help in their development.

In this perspective, in response of an external request, RobotSol felt the need to create an automatic system for an assembly application in order to prevent human collaborators from performing such a repetitive task without any evolution. The system also aimed at improving production prospects and reducing production cycle time.

The purpose of this dissertation is the simulation, programming and control of a fully automated system for mounting Kocher tweezers using industrial robots. The present work focused on the simulation of an industrial environment, programming of the cell control center and study and programming of a parallel robot for a pick and place and assembly task.

Based on the automation of the assembling task and controlling the rest of the cell environment, an autonomous system was developed that allowed to reach the defined productions with reduced cycle times.

For the implementation of this system it was necessary to carry out an analysis of pick-and-place and assembly solutions executed by similar robotic systems. The purpose of the study of the components similar solutions was to create bases of reasoning for the programming of the system and technical knowledge for the implementation of the cell. After this theoretical study, the programming and control of the system was implemented. After the assembly of the system and its implementation, tests were carried out for the validation of the final solution. During this phase was tested control system, the assembly, the communications and the capacity of rejection parts. At this phase was also analysed the cycle time of the system. These tests were carried out in simulation environment and in real context. The results were as expected, the system was

able to assemble 100 % of the parts in less than 12 s and if it was necessary to reject 100 % of the parts to be rejected.

With this dissertation it was possible to acquire knowledge about the market of this type of application, namely the client areas most conducive to the application of pick and place and assembly solutions, and acquired experience in configuration and programming of a parallel robot. Associated at these experiences, it was possible to acquire knowledge in the industrial automation area, control and security through relays. Finally, it was possible to gain experience in the field with customers and with the industrial environment.

Keywords

Industrial robotics, industrial automation, control, programmable logic controller, parallel robot, security, Fanuc, Omron, sensors, assembly, pick and place.

Conteúdo

Agradecimentos	v
Conteúdo	i
Lista de Figuras	vii
Lista de Tabelas	xi
Glossário	xiii
1 Introdução	1
1.1 Contextualização	1
1.2 Objectivos	2
1.2.1 Requisitos	2
1.2.2 Testes funcionais	3
1.3 Calendarização	3
1.4 Estrutura da dissertação	3
2 Estudo Conceptual da Robótica	7
2.1 Introdução à robótica	7
2.1.1 Terminologia e origem do conceito "Robótica"	7
2.1.2 Definição de robô	8
2.1.3 Evolução histórica da robótica	9
2.2 Constituição de um robô industrial	11
2.2.1 Elos	11
2.2.2 Juntas	12
2.2.3 Manipulador	14
2.2.4 Punho	15
2.2.5 Atuadores	16
2.2.6 Atuadores finais	17
2.2.7 Sensores	18

2.2.8	Controlador	19
2.2.9	<i>Teach pendant</i>	21
2.3	Conceitos básicos da robótica	21
2.4	Algoritmos de controlo	23
2.4.1	Controlo de sequência limitada	23
2.4.2	Controlo de reprodução ponto a ponto	23
2.4.3	Controlo de reprodução de trajetória contínua	23
2.4.4	Controlo inteligente	24
2.5	Métodos de programação de um robô	24
2.5.1	Programação <i>online</i>	24
2.5.2	Programação <i>offline</i>	26
2.6	Conclusão	28
3	Contexto Industrial	29
3.1	Aplicações industriais de robôs	29
3.1.1	Operações de manuseamento de materiais	31
3.1.2	Operações de montagem e inspeção	32
3.1.3	Operações de processamento	33
3.2	Controladores Lógicos Programáveis	35
3.2.1	Conceito	35
3.2.2	Partes constituintes de um PLC	37
3.2.3	Linguagens de programação	40
3.3	Processo de controlo com recurso a PLC	40
3.3.1	Sistemas de fabrico	41
3.3.2	Classificação dos sistemas de fabrico	42
3.3.3	Controlo dos processos através de PLC	42
3.3.4	Tipos de comunicação	43
3.3.5	Sistemas SCADA	45
3.4	Conclusão	45
4	Segurança e Aplicações Industriais	47
4.1	Aplicações de <i>pick and place</i> e montagem	47
4.1.1	Robôs de <i>pick and place</i> e montagem	47
4.1.2	<i>Software</i> de programação para aplicações de <i>pick and place</i>	52
4.1.3	Garras para aplicações de <i>pick and place</i>	55
4.2	Segurança em Ambiente Industrial	57
4.2.1	Perigos do uso de robôs	57
4.2.2	Normas e legislação	58
4.2.3	Medidas de proteção	59
4.2.4	Interface Homem-robô	60
4.3	Casos de estudo da aplicação <i>pick and place</i>	61
4.3.1	ABB - <i>picking pizza pick and place</i>	61

4.3.1.1	Problema	61
4.3.1.2	Componentes da solução	62
4.3.1.3	A solução apresentada pela Vortex	63
4.3.2	Uma solução para a indústria do chocolate	63
4.3.2.1	Problema	63
4.3.2.2	Componentes da solução	64
4.3.2.3	A solução apresentada pela Han Shin Power Tech Company	64
4.3.3	Redução do tempo de programação do robô	65
4.3.3.1	Problema	65
4.3.3.2	Componentes da solução	65
4.3.3.3	A solução apresentada pela R+D Custom Auto- mation	66
4.4	Casos de estudo de paletização	67
4.4.1	Embalar e paletizar café: Löfbergs Lila	67
4.4.1.1	Problema	67
4.4.1.2	Componentes da solução	67
4.4.1.3	A solução apresentada pela ABB	67
4.4.2	Robô de longo alcance da KUKA melhora o desempenho da paletização por camadas	68
4.4.2.1	Problema	68
4.4.2.2	Componentes da solução	68
4.4.2.3	A solução apresentada pela KUKA	68
4.4.3	Integração de um robô colaborativo numa aplicação de pa- letização	69
4.4.3.1	Problema	69
4.4.3.2	Componentes da solução	69
4.4.3.3	A solução apresentada pela Universal Robots	70
4.5	Conclusão	70
5	Arquitetura do Sistema	71
5.1	Exposição do problema	71
5.2	Composição do sistema	73
5.3	Hardware	74
5.3.1	FANUC M20iA/20M	74
5.3.2	FANUC M2iA/3SL	75
5.3.3	Máquina de injeção	78
5.3.4	Mesa de montagem	78
5.3.5	Unidade de controlo	82
5.3.6	Interface Homem-Máquina	84
5.3.7	Segurança	85
5.4	Software	87

5.4.1	Visual Components	87
5.4.2	Roboguide	88
5.4.3	Sysmac Studio	89
5.4.4	NB Designer	90
5.5	Protocolo Ethernet/IP	90
5.6	Cenários analisados	90
5.6.1	Organização base do cenário	90
5.6.2	Peça de 13 cm de comprimento	91
5.7	Conclusão	93
6	Implementação Prática	95
6.1	Modelação e simulação	95
6.1.1	Exportar ficheiros CAD do projeto em Inventor	95
6.1.2	Esquema da célula	96
6.1.3	Modelação e simulação em Visual Components	98
6.1.3.1	Modelação	98
6.1.3.2	Simulação	99
6.1.4	Modelação e simulação em Roboguide	101
6.1.4.1	Modelação	101
6.1.4.2	Simulação	102
6.2	Programação do robô	104
6.2.1	Configuração dos sinais utilizados pelo robô	104
6.2.2	Arranque remoto do robô	111
6.2.3	Configuração da ferramenta e da carga a movimentar	113
6.2.4	Configuração dos sistemas de coordenadas	115
6.2.5	Configuração do espaço partilhado entre robôs	118
6.2.6	Configuração da rede	120
6.2.7	Criação do conjunto de programas a executar	122
6.3	Controlo da célula	125
6.3.1	Configurações no PLC	125
6.3.2	Lógica implementada no controlo	127
6.4	Conclusão	133
7	Testes e Resultados	135
7.1	Testes em ambiente simulação	135
7.1.1	Teste de validação da mesa, bases e garra	135
7.1.2	Teste do arranque do robô	136
7.1.3	Testes de movimento controlado por sinais	137
7.1.4	Teste do tempo de movimento do robô	138
7.2	Testes da comunicação e da componente física	139
7.2.1	Teste de comunicação entre PLC e robôs	139
7.2.2	Testes de comunicação entre PLC e HMI	140

7.2.3	Teste da leitura dos sinais da mesa	141
7.2.4	Testes da atuação de saídas da mesa	142
7.2.5	Testes aos sinais de interface entre robô e atuador final	143
7.3	Testes ao sistema	144
7.3.1	Testes com amostras de peças	144
7.3.2	Testes em produção	147
7.4	Conclusão	149
8	Conclusão	151
8.1	Solução implementada	151
8.2	Melhorias futuras	153
8.3	Considerações finais	154
	Referências Bibliográficas	155
A	Variáveis de comunicação entre Robô e o PLC	159
A.1	Mapeamento padrão das entradas e saídas da RobotSol	160
A.2	Sentido R2 -> PLC	161
A.3	Sentido PLC -> R2	162
B	Variáveis de comunicação entre R1 e PLC	166
B.1	Mapeamento padrão das entradas e saídas da RobotSol	166
B.2	Sentido R1 -> PLC	167
B.3	Sentido PLC -> R1	168
C	Programação Robô2	169
C.1	Fluxograma do programa principal	170
C.2	Fluxograma do programa de manutenção	171
C.3	Fluxograma do programa da peça 13cm	172
C.4	Fluxograma base da lógica seguida pelo robô	174
C.5	Fluxograma base da lógica seguida pelo robô na base 1	175
C.6	Fluxograma base da lógica seguida pelo robô na base 2	176
D	Configuração do PLC	177
D.1	Configuração da rede EtherCat	178
D.2	Mapeamento das entradas e saídas	180
D.3	Seleção dos programas a executar	181
D.4	Configuração da rede de comunicação entre o PLC e os robôs	183
E	Lógica de controlo	187
E.1	Leitura das entradas do PLC	187
E.2	Arranque do sistema	189
E.3	Rotinas de arranque	190

E.4	Rotinas de carregar receita	192
E.5	Arranque do robô de descarga	193
E.6	Arranque do robô delta	194
E.7	Comunicação com a HMI	195
E.8	Ciclo base do sistema	196
E.9	Ciclo inicial da montagem	197
E.10	Ciclo de montagem do robô 2	199
E.11	Ciclo de escrita nas saídas do PLC	200

Lista de Figuras

1.1	Calendarização do projeto	5
2.1	Elos e juntas de um robô	11
2.2	Configuração de um robô paralelo.	12
2.3	Exemplos do tipo de juntas.	12
2.4	Configurações dos robôs série.	14
2.5	Manipulador robótico FANUC M20iA	15
2.6	Punho de um braço robótico	15
2.7	Blocos de funcionamento de um controlador	20
2.8	Linha de controladores de robôs Kuka	21
2.9	<i>Teach Pendant</i> da FANUC	21
2.10	<i>Leadthrough</i> manual	26
2.11	Exemplo de simulação	28
3.1	Exemplo de uma aplicação industrial: indústria automóvel	29
3.2	Exemplo de uma aplicação de transferência de materiais	32
3.3	Diagrama de constituintes de um PLC	37
4.1	a) Manipulador robótico IRB 140. b) Manipulador robótico paralelo IRB360. c) Manipulador robótico SCARA IRB910sc.	50
4.2	a) Manipulador robótico LBR iiwa. b) Manipulador robótico KR 5 sixx. c) Manipulador robótico KR 6 R900 sixx.	51
4.3	a) Manipulador robótico M-2iA/3SL [1]. b) Manipulador robótico CR-35iA. c) Manipulador robótico SR-3iA.	52
4.4	Garras paralelas da Schunk	55
4.5	Garras concêntricas da Schunk	56
4.6	Garras concêntricas de quatro dedos da Schunk	56
4.7	Imagem da linha implementada	62
4.8	Linha implementada pela OMRON	65
4.9	Linha de <i>pick and place</i> adaptável	66

5.1	Diagrama de blocos do sistema global	73
5.2	Robô FANUC M20iA/20M	74
5.3	Robô FANUC M2iA/3SL	75
5.4	Volume de trabalho do robô FANUC M2iA/3SL	76
5.5	Diagrama de carga do modo de inércia padrão	77
5.6	Diagrama de carga do modo de alta inércia	78
5.7	Esquema da mesa de montagem de pinças	79
5.8	Conjunto de eletroválvulas biestável da Festo	80
5.9	Sensor de proximidade da Festo	80
5.10	Sensor indutivo da Contrinex	81
5.11	Sensor fotoelétrico da Contrinex	81
5.12	Pressoestato da Festo	81
5.13	Sensor fotoelétrico da Contrinex para detecção de peça	82
5.14	PLC NX1P2-9024DT1 da OMRON	82
5.15	Fonte de alimentação da Phoenix Contact	83
5.16	Carta de entradas da OMRON	83
5.17	Carta de saídas da OMRON	84
5.18	Módulo EtherCAT da OMRON	84
5.19	HMI NB10W-TW01B da OMRON	85
5.20	a) Botoneira de segurança Siemens. b) Botoneiras de emergência do controlador e <i>teach pendant</i> . c) Botoneira de emergência da PSEngate.	85
5.21	Relé de segurança G9SE-401 da OMRON.	86
5.22	PSEngate da PILZ.	86
5.23	PSEngate da PILZ.	87
5.24	Exemplo de uma aplicação de <i>pick and place</i> no Visual Components.	88
5.25	Exemplo de modelação e simulação do cenário a implementar	89
5.26	a) Vista frontal superior da célula. b) Vista traseira superior da célula.	91
5.27	Bases para as peças de 13 cm de comprimento.	91
5.28	Localização dos sensores na base.	92
5.29	Exemplo da peça de 13 cm.	93
6.1	Exportar ficheiros CAD para modelação.	96
6.2	Esquema da célula e medidas das redes.	97
6.3	Importar ficheiros CAD para modelação no Visual Components.	98
6.4	Criação de um programa no Visual Components.	100
6.5	Modelação da célula em Roboguide.	102
6.6	Simulação da célula em Roboguide.	103
6.7	Simulação da célula em Roboguide - <i>work envelope</i> e configuração de sinais de colisão.	104
6.8	Menu na consola dos sinais de I/O.	105
6.9	Configuração dos sinais de I/O digitais.	106
6.10	Configuração dos sinais de I/O de grupo.	107

6.11	Sinais de I/O do robô.	108
6.12	Sinais de I/O do robô de interface com PLC.	109
6.13	Sinais específicos para interface PLC e robô.	111
6.14	Configurações para arranque através dos sinais UI.	112
6.15	Configurações para arranque através dos sinais RSR.	113
6.16	Menu de definição das ferramentas a utilizar.	114
6.17	Configuração das ferramentas a utilizar.	115
6.18	Menu dos sistemas de coordenadas de utilizador.	116
6.19	Configuração dos sistemas de coordenadas.	117
6.20	<i>User frames</i> marcadas em simulação e na prática.	118
6.21	Menu de <i>space function</i>	119
6.22	Menu de <i>space function</i>	120
6.23	Configuração dos dados a transmitir.	121
6.24	Configuração dos endereços.	121
6.25	Diagrama de árvore das rotinas.	122
6.26	Esquema das bases para montagem das pinças.	124
6.27	Ambiente de trabalho do Sysmac Studio.	126
6.28	Organização da programação do controlo.	128
6.29	Organização de cada programa do controlo.	129
6.30	Esquema do programa General_Control.	130
7.1	Simulação de colisão no Visual Components.	136
7.2	Simulação de arranque do robô via RSR no roboguide.	137
7.3	Simulação dos movimentos do robô no roboguide.	138
7.4	Simulação do tempo de movimento do robô no roboguide.	139
7.5	Teste de comunicação entre R2 e PLC.	140
7.6	Excerto do esquema elétrico.	142
7.7	Exemplo da base na qual foram efetuados os testes.	143
7.8	Garra testada.	144
C.1	Fluxograma do programa principal do robô.	170
C.2	Fluxograma da rotina de manutenção do robô.	171
C.3	Fluxograma da rotina da peça 1 do robô.	172
C.4	Fluxograma base da lógica seguida pelo robô.	174
C.5	Fluxograma base da lógica seguida pelo robô ao montar peças na base 1.	175
C.6	Fluxograma base da lógica seguida pelo robô ao montar peças na base 2.	176
D.1	Menu de configuração dos módulos EtherCAT.	178
D.2	Nó da rede EtherCAT para ligação das cartas de entradas e saídas.	179
D.3	Janela de mapeamento das entradas e saídas.	180

D.4	Menu de seleção de tarefas a executar.	181
D.5	Nó de comunicação entre o PLC e os robôs.	183
D.6	Robôs ligados ao PLC.	184
D.7	Definição da comunicação.	185
D.8	Configuração da rede no Network Configurator.	186
E.1	Leitura da informação proveniente do Robô.	187
E.2	Diagrama de blocos do programa Input.	188
E.3	Fluxograma do controlo do arranque do sistema.	189
E.4	Fluxograma das rotinas de controlo de pedido de acesso, controlo da psengate, controlo de ar, controlo do volume de trabalho, interface entre sensores e o robô delta.	190
E.5	Carregar receita proveniente do HMI.	192
E.6	Arranque do robô de descarga.	193
E.7	Arranque do robô delta.	194
E.8	Fluxograma da comunicação entre PLC e HMI.	195
E.9	Fluxograma do ciclo base do sistema.	196
E.10	Fluxograma do ciclo inicial da montagem.	197
E.11	Fluxograma do ciclo de montagem do robô 2.	199
E.12	Fluxograma do ciclo de escrita das saídas.	200

Lista de Tabelas

4.1	Robôs de <i>pick and place</i> da ABB [2, 3, 4].	49
4.2	Robôs de <i>pick and place</i> da KUKA [5, 6, 7].	50
4.3	Robôs de <i>pick and place</i> da FANUC [1, 8, 9].	51
6.1	Significado do valor recebido através do código do programa	123
6.2	Esquema de cores e significados da torre luminosa	133
7.1	Testes de montagem por parte do robô e escoamento por parte da mesa	145
7.2	Teste da rejeição das peças	146
7.3	Teste em produção com matéria prima antiga	148
7.4	Testes em produção com a nova matéria prima	149
A.1	Divisão das entradas e saídas do Robô 2 por grupos	160
A.2	Variáveis utilizadas para as saídas do Robô 2 e entradas do PLC	161
A.3	Variáveis utilizadas para as entradas do Robô 2 e saídas do PLC	162
A.4	Variáveis utilizadas para as entradas do Robô 2 e saídas do PLC - sensores	163
B.1	Divisão das entradas e saídas do Robô 1 por grupos	166
B.2	Variáveis utilizadas para as saídas do Robô 1 e entradas do PLC	167
B.3	Variáveis utilizadas para as entradas do Robô 1 e saídas do PLC	168

Glossário

Abreviatura	Descrição
AC	<i>Alternate Current</i>
AGV	<i>Automated Guided Vehicle</i>
CAD	<i>Computer Aided Design</i>
CAM	<i>Computer Aided Manufacturing</i>
CIP	<i>Common Industrial Protocol</i>
CPU	<i>Central Processing Unit</i>
DC	<i>Direct Current</i>
DOF	<i>Degrees of Freedom</i>
EDS	<i>Electronic Device Data Sheets</i>
EU-OSHA	Agência Europeia para a Segurança e Saúde no Trabalho
HMI	(Human Machine Interface)
I/O	<i>Input/Output</i>
ISEP	Instituto Superior de Engenharia do Porto
LED	<i>Light Emitting Diode</i>
MIT	Massachusetts Institute Technology
PC	Personal Computer
PDF	<i>Portable Document Format</i>
PID	Proporcional-Integral-Derivativo
PLC	Programmable Logic Controllers
PUMA	Programmable Universal Machine for Assembly
RIA	Robotics Industrial Association
RUR	Rossum's Universal Robots
SCADA	<i>Supervisory Control and Data Acquisition</i>
SOP	<i>System Operator Panel</i>
STL	<i>Standard Triangle Language</i>
TCP	<i>Tool Center Point</i>
TFT LCD	<i>Thin Film Transistor Liquid Crystal Display</i>
UI	<i>User Input</i>
UO	<i>User Output</i>
UOP	<i>User Operator Panel</i>

Abreviatura	Descrição
USB	<i>Universal Serial Bus</i>

Capítulo 1

Introdução

A indústria médica é parte de extrema importância em qualquer país. Esta indústria tem por consumidor final os prestadores de serviços médicos. Os prestadores dos serviços médicos, neste momento, são uma importante parte de todas as sociedades modernas devido ao envelhecimento da população e aos avanços na medicina.

Para a prestação deste tipo de serviços são necessários vários tipos diferentes de utensílios para auxiliar a atividade. Estes utensílios podem ser produzidos em vários materiais, tendo sempre em conta a necessidade de evitar materiais possivelmente nocivos para os pacientes.

1.1 Contextualização

Devido à necessidade de otimizar a produção em ambiente industrial, recorre-se frequentemente à automatização dos processos. A automatização de um processo pode ser efetuada através de uma automação clássica ou através do uso dos robôs. A utilização de robôs em tarefas repetitivas permite aumentar a eficácia e a eficiência da tarefa através de um investimento, feito com base num estudo que o demonstre positivo, pode tornar lucrativo a médio e longo prazos o investimento.

A produção de utensílios para auxílio médico é um processo repetitivo que implica a execução de um mesmo número de tarefas de forma cíclica. Este tipo de tarefas podem incluir carga e descarga de máquinas, o manuseamento de materiais, montagem, empacotamento e transporte. Grande parte destas tarefas pode levar a um défice de atenção na sua execução por parte do operador devido à sua repetibilidade e monotonia.

Deste modo, quando se justifica, é vulgar automatizar-se este tipo de tarefas libertando o operador para tarefas com um grau de dificuldade superior e um grau de monotonia inferior. Uma vez que a finalidade desta tese é automatizar o processo de manuseamento e montagem de peças, recorre-se à automação e robótica para o desenvolvimento de um sistema que cumpra as tarefas necessárias.

1.2 Objectivos

O objetivo principal deste trabalho é desenvolver um sistema que permita efetuar a montagem de pinças Kocher através do recurso a um robô paralelo, para efetuar o manuseamento das peças e a sua montagem.

Este sistema tem também por objetivos:

- controlo total do sistema através de um autómato programável e com comunicação com um sistema de descarga.
- controlo da produção através dados recebidos através de uma interface com o operador;
- flexibilidade do tipo de peça a montar;
- definição e controlo da peça em produção pelo robô através do PLC;
- autonomia total, seguindo a parametrização feita pelo operador;
- reduzir tempo de ciclo de montagem e embalagem;
- aumentar a produtividade do sistema de montagem;
- controlo da segurança da célula.

1.2.1 Requisitos

O sistema a ser implementado e abordado por esta dissertação será composto por dois robôs, o modelo M2iA-3SL e o modelo M20iA/20M da Fanuc, um autómato programável e uma consola de interface humana da OMRON e uma mesa de montagem mecânica e automatizada. A máquina de injeção de plástico e o robô do modelo M20 correspondem à entrada do sistema a ser implementado por esta tese. Serão também utilizados dois programas de simulação e programação *offline* de robôs, Visual Components e Roboguide, e também o *software* específico da OMRON para a programação do autómato programável. A garra utilizada pelo robô M2iA é da marca Schunk, e possui dois dedos paralelos desenhados na RobotSol especificamente para a tarefa a executar. Por fim, é necessária uma rede sensorial para acompanhar cada movimento dos constituintes da mesa e a abertura e fecho da garra.

1.2.2 Testes funcionais

Os testes a serem executados começarão pela simulação do sistema nos programas de simulação de soluções robóticas e pela simulação do código produzido para controlo da célula via autómato programável. De seguida proceder-se-á a testes de funcionamento da mesa e de cada um dos seus componentes, o designado teste I/O. Posteriormente serão executados testes ao funcionamento da tarefa programada em relação ao robô e de todos os pontos marcados. Para terminar serão executados testes produtivos conjugando o robô e a mesa automática.

1.3 Calendarização

Esta dissertação foi efetuada ao longo de 10 meses. Durante os primeiros dois meses o trabalho efetuado residiu num estudo acerca do sistema e da componente teórica ligada ao mesmo. O estudo teórico foi desenvolvido ao longo de 7 meses nos quais estão incluídos os estudos e correções. Durante este estudo teórico, e com o auxílio do mesmo, foi efetuada a componente prática do projeto na qual se insere a programação do sistema e os seus testes que ocuparam 3 meses da elaboração desta dissertação. Ao longo de todo este período temporal o documento presente foi escrito como base em ambas as componentes. Na Figura 1.1 está ilustrada toda a calendarização da presente dissertação.

1.4 Estrutura da dissertação

Esta dissertação está organizada em oito capítulos. Três desses oito capítulos são referentes a um estudo teórico e mais três desses oito são referentes à implementação prática do trabalho da dissertação.

O Capítulo 1 é a introdução da tese no qual é feito um preâmbulo referindo todos os objetivos da tese, os seus requisitos e os testes que se pretendem fazer. É apresentada uma visão bastante geral do documento em si.

No Capítulo 2 é apresentado o resultado do estudo de conceitos acerca da robótica em geral e da robótica industrial. Neste capítulo são abordados conceitos teóricos diretamente ligados com a definição e constituição de um robô industrial e também os conceitos ligados à sua programação e movimentos.

Segue-se o Capítulo 3 no qual é apresentado o estudo do ambiente industrial no qual podem ser aplicados robôs e automação convencional.

O Capítulo 4 fecha o estado da arte, sendo que neste capítulo é exposto um estudo sobre segurança em ambiente industrial e uma série de casos de estudo nos quais são aplicadas a robótica e a automação industrial.

No Capítulo 5, primeiro da parte prática que constitui esta dissertação, é feita uma exposição da arquitetura do sistema e uma abordagem ao problema

a solucionar. Neste capítulo são também abordados os cenários analisados e os programas utilizados para o desenvolvimento do projeto.

O capítulo que se segue é o 6 no qual estão expostas as diversas fases práticas do projeto bem como o raciocínio inerente a cada uma. Neste capítulo é explicitado todo o algoritmo que controla o sistema e cada um dos seus componentes.

No Capítulo 7 estão expostos os testes efetuados ao sistema bem como os resultados inerentes a esses mesmos testes.

Após todos estes capítulos segue-se o capítulo de conclusões no qual está exposta uma breve análise ao trabalho efetuado e os possíveis melhoramentos.

Para fechar o documento seguem-se os anexos nos quais estão contidos documentos importantes para a concretização do projeto.

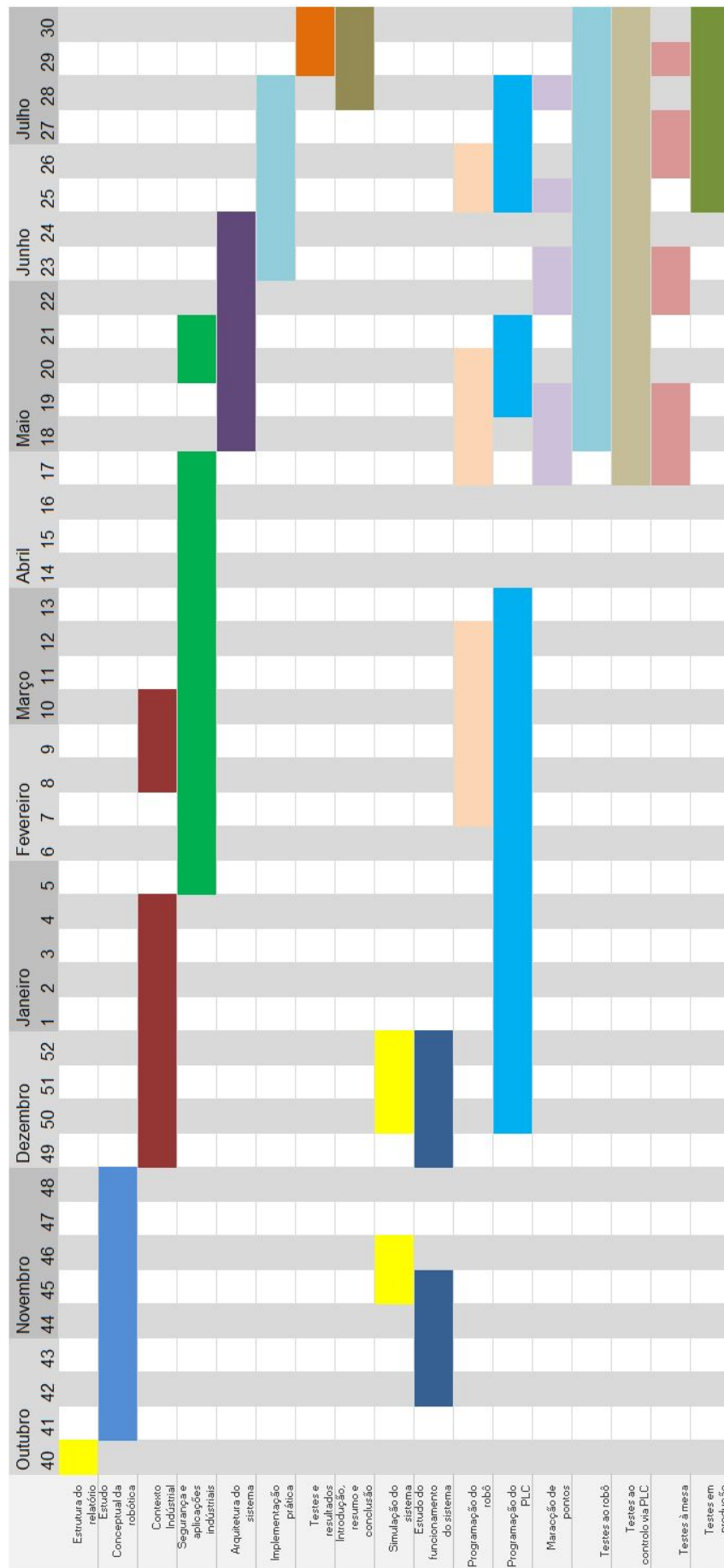


Figura 1.1: Calendarização do projeto.

Capítulo 2

Estudo Conceptual da Robótica

Neste capítulo é apresentada uma visão geral sobre o mundo da robótica com incidência em pormenores necessários para a percepção do conceito. A informação presente neste capítulo tem por base fatores históricos e técnicos. A visão histórica será apresentada cronologicamente e concetualmente. A evolução tecnológica destes dispositivos será também apresentada cronologicamente. Ainda neste capítulo serão também abordados alguns benefícios e contrapartidas da robótica.

2.1 Introdução à robótica

2.1.1 Terminologia e origem do conceito "Robótica"

O termo robô, vulgarmente utilizado nos dias de hoje, tem origem numa peça de teatro do dramaturgo checo Karel Capek. O título da peça é Rossum's Universal Robots (R.U.R) e o tema abordado é a perda da influência humana na sociedade tecnologicamente evoluída. A palavra robô tem origem na palavra checa "robota", enunciada na peça R.U.R, que significa trabalho forçado ou servo. De salientar, que os robôs idealizados por Capek não eram mecânicos, mas sim criados através de processos químicos sendo que ao longo da peça ganham inteligência e revoltam-se contra a raça criadora, a raça humana [10].

Mais tarde a ficção científica abordou a robótica de uma forma mais exaustiva. Em 1941 Isaac Asimov enuncia no livro "Runaround" aquilo que se conhece como as três leis da robótica. As leis enunciadas por Asimov são as seguintes:

- I Um robô não pode fazer nenhum mal a um humano e nem, por inação, permitir que algo de mal lhe aconteça.

- II Um robô deve obedecer às ordens dos humanos, exceto quando contrariarem a primeira lei.
- III Um robô deve proteger a sua integridade física, desde que isso não contrarie as duas leis anteriores.

Mais tarde acabaria por ser introduzida a lei zero que refere que nenhum robô pode fazer mal à humanidade nem permitir que algo de mal lhe aconteça [11]. Desde então o assunto robótica tem fascinado a ficção levando ao aparecimento de grandes êxitos como Star Wars, Wall-E e Westworld.

2.1.2 Definição de robô

A definição do termo robô é um assunto algo controverso e tem como consequência a ausência de uma definição universalmente aceite. Existem várias definições, mais ou menos complexas, que conseguem transpor o conceito base do robô. De forma simples, um robô pode ser definido como uma máquina automática que executa funções atribuídas a seres humanos [12].

De acordo com o dicionário de Oxford um robô possui várias definições [13]. Numa primeira instância, pode ser definido como uma máquina com características humanas e capaz de replicar alguns movimentos e funções humanas de forma automática. Pode ser definido também como uma máquina capaz de executar uma série de ações complexas de forma automática, principalmente as que são programadas através de um computador [13].

De acordo com o dicionário *Priberam*, um robô é um aparelho capaz de agir de maneira automática numa dada função ou um autómato com figura humana [14].

De acordo com a Robotic Industries Association (RIA), um robô industrial é definido como sendo um manipulador universal, reprogramável e controlado de forma automática. Possui a capacidade de ser programável em três ou mais eixos e pode estar ou não fixo num ponto e é utilizado em atividades de cariz industrial [15]. A RIA define a robótica industrial como sendo a aplicação de um dispositivo mecânico programável em detrimento de uma pessoa para executar atividades perigosas ou repetitivas com um elevado grau de precisão [16].

O autor Mikell P. Groover define robô industrial como uma máquina de uso geral e programável possuidora de certas características antropomórficas. Uma das características mais óbvia é o braço mecânico usado para cumprir várias tarefas industriais. Os robôs industriais têm a capacidade de responder a impulsos, comunicar com outras máquinas e tomar decisões através do recurso a sensores [17].

2.1.3 Evolução histórica da robótica

Os primeiros estudos, dos quais se possui conhecimento, no âmbito da robótica datam de finais do século XV e início do século XVI. Esses estudos foram realizados por Leonardo da Vinci e registrados no seu livro *Codex Atlanticus*. Leonardo da Vinci possuía conhecimentos nas mais variadas áreas, nomeadamente nas áreas como o desenho, a anatomia, o projeto, a produção de ferramentas e a fundição. Este conjunto de conhecimentos é necessário à construção de um protótipo de um robô. Apesar deste amplo conhecimento não foi descoberto nenhum protótipo construído por Leonardo da Vinci [18].

Em meados do século XVII, Blaise Pascal produziu avanços na área das máquinas associadas ao cálculo matemático com a criação da Pascaline. Esta máquina foi criada com o intuito de auxiliar nos cálculos de taxas, sendo mais tarde melhorada por Samuel Morland. Samuel Morland reduziu o tamanho da Pascaline e permitiu que a mesma funcionasse "sem carregar a memória, perturbar a mente ou expor as operações a qualquer incerteza" [19].

No século XVIII, os robôs miniatura tornaram-se brinquedos populares para as famílias mais ricas da época. Estes robôs eram construídos com formas humanas ou de animais e executavam cópias dos seus movimentos [19]. Ainda neste século, Jacques de Vaucanson criou uma máquina conhecida como "*The Duck*" (o pato). Esta máquina possuía a capacidade de mexer as asas, comer e digerir grão. Cada asa desta máquina possuía cerca de 400 partes móveis [19].

No início do século XIX, Joseph-Marie Jacquard cria o primeiro tear programável, uma máquina com a capacidade de criar designs para serem impressos em tecido [18, 19]. Ainda neste século, John Brainerd cria o robô cujo nome era "Steam Man" e o objetivo deste robô era de puxar carroças [19, 20]. Anos mais tarde, Frank Reade Jr. construiu o "Electric Man" que era uma versão elétrica do "Steam Man" [19, 21].

No início do século XX há o aparecimento da palavra checa "*Robota*", que significa "trabalhador forçado", na peça R.U.R de Karel Capek. Surgem, poucos anos mais tarde, os robôs Elektro e Sparko no filme *Metrópolis*, tendo Elektro se tornado o primeiro robô celebridade [18]. Neste século surgem os manipuladores mecânicos remotos para materiais radioativos desenvolvidos por Oak Ridge e Argonne National Labs. Neste século também, Isaac Asimov introduz a palavra "robótica", por forma a denominar a ciência vocacionada para os robôs e previu o crescimento de uma poderosa indústria de robôs [10, 18].

O primeiro mecanismo programável de pintura foi criado por Willard Pollard e por Harold Roselund para a DeVilbiss Company, no início dos anos 40. No fim dos anos 40 dá-se o aparecimento do comando numérico, do telecomando e também dos primeiros robôs de W. Grey Walter. Estes robôs denominados

de Elmer e Elsie, também conhecidos de robôs tartaruga, tinham a capacidade de localizar a estação de carregamento quando a sua bateria estava com pouca carga. Em meado dos anos 50 George Devol cria o primeiro robô programável nomeando-o de ULTIMATE para integração na Universal Automation e tenta patentear-lo. Mais tarde George Devol e Joseph Engelberg criam a primeira empresa de robótica denominada de "Unimation". Já no fim dos anos 50 é introduzido pela Planet Corporation o primeiro robô comercial, controlado por sensores de fim de curso e excêntricos [10, 18]. Também nesta altura é enviado para o espaço, pela União Soviética, o primeiro satélite artificial e autónomo, o Sputnik I [19]. No início dos anos 60 é introduzido o primeiro robô industrial com princípios de controlo numérico e atuadores hidráulicos, o Unimate da Unimation, INC. Ainda no início dos anos 60 é criada a primeira mão mecânica, com sensores tácteis, controlada por computador. Em meados desta mesma década é criado, no Massachusetts Institute of Technology (MIT), o primeiro laboratório de investigação de inteligência artificial. Nesta altura é também criado o Robotics Institute na universidade de Carnegie Mellon. No final dos anos 60 é desenvolvido no Stanford Research Institute (SRI) o primeiro robô móvel já com câmara de vídeo e sensores de contacto acoplados [18, 19]. No fim dos anos 60 e início dos anos 70, Victor Scheinman criou o Stanford Arm, que foi o primeiro braço robotizado e controlado por computador com atuação eléctrica. Nesta mesma década foi criada a primeira linguagem de programação de robôs, a linguagem WAVE e em seguida a linguagem AL. Mais tarde as duas linguagens deram origem à linguagem VAL. Em meados desta mesma década são utilizados manipuladores robóticos nas sondas enviadas em missão para Marte. Já nos finais desta década dá-se o aparecimento do Programmable Universal Machine for Assembly (PUMA) e do primeiro robô SCARA, Selective Compliance Assembly Robot Arm, desenvolvido na universidade de *Yamashi* [18, 19]. Na década de 80 dá-se o aparecimento do primeiro robô com atuação direta desenvolvido na Carnegie Mellon University (CMU) e o projeto de uma linha flexível de fabrico com a utilização de robôs. A indústria da robótica cresce graças à indústria automóvel que tinha necessidade de automatizar os seus processos. Nesta década é criado também o primeiro robô aquático que tinha a capacidade de se locomover debaixo de água, criado pela Robotics Laboratory no ministérios dos transportes do Japão [19]. Nos anos 2000 a Honda apresenta o robô humanoide com capacidade de locomoção e de sentidos como visão e tato e ainda a capacidade de comunicar através de sinais de voz. A Sony apresenta o primeiro robô humanoide com a capacidade de correr. São apresentados os braços robóticos modernos com capacidades e desempenho superiores através da evolução de controladores e linguagens de programação, melhores mecanismos de movimento, integração de sensores e sistemas de controlo [18, 19].

2.2 Constituição de um robô industrial

É possível subdividir qualquer sistema em partes por forma a facilitar a sua compreensão. Esta técnica permite ter a percepção da funcionalidade de cada subsistema e da sua interligação para que o sistema global funcione de acordo com o cenário idealizado.

Um robô industrial tem por base na sua constituição a estrutura mecânica, o atuador final, os atuadores, o controlador e a consola de programação. Cada uma destas partes integrantes desempenha uma função específica. São necessárias para manter a capacidade do robô executar as tarefas pré-programadas.

A estrutura mecânica de um robô pode ainda ser dividida em pequenas partes como a base, as juntas, os elos, entre outros.

2.2.1 Elos

Os elos, representados na Figura 2.1, correspondem às suas partes rígidas de um robô. Os elos apresentam um movimento relativo a outros elos ou à base do robô. Do ponto de vista da cinemática, um elo pode ser constituído por dois membros ou mais que estão ligados entre si, mas estes membros não realizam movimento relativo entre eles. Os elos estão conectados a juntas, juntas estas que permitem que os elos realizem movimento entre si [22, 23].

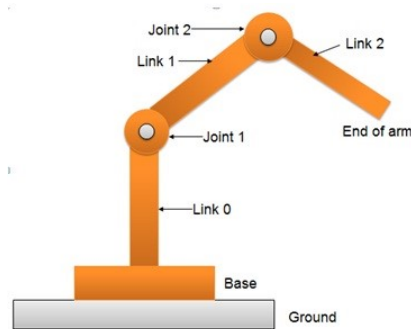


Figura 2.1: Elos e juntas de um robô [24].

Um robô pode ser classificado de acordo com o posicionamento dos seus elos. Se os elos estiverem ligados de forma paralela entre si o robô diz-se manipulador paralelo (como se verifica na Figura 2.2). O robô do tipo paralelo é composto por duas plataformas (uma fixa e uma móvel) ligadas por duas ou mais cadeias cinemáticas. No caso do robô ter os seus elos ligados em série, ou seja, se cada junta só possuir dois elos ligados à mesma e formarem uma só cadeia cinemática, o robô designa-se por manipulador série [25]. Os robôs série caracterizam-se por seguirem uma sequência de elos e juntas que se inicia na

base e termina no atuador final. Estes elos e juntas estão configurados por forma a permitir translação e rotação. No caso dos robôs paralelos a ferramenta está colocada na base móvel do robô [26, 23].

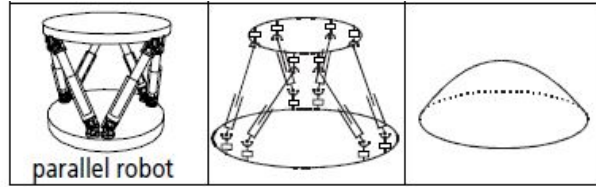


Figura 2.2: Configuração de um robô paralelo [26].

2.2.2 Juntas

As juntas são os constituintes do robô que permitem a ligação de dois ou mais elos por forma a que estes executem movimentos relativos entre si. Os movimentos relativos que os robôs industriais executam, proporcionados pelas juntas, são: os movimentos rotacionais e prismáticos. Em consequência as respectivas juntas denominam-se de rotacionais e prismáticas. Existem vários tipos de juntas, como se observa na Figura 2.3, desde as já referidas, mas ainda as juntas helicoidais, cilíndricas, esféricas, planas e universais. As juntas são o elemento do robô que lhe confere “graus de liberdade” [23].

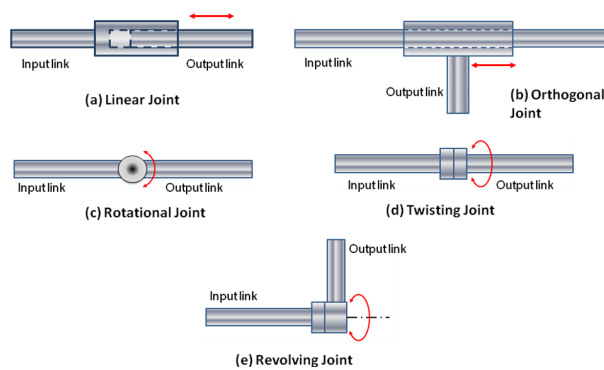


Figura 2.3: Exemplo do tipo de juntas [24].

As juntas do tipo rotacional, vulgarmente conhecidas como juntas do tipo R, como a designação sugere, permitem que dois elos executem um movimento rotacional entre si. Esta rotação é executada em torno do eixo de rotação da própria junta. Este tipo de junta apenas confere um movimento angular dos elos em relação a um eixo definido. Este tipo de juntas são designadas, de forma coloquial, de dobradiça ou pino [23].

As juntas do tipo prismático, vulgarmente conhecidas como juntas P e denominadas de juntas deslizantes, permitem apenas o deslizamento de um elo em relação a outro. Neste tipo de juntas por vezes um dos elos está no interior do outro [23].

As juntas do tipo cilíndrico, juntas do tipo C, são formadas por dois elos que giram entre si e em que um destes se encontra na superfície interior e o outro na superfície exterior. Este tipo de juntas permite que ambos os elos rodem entre si e executem um movimento prismático paralelo ao eixo de rotação [23].

As juntas do tipo esférico, juntas do tipo S, são formadas por duas superfícies esféricas e congruentes. Estas superfícies encontram-se colocadas uma no interior da outra, sendo uma a superfície exterior e a outra superfície interior. Uma das características deste tipo de junta é permitir a rotação em qualquer direção em volta do centro da esfera. Permite uma rotação independente em relação aos três eixos cartesianos e apresenta três graus de liberdade. Este tipo de junta não possui singularidades [23].

As juntas do tipo planar são formadas por superfícies planas de contacto e assim como as juntas do tipo esférico possuem três graus de liberdade. Tal como as juntas do tipo esférico não possui singularidades [23].

Os robôs podem também ser classificados de acordo com os graus de liberdade que as juntas que os constituem lhes conferem. A maioria dos robôs industriais possui seis graus de liberdade, ou seja, seis juntas. Este tipo de classificação é usual nos robôs série.

Nos robôs série as principais configurações são: a configuração cartesiana, a configuração cilíndrica, a configuração esférica, a configuração articulada e a configuração SCARA, como se observa na Figura 2.4 [26, 23].

Os robôs cartesianos possuem três juntas prismáticas, cujos eixos são coincidentes com o sistema de coordenadas cartesiano. É usual que a montagem dos robôs cartesianos seja feita sob a forma de pórtico. Estes manipuladores são frequentemente utilizados para tarefas como paletização, armazenagem e *picking* de encomendas ou em tarefas especiais como o corte a laser e a jato de água. Grande parte destes robôs seguem uma estrutura de sistema modular [26, 23].

Os robôs cilíndricos possuem um braço composto por duas juntas prismáticas e uma rotacional. Os robôs esféricos possuem um braço composto por duas juntas rotacionais e uma junta prismática. O braço de um robô cilíndrico forma um sistema de coordenadas cilíndrico, por outro lado o braço de um robô esférico forma uma eixo de coordenadas esféricas. Estes robôs são utilizados para paletização, carga e descarga de máquinas [26].

Os robôs SCARA representam uma subclasse dos robôs cilíndricos e consistem em robôs com duas juntas rotacionais paralelas que conferem a capacidade


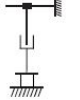
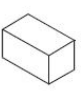



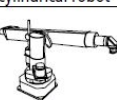


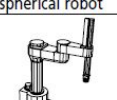
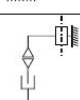

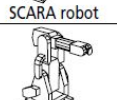
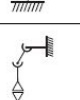
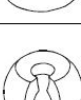
Robot	Axes		Wrist (DOF)		
	Principle	Kinematic Chain			
 cartesian robot			1	1	2
			2	3	3
 cylindrical robot			1	1	2
			2	3	
 spherical robot			1	2	3
			3	3	3
 SCARA robot			1	2	2
			2		
 articulated robot			2	3	3
			3	3	3

Figura 2.4: Configurações dos robôs com estrutura série [26].

de executar um movimento rotativo no plano horizontal com um movimento linear vertical. Este tipo de robôs é rígido na direção vertical porém bastante móvel no plano horizontal do espaço de trabalho, o que facilita em tarefas de montagem, pois são normalmente mais rápidos e precisos que manipuladores de seis eixos. [26, 23].

Os robôs articulados são os robôs cuja configuração é a mais comum e são constituídos por, pelo menos, três juntas rotacionais. Estes robôs normalmente possuem um longo alcance e elevadas capacidades de carga. Em alguns casos contrabalançam a sua estrutura com pesos [26, 23].

2.2.3 Manipulador

Entende-se por manipulador, presente na Figura 2.5, o conjunto do corpo do robô que inclui os elos, as juntas e todos os outros elementos estruturais. O manipulador torna-se um robô quando a este são adicionados o atuador final e o sistema de controlo. Na literatura, por vezes ambos os termos são utilizados para referir um robô [22]. Um dos manipuladores mais frequentemente utilizado atualmente é braço robótico, cuja constituição é baseada no braço humano.

O braço do manipulador está associado ao posicionamento do robô no seu espaço físico [25].



Figura 2.5: Manipulador robótico FANUC M20iA [27].

2.2.4 Punho

O punho, como se observa na Figura 2.6, consiste nas juntas da cadeia cinemática do robô que se encontram localizadas entre o atuador final e o braço robótico [22].

Normalmente são utilizadas juntas esféricas na concepção do punho do robô para aumentar a mobilidade do atuador final.

O punho tem como função orientar o atuador final para a execução de determinada função. Este controla essencialmente a orientação angular do atuador final [25].

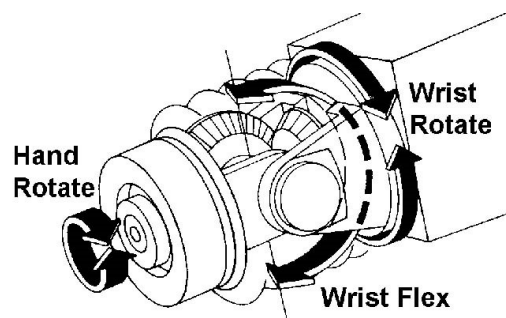


Figura 2.6: Punho de um braço robótico [28].

2.2.5 Atuadores

Os atuadores funcionam como os “músculos” dos robôs contribuindo para a alteração da posição, ou seja, para que se movimentem. Estes permitem que a estrutura mecânica contrarie forças como a gravidade, inércia e outras forças caracterizadas por serem contra o movimento. Os atuadores podem ser de vários tipos, desde hidráulicos, pneumáticos e até elétricos e precisam de ser controláveis. A maior parte dos atuadores presentes no mercado são modificáveis e adaptáveis, quando necessário, à aplicação em específico [23].

Os atuadores hidráulicos foram inicialmente escolhidos como fontes de energia para os primeiros robôs industriais. Ofereciam a possibilidade de se exercerem forças elevadas e proporcionavam uma maior capacidade de carga. Num sistema hidráulico a energia é fornecida mecanicamente através de um motor elétrico ou outro tipo de motor responsável por controlar o bombeamento do fluido a altas pressões. Os atuadores hidráulicos mais comuns são os cilindros hidráulicos, atuadores de palhetas rotativas e motores hidráulicos. O controlo destes atuadores é executado por válvulas indutivas ou servo-válvulas que são controladas eletricamente por circuitos eletrónicos de controlo. A fonte de energia para um sistema hidráulico é de elevadas dimensões, bem como os seus custos. A sua manutenção e perigo de fugas limitou o uso deste tipo de atuadores [26, 23].

Os atuadores pneumáticos são usados principalmente em manipuladores mais simples. Estes atuadores fornecem um movimento incontrolável do ponto de vista de uma paragem mecânica. Este tipo de atuadores são de simples utilização e de baixo custo, o que permite que alguns atuadores de pequenas dimensões possam funcionar com os sistemas de ar já existentes nas fábricas. Se não for possível a utilização dos sistemas de ar previamente existentes, devido à utilização de uma elevada gama de robôs pneumáticos ou outras razões que determinem a incapacidade do sistema de ar existente suprir as necessidades, é necessário proceder à instalação de fontes de ar comprimido para melhorar o sistema existente. Este tipo de atuadores do ponto de vista energético é pouco eficiente. Revelam-se úteis para utilizar em ambientes em que as condições de segurança e da própria aplicação colocam de parte os elétricos.

Os atuadores elétricos são os mais utilizados nos dias de hoje. Estes atuadores podem classificar-se de acordo com o tipo de motor utilizado.

No caso dos atuadores com motor de passo a sua utilização é remetida para os robôs mais simples e de pequenas dimensões, como robôs dispensadores de etiquetas. Neste tipo de motores é utilizado um controlo em malha aberta de velocidade e posição. Motores com a capacidade de micro passo permitem mais de 10000 posições de juntas do robô. Por outro lado, em malha aberta estes

motores conferem ao robô tempos de estabelecimento significativos que podem complicar a aplicação.

Se forem utilizados motores de corrente contínua de ímanes permanentes, o preço desse tipo de atuadores é baixo. Este tipo de atuadores é utilizado em brinquedos robóticos ou robôs de passatempo. Este tipo de motores tem como vantagens a sua baixa indutância, baixo atrito e a ausência de binário parasita nas engrenagens. Apresentam como desvantagem as perdas elevadas de energia por efeito de Joule quando são exigidos elevados binários.

Podem ainda ser utilizados motores sem escovas que podem ser servo motores de corrente alternada (AC) ou motores sem escovas de corrente contínua (DC). Devido à baixa complexidade construtiva este tipo de motores conseguem um bom desempenho a baixo custo [26, 23].

2.2.6 Atuadores finais

Estes dispositivos são ligados ao punho do robô e são responsáveis por executar a tarefa pretendida. Devido à vasta variedade de tarefas existem vários atuadores finais por forma a que cada tarefa possua o atuador final mais apropriado. É possível dividir-se estes atuadores finais em dois grupos: garras e ferramentas [17].

As garras são atuadores finais com o objetivo de pegar e manipular objetos, movendo-os dentro da área de trabalho, durante o ciclo de trabalho. Um exemplo deste tipo de atividade são a carga e descarga de uma máquina. As garras têm formas e tamanhos diferentes dependendo do objeto a mover e, consequentemente, pesos diferentes. Existem vários tipos de garras: garras mecânicas, garras a vácuo, garras magnéticas, garras adesivas e garras de mecânica simples. As garras mecânicas são formadas por dois ou mais "dedos" por forma a pegar no objeto. Este tipo de garra é o mais comum. A garra a vácuo possui ventosas que fazem a sucção do objeto para transporte do mesmo. As garras magnéticas baseiam-se no efeito eletromagnético e são utilizadas para mobilização de objetos ferrosos. As garras adesivas têm uma superfície adesiva cuja finalidade é pegar em materiais flexíveis. Por fim, as garras mecânicas simples correspondem a ganchos ou colheres [17].

As ferramentas são de uso comum em operações de processamento ou maquinação de uma matéria prima ou objeto. O robô tem a tarefa de mover a ferramenta relativamente a um objeto parado ou em movimento lento, ou ainda, mover o objeto relativamente a uma ferramenta parada. As ferramentas mais comuns são: pinças de soldadura por pontos, tochas para soldadura por arco elétrico, brocas de furação, ferramentas de montagem, tochas de aquecimento e pistolas de corte a jato de água. Em cada um dos casos o robô não tem só a função de controlar a posição relativa da ferramenta mas também de controlar a

própria ferramenta em si por forma a executar a tarefa. Para que tal seja possível o robô envia sinais de controlo à ferramenta. É possível utilizarem-se várias ferramentas por ciclo de trabalho e que o robô alterne entre as mesmas [17].

2.2.7 Sensores

Controlar um sistema robótico seria simples caso os seus atuadores executassem comandos de movimento de forma perfeita, bem como o modelo do espaço ocupado pelo robô fosse ideal; porém, na realidade isto não é possível e para que o robô possua toda a informação para operar de forma ideal compensa-se esta lacuna com o uso de sensores. Os sensores têm como função transmitir informação complementar àquela que o robô já possui. O processo de percepção tem como entradas a informação proveniente dos sensores e a informação proveniente do modelo do espaço físico do robô. Os sensores podem estar conetados ao próprio robô ou a uma unidade de controlo, por exemplo um autómato [23].

Os sensores são responsáveis pela atividade de detetar e recolher informação acerca do próprio sistema robótico e do ambiente que o envolve. Na robótica industrial a posição das juntas, a velocidade, a aceleração e a força são as variáveis mais importantes a serem medidas através dos sensores. Desta forma, os sensores são responsáveis por enviar ao controlador informações que após analisadas por este irão determinar a configuração do manipulador[22]. Os sensores podem ser classificados de diferentes formas. Na área da robótica é comum classificarem-se os sensores de acordo com o tipo de dispositivo e objetivo de medição. É possível classificar-se os sensores em:

- Sensores de fim de curso mecânico: este tipo de sensores emite um sinal cada vez que o robô atinge um determinado ponto. Frequentemente utilizados para impedir que o robô execute movimentos nas suas juntas que lhe causem dano próprio;
- sensores fotoelétricos: sensores cujo modo de funcionamento tem por base o uso de luz. Existe um emissor de feixe de luz e um recetor; este sensor tem a capacidade de detetar a interrupção do feixe proveniente do emissor e envia a informação relativa a essa interrupção. Existem recetores com a capacidade de ler feixes de luz de diferentes fontes. Outros, porém, têm a capacidade de ler a reflexão de um feixe de luz emitido pelo mesmo [26];
- sensores de proximidade: tiram vantagem das propriedades do material que permitam que este altere o seu campo elétrico ou magnético [26];
- sensores de força e binário: sensores para medida de forças e binários no robô ou aplicados pelo robô. Estes sensores baseiam-se no princípio da deflexão mecânica de um material sólido, princípio esse descrito pela lei

de *Hooke*, que enuncia que a deflexão dos materiais sólidos é mensurável e diretamente proporcional à força exercida. Este tipo de sensores é mais utilizado em operações de montagem [26];

- sensores táteis: estes sensores são compostos por vários sensores de força cuja informação proveniente dos mesmos é guardada num vetor. Estes sensores geram um conjunto de informação sob a forma binária por forma a gerar um vetor bi-dimensional. Através da análise de todos os pontos de dados e da sua disposição no vetor permite retirar informação acerca do tamanho, forma e orientação de um objeto seguro pela garra do robô [26];
- sensores ultrassónicos: sensores cujo funcionamento se baseia na emissão e receção de uma onda sonora que é refletida numa superfície, ou seja, a informação é obtida através do tempo de voo da onda sonora [26];

2.2.8 Controlador

Um controlador é constituído por: uma interface com o utilizador, a zona de memória programável e controlável e o interpretador do programa. Esta constituição pode ser dividida em blocos, como se verifica na Figura 2.7. O bloco de interface transmite a informação proveniente do utilizador/programador à memória por forma a ser analisada e executada. O bloco de controlo das entradas e saídas, do inglês *inputs* e *outputs* (I/O), interliga o programa produzido pelo programador e os dispositivos externos. O bloco de controlo dos sensores é responsável pela interface entre os sensores e a unidade responsável pela execução das instruções do programa. O bloco de controlo de movimento pode ser interligado ao bloco de controlo dos sensores, sendo que faz a interface entre o robô e o programa. Os blocos de execução do programa e de controlo da memória têm a capacidade de ligação a um bloco de comunicação em rede para a transferência de informação [26].

O controlador do robô possui como unidade de processamento um microprocessador. O microprocessador pode ser programado com as instruções pretendidas para a descrição, execução e controlo das operações do robô. O controlador de robô é responsável pelo acionamento dos motores de cada junta. Pode conter amplificadores de sinal para auxílio no acionamento de atuadores e motores, de acordo com os comandos programados.

O controlador possui também a tarefa de gerir as coordenadas de movimento por forma a controlar o *tool center point* (TCP) que se localiza no fim do braço robótico. Gere também os sinais de entrada e saída. Estes sinais podem ser digitais ou analógicos. Com base nestes sinais controla os dispositivos externos, como por exemplo, a ferramenta a utilizar, e tem por base uma sequência sincronizada com o robô.

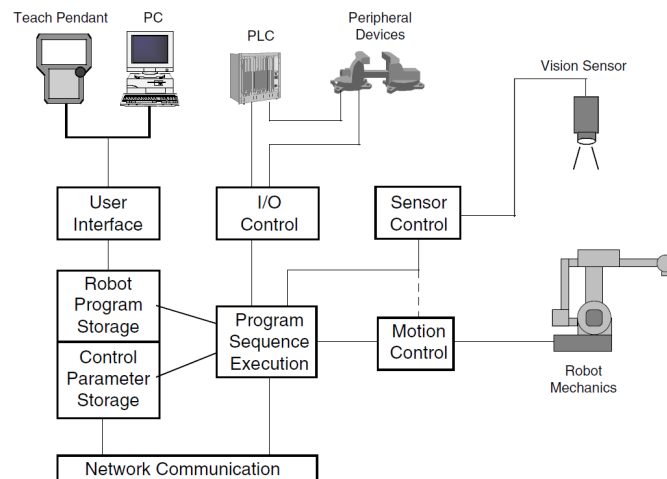


Figura 2.7: Blocos de funcionamento de um controlador [26].

O controlador tem ainda a capacidade de comunicar com outros controladores, por exemplo um computador, por forma a obter ainda mais informação ou enviar informação para ser posteriormente analisada [26].

Estas unidades de controlo, exemplificadas na Figura 2.8, assumem três papéis específicos:

1. papel informativo: consiste na leitura e processamento da informação proveniente dos sensores [22];
2. papel decisivo: consiste no conjunto de ordens de movimentos no plano geométrico enviadas para o robô para que este se movimente [22];
3. papel comunicativo: consiste na organização das transferências de informação entre o robô e o seu ambiente envolvente [22];

As trajetórias executadas pelos robôs industriais são controladas pelo controlador do robô de forma concordante com o programa a executar. Este permite que o robô execute uma determinada trajetória do TCP no espaço, pré-programada através da gravação de pontos discretos.

O programa tem na sua génese dois tipos de instruções: instruções de movimento e instruções de não movimento.

As instruções de movimento são responsáveis pelo movimento dos eixos do robô que, por consequência, são responsáveis pelo movimento do mesmo ao longo de uma determinada trajetória.



Figura 2.8: Linha de controladores de robôs Kuka [29].

As instruções de não movimento são responsáveis pelo controlo dos sinais de I/O e por executar uma sequência de instruções no programa do robô.

2.2.9 *Teach pendant*

O *teach pendant*, presente na Figura 2.9, é um dos constituintes do controlador e corresponde a uma consola de programação que permite programar diretamente o controlador ou enviar instruções para o mesmo, permitindo assim uma interface entre o utilizador, ou programador, e o controlador. O *teach pendant* permite programar diferentes parâmetros.



Figura 2.9: *Teach Pendant* da FANUC [30].

2.3 Conceitos básicos da robótica

Como em qualquer área, a robótica possui conceitos básicos, terminologias aceites por forma a facilitar a comunicação dos integrantes da área. Principalmente a nível industrial, é necessário manter uma base de conceitos fixa para

facilitar a comunicação e a implementação deste tipo de sistemas, sem erros e em todas as fases do projeto.

Volume de trabalho

O volume de trabalho, também conhecido por *workspace*, corresponde a todos os pontos que o robô consegue alcançar, ou seja, representa todas as posições que o robô alcança no espaço tridimensional [23, 25].

Espaço morto

O espaço morto corresponde aos pontos que o robô não consegue alcançar e é possível ser programado através da limitação do movimento das juntas [23, 25].

Alcance

O alcance corresponde ao ponto mais longínquo que a garra ou ferramenta do robô alcança no espaço, em relação à base [23, 25].

Graus de liberdade (*degrees of freedom, DOF*)

Número de movimentos independentes que o manipulador é capaz de executar no total [23, 25].

Graus de mobilidade

O número de graus de mobilidade é igual ao número de juntas do manipulador [23, 25].

Velocidade de operação

Velocidade à qual o manipulador executa a tarefa pré-determinada [23, 25].

Capacidade de carga

A capacidade de carga corresponde ao valor máximo de carga que o manipulador pode transportar sem que se perca precisão [23, 25].

Resolução

Corresponde ao incremento mínimo de distância que o manipulador pode efetuar, que corresponde ao menor movimento incremental de uma junta [25].

Precisão

A precisão de um manipulador corresponde à capacidade que este tem de se colocar num ponto pré-definido no volume de trabalho. A precisão é tanto maior quanto menor for a discrepância entre o ponto efetivo e o programado [23, 25].

Repetibilidade

A repetibilidade caracteriza-se pela capacidade do manipulador recolocar-se num ponto atingido anteriormente. É tanto maior quanto menor for a discrepância entre o conjunto de pontos efetivos e o ponto desejado [23, 25].

2.4 Algoritmos de controlo

A atuação das juntas de forma individual deve ser controlada coordenadamente por forma a que o manipulador execute o movimento pretendido. Cada uma das juntas tem o seu próprio sistema de controlo de realimentação. O controlador coordena a combinação de atuação das juntas de acordo com o programa produzido. Diferentes aplicações exigem métodos de controlo diferentes. Os algoritmos na base dos mesmos podem ser classificados em quatro categorias: controlo de sequência limitada, controlo de reprodução ponto a ponto, controlo de reprodução de trajetória contínua e controlo inteligente.

2.4.1 Controlo de sequência limitada

O controlo de sequência limitada é o controlo mais básico e utilizado para ciclos de movimento que sejam bastante simples, como por exemplo o *pick-and-place*. Normalmente é implementado através da utilização de sensores de fim de curso para cada junta e com atuação sequencial. A malha de realimentação indica normalmente que a atuação de uma junta ocorreu sem problema, para que se possa executar o passo que se segue na sequência. Contudo, não existe controlo a nível dos servo motores para que o posicionamento da junta seja preciso.

2.4.2 Controlo de reprodução ponto a ponto

O controlo de reprodução ponto a ponto está presente em robôs mais evoluídos que os que apresentam um controlo de sequência limitada. No controlo de reprodução existe uma memória que guarda a sequência de movimentos do ciclo de trabalho, bem como as coordenadas, entre outros parâmetros associados a cada movimento. No controlo ponto a ponto cada posição individual do robô é guardada em memória. Estas posições não são limitadas pelos fins de curso mecânicos de cada junta como no controlo de sequência limitada. Em oposição, cada ponto do programa representa uma localização ao alcance de cada junta do robô. Por cada ponto definido, as juntas são atuadas por forma a atingir esse ponto, colocando as mesmas numa posição específica que o permita. A malha de realimentação é utilizada durante o movimento para confirmar que as posições de cada junta estão de acordo com o programa [17].

2.4.3 Controlo de reprodução de trajetória contínua

Os robôs com controlo de trajetória contínua têm uma capacidade de reprodução semelhante aos de controlo ponto a ponto. Estes robôs têm a capacidade de executar o controlo de dois ou mais eixos de forma contínua. Um manipulador com este tipo de controlo possui pelo menos uma, ou ambas, as vantagens: grande capacidade de memória e cálculo de interpolações. Quando possui uma

grande capacidade de memória, o manipulador possui a capacidade de memorizar um maior número de pontos que constituirão o ciclo de movimento. Estes pontos podem ser gravados muito próximos uns dos outros por forma a permitir que o robô execute um movimento contínuo mais suave. No controlo contínuo toda a trajetória é controlada ao longo do movimento, em contraste ao que acontece no controlo ponto a ponto, na qual apenas o ponto de chegada é controlado. No cálculo de interpolações o controlador programa a trajetória entre os pontos inicial e final através do uso de rotinas de interpolação, quer linear, quer circular [17].

2.4.4 Controlo inteligente

Um robô com controlo inteligente tem a capacidade de interagir com o seu ambiente envolvente, tomar decisões quando algo inesperado ou errado acontece durante o ciclo, comunicar com o operador, executar tarefas computacionais durante o ciclo de movimento e responder a sensores em avanço. Estes robôs usam os controlos ponto a ponto e contínuo para controlar a sua capacidade de reprodução do programa. Este tipo de controlo exige um nível computacional de controlo elevado e uma linguagem de programação avançada [17].

2.5 Métodos de programação de um robô

Com o aumento da produção, os sistemas robóticos tornaram-se parte importante da indústria, uma vez que estes sistemas apresentam uma grande versatilidade em termos de operações que podem executar. Sendo um robô um mecanismo automático com a função de movimentar objetos ou ferramentas segundo uma trajetória definida e executar um conjunto de operações, torna-se necessário enviar esta informação ao robô. A forma de transferir ao robô a informação sobre a tarefa a executar é a programação. Cria-se um programa que corresponde a um conjunto de ações a executar. Existem duas formas de programação de manipuladores robóticos: programação *online* e a programação *offline*, que serão abordadas nesta secção [26].

2.5.1 Programação *online*

Os métodos de programação *online* são métodos de programação de "ensino", ou seja, programa-se diretamente no controlador a tarefa desejada através da "mímica", de movimentos e posições.

Para se proceder a essa programação, o robô é movido manualmente ao longo do conjunto de pontos pretendido, por forma a executar os movimentos desejados. O processo de programação envolve as etapas de ensino, edição e reprodução da trajetória pretendida. A parte do ensino corresponde à etapa em que o programador posiciona em pontos específicos o robô e os grava, por

forma a criar uma trajetória. A etapa de edição é utilizada para acrescentar informação ao programa de controlo do movimento, por forma a que o manipulador execute de forma automática o programa criado. Permite ainda corrigir ou modificar um programa já existente. A fase de reprodução permite verificar a trajetória programada, quer seja no final da programação, quer seja ao longo da própria programação [26].

Existem várias formas de programar *online*. Uma das formas de programação é através do *teach pendant*, também conhecido como *leadthrough* automático [26]. Desta forma, é possível programarem-se os pontos num referencial de coordenadas através da movimentação do robô ao longo desses pontos e da gravação dos mesmos. Existe ainda a capacidade de se alterarem os dados do funcionamento do robô, como a velocidade, e todos estes dados podem ser inseridos através da consola. É possível efetuar várias programações em diferentes sistemas de eixos coordenados (retangular, esférico e cilíndrico), quer sejam em relação ao mundo do robô, quer sejam em relação à ferramenta. Ainda é possível programar a interface com sinais externos que controlarão o robô [26].

Outro método de programação *online* é o método de *leadthrough* manual que consiste num processo de ensino da máquina por parte de um operador. Este faz o robô deslocar-se sobre a trajetória pretendida através do uso de um *teach handle* colocado diretamente no braço do robô. Através deste *teach handle* é possível gravar a trajetória pretendida, bem como a velocidade relativa entre pontos. Enquanto o operador efetua o ensino da trajetória os transdutores medem os ângulos das juntas associados a cada eixo do robô; essa informação é gravada pelo controlador para posteriormente ser executada. Este tipo de programação é mais simples que a programação através do *teach pendant*. É utilizada na programação de tarefas cuja trajetória não é regular, nem é necessária muita precisão na mesma, como é exemplo, a pintura. As dimensões do robô não podem ser elevadas, bem como o seu peso, uma vez que é o operador que tem de o conduzir ao longo da trajetória pretendida, como se observa na Figura 2.10 [26, 17].



Figura 2.10: *Leadthrough* manual [31].

Estes métodos têm como desvantagem o facto de ser necessário parar a linha de produção para programar a trajetória. A principal vantagem resume-se à simplicidade de aprendizagem destes métodos [32].

2.5.2 Programação *offline*

A programação *offline* de um robô corresponde ao desenvolvimento de programas, de forma parcial ou completa, sem a necessidade de recorrer ao manipulador. Isto inclui a geração dos pontos coordenados, das funções e do ciclo de execução. Este tipo de programação tem como vantagens a redução dos tempos de paragem do robô, ou seja, reduzir os tempos em que o robô não está em produção. Permite também reduzir os acidentes de comportamento imprevisível por parte do robô durante a programação *online*. Esta programação confere a possibilidade de programar uma enorme variedade de robôs sem a necessidade de conhecer ao pormenor o controlador de cada um. Permite também a integração com sistemas de *Computer Aided Design/ Computer Aided Manufacturing* (CAD/CAM). Ainda permite a simplificação e verificação das tarefas a executar [26, 32]. Estes métodos têm como desvantagem algumas discrepâncias entre o modelo utilizado e a realidade. Estas discrepâncias podem ser, por exemplo, erros de medição.[32].

A programação *offline* possui dois formatos: programação através da simulação do cenário e programação através de linguagens de programação [26, 32].

A programação através da simulação do cenário segue um algoritmo que se baseia em três etapas. A primeira etapa é a da criação do modelo do cenário, segue-se a segunda etapa que é a programação do robô propriamente dita e a terceira corresponde à passagem do programa para o mundo real [26, 32].

A programação através de linguagens de programação é também conhecida como programação *offline* textual. Consiste na criação de um documento de texto com as operações a executar, que será posteriormente traduzido pelo compilador e interpretado pelo controlador. Existem diversas linguagens de programação de robôs, como é o caso das linguagens RAPID da ABB e Karel da FANUC. Este método apresenta algumas limitações. É difícil a especificação dos pontos de movimento sem a presença do robô, uma vez que é difícil ao operador conhecer com precisão as coordenadas dos pontos para os quais o robô se deve movimentar. Para contrariar este facto o usual é criar o programa em texto e no fim fazer uma memorização de todos os pontos necessários através de um método de programação *online* [17, 26, 32].

A programação através da simulação de cenário é usualmente executada através de *software* provido pelos próprios fabricantes, como se verifica no exemplo da Figura 2.11 em que o *software* roboguide simula a célula e a consola presente no controlador. Neste tipo de programação simula-se uma célula, o mais fiel possível à real, por forma a simular os movimentos e ações que o robô irá executar. Em ambiente de simulação é também possível testar o programa escrito textualmente, uma vez que o ambiente de simulação é uma cópia fiel do ambiente de implementação. Através da simulação é possível perceber erros de programação, como erros nos movimentos do robô, bem como erros no próprio código desenvolvido. Este tipo de programação permite que se desenvolva todo o *layout* referente à tarefa a executar, e toda a programação do manipulador, sem ser necessário uma ligação ao robô, ou retirar o robô da atividade que esteja a executar. Na prática são necessários alguns ajustes de calibração para que o robô execute na realidade a tarefa pretendida [17, 26, 32].

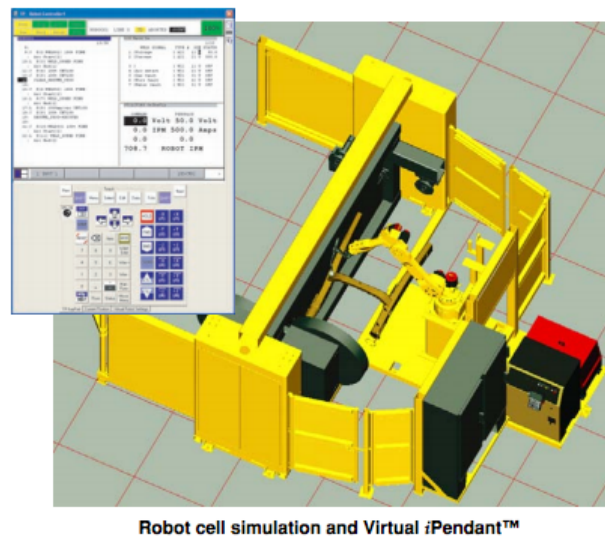


Figura 2.11: Exemplo de simulação [33].

2.6 Conclusão

Neste capítulo foram abordados conceitos teóricos básicos da robótica e também a constituição de um robô. Conceitos, como juntas e elos por exemplo, que são necessários conhecer para proceder ao estudo e implementação de um sistema robotizado. Neste capítulo também foram abordados os conceitos de atuadores e atuadores finais que variam de acordo com a operação a exercer como é apresentado no capítulo seguinte. Também foram abordados métodos de programação cujas vantagens variam também com a operação a ser executada pelo sistema robótico.

Capítulo 3

Contexto Industrial

Neste capítulo é apresentada uma breve descrição das aplicações mais comuns de robôs, uma abordagem à segurança necessária na sua utilização e sua metodologia de implementação. É apresentada também uma referência aos autômatos programáveis, suas funcionalidades e aplicações. Também é efetuada uma abordagem ao ambiente industrial, interligação entre robôs, autômatos e os restantes integrantes de uma célula de fabrico.

3.1 Aplicações industriais de robôs

Uma das primeiras aplicações de robôs em ambiente industrial data do ano de 1961, implementada numa operação de fundição. O robô tinha a função de descarregar uma máquina de fundição. Isto devia-se ao facto do ambiente de uma célula de fundição ser nocivo para os colaboradores. Neste momento são usados em todo o tipo de indústrias, por exemplo, na automóvel, como ilustra a aplicação presente na Figura 3.1

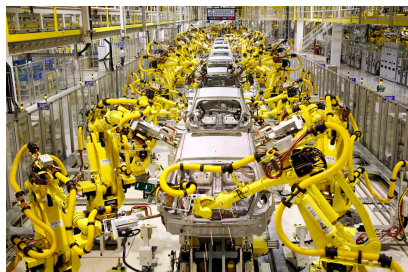


Figura 3.1: Exemplo de uma aplicação industrial: indústria automóvel [34]

Os robôs são utilizados em detrimento de colaboradores humanos por várias razões, de entre as quais [17]:

1. ambiente de trabalho perigoso ou nocivo para a saúde

Quando o ambiente é inseguro, perigoso para a saúde, desconfortável ou de alguma forma desagradável para o operador colocam-se, normalmente, robôs a laborar nesses ambientes em detrimento de colaboradores humanos.

2. ciclos de trabalho repetitivos

O facto de num ciclo de trabalho se verificar um conjunto de tarefas semelhantes e repetitivas, estas podem ser executadas por um robô. O facto de um operador executar sempre o mesmo conjunto de tarefas é nocivo para o mesmo, uma vez que estagna a sua capacidade de aprendizagem laboral. Outra razão é o facto de não se conseguir retirar toda a capacidade potencial do colaborador quando este é obrigado a executar tarefas repetitivas. Deste modo, em tarefas deste tipo tende-se a robotizar.

3. difícil manipulação para o colaborador

Tarefas cujas partes sejam de difícil manipulação para o operador, devido ao seu peso ou outras razões, são, por norma, robotizadas. Desta forma simplifica-se a tarefa e são reduzidas as probabilidades de acidente por incapacidade do operador ou por extremo esforço por parte do mesmo.

4. operações multi-turno

No caso de uma tarefa necessitar de dois ou mais turnos tende a ser repetitiva. Deste modo, a aplicação de um robô evita que um operador efetue esse tipo de tarefa e reduz os gastos da empresa com colaboradores. Esta redução de gastos deve-se ao facto do robô efetuar a tarefa em detrimento do colaborador e do turno.

5. mudanças de local de trabalho pouco frequentes.

A mudança na área de trabalho causa tempos improdutivos. Efetuar a transição definitiva de um robô é um processo demorado o que tem como consequência tempos improdutivos. Deste modo, para trabalhos estacionários no espaço é benéfico o uso de robôs articulados, em oposição a trabalhos que exijam mobilidade, ao longo de um amplo espaço de trabalho. Em trabalhos cuja mobilidade seja necessária pode ser benéfico o recurso a veículos automatizados guiados, do inglês *automated guided vehicle* (AGV) ou soluções que permitam a mobilidade de um robô articulado.

6. quando a posição do componente e a sua orientação são pré-definidas em relação à célula de trabalho

A maioria dos robôs não possui sistemas de visão artificial embutidos. Devido a este facto, os robôs efetuam tarefas nas quais estejam bem definidos os pontos de ação. É necessário definir bem a posição e orientação dos pontos de trabalho e do objeto. Desta forma consegue-se colocar sempre no mesmo ponto uma determinada peça, ou executar determinada tarefa ao longo de um mesmo conjunto de pontos.

Perceciona-se que a utilização de robôs nas mais variadas tarefas tem por finalidade auxiliar e proteger o ser humano. O robô atinge esta finalidade substituindo o ser humano quando há elevado risco associado a determinada tarefa e permitindo que este execute tarefas de maior grau de complexidade.

3.1.1 Operações de manuseamento de materiais

Operações de manuseamento de materiais, mais conhecidas por *pick and place*, correspondem a aplicações cuja finalidade é mover um determinado objeto de um ponto inicial para um ponto final, sendo este o material certo e sendo transportado para o local correto, com a orientação certa no momento temporal correto. Para executar este tipo de operação o atuador final escolhido para o robô é a garra. Esta deve ser pensada, desenhada e construída para mover a peça específica da operação. O uso de robôs neste tipo de tarefas permite manipular cargas elevadas sem operadores, aumentar a segurança entre máquinas e colaboradores e ainda permite repetir a tarefa de forma contínua. Este tipo de aplicações podem ser divididas em três categorias: transferência de materiais, carga/descarga de máquinas e paletização. É necessário ter em atenção o volume de trabalho do robô, capacidade de carga, repetibilidade, precisão, graus de liberdade e velocidade de operação, antes de designar o robô para cada uma destas aplicações [17, 26, 35].

Nas aplicações de transferência de materiais o objetivo principal reside, como referido anteriormente, em mover um objeto no espaço (como se verifica na Figura 3.2), entre pontos definidos. Algumas vezes é necessário reorientar a peça antes de a recolocar. Neste tipo de aplicações não são necessários robôs de tecnologia muito avançada. Existem exceções, nomeadamente quando o robô necessita de fazer o seguimento de uma peça num tapete [17, 35].

Nas aplicações de carga/descarga de máquinas um robô efetua o transporte das peças, quer seja para o interior da máquina, quer seja para o exterior da mesma. Neste tipo de aplicações existem três casos possíveis. Dois dos casos são aqueles nos quais o robô é responsável apenas pela carga ou descarga da máquina, no terceiro caso o robô é responsável por ambas as tarefas. Sendo uma tarefa monótona, por vezes em ambientes prejudiciais e de elevado esforço físico, é benéfico ser um robô a fazê-lo [17, 35].

Nas aplicações de paletização o objetivo é empilhar caixas numa palete que previamente foram lotadas com o material produzido. Esta tarefa é complexa pois o robô deve ser capaz de transportar as caixas e de as colocar em múltiplas posições na palete [17, 35].

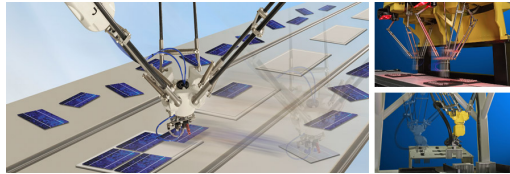


Figura 3.2: Exemplo de uma aplicação de transferência de materiais [36].

3.1.2 Operações de montagem e inspeção

Não existe uma diferença perceptível entre operações de montagem e manuseamento. As atividades de montagem e inspeção são formas híbridas do manuseamento de materiais. A montagem e inspeção podem implicar tanto o manuseamento de materiais, como o manuseamento da garra. A montagem consiste no acréscimo de partes a um componente por forma a construir um elemento mais complexo. Este tipo de atividades são intensas para os colaboradores pois são bastante repetitivas e cansativas. A inspeção requer grande precisão e paciência para perceber se o produto possui as especificações [17, 35].

Estas operações são robotizadas devido ao seu impacto financeiro. A utilização de robôs em aplicações de montagem torna-se apelativa quando é preciso montar peças diferentes, nomeadamente de dimensões diferentes. Também é apelativo quando numa mesma célula são produzidos produtos semelhantes ou de diferentes modelos. O facto dos robôs possuírem a capacidade de executar variações, pré-programadas, do programa original no ciclo de trabalho, permite que estes se adaptem a produtos de configurações diferentes. Neste tipo de aplicação são utilizados, tipicamente, robôs de dimensões menores e menores capacidades de carga, com configurações como a cartesiana e a SCARA. Estes robôs devem apresentar níveis elevados de precisão e repetibilidade [17, 35].

Na operação de inspeção existe a necessidade do sistema perceber se determinada condição foi satisfeita. Estas operações devem ter em conta que o processo deve ser completado, ter em conta que todas as partes foram adicionadas de acordo com o especificado e determinar as falhas que existiram, tanto a nível do trabalho das matérias primas, como ao nível das peças já acabadas [17, 35].

3.1.3 Operações de processamento

As operações de processamento são aquelas em que o robô executa uma operação sobre a peça, como por exemplo a pintura. Para a execução do processo, o robô deve ser capaz de mover a ferramenta em relação à peça a ser trabalhada ao longo de todo o ciclo de trabalho. Em certas operações de processamento pode haver necessidade de troca de ferramenta por parte do robô, utilizando-se um sistema para troca rápida de ferramenta. Exemplos deste tipo de operações são: a pintura, a soldadura, maquinação e operações de corte [17].

As operações de pintura possuem um ambiente bastante próprio. Este ambiente é bastante nefasto para o ser humano devido aos gases e partículas libertados durante a operação. A toxicidade deste ambiente exige que um operador trabalhe com máscara e roupa protetora. É exigido também que exista uma boa capacidade de exaustão e ventilação, por forma a reduzir os perigos deste mesmo ambiente. A operação de pintura é executada, frequentemente, através de uma ferramenta denominada de pistola de pintura. Um sistema de pintura deve integrar um sistema de transporte de peças entre o interior e exterior da cabine de pintura. Dentro da cabine coloca-se um robô a executar a tarefa de pintura e, deste modo, evita-se que o operador se dirija ao interior da célula. Em alguns tipos de células de pintura existe um sistema de reconhecimento de peças que permite adaptar o ciclo de pintura à peça específica a pintar. Neste tipo de operações os objetos são pulverizados, sendo que a tinta é aplicada na face do objeto. Nestas aplicações os robôs têm a capacidade de produzir movimentos suaves, normalmente programados através de métodos *leadthrough*. Estes robôs normalmente apresentam uma altura semelhante à de um ser humano, possuem seis graus de liberdade, com juntas de revolução e com acionamento hidráulico ou elétrico. Este tipo de robôs normalmente precisam de um revestimento próprio para evitar depósitos de partículas nas juntas do mesmo. Estes depósitos de partículas reduzem o tempo de vida útil, causando desgaste e avarias [17, 26, 35].

As operações de soldadura são uma grande percentagem das aplicações industriais do âmbito da robótica industrial. Os robôs na soldadura permitem racionalizar o processo. Esta racionalização deve-se à redução do espaço de trabalho, à flexibilidade do robô e à organização do processo. A utilização de robôs nesta operação permite também uma melhor segurança e qualidade, evitando que um operador frequente um ambiente hostil.

É possível dividir-se a soldadura em duas categorias: soldadura por arco elétrico e soldadura por pontos.

A soldadura por arco elétrico é também conhecida por soldadura contínua, uma vez que une duas superfícies através de uma união contínua num sentido referencial. Neste método de soldadura a resistência da zona de soldadura é su-

perior à da soldadura por pontos. Assim como nas operações de pintura, para um operário executar esta tarefa necessita de equipamento de proteção específico para essa tarefa. Neste tipo de aplicação é normal usarem-se robôs articulados ou cartesianos com cinco ou seis eixos, com boa repetibilidade e precisão. Para aumentar a durabilidade do material e qualidade do processo é necessário proceder a limpezas frequentes ao material. No caso da soldadura por arco elétrico é necessário uma limpeza periódica da tocha por forma a eliminar resíduos que se depositam durante o processo.

A soldadura por pontos é caracterizada pela união de dois materiais através de pontos específicos, designados por pontos de contacto. Neste tipo de soldadura são utilizados dois eléctrodos, feitos à base de cobre, através dos quais se fazem percorrer elevadas correntes de forma a derreter e soldar o material colocado no meio dos eléctrodos. Chama-se ao conjunto dos eléctrodos e do restante atuador pistola de soldadura. Neste tipo de soldadura, normalmente, as ferramentas de soldadura são demasiado pesadas para que um operador as consiga manipular sem dificuldade, pelo que neste género de soldadura são utilizados robôs com grande capacidade de carga e de cinco ou seis eixos. A sequência deste tipo de soldadura é composta por quatro fases: aperto, soldadura, espera com os eléctrodos fechados e a fase de libertar o material [17, 26, 35].

As operações de maquinagem podem ser divididas em três categorias: operações de maquinagem propriamente ditas, as operações de polimento e as operações de rebarbagem.

Nas operações de maquinagem propriamente ditas são utilizados robôs com tamanho humano, capacidades de programação avançadas, boa repetibilidade e servo-controlados. Estas operações libertam grandes níveis de poeira e produzem elevados ruídos sonoros que podem ser nocivos aos colaboradores. Neste tipo de aplicações existe uma elevada taxa de acidentes. Com a aplicação de robôs é possível aumentar a segurança dos colaboradores, garantir as tolerâncias devido à sua precisão e aumentar a produção devido à capacidade de trabalho de forma quase ininterrupta. O colaborador passa a ter como funções alimentar a linha com as peças, verificar a sua qualidade e substituir as ferramentas. A troca de ferramenta pode ser também executada pelo próprio robô. Existem duas hipóteses de maquinação por parte do robô: o robô segura a peça ou o robô segura a ferramenta. No primeiro caso o robô aproxima a peça de uma ferramenta em funcionamento por forma a executar a tarefa; no segundo caso o atuador final do próprio robô é a ferramenta. Este tipo de aplicações são de difícil implementação, uma vez que requerem um elevado número de ensaios, é necessário considerar elevado número de parâmetros e a pouca repetição das operações contribui para a difícil percepção da tarefa. Estes fatores têm como consequência uma difícil implementação robótica, devido ao elevado número de sensores necessário [17, 35].

As operações de polimento utilizam, normalmente, robôs de seis eixos colocados em posição invertida. Neste tipo de operações o robô manipula uma ou mais ferramentas de polimento cuja extremidade é, por norma, um disco de borracha. A pressão exercida e a velocidade são controladas através do atuador intrínseco ao robô. Nas operações de rebarbagem o robô pode possuir como atuador final um disco de rebarbagem rotativo para corte ou rebarbagem de uma peça [17, 35].

As operações de corte podem ser divididas em três categorias: corte com ferramentas tradicionais, corte laser e corte através de jato de água. A este tipo de operação estão associadas características como: ambiente de trabalho pouco agradável, perigosidade, elevada repetição e a inexistência de pessoal especializado para executar. A este tipo de tarefa podem estar associados sistemas de visão, uma vez que as peças podem variar entre si.

Nas operações de corte com ferramentas tradicionais são utilizadas ferramentas mecânicas que eram utilizadas pelos próprios operadores e foram adaptadas para uso do robô. Estas ferramentas estão sujeitas a grande desgaste, requerem maior manutenção e provocam efeitos indesejáveis no produto.

No caso do corte a laser o atuador final é uma ferramenta laser para corte. Este tipo de sistema possui grande flexibilidade espacial e precisão, assim como a flexibilidade de processamento do laser. Utilizados normalmente para corte de metal, podendo ser utilizados também para a sua fusão e conseqüente soldadura. Neste tipo de aplicação é necessário o recurso a sensores para a percepção da tarefa e da sua execução. O corte por jato de água é usual na indústria alimentar, na indústria do papel e na indústria metalúrgica. Neste tipo de aplicação o atuador final corresponde a uma ferramenta que emite um jato de água a pressão elevada, produzindo um corte preciso [17, 35].

3.2 Controladores Lógicos Programáveis

3.2.1 Conceito

Os controladores lógicos programáveis, do inglês *Programmable Logic Controllers* (PLC), são neste momento a tecnologia mais usada para controlo a nível industrial. Entende-se por PLC um computador de nível industrial com a capacidade de ser programado para executar funções de controlo, baseado em microprocessador, que utiliza uma memória programável para guardar instruções e para implementar determinadas funções. Este controlador diz-se por lógico pois de uma forma primária a sua programação assenta em princípios de lógica e de *switching*. A utilização de um controlador programável permite que se elimine o excesso de cablagem associado aos circuitos convencionais compostos por relés. Estes equipamentos permitem ainda uma resposta rápida, possuem uma

programação mais fácil bem como a sua instalação, permitem um controlo de velocidade, permitem a ligação a redes e são bastante fiáveis e com capacidade de teste e solução de problemas. O PLC está pensado para várias combinações de entradas e saídas, para trabalhar em escalas alargadas de temperatura, possui imunidades a ruído elétrico e é resistente à vibração e a danos por impacto. De forma resumida, um PLC é um computador digital para utilização no controlo de máquinas. Estes dispositivos foram inicialmente criados para substituir a lógica dos relés mas graças ao aumento constante das suas capacidades começaram a ser usados em aplicações mais complexas. Como os princípios de funcionamento de um PLC são semelhantes aos de um computador, estes possuem outras capacidades como: contagem, temporização, cálculo, comparação e análise de sinais analógicos [37, 38].

Os PLC possuem as seguintes vantagens [37]:

- maior confiabilidade: após a escrita e teste do programa este pode ser facilmente descarregado para o PLC. Uma vez que a lógica, o programa, esteja escrita em memória não existe forma de ocorrerem erros de contactos. O programa substitui uma grande parte da possível cablagem exterior;
- maior flexibilidade: é mais simples criar ou mudar um programa num PLC do que alterar as ligações de um circuito. O utilizador define as relações entre as entradas e as saídas através do programa inserido no PLC. Podem ser efetuadas melhorias através da criação de novos programas. Os utilizadores podem modificar o programa e ainda as definições de segurança diretamente no terreno;
- baixo custo: inicialmente os PLC foram pensados para substituírem os relés, pois apresentam um custo mais baixo. Em aplicações industriais é comum a utilização de um número elevado de relés. Desta forma, é compensatório o uso de um PLC em substituição do elevado número de relés;
- capacidade de comunicação: os PLC conseguem comunicar com outros controladores ou computadores para execução de tarefas de controlo, transferência de dados, monitorização e processamento. Podem ainda fazer *download* e *upload* do programa;
- tempo de resposta rápido: os PLC foram criados para executarem tarefas que necessitem de velocidades mais elevadas e aplicações em tempo real. O processamento de milhares de itens por segundo, por parte de uma máquina, e a leitura a uma taxa elevada de objetos por parte de um sensor, requer uma capacidade rápida de resposta;
- fácil solução de problemas: os PLC possuem conjuntos de funções que permitem aos utilizadores seguirem e corrigirem erros de *software* e *hardware*.

Para identificar e resolver problemas, os utilizadores podem executar o programa de monitorização e verificar em tempo real durante a execução,

- maior facilidade de teste de dispositivos: o painel de controlo do PLC possui a capacidade de verificar os dispositivos;

3.2.2 Partes constituintes de um PLC

Um PLC pode ser dividido em diversas partes fundamentais, como está representado na Figura 3.3: a unidade central de processamento, do inglês *central processing unit* (CPU), as unidades de entradas e saídas, a fonte de alimentação, o dispositivo de programação, a unidade de memória e a interface de comunicação. Um PLC semelhante a um computador. Um computador é otimizado para a execução de vários programas ou tarefas em simultâneo, ao contrário do PLC que é otimizado para tarefas industriais e de controlo, possuindo internamente a interface para as entradas e saídas [37, 38].

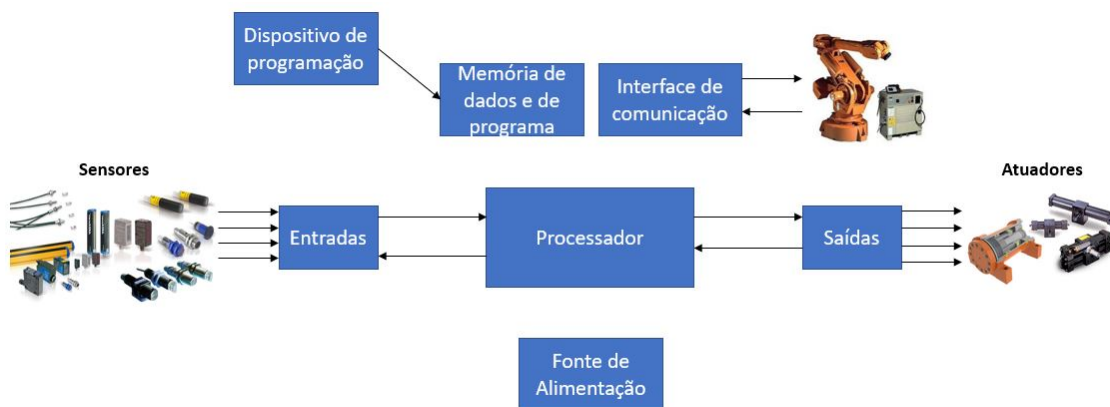


Figura 3.3: Diagrama de constituintes de um PLC.

1. Entradas e saídas (I/O)

As entradas e saídas podem ser ligadas ao PLC de forma: fixa ou modular. As entradas e saídas fixas são típicas de PLC mais pequenos que se apresentam com todos os módulos necessários sem unidades removíveis. O processador e as cartas de entrada e saída aparecem ligadas no mesmo espaço físico e os terminais de entrada e saída apresentam um número fixo de conexões. A principal vantagem deste tipo de PLC é o baixo preço. Atualmente é possível aumentar o número de entradas e saídas acrescentando cartas externas. Uma desvantagem é a reduzida flexibilidade, uma vez que as entradas e saídas são limitadas.

As entradas e saídas modulares são divididas por módulos individuais que podem ser ligados entre si e à unidade central. Este tipo de soluções aumentam a flexibilidade, uma vez que, permite associar vários módulos do fabricante de forma a encontrar a solução desejada. Um controlador modular básico consiste numa *rack*, uma fonte de alimentação, um módulo de processamento, cartas de entradas e saídas e um módulo de interface de programação e monitorização. Os módulos são ligados todos à *rack*, onde é efetuada uma ligação elétrica entre a *rack* e o módulo.

2. Fonte de alimentação

A fonte de alimentação é responsável por converter a tensão alternada em tensão contínua que irá fornecer posteriormente aos módulos ligados na *rack*. Em sistemas PLC maiores os dispositivos de campo podem ter uma alimentação própria, ao contrário de aplicações mais pequenas nas quais esta fonte de alimentação pode alimentar os dispositivos externos.

3. Processador (CPU)

O processador é o "cérebro" do PLC e tipicamente consiste num micro-processador desenvolvido para implementação de lógica e controlo de comunicações entre módulos. Esta unidade requer memória para guardar as instruções do programa, valores numéricos e estado dos dispositivos de I/O. Esta unidade é responsável pelo controlo da atividade do PLC e permite ao utilizador colocar o programa desejado sob a forma de lógica *ladder*. Um programa de um PLC é executado de forma cíclica e repetitiva associado a um varrimento. Um PLC tipicamente inicia com o CPU a ler o estado das entradas e de seguida a aplicação é executada. Uma vez executado todo o programa, o estado de todas as saídas é atualizado e, de seguida, o CPU executa as tarefas de diagnóstico interno e de comunicação. Este processo ocorre repetidamente enquanto o PLC se encontra em modo de execução.

4. Dispositivo de programação

O dispositivo de programação é utilizado para descarregar o programa na memória do processador. Este programa pode ser escrito em várias linguagens, dependendo das capacidades do compilador e do PLC. Os PLC são, normalmente, programados através de um computador pessoal, do inglês *personal computer* (PC). Este tipo de programação é, em muitos casos, efetuado através de *software* disponibilizado pela marca do próprio PLC. Estes programas não permitem só programar, mas também criar documentos e executar o algoritmo para solução de problemas.

5. Unidade de memória

A unidade de memória é a unidade responsável por armazenar informa-

ção, programas e dados no PLC. A memória do utilizador do PLC contém espaço para o armazenamento de programas, bem como para armazenamento de dados. O espaço de memória ocupado por um programa depende da sua complexidade. A memória do PLC pode ser dividida em duas categorias: volátil e não volátil. A memória volátil é aquela que perde os dados cada vez que há uma quebra de energia e é facilmente alterada. A memória não volátil tem a capacidade de reter a informação guardada, mesmo que haja uma quebra de energia.

6. Interface de comunicação

A interface de comunicação é utilizada para transferências de dados através de redes entre PLC em ambos os sentidos de comunicação. É necessário ter atenção à verificação do dispositivo, aquisição dos dados, sincronização entre aplicações e ao controlo da conexão.

7. *Human machine interface, HMI*

O HMI, vulgarmente conhecido como consola, é o constituinte do PLC que permite a interface direta entre o autómato e o operador. Estas consolas vieram substituir o uso de botões e equipamentos analógicos, anteriormente utilizados para a interface. Permitem visualizar as operações através de gráficos, fazer a gestão de alarmes e alterar temporizadores, contadores e outras variáveis.

O funcionamento do PLC baseia-se em três fases cíclicas: leitura das entradas, execução do programa e alteração das saídas. Através da leitura das entradas o PLC adquire a informação necessária para monitorização de um determinado processo. Após esta leitura, o CPU procede à execução do programa, guardado em memória, e processa a informação proveniente das entradas. Durante a execução do programa pode ser necessário guardar informação em memória. Após a execução do programa, conforme o código produzido, procede à atualização das saídas. Essas saídas serão responsáveis por ativar componentes constituintes do processo por forma a que este se processe da forma desejada e previamente programada [37, 38].

Associado a uma base de funcionamento simples o PLC possui também a característica de permitir que facilmente se altere a função que este executa, ou seja, é fácil alterar o código programado no PLC [37, 38]. Os PLC possuem ainda a característica de poderem ser ligados a consolas (HMI). Nestas consolas é possível visualizar algo que o operador deseje, e que tenha sido programado, e inserir dados necessários ao funcionamento do programa; basicamente permite uma interface entre o operador e o PLC [37, 38].

3.2.3 Linguagens de programação

É possível programar um PLC com o recurso às linguagens: o ladder, o texto estruturado, os gráficos de funções estruturais, os blocos de funções e as listas de instruções.

O ladder é uma linguagem de programação baseada numa representação simbólica com recurso à analogia dos circuitos elétricos, imitando a lógica dos relés. Cada bloco do ladder assemelha-se a um contacto tornando a programação em ladder semelhante à construção de um circuito elétrico. Na linguagem ladder o código é escrito por degraus similares. As linhas verticais representam as linhas de potência através das quais se conectam os circuitos. Cada degrau que constitui o código ladder (denominado de *rung*) representa uma operação no controlo do processo. Cada degrau deve começar com a leitura de uma entrada e terminar com a ativação de uma saída, podendo estas entradas e saídas serem variáveis internas, como por exemplo contadores [37][38].

O texto estruturado é uma linguagem alto nível utilizada principalmente para implementar procedimentos que não são facilmente implementados através de linguagens gráficas. Esta linguagem assemelha-se bastante à linguagem de programação Pascal. É utilizada escrevendo linhas de texto separadas por ponto e vírgula. Nestas linhas podem ser utilizadas linhas de código pré-definidas ou sub-rotinas para se alterarem valores de variáveis. O texto estruturado não é uma linguagem *case sensitive*. Esta linguagem permite o uso facilitado de condições *if*, *while* e *for* [37][38].

Os gráficos de funções estruturais, do inglês *sequential function chart* (SFC), são uma forma de programação gráfica com recurso a estados, ações e transições como acontece no *GRAFSET*, para descrever a sequência de eventos de uma operação [37][38].

Os blocos de funções, do inglês *function block diagrams* (FBD), correspondem a programas do PLC descritos por blocos gráficos interligados de forma simples ou complexa. o FBD é descrito como uma linguagem gráfica para descrever fluxos de sinal e dados através de blocos, para o desempenho de certas funções. OS FBD típicos incluem lógica, temporizadores e contadores[37][38].

As listas de instruções podem ser vistas como a entrada de um programa em ladder utilizando texto [37][38].

3.3 Processo de controlo com recurso a PLC

O controlo de um processo consiste no controlo automatizado de um processo com recurso à leitura de sinais externos. A capacidade de um PLC executar funções matemáticas e de utilização de sinais analógicos torna-o ideal para

controlar um processo de fabrico. Num ciclo de fabrico de uma peça são aplicados vários processos às matérias primas.

3.3.1 Sistemas de fabrico

O sistema de fabrico é onde o trabalho de valor acrescentado é executado numa parte ou produto. Um sistema de fabrico tem na sua constituição as máquinas de produção, o sistema de manuseamento de materiais, o sistema de controlo e os recursos humanos [17].

As máquinas podem ser divididas em três categorias: operadas manualmente, semi-automatizadas e automatizadas. No caso das máquinas operadas manualmente, estas são operadas e supervisionadas por pessoas. Nas máquinas semi-automatizadas parte do processo é executado de forma automática e outra através do operador humano. Nas máquinas automatizadas todo o processo é executado de forma automática e controlado por um programa de controlo. Uma das diferenças entre as máquinas automatizadas e semi-automatizadas é o facto de nas semi-automatizadas a carga e descarga ser feita por um operador e nas automatizadas ser feita pela máquina. Outra diferença é o tempo que podem operar sem um operador humano ter contacto com a máquina que no caso das automatizadas é bastante superior [17].

O sistema de manuseamento de materiais realiza um conjunto de operações necessárias em vários processos e que implicam movimento de peças. As operações desenvolvidas são: carga e descarga das estações de trabalho, posicionamento das unidades de trabalho em cada estação, transporte entre estações e armazenamento temporário das peças. Estes sistemas permitem reduzir o esforço humano neste tipo de tarefas. Uma das desvantagens é a sua reduzida adaptabilidade, normalmente são específicos para o sistema de fabrico para o qual foram projetados [17].

O sistema de controlo normalmente é um sistema computadorizado nos sistemas de fabrico automatizados por forma a aumentar a eficiência no controlo e coordenação do sistema de fabrico. O sistema de controlo tem na base das suas funções as seguintes: comunicação de instruções para os trabalhadores (apresentando o esquema de trabalho que devem executar, através de uma interface pré-configurada), envio e gestão de programas para as máquinas que controla, controlar o sistema de manuseamento de materiais, programar a produção, diagnosticar falhas no sistema produtivo, monitorizar a segurança, monitorizar o controlo de qualidade e gerir operações [17].

Os recursos humanos correspondem aos operadores presentes no sistema produtivo. Nos sistemas de fabrico, os operadores executam grande parte, ou todos, os trabalhos que acrescentam valor ao produto, sendo nestes casos designados de trabalhadores diretos. Estes trabalhos podem ser executados manual-

mente ou com auxílio de máquinas que controlam. Mesmo nos sistemas completamente automatizados é necessário ter operários para executarem tarefas de carga e descarga do sistema, tarefas de mudança de ferramentas, tarefas de restauro de algumas ferramentas e outras funções semelhantes. Os operadores possuem ainda tarefas de manutenção e controlo do sistema, como programadores, operadores de máquinas computadorizadas, operacionais de manutenção do sistema e outras tarefas similares de trabalho indireto [17].

3.3.2 Classificação dos sistemas de fabrico

Devido ao vasto número de sistemas de fabrico diferentes existentes na indústria, existe uma necessidade de os classificar por razões organizativas. Os sistemas de fabrico podem ser classificados através de quatro fatores: tipos de operações executadas, número de estações de trabalho e *layout* do sistema, nível de automatização e variedade de partes ou produtos [17].

De acordo com o tipo de operações executadas os sistemas podem ser divididos em: operações de processamento em unidades de trabalho individuais e operações de montagem. Além desta distinção, podem ser distinguidos também através das tecnologias utilizadas, bem como os parâmetros do produto [17].

O fator de número de estações de trabalho é bastante importante na classificação dos sistemas. Este fator encontra-se diretamente ligado ao desempenho do sistema através dos termos: capacidade de produção, produtividade, custo por unidade e manutenção. Uma forma de medir o tamanho do sistema de fabrico é através do número de estações de trabalho. O autor Mikell Groover organiza este tipo de classificação em três tipos numerados: tipo I, tipo II e tipo III. No tipo I estão as unidades compostas por uma simples estação, no tipo II os sistemas multi-estação com rota variável e no tipo III os sistemas multi-estação com rota fixa [17].

A classificação através do nível de automatização assenta no critério anterior completando o mesmo, ou seja, são utilizados os mesmos três tipos e divididos em três grupos. Esses três grupos são: sistemas completamente automatizados, manuais ou híbridos [17].

O quarto fator corresponde à capacidade que um sistema possui para lidar com variações nas partes ou produtos que o mesmo produz [17].

3.3.3 Controlo dos processos através de PLC

Os PLC são utilizados para controlar os processos desenvolvidos nos sistemas de fabrico. O controlo de processo designa um controlo automático de um processo, por forma a que este decorra dentro de parâmetros previamente definidos. Neste tipo de sistemas existe uma troca de sinais analógicos entre o

processo e o centro de controlo, através de sensores. Em ambiente industrial existem três tipos de processos: processo contínuo, processo em lote e fabrico discreto [37].

Um processo contínuo é caracterizado pela entrada das matérias primas num extremo do sistema e saída do produto final no outro extremo, sendo que o processo entre extremos acontece de forma contínua. No processo em lote não existe um fluxo de materiais do produto de uma secção do processo para outra. Em contrapartida, é definida uma quantidade a ser recebida no início do processo sob a forma de lote e é executado um processo sobre esse lote. O fabrico discreto é caracterizado por uma produção individual ou em baixas quantidades de um produto [37, 38].

O sistema de controlo de um sistema de fabrico pode ser centralizado ou distribuído. Num sistema de controlo centralizado um conjunto de máquinas ou processos são controlados por um controlador central. O sistema distribuído é baseado em redes. Estes sistemas são compostos por duas ou mais unidades de controlo que comunicam entre si para executarem a tarefa de controlo. Cada unidade controla diferentes processos localmente e troca constantemente informação com outras unidades. Estes sistemas têm na sua composição o PLC, HMI, sensores, atuadores e o condicionamento de sinal. Neste tipo de sistemas o HMI é útil para se visualizarem sumários da operação, configurar o processo, visualizar o sumário de alarmes, efetuar controlo manual, tarefas de diagnóstico e verificação de valores. Estes sistemas são utilizados em vários tipos de controlo como os seguintes: controlo ON/OFF, controlo proporcional-integral-derivativo (PID), controlo de movimento e gestão de tarefas [37, 38].

3.3.4 Tipos de comunicação

Os tipos de comunicação utilizados nestes sistemas são: comunicação série, data highway, deviceNet, controlnet, ethernetIP, modbus, field bus, profibus, sercos e hart.

Comunicação série é implementada através de normas como RS-232, RS-442 e RS-485. Estas normas definem as características elétricas, mecânicas e funcionais da comunicação. O módulo de comunicação pode ser integrado no processador ou ser um periférico, no caso de um PLC uma carta. Este tipo de comunicação consiste no envio de dados, bit a bit, através de um canal de comunicação. Esta comunicação permite trocas de informação com vários tipos de dispositivos como: sistemas de visão, leitores de códigos, entre outros [37].

O *data highway* é um tipo de rede característico da Allen-Bradley, sendo uma rede de proprietário. Utiliza a comunicação *peer-to-peer* através da implementação de *tokens* [37].

O device net é um tipo de rede aberta ao nível dos dispositivos. Apresenta velocidades baixas mas é bastante eficiente na gestão de pequenas mensagens trocadas com os dispositivos de entrada e saída. O device net permite reduzir custos, uma vez que permite integrar todas as entradas e saídas numa rede composta por 4 fios de dados e alimentação, no mesmo cabo. Este tipo de conexão reduz custos e consumo de tempo. A sua função principal é a troca de dados entre o controlador e os dispositivos de campo. O device net utiliza um protocolo denominado de *Common Industrial Protocol* (CIP), que é orientado a objetos, objetos esses que possuem atributos, serviços e comportamentos [37].

O controlNet está localizado um nível acima do device net. Utiliza o protocolo CIP para combinar as funcionalidades de uma rede *peer-to-peer* com as funcionalidades de uma rede de entradas e saídas, permitindo executar ambas a velocidades elevadas. Este tipo de rede é determinística, permite prever quando os dados serão entregues, e possui repetibilidade, pois apresenta tempos de transmissão de dados constantes e independentes de conexões e quebras de conexão de outros dispositivos. Para a configuração de cada dispositivo ControlNet, são necessários os Electronic device data sheets (EDS), de cada dispositivo [37].

O ethernet/IP é um protocolo aberto, baseado no protocolo CIP utilizado no deviceNet e no ControlNet. Este protocolo permite troca de informação entre dispositivos sem que seja necessário *hardware* personalizado para o processo. A partilha da camada de aplicação por parte do ethernet/IP, deviceNet e ControlNet, permite criar uma interação *plug-and-play* entre dispositivos complexos através do Ethernet/IP. O Ethernet/IP permite também uma comunicação *full-duplex*. Apresenta também a capacidade de interação entre dispositivos e equipamentos de controlo na mesma rede [37].

O Modbus é um protocolo de comunicação série desenvolvido pela Modicon para uso em PLC, consistindo num método de transmissão de informação entre dispositivos electrónicos através de linhas série. O dispositivo que requer informação é denominado de *Master* os que providenciam a informação são denominados de *Slave*. Este protocolo é um protocolo aberto [37].

O FieldBus é um protocolo série, aberto e bi-direcional, que interliga sistemas de medição e controlo. É utilizado no nível "campo" para controlo de processos, podendo ter diversas tipologias [37].

O Profibus-DP é um tipo de comunicação fieldBus aberta que suporta sinais analógicos e digitais. As suas funcionalidades assemelham-se ao *device net* [37].

Sercos, do inglês *serial real-time communication system*, é uma comunicação padrão para controlo de movimento. Este tipo de comunicação possibilita o uso de dispositivos de vários fabricantes. Esta comunicação em rede está desenvolvida para comunicações série de velocidades elevadas em tempo real, com transmissão de dados padronizados em *closed-loop* [37].

O Hart é um protocolo de comunicação aberta do tipo *master-slave* desenvolvido para comunicação com dispositivos inteligentes que se encontram no "campo", ou seja, diretamente no ambiente industrial. Este tipo de dispositivos contém mais informação e podem possuir algumas funções originalmente programadas no PLC. Este protocolo permite comunicações analógicas e digitais em simultâneo. O protocolo hart permite a comunicação com um número mais alargado de dispositivos [37].

3.3.5 Sistemas SCADA

Os sistemas *supervisory control and data acquisition* (SCADA) são utilizados quando é necessário que o PLC, para além das normais funções de controlo, seja responsável pela aquisição de dados, os processe e os estruture para a geração de relatórios. Um sistema SCADA simplifica a aquisição de dados e facilita o fluxo e troca de informação entre níveis inferiores e superiores. Este tipo de sistemas efetuam trocas de informação com os sistemas a gerir, em tempo real, mas não os controlam. Possuem a capacidade de supervisionar as saídas de controlo, o que permite proceder a ajustes nos processos com uma precisão maior, e aumenta a eficiência. Tipicamente, num sistema SCADA os PLC são independentes e executam funções de controlo enquanto são supervisionados por um *software* SCADA. Uma grande vantagem de um sistema SCADA é o facto de armazenar os dados automaticamente em formatos para posterior análise, sem erros ou trabalho adicional. Estes sistemas têm uma importante aplicabilidade na gestão de alarmes [37].

3.4 Conclusão

Em suma, existe uma necessidade de controlo das diversas aplicações industriais e operações a elas associadas. Para se efetuar esse controlo recorrem-se a controladores discretos como é o caso dos PLC, microcontroladores e computadores. Na realidade industrial o mais regularmente utilizado é o PLC devido à sua resistência e outras características, como as linguagens de programação gráficas e outras semelhantes a linguagens de programação de mais alto nível, que o tornam a opção mais viável em ambiente industrial. Para aumentar a eficácia do controlo e centralizá-lo recorre-se a sistemas de comunicação que permitem a transferência de informação entre os centros de controlo e as estações a controlar. Por fim estes sistemas de controlo através de PLC podem recorrer a sistemas SCADA por forma a armazenar dados que sejam relevantes futuramente.

Capítulo 4

Segurança e Aplicações Industriais

No início deste capítulo são abordadas aplicações de pick and place e de montagem, nas quais são especificados os robôs e atuadores mais indicados bem como software. De seguida é abordada a questão da segurança em ambientes robotizados e no seguimento são analisados diversos casos de estudo associados ao controlo de sistemas através de autómatos e de aplicações robóticas. As aplicações robóticas analisadas neste capítulo são o pick and place e a paletização e os casos de estudo analisados neste capítulo pertencem a diversos fabricantes, quer de PLC, quer de robôs.

4.1 Aplicações de *pick and place* e montagem

As aplicações de *pick and place* consistem no manuseamento de um ponto inicial para o final, sendo que no ponto final pode existir a montagem quando se dá a união de duas peças. Um exemplo clássico de uma operação de *pick and place* é o manuseamento de materiais entre um tapete de entrada e um tapete de saída que não estejam ligados.

A aplicação a desenvolver tem por objectivo proceder ao *pick and place* de peças e conseqüente montagem. Pretende-se reduzir o tempo de ciclo desta tarefa sem perder a qualidade da mesma.

4.1.1 Robôs de *pick and place* e montagem

Os robôs de *pick and place* podem ser utilizados numa grande variedade de tarefas de manuseamento de materiais, como por exemplo, paletização, manuseamento de produtos e carga e descarga de máquinas. Estes robôs podem ser utilizados numa configuração circular, que consiste em colocar os postos de *pick*

e de *place* em forma circular dentro da área de trabalho do robô, podendo interagir com vários postos de pega de produto e de depósito de produto. Podem ainda executar várias funções diferenciadas através da possibilidade atuar como interface entre tapetes e máquinas, através da carga e descarga de máquinas ou através da transferência de partes de uma máquina para outra [26].

Os robôs de *pick and place* possuem várias características específicas como:

- precisão: estão equipados com braços longos e finos, possuem repetibilidade constante e podem ser equipados com ferramentas bastante precisas;
- flexibilidade: devido à facilidade de alteração do programa de movimentos do robô, estes assumem uma elevada flexibilidade de tarefas. Esta flexibilidade confere a capacidade de lidar com múltiplos produtos, com formas diferentes. Possuem ainda uma elevada flexibilidade de movimentos;
- aumenta a consistência: estes robôs conferem a capacidade de aumentar a qualidade dos produtos e de reduzir o tempo de ciclo da tarefa. Devido à sua repetibilidade, o resultado dos movimentos, e conseqüentemente da tarefa, é sensivelmente constante. Esta consistência contribui para o aumento da qualidade.
- eficiência espacial: devido ao seu formato compacto, estes robôs ocupam um espaço reduzido. Podem ainda operar em espaços restritos e espaços de trabalho reduzidos.
- maximização da segurança: através da substituição de colaboradores humanos em tarefas repetitivas contribuem para o aumento da segurança, pois a monotonia associada às mesmas pode resultar em acidentes. Os robôs não sofrem de *stress* nem cometem erros por repetição de tarefas.
- baixar os custos: a sua precisão e repetibilidade aumentam a eficiência da produção e podem contribuir para a redução do desperdício de materiais. Desta forma o investimento é rapidamente recuperado.

Existem dois tipos de robôs bastante utilizados em tarefas de *pick and place* de materiais com pouca massa: os robôs SCARA e os robôs paralelos, mas também podem ser utilizados outros manipuladores robóticos com configuração série de mais eixos (normalmente de 4 a 6 eixos) dependendo da aplicação em específico. Quando é necessário manipular materiais mais pesados recorre-se a braços robóticos de 4 a 6 eixos. Os robôs de 4 eixos são muitas vezes utilizados em tarefas de paletização.

As aplicações de montagem possuem na sua génese a função de unir dois produtos num só. Este tipo de aplicações podem dar origem a produtos intermédios ou finais numa linha de produção. Desta forma, esta aplicação, na

maioria dos casos, interage diretamente com os pedidos dos clientes. Como consequência, a montagem deixa de ser apenas a função de unir peças e tende a ir para além disso. Através do planeamento da montagem de um produto é possível atender uma variedade de necessidades e negócios [26].

Os robôs foram associados a estes processos com o objetivo de aumentar a sua eficiência através da redução dos gastos e aumento da produção. A utilização de um robô num processo de montagem pode aumentar de forma exponencial a velocidade de produção da linha e a sua consistência. Os robôs permitem também substituir as pessoas em trabalhos repetitivos, como a montagem, que podem incorrer em erros devido à falta de atenção. O atuador final pode variar e ser adaptado a cada operação de montagem por forma a retirar o melhor desempenho do sistema. Para melhorar ainda mais o desempenho podem ser acrescentados sistemas de visão. Os sistemas de visão aumentam a eficiência e precisão do sistema [39].

A ABB possui três robôs, entre outros, cujas características são benéficas para a aplicação, de *pick and place*: IRB 140, IRB 360 e o IRB 910SC. O conjunto de características mais importante está representado na Tabela 4.1.

Tabela 4.1: Robôs de *pick and place* da ABB [2, 3, 4].

Robô	Número de eixos	Capacidade de Carga	Alcance	Modo de montagem	Controlador	Velocidade máxima	Repetibilidade
IRB 140	6	6 kg	810 mm	Qualquer	IRC5	1000 mm/s	±0.4 mm a 100 mm/s ±0.7 mm a 300 mm/s
IRB 360	3 ou 4	1 kg 3 kg 6 kg 8 kg	800 mm 1130 mm 1600 mm	Superior (em estrutura)	IRC 5	10000 mm/s	±0.04 mm
IRB 910 sc	4	6 kg	450 mm 550 mm 650 mm	Inferior	IRC 5 Compact	6200 mm/s 6900 mm/s 7600 mm/s	±0.015 mm

De acordo com o seu diagrama de carga do IRB 140, presente na figura 4.1, a sua capacidade de carga corresponde à anteriormente referida desde que o centro de gravidade do objeto a manusear se encontre num intervalo de 50 mm de distância à origem do eixo dos z. Com o aumento dessa distância a capacidade de carga cai para os 1,5 kg [2].

O IRB 360 FlexPicker, desenvolvido especificamente para tarefas de *pick and place* e ilustrado na Figura 4.1, possui a sua capacidade de carga máxima quando o objeto possui o seu centro de massa a menos de 20 mm da origem do eixo z com o plano x-y. Consoante esta distância aumenta, a capacidade de carga diminui até ao valor de 1/10 da capacidade carga máxima [3].

O IRB 910 SC, visualizado na Figura 4.1, é um robô SCARA da ABB desenvolvido para executar tarefas de *pick and place* e de montagem de produtos [4].



Figura 4.1: a) Manipulador robótico IRB 140 [2]. b) Manipulador robótico paralelo IRB360 [3]. c) Manipulador robótico SCARA IRB910sc [4].

A Kuka possui vários modelos com capacidade de executar tarefas de *pick and place* e de montagem, porém nenhum modelo é específico para este tipo de tarefas. Este robôs possuem características que lhes permitem, alterando o seu atuador final, executar várias tarefas diferentes. Três possíveis robôs para *pick and place* e montagem que a KUKA oferece são: o LBR iiwa, o KR 5 sixx e o KR 6 R900 sixx [40]. As características mais importantes estão descritas na Tabela 4.2.

Tabela 4.2: Robôs de *pick and place* da KUKA [5, 6, 7].

Robô	Número de eixos	Capacidade de Carga	Alcance	Modo de montagem	Controlador	Velocidade máxima	Repetibilidade
LBR iiwa	7	7 kg 14 kg	800 mm 820 mm	Inferior	KUKA Sunrise Cabinet	85 °/s a 180 °/s	±0.1 mm ±0.15 mm
KR 5 sixx	6	5 kg	1,078 mm 1,453 mm	Qualquer	KR C2 sr	250 °/s e os 660 °/s	±0.02 mm ±0.03 mm
KR 6 R900 sixx	6	6 kg	901.5 mm	Inferior	KR C4 smallsize-2 KR C4 compact	360 °/s a 518 °/s	±0.03mm

O LBR iiwa, ilustrado na Figura 4.2 divide-se em dois modelos: o LBR iiwa 7 R800 e o LBR iiwa 14 R820. As diferenças entre os modelos resumem-se à capacidade de carga, alcance e repetibilidade. A sua precisão é de ± 0.1 mm [5].

O KR 5 sixx, que se visualiza na Figura 4.2, divide-se também em dois modelos: o KR 5 sixx R650 e o KR 5 sixx R850. As diferenças entre estes dois modelos são o alcance e a repetibilidade [6].

O KR 6 R900 sixx, presente na Figura 4.2, é um modelo da série KUKA AGILUS [7].



Figura 4.2: a) Manipulador robótico LBR iiwa [5]. b) Manipulador robótico KR 5 sixx [6]. c) Manipulador robótico KR 6 R900 sixx [7].

A FANUC possui uma elevada variedade de robôs com características, para a execução de tarefas de *pick and place* e montagem, de entre os quais o M-2iA, o CR35iA e os SR-3iA.

Tabela 4.3: Robôs de *pick and place* da FANUC [1, 8, 9].

Robô	Número de eixos	Capacidade de Carga	Alcance	Modo de montagem	Controlador	Velocidade máxima	Repetibilidade
M-2iA/3SL	4 a 6	3 kg	1130 mm	Superior (numa estrutura)	R-30iB	10000 mm/s 3500 °/s	±0.1 mm
CR-35iA	6	35 kg 3 kg	1813 mm	Inferior	Without	750 mm/s	±0.08 mm
SR-3iA	4	6 kg	400 mm 650 mm	Inferior	R-30iB Compact Plus controller	12 rad/s a 52,35 rad/s	Varia consoante o eixo

O robô M-2iA/3SL, ilustrado na figura 4.3 é um robô do tipo paralelo da Fanuc [1].

O robô da FANUC CR-35iA, presente na Figura 4.3, é um robô colaborativo [8].

O robô SR-3iA, ilustrado na Figura 4.3, é um robô do tipo SCARA da Fanuc [9].

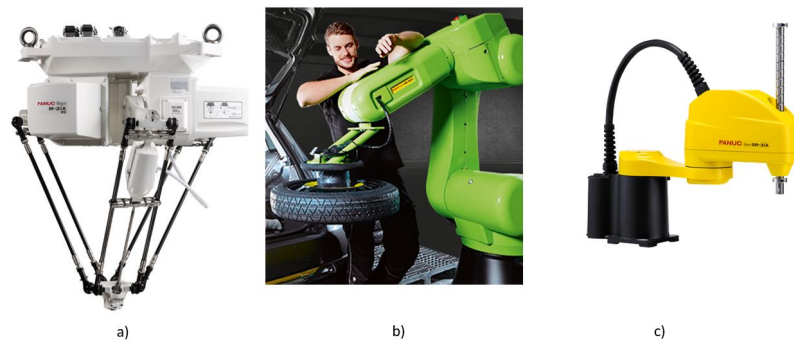


Figura 4.3: a) Manipulador robótico M-2iA/3SL. b) Manipulador robótico CR-35iA [8]. c) Manipulador robótico SR-3iA [8].

Com base no estudo dos modelos de robôs de *pick and place* que as marcas ABB, KUKA e FANUC conclui-se, que estes robôs apresentam capacidades de carga reduzidas, velocidades de operação elevadas, repetibilidade e precisão na ordem das décimas ou centésimas de milímetro e possuem alcances reduzidos, normalmente abaixo dos 1500 mm.

4.1.2 Software de programação para aplicações de *pick and place*

Os *softwares* utilizados para aplicações de *pick and place* são os mesmos utilizados em aplicações de paletização, uma vez que a paletização é um caso específico de *pick and place*. Existem marcas que desenvolvem *software* específico para *pick and place*, outras desenvolvem *software* aplicável em tarefas de *pick and place* e paletização.

No caso da ABB o *software* utilizado em aplicações de *pick and place* é o PickMaster. Este *software* é aplicável em operações de *pick and place*, embalagem, paletização e transporte de materiais. O PickMaster permite reduzir o tempo de programação dos robôs, nas tarefas enunciadas, através das ferramentas que possui. Existem dois produtos PickMaster: o PickMaster 3 e o PickMaster 5, ambos utilizam o mesmo ambiente gráfico e a mesma estrutura [41].

O PickMaster 3 possui a funcionalidade de utilizar técnicas de visão avançadas e a capacidade de acompanhar peças em tapetes. O sistema de visão integrado no *software* PickMaster 3 é avançado e permite a comunicação com sensores externos. Este *software* é integrado com o controlador IRC5 por forma a permitir que os robôs sejam guiados de forma mais eficaz nas tarefas de *pick and place*. Permite o controlo de um sistema de visão, desenvolvido pela ABB, que tem a capacidade identificar objetos num ângulo de 360 graus, num tempo compreendido entre os 50 e os 200 ms. Este sistema pode ser monocromático

ou policromático. Os projectos desenvolvidos nesta aplicação necessitam de ser configurados e calibrados com elevada precisão através do recurso ao seu ambiente gráfico. A aplicação PickMaster pode controlar até 10 robôs e 10 câmaras, operando em conjunto ou em processos distintos. Permite ainda o controlo periférico da operação através da conexão a um PLC ou PC [41].

O PickMaster 5 é um *software* desenvolvido para melhorar aplicações de paletização. Esta aplicação permite uma instalação sem necessidade de recorrer à programação propriamente dita. Permite configurar *offline* a aplicação a desenvolver sendo o código desta depois transferido para IRC5. Após esta ser transferida para o controlador corre de forma independente ao PC integrado no controlador IRC5. Esta aplicação corre em paralelo na consola facilitando assim o início, fim e acompanhamento da tarefa de paletização. Permite o recurso a várias entradas e saídas no processo de paletização, permite também a conjugação de vários produtos diferentes e de entradas diferentes na mesma paleta [41].

No caso da KUKA não existe um *software* específico para aplicações de *pick and place*. Para este tipo de aplicações são utilizados o KUKA.SIM, que é o *software* de simulação e programação *offline* da KUKA, ou o KUKA.FlexPal que é o *software* utilizado em aplicações de paletização.

O KUKA.Sim é o *software* de simulação e programação da KUKA. Este *software* permite a criação de *layouts* através do uso da sua funcionalidade gráfica e da biblioteca de objetos que possui. A sua biblioteca possui um catálogo, designado por eCatalog, que contém modelos de componentes electrónicos parametrizáveis. Este catálogo contém desde tapetes, a atuadores finais e outros componentes industriais. O KUKA.Sim possui a capacidade de importar ficheiros CAD de diferentes formatos por forma a melhorar a simulação com modelos dos componentes reais que não se encontram no catálogo. Permite avaliar alcances e colisões no ambiente simulado. Permite ainda a programação de robôs KUKA *offline* e encontra-se preparado para a indústria 4.0, permitindo a avaliação de tempos de ciclo, por exemplo [42].

O KUKA FlexPal é um *software* desenvolvido para aplicações de paletização. Este consiste na configuração de duas ferramentas: o KUKA.FlexPal Editor e o KUKA.FlexPal RT. O editor permite definir pacotes, paletes, atuadores finais, folhas intercalares, padrões de camada e planos de empilhamento. Com algumas expansões permite também definir espaços de trabalho e movimentos. O KUKA.FlexPal RT é o *software* que suporta as configurações mais simples e a programação de tarefas de paletização e despaletização no controlador KR C4. O KUKA.FlexPal permite definir o comprimento, a largura, a altura e o peso de uma paleta bem como atribuir-lhe um código. Sendo que quando aplicado em tarefas de *pick and place* sem recurso a paletes, estas paletes definidas no programa corresponderão às zonas de *pick* e de *place* da aplicação. Este *software* tem

como vantagens a definição dos vários componentes utilizados na paletização três tipos de atuadores finais (sucção, pinças e garfos) [43].

A FANUC, à semelhança da KUKA, oferece a possibilidade do uso de um *software* genérico para aplicações, o ROBOGUIDE. Este torna-se mais versátil quando são acrescentadas algumas extensões como: o ROBOGUIDE Handling-Pro, e o outro mais específico ainda para *pick and place*, o ROBOGUIDE-PickPro.

O *software* ROBOGUIDE é a aplicação desenvolvida pela FANUC para simulação e programação *offline*. Este *software* permite programação através da linguagem KAREL ou através da marcação de pontos. Nesta aplicação é simulada a programação *online* pois apresenta um emulador da consola presente nos controladores da FANUC. Permite a importação de ficheiros CAD externos por forma a melhorar a simulação do cenário pretendido. Permite também perceber alcances, determinar colisões, apurar tempos de ciclo e definir trajetórias para os robôs. Através do ROBOGUIDE é possível criar, alterar e descarregar um programa para um robô [44].

A extensão HandlingPRO permite simular aplicações de transferência de materiais, como é o caso de aplicações de *pick and place*. Permite a programação de uma trajetória, simular e programar o acompanhamento de um produto num tapete, criar modelos de máquinas e programar a aplicação [45].

A extensão PickPRO permite simular e analisar processos de *pick and place* de elevada cadência. Permite também analisar sistemas multi-tarefa, onde múltiplos robôs executam tarefas de *pick and place* entre várias entradas e saídas. O PickPRO ajuda a perceber o impacto, na aplicação, de parâmetros como: a velocidade dos tapetes, espaço entre produtos e a sua orientação, partilha do espaço de trabalho feita por vários robôs, direcção do fluxo de produtos no tapete, o robô em si, o sistema de localização de produto, a influência de atuadores finais multifacetados e interrupções de produção [44].

A FANUC apresenta ainda um *software* específico para tarefas de *pick and place*, com recurso ou não a sistemas de visão, o iRPickTool. Este *software* permite a gestão de múltiplos robôs para se efetuarem mais de 2000 *picks* por minuto. Permite a inclusão até 32 robôs, 16 zonas de *pick and place* podendo estas conter até 160 peças, permite incluir até 32 sensores, permite até 32 tapetes e ainda permite a inclusão de até 84 estações de trabalho. A sua integração com o *software* iRVision permite o acompanhamento visual do produto no tapete, executar tarefas de inspecção e controlo de qualidade, efectuar a leitura de códigos de barras e permite ainda o rastreio rápido dos tapetes. Desta forma, o iRPickTool permite melhorar o desempenho da tarefa através da melhoria do desempenho dos robôs e uma mais fácil e rápida programação dos mesmos [46].

4.1.3 Garras para aplicações de *pick and place*

O atuador final mais indicado para aplicações de *pick and place* é a garra, visto que esta tem como objetivo agarrar o produto permitindo ao robô movimentá-lo no espaço.

A escolha do tipo de garra a utilizar em determinada tarefa de *pick and place* depende do produto a transportar. No caso de ser sólido e indeformável poder-se-á utilizar garras mecânicas de dois ou três dedos; no caso de poder ser deformável poder-se-á utilizar, por exemplo, garras que operem a vácuo.

As garras mecânicas podem ser atuadas de forma pneumática, eléctrica ou hidráulica, sendo o mais comum a atuação pneumática. Estas garras são constituídas por dedos denominados de pinças, que efetuam o contacto com o produto a transportar. Estes dedos podem ou não fazer parte do mecanismo da garra [47].

A título de exemplo, a marca Schunk apresenta unidades desenvolvidas para tarefas de *pick and place*, unidades estas cujo atuador final é uma garra mecânica. Para soluções robóticas a Schunk apresenta garras mecânicas de dois, três e quatro dedos [48].

As garras de dois dedos paralelos permitem transportar materiais com uma massa de poucos gramas até materiais com massa de 80 kg, dependendo do modelo da garra. Estas garras permitem a adição de sensores consoante a aplicação. Na Figura 4.4 estão representados vários modelos da Schunk que variam principalmente na sua dimensão e forma da garra, apresentando relevos diferentes na zona de pega. Esta zona de pega é por vezes desenhada e construída de acordo com o produto a manipular [49].



Figura 4.4: Garras paralelas da Schunk [49].

As garras de três dedos permitem movimentar produtos cuja massa varie entre os poucos gramas e os 100 kg. Estas garras são maioritariamente utilizadas para transporte de produtos de forma cilíndrica ou esférica. Na Figura 4.5, à semelhança das paralelas, diferem entre si em determinadas características, sendo o comprimento e o formato da zona de pega o mais evidente. Este formato tem que ser adaptável ao tipo de material a pegar e à sua resistência mecânica [50].



Figura 4.5: Garras concêntricas da Schunk [50].

As garras de quatro dedos concêntricos têm por objetivo a sua utilização em tarefas de precisão como posicionamento, montagem, entre outras. A Schunk apresenta apenas um modelo com várias variantes deste tipo de garra presente na Figura 4.6. Isto deve-se ao facto de ser um modelo utilizado em aplicações bastante específicas [51].



Figura 4.6: Garras concêntricas de quatro dedos da Schunk [50].

As garras por sucção permitem o manuseamento de objetos planos, lisos e limpos. Se o material do produto a manusear for rígido as ventosas utilizadas devem ser de um material elástico por forma a se adaptarem melhor ao objeto. Se o material for macio as ventosas devem ser constituídas por um material mais duro por forma ao material se ajustar às ventosas [47].

4.2 Segurança em Ambiente Industrial

A segurança é algo fundamental em qualquer aplicação automatizada. A aposta na segurança permite que se diminuam perigos com a intenção de eliminação de acidentes que envolvam colaboradores e máquinas [52].

De acordo com a Agência Europeia para a Segurança e Saúde no Trabalho (EU-OSHA), a cada três minutos e meio morre uma pessoa na União Europeia devido a causas relacionadas com o trabalho. Todos os anos morrem na União Europeia 142 400 trabalhadores devido a doenças profissionais e 8 900 devido a acidentes de trabalho, e um terço das mortes por doença podem ser atribuídas ao uso de substâncias perigosas no trabalho. Por exemplo, o uso do amianto é responsável por mais de 21 000 mortes [52].

Portugal apresenta um número elevado de mortes relacionadas com o trabalho, e cerca de metade ocorre na construção civil [52]. As principais causas de acidentes de trabalho são: o não cumprimento ou desconhecimento das normas de segurança, utilização incorreta ou nula dos equipamentos de proteção, mau estado de conservação das instalações e equipamentos de trabalho, falhas de comunicação, riscos inerentes à tarefa, não cumprimento do método de trabalho e *stress* e cansaço [52].

4.2.1 Perigos do uso de robôs

Um perigo é tudo aquilo com potencial de causar um dano [52]. Com base nisto, os robôs e o seu *software* são desenvolvidos para uso em aplicações específicas e os fabricantes não se responsabilizam pelo uso impróprio. O seu uso impróprio pode provocar perigo para os colaboradores, robô e outras máquinas e ao próprio processo [53].

Os principais perigos dos robôs são [53]:

- impacto provocado pelo robô ou pela ferramenta,
- pontos de reclusão criados pelo movimento do robô;
- choques elétricos, queimaduras, radiações, entre outros.

Um risco é a possibilidade que alguém tem de sofrer um dano provocado por um perigo [52].

As causas dos riscos associados aos robôs são: a invasão não autorizada da área de trabalho por parte de colaboradores, os erros de programação e manutenção, falhas associadas ao acionamento, erros de controle, falhas mecânicas, falhas ambientais e possíveis incêndios [53].

Para evitar possíveis acidentes, é preciso ter em mente as regras de ouro da robótica que são as seguintes [53]:

1. se o robô não se encontrar em movimento, não assumir que não o irá fazer;
2. se o robô está a executar um padrão, não assumir que o continuará a executar;
3. manter respeito pelo robô e pelo que pode vir a fazer.

4.2.2 Normas e legislação

Para a proteção das pessoas são criadas normas, cujo objetivo é aumentar a segurança e evitar acidentes. Estas normas podem ser nacionais ou internacionais. O conjunto de normas mais seguido na área de automação é a diretiva máquinas que legisla a segurança de soluções automatizadas. No caso da robótica as normas mais conhecidas são as normas ANSI/RIA que legislam o cuidado a ter com o uso de robôs [54].

As normas ANSI/RIA foram lançadas no ano de 1999 e sofreram uma atualização no ano de 2012. Este conjunto de normas abrange todos os intervenientes das soluções robóticas. Os intervenientes das soluções robóticas são os fabricantes, os integradores e os utilizadores. ANSI/RIA têm como objetivo definir um conjunto de regras para a segurança dos utilizadores de robôs. Pode ser dividido em três documentos: avaliação de riscos baseado em tarefas, segurança e sistemas já existentes. Nestes documentos são abordados os modos de operação, as funções de paragem dos robôs, o controlo da sua velocidade, o controlo dos atuadores, a diminuição dos riscos, entre outros parâmetros.

De acordo com as normas ANSI/RIA um robô corresponde ao manipulador sem o atuador final e quando estes se juntam formam um sistema robótico. A célula robótica é constituída pelo sistema robótico e pelas medidas de proteção.

ANSI/RIA regulamenta o uso de robôs colaborativos e o tipo de células nas quais podem ser inseridos, bem como os tipos de operações colaborativas existentes. Estas normas definem também como devem ser projetados e instalados os circuitos de emergência dos robôs bem como o seu sistema de paragem [54, 55].

É previsto nas normas ANSI/RIA, para avaliação do risco, a avaliação da gravidade da lesão. Caso uma lesão aconteça a avaliação é feita com recurso

a três níveis: lesões pequenas, lesões moderadas e lesões graves. É previsto e qual o grau de exposição, a forma de evitar o perigo e também o nível de risco resultante [55].

De acordo com estas normas, para a identificação dos perigos e avaliação dos riscos, é necessário ter em conta vários tipos de características.

As características gerais referem que operacionalmente os robôs diferem das outras máquinas devido à sua área de trabalho, referem que é necessário determinar os riscos e os perigos de forma prévia para escolha e concepção de medidas de segurança e referem também que a segurança está ligada à interação dos robôs com outras máquinas. Após a reflexão nestes conceitos é efetuado o desenho do *layout*. Após o desenho do *layout* é efetuada a avaliação de todos os riscos. Terminado este processo são identificados todos os perigos associados ao sistema. Após a identificação dos perigos é identificada a tarefa a executar e são pensadas medidas para se eliminarem os perigos e reduzirem os riscos. Para se reduzirem os riscos são analisados os requisitos de segurança e implementadas medidas de proteção. Por fim são dimensionados sistemas de controlo de segurança, devendo estes ser dimensionados com base nas condições do ambiente a operar, ter em atenção a localização dos atuadores de segurança, bem como ter atenção ao controlo da energia. Ainda no dimensionamento de segurança, é necessário ter em atenção as funções de paragem dos robôs e do sistema, bem como os processos de recuperação do sistema [55].

4.2.3 Medidas de proteção

As medidas de proteção são tomadas para evitar o máximo de riscos possíveis podendo ser passivas ou ativas [53].

As medidas de proteção passivas correspondem a um conjunto de regras que visam evitar situações que possam provocar acidentes (regras empíricas e de bom senso). Definir o tamanho das zonas de perigo do robô, limitar o espaço de trabalho dos robôs ao mínimo necessário, definir zonas de segurança e de trabalho, marcar todas as zonas, especialmente as de perigo, ter em atenção às condições de ordenamento e limpeza da célula, não operar manualmente o robô exceto em situações de programação ou erro, evitar roupas ou acessórios que possam ficar presas em algum ponto da célula e usar equipamento de proteção individual adequado, são alguns exemplos de medidas. Estas regras têm como objetivo formar os colaboradores, atribuir responsabilidades na operação e utilização dos robôs e definir posições e ambientes permitidos à montagem de robôs [53].

Para além das medidas passiva, num contexto de redução dos danos produzidos, podem ser aplicadas as medidas de proteção ativa.

As medidas de proteção ativas servem para minimizar os efeitos, ou consequências, dos acidentes quando as medidas passivas não conseguem evitá-los. O circuito de paragem de emergência, a pré-fixação de saídas, os sistemas para impedir o acesso à área dos robôs e a utilização de sistemas de proteção contra colisões são exemplos deste tipo de medidas [53].

No circuito de paragem de emergência existem dois tipos de paragem: a paragem de emergência e a paragem de segurança. No caso de uma paragem de emergência é necessário intervir manualmente e reprogramar antes de iniciar a operação. Na paragem de segurança não há necessidade de reprogramar. Os circuitos de paragem de emergência ativos são necessários tanto nos robôs como nos controladores e cada estação robotizada deve dispor do seu próprio dispositivo de paragem de emergência [53].

Na pré-fixação de saídas, fixam-se as saídas do controlador do robô que são mais relevantes para a segurança da célula [53].

Os sistemas mais utilizados para impedir o acesso à área dos robôs são as vedações da célula. Estas têm como objetivo reduzir ao mínimo possível a área de trabalho de robô e isolar as zonas de perigo. O número de entradas na vedação deve ser o mínimo possível, devem estar ligadas ao sistema de paragem de emergência e as portas devem abrir na direção contrária à da entrada do operador na célula. No caso da utilização de barreiras luminosas, cortinas de luz ou *scanners* de zona, esta deve ser feita em adição ou substituição das cancelas de segurança e devem ser ligadas ao sistema de paragem de emergência [53].

4.2.4 Interface Homem-robô

As fases em que os colaboradores estão mais sujeitos a acidentes são: a fase de programação e testes, o arranque, a manutenção e a reparação. O controlador deve ter intrínseca a funcionalidade de distinguir entre modo automático e modo de programação [53]. A fase menos propícia a acidentes é a fase de operação na qual existe pouca proximidade entre o robô e o operado. Esta proximidade só é conseguida no caso de se utilizar um robô colaborativo.

No caso da programação ser *online* os movimentos do robô devem ser efetuados a baixa velocidade de modo a permitir que o colaborador se desvie caso haja algum movimento imprevisível. Antes de iniciar o arranque do sistema e posteriormente à programação devem ser efetuados testes ao programa a baixa velocidade por forma a perceber se não vão ocorrer problemas posteriormente. O arranque do programa só deve ser permitido com recurso à função "*hold to run*" associada ao botão "*dead man jog*" presente na consola de programação. Após a programação e teste a baixa velocidade e o colaborador se colocar numa localização fora do volume de trabalho do robô, deve ser efetuada a execução do programa à velocidade de trabalho [53].

Durante o arranque do robô todas as proteções devem estar instaladas e funcionais. Deve-se verificar todos os elementos do sistema, procurando corpos estranhos, pessoas e objetos. Estes elementos não se devem encontrar na zona de perigo e deve ter-se em atenção movimentos do robô que podem danificar a célula [53].

No modo automático devem ser ativados os sistemas de segurança. O acionamento do robô através da consola deve estar impossibilitado. Este modo só deve poder ser ativo caso a célula esteja completamente em segurança, sem a presença de colaboradores no seu interior [53].

Na fase de operação devem ser cumpridas todas as normas de segurança e toda a célula deve estar sensorizada, para evitar possíveis acidentes. Esta fase pode incluir o recurso a colaboradores. Nesse caso, é necessário dimensionar a célula e o sistema de segurança para privilegiar a segurança do trabalhador, por forma a evitar que ocorra qualquer acidente. Nesta fase o robô pode necessitar de mudar a ferramenta de trabalho; para tal deve efetuá-lo numa zona específica para o trabalho [53].

Para se evitarem os acidentes durante as fases de manutenção e reparação, devem ser seguidas um conjunto de normas que tornem mais seguras estas fases. Nestas fases o robô deve estar desligado e o seu arranque deve ser prevenido. O robô deve ser movimentado apenas em modo manual e a baixa velocidade. De preferência apenas um colaborador deve estar na zona de perigo do robô. Nenhum equipamento de segurança deve ser desativado ou retirado sem autorização prévia. Não deve ser efetuada nenhuma conversão ou modificação ao sistema sem autorização. Após tudo isto o sistema deve ser reiniciado com todos os sistemas de segurança ativos [53].

4.3 Casos de estudo da aplicação *pick and place*

4.3.1 ABB - *picking pizza pick and place*

4.3.1.1 Problema

Este caso de estudo tinha como desafio a criação de um sistema automatizado de formação de embalagens de pizza. O sistema foi desenvolvido pela empresa integradora italiana Vortex para a empresa produtora de pizzas congeladas, Panidea.

As pizzas percorreriam uma linha de produção provenientes de um frigorífico e seriam carregadas numa máquina de *flow-pack* antes de prosseguirem para uma máquina de embalagem. O maior desafio inerente a este caso consistia no manuseamento de produtos de diferentes formas: pizzas triangulares, ovais e circulares. A empresa Panidea pretendia um sistema totalmente flexível e escalá-

vel, em que novos produtos, com novos tamanhos, formas e tipos, pudessem ser introduzidos e também que fosse possível aumentar a capacidade do sistema.

O desafio inerente às pizzas residia no facto de não estarem colocadas perfeitamente em linha e de não possuírem forma uniforme. Havia também necessidade de manusear as pizzas com cuidado, tendo em conta a necessidade de uma garra específica. Para a resolução deste problema, a Vortex optou pelo uso do sistema FlexPicker da ABB e do *software* PickMaster para orientação visual do sistema [56].

4.3.1.2 Componentes da solução

O FlexPicker é um robô paralelo que possui como qualidades a flexibilidade e a velocidade de operação. Este robô possui rácios de *picking* que excedem as 120 peças por minuto, sendo que as peças são apanhadas e largadas uma a uma. Devido ao facto dos motores e engrenagens se encontrarem fixados à base do robô, este possui uma capacidade de carga reduzida, o que pode permitir acelerações acima dos 10 g. Este robô possui um projeto higiénico, sem superfícies pintadas, sendo que a proteção que possui pode ser lavada, desde que a baixas pressões, tornando-o ideal para a aplicação. Como este é montado no topo de uma estrutura elimina as restrições de acesso à linha. O controlador utilizado, IRC5, possui capacidade de rastreio do tapete da linha, permitindo o *pick and place* enquanto o tapete se movimenta. Assim eliminam-se os tempos perdidos por paragem e acionamento do tapete [56].

O PickMaster é um sistema de visão que fornece ao robô uma capacidade visual. Devido ao facto dos produtos estarem dispostos no tapete de forma aleatória, é necessário dar ao robô a informação sobre a sua localização no tapete, como se observa na Figura 4.7. O PickMaster é um *software* que incorpora um sistema de visão da Cognex. Este *software* facilita a programação de vários robôs, câmaras e tapetes [56].



Figura 4.7: Imagem da linha implementada [56].

4.3.1.3 A solução apresentada pela Vortex

A solução apresentada pela Vortex assenta numa parceria entre a empresa e a ABB. Inicialmente a Vortex tinha o seu mercado na indústria automóvel, na qual integrava robôs da ABB. Mais tarde alargou o mercado à indústria alimentar. A Vortex optou por iniciar uma parceria com a ABB com o objetivo de competir com as empresas integradoras alemãs e suíças [56].

O sistema apresentado pela Vortex, na fase de carga, é composto por dois robôs paralelos ABB FlexPicker e um sistema de visão PickMaster constituído por uma câmara por cada robô. Caso a Panidea pretenda melhorar o sistema, este vem preparado com espaço para acrescentar mais um robô no fim da linha. Este sistema foi pioneiro em Itália aquando da sua implementação. O sucesso da solução reside na forma da garra utilizada. Apesar das garras individuais se apresentarem mais económicas que os sistemas multi-garra, era necessário ter em conta a capacidade e a variação do produto que influenciam a escolha da garra e torna essa tecnologia muito importante. Neste projecto são utilizadas duas garras dependendo do produto em causa. Em alguns produtos é utilizada uma garra composta por dedos e noutros utilizada uma garra a vácuo. A flexibilidade era um dos objectivos do cliente.

Nesta linha são produzidos e empacotados quatro tipos de produtos diferentes, e a mistura de produtos por caixa é configurada pelo utilizador. Cada robô possui a capacidade de movimentar 60-80 pizzas por minuto, dependendo da garra que está em utilização. Caso haja a adição de um terceiro robô, a capacidade da linha sobe para 240 pizzas por minuto. As câmaras e o sistema de visão da Cognex fornecem ao controlador dos robôs as posições das pizzas, evitando assim a necessidade de guiar as pizzas através de componentes de automação tradicional. O resultado é um sistema baseado em robótica, sem recurso a periféricos mecânicos. Em suma, os benefícios que apresenta são: flexibilidade, redução de custos com trabalhadores e capacidade de rápida mudança do sistema [56].

4.3.2 Uma solução para a indústria do chocolate

Esta aplicação tem por base o transporte de materiais e o empacotamento, na indústria alimentar.

4.3.2.1 Problema

O cliente da empresa Han Shin Power Tech localiza-se no mercado dos chocolates e possuía um sistema manual, com recurso a colaboradores, de empacotamento de chocolates, que chegavam aos colaboradores através de um tapete de alta velocidade. Devido ao facto deste cliente produzir vários produtos diferentes em grandes volumes sentiu necessidade de automatizar o seu processo.

Esta necessidade surge em consequência do facto da chegada dos chocolates a alta velocidade contribuir para um elevado *stress* na tarefa executada pelos trabalhadores. Esta tarefa de *pick and place* manual a alta velocidade apresentava uma taxa de erro de 2% e uma elevada probabilidade de dano no produto. A qualidade dos pacotes baixava com a chegada ao fim de cada turno. O ponto positivo da tarefa executada por colaboradores era a flexibilidade de trabalhar com diferentes produtos [57].

4.3.2.2 Componentes da solução

Nesta solução são utilizados o robô Adept Quattro s650HS, o controlador da Adept SmartController CX e o pacote de *software* ACE PackXpert [57].

O robô Quattro s650HS da Adept é um robô paralelo de quatro eixos. Este robô tem uma capacidade de carga de 2 kg podendo, no limite, manipular 6 kg de carga e tem como principais características a elevada velocidade de execução e uma área de trabalho alargada. Este equipamento possui qualidades que permitem que não haja desenvolvimento de culturas bacterianas ou contaminantes. O s650HS possibilita ainda o rastreio de produtos através do uso de sistemas de visão e possui também a capacidade de comunicar por Ethernet/IP e DeviceNet [57].

O controlador SmartController CX é um controlador ultra compacto, com elevado desempenho no controlo de movimentos de um robô. Este controlador possui uma arquitetura distribuída com *software* embebido, possui integrado um sistema de rastreio de tapetes e um módulo de comunicações de alta velocidade para integração entre o sistema de automação e um *software* baseado em PC. Este controlador pode controlar todo um processo, incluindo outros robôs e periféricos. Tem capacidade para controlar até 15 robôs e a capacidade de integrar com sistemas de visão. O SmartController CX possui integração com o *software* Adept ACE para permitir o controlo de mecanismos complexos através de um *software* de uso amigável [57].

O *software* Ace PackXpert da Adept foi projetado para gestão de linhas de empacotamento, desde a integração e implementação através da operação. A aplicação permite que o utilizador navegue por diferentes configurações, ao definir parâmetros específicos do processo, como controladores, robôs e tapetes. Este *software* providencia uma programação de alto nível baseada em configurações de sistema [57].

4.3.2.3 A solução apresentada pela Han Shin Power Tech Company

A solução apresentada, como se verifica na Figura 4.8, baseia-se na utilização de robôs que cumprem regras específicas de higiene para efetuar a tarefa de embalagem. Foi desenvolvido um sistema compacto e simples com recurso a

robôs do tipo paralelo. Foram também desenvolvidas várias receitas por forma a satisfazer as variações de produto, através de simples cliques na interface com o utilizador. Este sistema foi implementado em três meses, automatizando completamente a linha. Nesta solução os robôs paralelos da Adept são controlados pelo controlador anteriormente referido e geridos pelo *software* da Adept que permitiu o controlo também de tapetes, câmaras e ferramentas [57].



Figura 4.8: Linha implementada pela OMRON [57].

4.3.3 Redução do tempo de programação do robô

Este caso de estudo nasce da parceria das empresas Mitsubishi Electric e R+D Custom Automation com o objetivo de desenvolvimento de soluções de fácil aplicação e de uso generalizado.

4.3.3.1 Problema

A Mitsubishi Electric e a R+D Custom Automation decidiram tentar ultrapassar um desafio da robótica que consistiu em criar um sistema de *pick and place* guiado por visão artificial personalizado. Esta colaboração permitiu a ambas as empresas aprofundarem as suas relações, dando a oportunidade da R+D trabalhar com os melhores produtos da Mitsubishi, e deu a oportunidade à Mitsubishi de ver a sua tecnologia aplicada por um integrador [58].

4.3.3.2 Componentes da solução

Neste caso foram utilizados dois robôs SCARA da Mitsubishi: RH-3FH35 e RH-3SDHR, e um sistema de visão Cognex In-Sight 7000 [58].

O RH-3FH35 é um robô horizontal SCARA de montagem no chão. Possui uma capacidade de carga de 3 kg e 4 eixos. Com uma carga de 2 kg o seu tempo de ciclo é de 0,41 s e pode atingir velocidades de 6800 mm/s. Tem um alcance máximo de 125 mm, com uma área de cobertura de 350 graus [58].

O RH-3SDHR é um robô SCARA montado na parte superior de uma superfície de pequenas dimensões, por forma a aproveitar melhor o espaço em comparação com um robô paralelo. Este robô apresenta o controlador integrado.

É um robô de 4 eixos, com alcance de 175 mm e capacidade de carga de 3 kg. Com uma carga de 1 kg apresenta um tempo de ciclo de 0,32 s e pode atingir velocidades de 6267 mm/s [58].

O sistema de visão Cognex In-Sight 7000 corresponde a um sistema composto por câmaras monocromáticas ou a cores, com capacidade de aquisição de 53 a 217 frames por segundo, e possuem a capacidade de ligação em rede com outros dispositivos [58].

4.3.3.3 A solução apresentada pela R+D Custom Automation

A R+D apresentou um sistema que apresentava robustez e flexibilidade. Foram utilizados os robôs SCARA da Mitsubishi com o objetivo de aumentar a velocidade do sistema no processo de *pick and place* entre os tapetes de entrada e saída, e estes robôs ajudam a minimizar o movimento do trajeto aumentando o rendimento. Os produtos provenientes do tapete de entrada tinham disposições aleatórias. Como consequência deste facto, foi utilizado o sistema de visão por forma a fornecer as coordenadas dos produtos. O RH-3SDHR é utilizado em sincronismo com o tapete de entrada levando os produtos, um de cada vez, para serem colocados no transportador de saída. Os produtos viajam em direção ao RH-3FH35 que efetua o *pick and place* para serem orientados de forma aleatória no tapete de entrada. Este processo repete-se à medida que os produtos circulam na máquina [58].

Esta solução, cujos robôs se visualizam na Figura 4.9, permitiu a criação de um modelo de solução de *pick and place* que aumenta a produtividade com operações de velocidades elevadas, com uma aplicação e programação otimizadas e possibilidade de aumentar a eficiência das operações [58].



Figura 4.9: Linha de *pick and place* adaptável [58].

4.4 Casos de estudo de paletização

4.4.1 Embalar e paletizar café: Löfbergs Lila

A Löfbergs Lila é uma empresa Sueca presente no mercado do café, sendo o segundo maior produtor na Suécia. Esta empresa possui uma das fábricas de produção mais modernas da Europa.

4.4.1.1 Problema

A Löfbergs Lila é um empresa bastante automatizada e com níveis de produção bastante elevados. Esta empresa embala e paletiza cerca de 16700 toneladas de café por ano. O peso de cada pacote varia entre os 50 g e os 6 kg. Estes pacotes são agrupados em caixas e paletizados para distribuição. Devido aos elevados níveis de produção e movimentação dos produtos por tapetes localizados em toda a fábrica, era necessário desenvolver um sistema que auxiliasse no empacotamento e paletização do café [59].

4.4.1.2 Componentes da solução

Nesta solução são utilizados três robôs da ABB: IRB 6650, IRB 640 e IRB 340 FlexPicker.

O IRB 6650 é um braço robótico com um alcance de 3,9 m e capacidade de carga de 90 kg a 200 kg, dependendo do modelo. Possui 6 eixos e tem a capacidade de os alinhar na horizontal ou vertical, esticando o braço nesses sentidos. Possui uma repetibilidade de 0,13 mm em termos de posição e de 0,9 mm em termos de trajetória [59].

O IRB 360 é um robô do tipo paralelo, com uma capacidade de carga que varia entre 1 kg e os 8 kg, dependendo do modelo. É um robô de 4 eixos que possui um diâmetro de alcance que varia entre os 800 mm e os 1600 mm, consoante o modelo escolhido. Possui uma repetibilidade de 0,1 mm no caso da posição e o seu tempo de ciclo varia entre os 0,3 s e os 0,92 s consoante o modelo e a carga. O IRB 640 é um robô articulado semelhante ao IRB 6650 [59].

4.4.1.3 A solução apresentada pela ABB

A ABB apresentou uma solução baseada num robô paralelo e num braço robótico para aumentar a flexibilidade na paletização [59].

O IRB 640 tem a função de pegar nas caixas de cartão no fim da linha e colocá-las numa europaleta até esta estar completa para a distribuição. Este robô foi programado para executar sete ciclos por minuto, podendo trabalhar ainda mais rápido e possui a capacidade de carregar paletes vazias. Estes aumentaram a flexibilidade na paletização uma vez que permitem paletizar diferentes tipos de caixas [59].

O IRB 6650 é utilizado num processo de despaletização. O IRB 6650, despaletizando os produtos e enviando-os, através de tapetes, para o IBR 640 para serem paletizados de acordo com os padrões definidos pelos clientes. O padrão é definido pelo envio de produtos por parte do IRB 6650 [59].

O IRB 360 FlexPicker efectua o *pick and place* de sacos de café de 250 g. O objetivo é colocar esses sacos em caixas de cartão, num total de 16 sacos por caixa. A caixa é enviada espalmada pelo tapete, na qual o IRB 360 coloca os sacos utilizando todo o espaço disponível da caixa que será montada adiante pela máquina [59].

Desta forma, ao conjugar os três robôs foi desenvolvido um sistema que pode operar com diversos produtos e paletizá-los de acordo com os padrões pretendidos pelos clientes [59].

4.4.2 Robô de longo alcance da KUKA melhora o desempenho da paletização por camadas

Um empresa alemã de produção de produtos à base de vinho, a Hauser GmbH, tinha a necessidade de melhorar o sistema de paletização que possuía.

4.4.2.1 Problema

A necessidade de instalar um robô para paletizar sacos caixas frágeis numa palete. Era necessário que este robô fosse flexível nas tarefas a executar, uma vez que há necessidade de paletizar dois tipos diferentes de produto. Era necessário que possuísse a capacidade de criar padrões diferentes de paletes [60].

4.4.2.2 Componentes da solução

Nesta solução é utilizado um robô da KUKA, o KR 160. Este robô tem um alcance máximo de 1573 mm e uma repetibilidade de 0,06 mm em termos de posição. Este robô possui 6 eixos e uma capacidade de carga de 160 kg e é de montagem no chão. O controlador utilizado com este robô é o KR C4 [60].

4.4.2.3 A solução apresentada pela KUKA

Inicialmente um colaborador utilizava um paletizador para construir as camadas de caixas na palete. Devido ao aumento de produção e necessidade de aumentar o rendimento, tornou-se necessário robotizar a linha. Com a introdução de novos produtos era necessário reconfigurar o paletizador. No caso do robô é necessário trocar apenas a pinça e o programa [60].

Para suprir tal necessidade foi utilizado o KR 160 que possui um bom alcance. Este robô foi utilizado para manipular caixas de cartão cheias de pacotes de vinho. Cada caixa pode incluir 12 pacotes de 1 l de vinho ou 8 pacotes de 1,5 l de chá gelado. No total, o robô consegue paletizar 1850 caixas de cartão por

hora. Foi implementada uma célula completa composta por tapetes automatizados, robô e seu controlador e também dispositivos de segurança. Os tapetes transportam as caixas desde a produção até à célula robótica para serem aí paletizadas. O robô fica entre ambos os terminos dos tapetes e paletiza segundo um padrão que consiste em duas filas de 6 caixas colocadas transversalmente e uma fila de 4 caixas colocada longitudinalmente, repetindo-se este padrão camada a camada. Toda a célula foi projetada pela KUKA [60].

Os critérios para a escolha deste robô residiram na proximidade geográfica entre as empresas e a relação qualidade/preço. Outro critério que satisfaz o cliente foi o facto de necessitar de pouca manutenção, o que contribui para menores tempos de paragem [60].

4.4.3 Integração de um robô colaborativo numa aplicação de paletização

4.4.3.1 Problema

Tipicamente os robôs de paletização necessitam de células de maiores dimensões. A empresa Nortura necessita de um processo de paletização contínuo, sem parar a linha de produção. Porém necessitava que a célula fosse flexível o suficiente para utilizar o espaço da paletização para outras tarefas, quando não existissem paletes naquele espaço. Com um limite no orçamento, a Nortura desafiou a Universal Robotics a encontrar um robô de 6 eixos que possuísse o alcance e a capacidade de carga necessária para a criação de pilhas de caixas, de tamanhos e massas diferentes, e alturas diferentes da paleta. Era necessário também que o robô fosse fácil de programar e integrar com sistemas de visão, e executasse a sua função com o mínimo de supervisão necessária [61].

4.4.3.2 Componentes da solução

Os componentes utilizados nesta solução são o robô UR10 da Universal Robots e o O2D222 que é um sensor de objetos da IFM [61].

O UR10 é um robô colaborativo, que possui uma repetibilidade de 0,1 mm, uma capacidade de carga de 10 kg, 6 eixos e um alcance de 1300 mm. Este robô é integrado sempre com o controlador e *teach-pendant* correspondentes [61].

O O2D222 é um sensor de objetos que pode operar entre os 50 mm e os 2000 mm. Em termos de distância de trabalho, o seu campo de visão varia entre os 32x24 mm e os 1320x945 mm, consoante a distância de trabalho, o menor objeto detetável varia entre os 0,3 mm e os 8,0 mm de acordo com a distância de trabalho e a repetibilidade varia entre os 0,075 mm e os 2,0 mm. A resolução da imagem capturada é 640x480 píxeis. A taxa de leitura máxima é de 20 Hz [61].

4.4.3.3 A solução apresentada pela Universal Robots

A Nortura pretendia um sistema de paletização otimizado. Não era possível a utilização de um robô de paletização vulgar, pois a célula de trabalho característica ocuparia demasiado espaço. A empresa Norwegian Robotics Integrator Rockerfarm AS dimensionou um sistema de visão personalizado para complementar o sistema proposto pela Universal Robots. O sistema da Universal Robots tem na base um robô colaborativo para uma abordagem inovadora que ocupe o menor espaço possível. A câmara é montada na célula, na zona superior ao robô que se encontra colocado no chão da célula. A garra utilizada para esta aplicação é uma garra de sucção. O espaço ocupado pelo robô é de 5 m². O espaço de trabalho utilizado está marcado no chão e corresponde a cerca de 10% a 20% do espaço utilizado numa célula de paletização vulgar. Quando uma palete vazia é colocada no pavimento da célula o sistema de visão deteta a mesma, e também as caixas que se movem pelo tapete, e dá início ao processo de paletização. O robô escolhido de 6 eixos possui um programa personalizável para a criação de camadas de tipos diferentes de caixas, num padrão que o cliente que irá adquirir a paleta desejar. O robô consegue formar 20 paletes por dia, ou seja, paletiza cerca de 1700 caixas por dia. Uma vantagem adicional que este sistema oferece é a capacidade de deteção de erros das máquinas que se encontram na linha [61].

O ponto forte deste sistema é a sua capacidade de trabalhar de uma forma segura ao lado dos colaboradores, uma vez que o robô escolhido é colaborativo. A Nortura prevê uma recuperação do investimento em 1 ano de operação [61].

Este sistema mostrou-se colaborativo e seguro, compacto e flexível, de custo reduzido e confiável [61].

4.5 Conclusão

Neste capítulo foi apresentado um conjunto de aplicações nas quais existe uma maior frequência na utilização de robôs. Deste modo verifica-se a sua versatilidade e que são úteis para tarefas repetitivas ou de elevado risco. De seguida conclui-se que a utilização de robôs acarreta risco para os seres humanos e é necessário evitar o contacto entre ambos quando a situação possa levar à lesão de um ser humano. Para evitar este tipo de situações foi criado um conjunto de normas que visam a proteção dos operadores e de tudo o que tenha contacto com os robôs. Por fim foram analisados casos de estudo relativos às tarefas de manipulação de materiais através dos quais é perceptível que os robôs mais apropriados para este tipo de tarefas são aqueles que permitem tempos de ciclo mais baixos e com elevada repetibilidade.

Capítulo 5

Arquitetura do Sistema

Neste capítulo, inicialmente, é exposto o problema a resolver e de seguida é exposta a arquitetura do sistema desenvolvido. Para tal é feita uma descrição de cada componente do sistema. Após a descrição dos blocos constituintes do sistema, procede-se à descrição do material utilizado, quer seja tangível ou intangível. Neste capítulo é também apresentado o conjunto de procedimentos adotados para a execução de todo o projeto

5.1 Exposição do problema

O sistema desenvolvido visa a montagem de 8 pinças kocher em 12 segundos de ciclo. A montagem manual ou através do auxílio de uma máquina que procede à montagem e fecho da pinça com atuação do operador colocando as peças na posição correta produz um tempo de ciclo superior. A temperatura elevada a que as peças saem da máquina de injeção e a reduzida folga existente no local de união entre as peças macho e fêmea criam dificuldades ao processo. Para se eliminarem estas dificuldades foi proposto um sistema composto por um robô e uma mesa de montagem que procedem à montagem das pinças, o seu fecho e colocação numa caixa de produção, no tempo de ciclo enunciado.

O processo anterior consistia na montagem de 4 pinças em 16 s por parte de um operador com o auxílio de uma máquina que procedia ao fecho das pinças. As peças constituintes eram injetadas num molde que permita o fabrico de 4 pinças por ciclo que eram montadas pelo operador e colocadas numa máquina que procedia ao seu fecho. O operador precisava de alimentar a máquina com 4 pinças para fecho que seriam posteriormente fechadas ao mesmo tempo.

As peças constituintes destas pinças são injetadas numa máquina de injeção de plástico. A máquina procede à injeção do plástico num molde por forma a produzir as peças que serão montadas posteriormente.

O processo de extração e montagem das pinças pode ser dividido em duas fases. Numa primeira fase o robô de seis eixos efetua a descarga da máquina de injeção e coloca as peças numa mesa de montagem, desenvolvida na RobotSol. Numa segunda fase o robô paralelo procede à montagem das pinças e o fecho e dispensação, para a caixa de produção, são efetuados pela base. O fecho é efetuado por pinos acionados pneumáticamente e a dispensação é efetuada por um varredor, também este atuado pneumáticamente.

A mesa é composta por duas bases por forma a permitir que ambos os robôs atuem em paralelo na mesa de montagem, ocupando uma base cada sem que nunca ocupem a mesma. Desta forma cada vez que o robô responsável pela descarga das peças, as descarrega e coloca numa base, o robô paralelo deve estar a proceder montagem das pinças na outra base. Tendo por objetivo a diminuição do tempo de ciclo do sistema.

No caso da implementação do sistema se mostrar bem sucedida, existe uma proposta de melhoria da flexibilidade do sistema. Para isso tomou-se em consideração a adaptação do sistema a mais dois tipos diferentes de pinça. Para cada modelo desenvolver-se-á um novo modelo de base de montagem e novos atuadores finais para ambos os robôs. Devido ao tamanho destes novos modelos de pinça existirá apenas uma base por modelo para a montagem das pinças.

5.2 Composição do sistema

O sistema global, esquematizado na Figura 5.1, é constituído por dois robôs, um autómato programável (PLC), uma máquina de injeção, componentes responsáveis pela segurança e uma mesa automática de montagem de pinças. A mesa de montagem é composta por duas bases de montagem controladas pelo PLC que, para além disso, faz a interface entre os robôs e a gestão do sistema global. Na máquina são colocados os moldes, para a produção das pinças, nos quais é injetado o plástico.

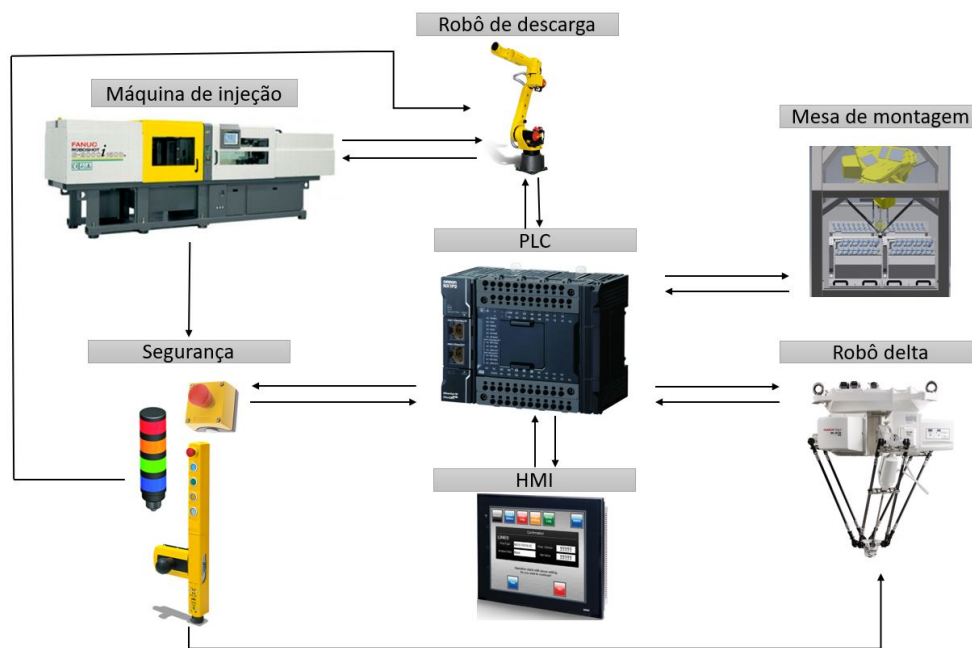


Figura 5.1: Diagrama de blocos do sistema global.

O subsistema desenvolvido para montagem de pinças é constituído por um robô e uma estrutura desenvolvida para a montagem. Esta subsecção do sistema total é alimentada por um manipulador robótico que executa o descarregamento de uma máquina de injeção de plástico. Nesta máquina são produzidas as peças que constituem o produto final e que serão transportadas pelo manipulador robótico para a estrutura de montagem na qual está inserido um robô delta que procederá ao *pick and place* das peças e à sua montagem.

Na mesa estão presentes um vasto conjunto de sensores cujo objetivo passa por detetar a posição das bases, gaveta, caixas e os componentes ativos de cada base (pinos de fecho, guias e empurrador) e detetar também a presença de peças nas bases.

Do ponto de vista da segurança este sistema conta ainda com barreiras para limitação do acesso ao espaço de trabalho dos robôs e uma porta de segurança para acesso a este mesmo espaço. O sistema de segurança inclui também as botoneiras de segurança presentes ao longo da célula e os sensores de porta colocados na porta de acesso à mesa. Estas botoneiras encontram-se ligadas a um relé de segurança. Da mesma forma, as portas de acesso à mesa e porta de segurança encontram-se ligadas também a um relé de segurança, bem como os sinais de segurança provenientes da máquina de injeção.

5.3 Hardware

Como referido o sistema é composto por dois robôs, uma máquina de injeção de plástico, uma mesa de montagem de pinças, componentes de controlo e interface e os componentes responsáveis pela segurança da célula. Os robôs utilizados são: o manipulador robótico série FANUC M20iA/20M e o manipulador robótico paralelo FANUC M2iA/3SL.

5.3.1 FANUC M20iA/20M

O robô M20iA/20M da marca FANUC, presente na Figura 5.2, possui as seguintes características [62]:

- Repetibilidade: $\pm 0,03$ mm
- Número de eixos: 6.
- Capacidade de carga: 20 kg.
- Alcance 1813 mm.
- Intervalo de temperatura ambiente suportado: 0 a 45 °C.
- Controlador: R30iB



Figura 5.2: Robô FANUC M20iA/20M [62].

5.3.2 FANUC M2iA/3SL

O robô M2iA/3SL, presente na Figura 5.3, é do tipo paralelo com as seguintes características [63]:

- Repetibilidade $\pm 0,01$ mm.
- Número de eixos: 4.
- Capacidade de carga: 3 kg.
- Alcance: 1130 mm.
- Intervalo de temperatura ambiente suportado: 0 a 45 °C.
- Controlador: R30iB.

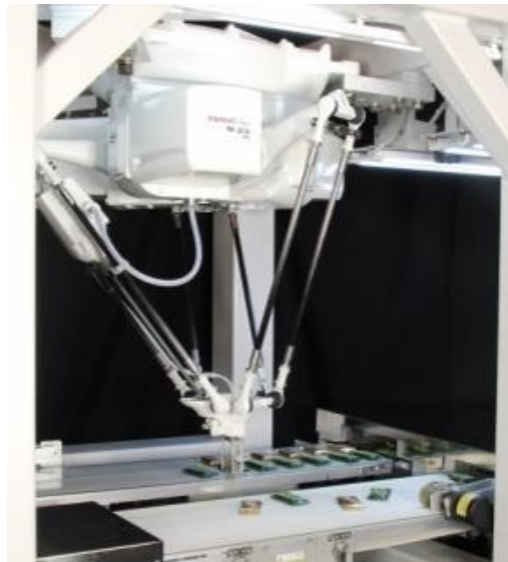


Figura 5.3: Robô FANUC M2iA/3SL.

O M2iA possui um volume de trabalho reduzido. Este volume é caracterizado pelo formato de um cilindro cujo diâmetro diminui entre os 250 mm e os 400 mm de distância do limite superior do volume de trabalho. Através da Figura 5.4, verifica-se que o diâmetro superior do volume de trabalho é de 1130 mm e o diâmetro inferior é de 830 mm. O volume de trabalho possui uma altura de 400 mm [63].

O corpo deste robô apresenta um grau de proteção de IP69K. Este código é atribuído quando o robô possui proteção contra poeiras e água proveniente de

jatos de vapor e alta pressão. Possui também uma proteção direcionada à área alimentar que lhe confere resistência contra desinfetantes ácidos ou alcalinos. Para evitar o desgaste, todas as juntas estão protegidas por estruturas rígidas que impedem o contacto das juntas com elementos abrasivos [63].

O M2iA é composto por elos de massa reduzida que lhe conferem elevada velocidade de movimento do punho. O punho possui massa reduzida e apresenta um orifício para permitir a passagem de tubagem e cablagem para o atuador final. Um característica do M2iA é a capacidade de atingir acelerações elevadas permitindo atingir a velocidade máxima num curto espaço de tempo [63].

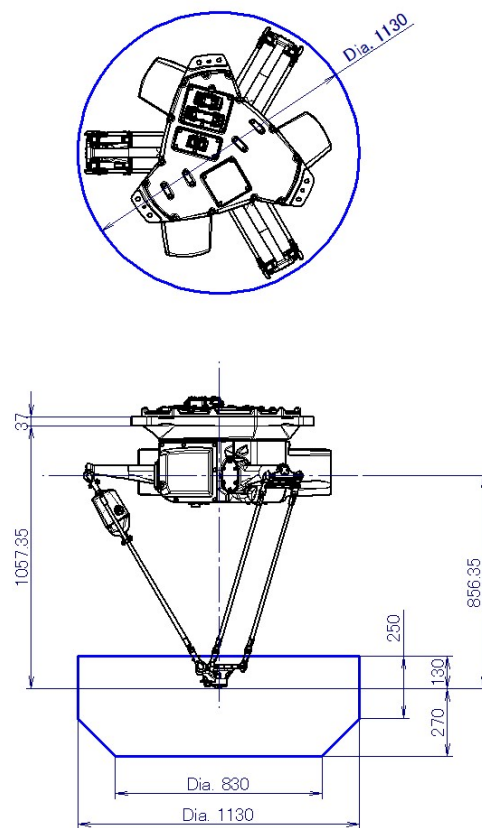


Figura 5.4: Volume de trabalho do robô FANUC M2iA/3SL [63].

Relativamente ao momento de inércia, o robô M2iA/3SL possui dois modos de operação: o modo de inércia padrão e o modo de alta inércia. Ambos os modos baseiam-se numa relação entre a variação do posicionamento do centro de massa e da capacidade de carga do robô [63].

No modo de inércia padrão verifica-se que o limite para a capacidade carga máxima se localiza a 1 cm de distância do centro em relação ao eixo dos Z e a

cerca de 5,5 cm em relação ao plano X,Y e com um momento de inércia relativo ao eixo dos Z menor que $0,0032 \text{ kg.m}^2$ [63].

Como se verifica na Figura 5.5, à medida que se aumenta a distância do centro de massa ao centro do punho existe uma diminuição da capacidade de carga máxima suportada, bem como do momento de inércia suportado. No limite o centro de massa pode encontrar-se a 7 cm de distância em relação ao eixo dos Z e a 15 cm em relação ao plano X,Y. Neste caso o momento de inércia máximo admissível é de $0,001 \text{ kg.m}^2$ e a capacidade de carga máxima é de 0,1 kg. Verifica-se que no limite a capacidade de carga diminui para cerca de 3% da capacidade de carga máxima do robô e o seu momento de inércia diminui para cerca de 31% do seu valor máximo admissível [63].

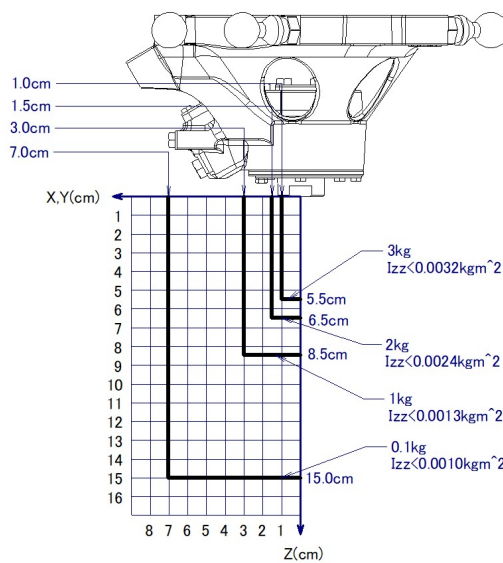


Figura 5.5: Diagrama de carga do modo de inércia padrão [63].

No modo de alta inércia verifica-se que o limite para a capacidade de carga máxima aumenta. A distância em relação ao eixo dos Z aumenta para 1,6 cm e a distância referente ao plano X,Y para os 11 cm, porém o momento de inércia máximo diminui para os $0,0022 \text{ kg.m}^2$ [63].

De forma semelhante ao modo de inércia padrão, no modo de alta inércia com o aumento da distância do centro de massa ao centro do punho diminui a capacidade de carga e o momento de inércia, como se verifica na Figura 5.6. Neste modo o momento de inércia tem de estar compreendido entre os valores 0 e $0,0022 \text{ kg.m}^2$. Porém o limite máximo em termos de distância localiza-se a 7 cm do centro em relação ao eixo dos Z e a 30 cm em relação ao plano X,Y. As capacidades de carga máxima referentes a cada conjunto de valores de distância

e de momento de inércia são semelhantes aos do modo de inércia padrão, sendo no limite a capacidade carga máxima 0,1 kg [63].

No caso da utilização deste modo existe uma contrapartida que se trata de existir uma diminuição da velocidade máxima atingível pela junta 4, que no caso está limitada aos 1750 °/seg [63].

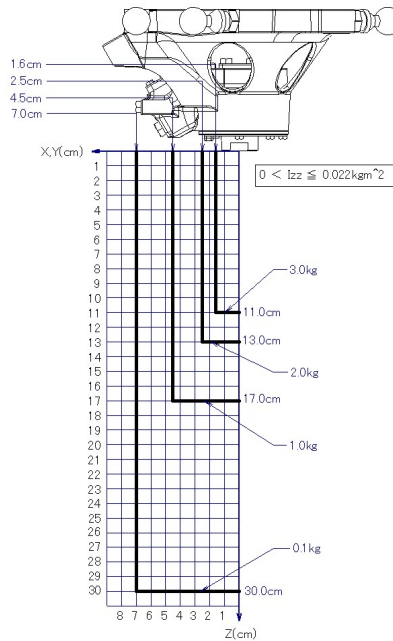


Figura 5.6: Diagrama de carga do modo de alta inércia [63].

5.3.3 Máquina de injeção

A máquina de injeção de plástico utilizada é a S-2000i150B da marca Fanuc. Esta máquina é responsável pela injeção do plástico no molde para a formação das pinças. Esta máquina possui um curso de injeção máximo de 150 mm e o seu ciclo de trabalho demora cerca de 12 segundos [64].

5.3.4 Mesa de montagem

A mesa de montagem é constituída por um conjunto de portas de acesso, duas bases de montagem de pinças com comprimento de 13 cm, gaveta para a colocação de caixas para produção e caixa de rejeição. A mesa possui ainda duas bases para troca rápida para a produção de mais dois tipos diferentes de peça.

Cada uma das bases para a montagem de pinças de 13 cm de comprimento é composta por duas bases de alumínio, duas guias, oito pinos de fecho e um empurrador. As bases de alumínio são o local onde são pousadas as pinças

fêmea e macho. As guias são utilizadas para guiar as peças macho, mantendo-as fixas em pontos específicos da base. Em cada uma destas guias estão montados quatro pinos para efetuar o fecho das pinças após a montagem. Associado a cada base de peças macho encontra-se um empurrador para conduzir as peças para a caixa. Cada um destes elementos encontra-se ilustrado na Figura 5.7.

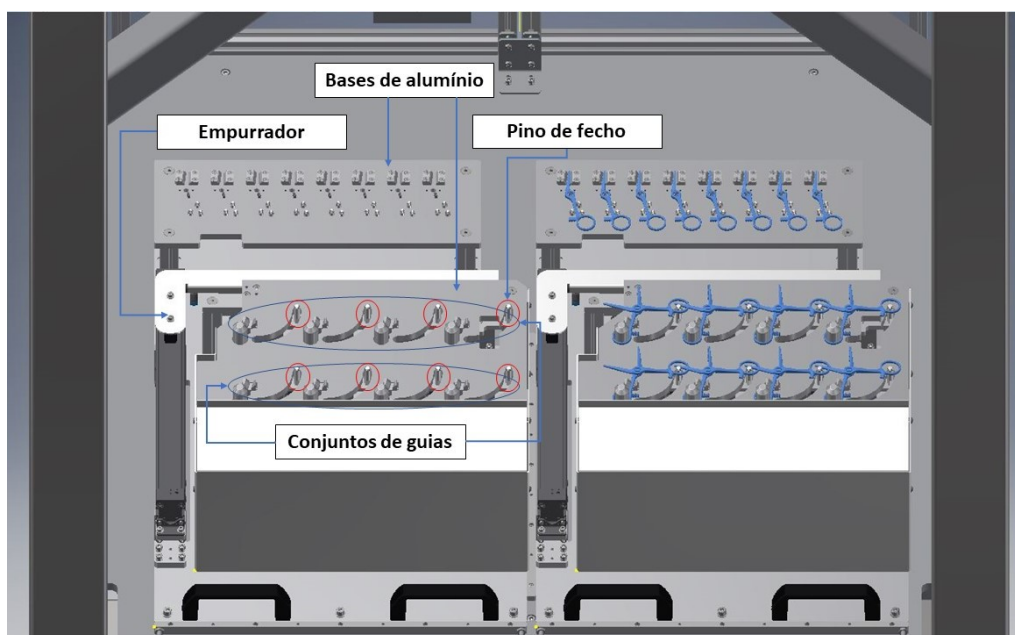


Figura 5.7: Esquema da mesa de montagem de pinças.

A cada elemento móvel de cada base estão associadas eletroválvulas para controlo da atuação dos cilindros. Em cada base estão presentes seis eletroválvulas biestáveis. Cada guia tem associada a si uma eletroválvula, cada conjunto de quatro pinos de fecho também tem associado a si uma eletroválvula e o empurrador tem associado a si as restantes duas eletroválvulas.

A eletroválvula utilizada é o modelo VUVG-BK10-B52-T-F-1R8L-S da Festo, ilustrada na Figura 5.8. Estas eletroválvulas são biestáveis, com tamanho de 10 mm e vedação em anel. Operam a 24 V DC e possuem dois conectores de 3 pinos M8 [65].

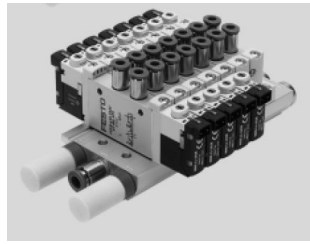


Figura 5.8: Conjunto de eletroválvulas biestável da Festo.

Para a percepção do movimento do cilindro são colocados sensores, a fim de perceber se este se encontra na posição pretendida. Através destes sensores é possível perceber se as guias se encontram numa posição superior ou inferior, se os pinos de fecho se encontram numa posição avançada ou recuada e se o empurrador se encontra numa posição superior ou inferior, avançado ou recuado.

O sensor utilizado para estas tarefas é o modelo SIES-8M-PO-24V-K-0,3-M8D da Festo, presente na Figura 5.9. Este sensor possui uma saída em PNP cuja tensão de saída é de 24V quando está ativo. Pode comutar entre estados a uma frequência máxima de 4500 Hz e pode operar numa gama de temperaturas compreendida entre os -25 e os 70°C [66].



Figura 5.9: Sensor de proximidade da Festo [66].

Para garantir a localização das bases e da gaveta é necessário a utilização de sensores. O sensor utilizado para este efeito é o sensor indutivo DW-AS-623-M8-001 da Contrinex, exemplificado na Figura 5.10 [67]. Este sensor do tipo PNP opera a um distância máxima de 2 mm, ou seja só deteta numa gama de 0 a 2 mm. Possui uma frequência de comutação máxima de 5000 Hz [67].



Figura 5.10: Sensor indutivo da Contrinex [67].

No caso da gaveta existe também a necessidade de detetar a existência ou não de caixas para colocar a produção; esta deteção é feita pelo sensor LHR-C23PA-PMS-403 da Contrinex, ilustrado na Figura 5.11. O intervalo de deteção deste sensor, do tipo PNP, localiza-se entre os 10 e os 300 mm e a deteção é feita através da emissão de um feixe de luz vermelha. A frequência máxima de comutação deste sensor é de 1000 Hz e pode operar em ambientes com temperaturas compreendidas entre os -25 e os 65 °C [68].



Figura 5.11: Sensor fotoelétrico da Contrinex [68].

Como grande parte dos constituintes utilizam a energia pneumática, é necessário ter em atenção à pressão de ar no sistema. Esta é medida através de um pressoestato SPAN-P10R-G18M-PNLK-PNVBA-L1 da Festo. Este sensor, presente na Figura 5.12, tem a capacidade de medir pressões entre os 0 e os 10 bar de pressão e permite operar num intervalo de temperatura compreendido entre os 0 e os 50 °C. Este sensor permite saída digital ou analógica [69].



Figura 5.12: Pressoestato da Festo [69].

Durante o ciclo de operação é necessário detetar o número de peças que são

colocadas sobre cada base por forma a desenrolar o normal funcionamento do sistema. Para a deteção de peça é utilizado o sensor LTK-1050-303-505 da Contrinex. É um sensor fotoelétrico, baseado na reflexão de um feixe de luz, que consegue detetar até 10 mm de distância, possui uma frequência máxima de comutação de 250 Hz e pode operar a temperaturas compreendidas entre os 0 e os 55 °C [70]. Este sensor está ilustrado na Figura 5.13.



Figura 5.13: Sensor fotoelétrico da Contrinex para deteção de peça [70].

Associada à mesa de montagem encontra-se o quadro elétrico secundário, no qual estão localizadas as cartas de entradas e saídas responsáveis pela leitura dos sensores e atuação das eletroválvulas. O quadro principal encontra-se à esquerda da mesa e no qual está inserido o PLC, a HMI e os botões para arranque, paragem e rearme da célula.

5.3.5 Unidade de controlo

A unidade de controlo inserida no sistema é o PLC NX1P2-9024DT1 da OMRON, ilustrado na Figura 5.14. Este PLC traz integradas 14 entradas digitais e 10 saídas digitais do tipo PNP, e possui também integrada uma porta EtherCAT e uma porta Ethernet. Possui uma memória de programa de 1,5 MB, 32 kB de memória retentiva para variáveis e 2 MB de memória não retentiva para variáveis. Permite que sejam diretamente acrescentadas até 8 cartas e no caso do uso da comunicação EtherCAT com as cartas, permite até 16 cartas. Este PLC pode operar em ambientes com temperaturas compreendidas entre os 0 e os 55 °C.



Figura 5.14: PLC NX1P2-9024DT1 da OMRON.

Para alimentar este PLC é utilizada uma fonte TRIO-PS/1AC/24DC/10 da Phoenix Contact, presente na Figura 5.15. Esta fonte pode operar numa gama de temperaturas compreendida entre os -25 e os 70 °C. Permite na sua entrada

de alimentação tensões compreendidas entre os 100 e os 240 V AC, colocando na saída 24 V DC. Tem um consumo de corrente de 3 A para os 100 V de entrada e de 1,5 A para os 240 V de entrada e coloca na saída no máximo 10 A. Possui um fator de potência de 0,99 e uma potência à saída de 240 W.



Figura 5.15: Fonte de alimentação da Phoenix Contact.

Por forma a controlar todo o sistema são utilizadas cartas digitais da OMRON. Para a leitura dos sensores são utilizadas cartas de entradas digitais e para a atuação das eletroválvulas é utilizada uma carta de saídas digitais da OMRON.

A carta de entradas digitais utilizada é a NX-ID6142-5 da OMRON, ilustrada na Figura 5.16. Esta carta possui 32 entradas digitais de comutação rápida, sendo o tempo máximo de flanco ascendente de 20 μ s e o tempo máximo de flanco descendente de 400 μ s. Em cada entrada permite uma tensão máxima de 24 V. Pode ainda operar num intervalo de temperatura compreendido entre os 0 e os 55 °C.



Figura 5.16: Carta de entradas da OMRON.

A carta de saídas digitais utilizada é a NX-OD6256-5 da OMRON, presente na Figura 5.18. Esta carta possui 32 saídas digitais de comutação rápida, possuindo um tempo de flanco ascendente de 0,5 ms e um tempo de flanco descendente de 1,0 ms. Em cada saída coloca uma tensão de 24 V. Possui ainda a capacidade de operar em temperaturas compreendidas entre os 0 e os 55 °C.



Figura 5.17: Carta de saídas da OMRON.

Estas cartas encontram-se num quadro secundário, ligadas a um módulo EtherCAT que faz a interface entre as cartas e o PLC. O módulo EtherCAT utilizado é o NX-ECC203, exemplificado na Figura 5.18, que fornece uma corrente máxima de 10 A às cartas nele montadas. Possui um ciclo de comunicação com tempos compreendidos entre os 125 e os 10000 μ s. O módulo NX-ECC203 tem a capacidade máxima de 63 unidades ligadas ao mesmo. Este módulo possui a capacidade de transferência de pacotes de dados com o tamanho máximo de 1024 bytes. O intervalo de temperatura de operação é dos 0 aos 55 °C.



Figura 5.18: Módulo EtherCAT da OMRON.

5.3.6 Interface Homem-Máquina

A interface homem máquina é executada através de um componente desenvolvido pela OMRON que se designa por *human machine interface*, vulgarmente apelidado de consola.

A *human machine interface*, HMI, utilizada é a NB10W-TW01B, presente na Figura 5.19. Esta consola de interface possui um ecrã de 10,1 polegadas *thin film transistor liquid crystal display*, TFT LCD, com a resolução de 800 x 480 pixels. O ecrã possui retroiluminação e um conjunto de 65536 cores. Garante 50000 horas de funcionamento a uma temperatura de 25 °C e o painel tátil garante resistência até 1 milhão de operações de toque.

Esta HMI possui uma memória interna de 128 MB e a capacidade de expandir esta memória através da ligação de um dispositivo de memória *universal*

serial bus, USB. A comunicação entre esta HMI e outros dispositivos pode ser efetuada através de uma ligação série, USB, conexão a impressora ou via ethernet.



Figura 5.19: HMI NB10W-TW01B da OMRON.

5.3.7 Segurança

Na componente responsável pela segurança do sistema estão presentes 4 componentes: botoneiras de emergência, relés de emergência, PSEngate e torre luminosa.

As botoneiras, ilustradas na Figura 5.20, são compostas por contactos normalmente fechados que estão ligados a um relé de segurança, enviando sinais do seu estado para o mesmo. Estas botoneiras abrem o circuito onde estão inseridas, impedindo assim o fluxo de corrente.

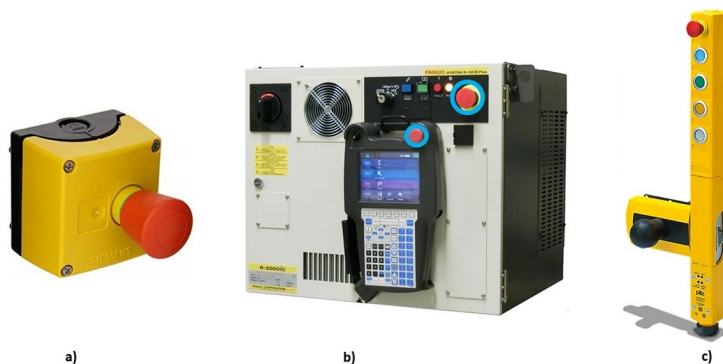


Figura 5.20: a) Botoneira de segurança Siemens. b) Botoneiras de emergência do controlador e *teach pendant*. c) Botoneira de emergência da PSEngate.

O relé de segurança utilizado é o modelo G9SE-401 da OMRON. Este relé possui saída em PNP, com a capacidade de alternar entre o modo ON e o modo OFF em 15 ms e vice-versa em 100 ms. Desta forma, quando na entrada não

estão presentes os sinais necessários, as saídas do relé são colocadas em circuito aberto. O relé, presente na Figura 5.21, apresenta 3 estados de erro identificados pelos diodos emissores de luz, LED do inglês *light emitting diode*, inseridos no mesmo. Os estados são de curto circuito nas entradas do relé, falha interna do relé e erro de atraso de abertura de saída.



Figura 5.21: Relé de segurança G9SE-401 da OMRON.

A PSENGate, ilustrada na Figura 5.22, é um sistema que combina a monitorização da segurança de porta, fecho de proteção e elementos de controlo. Este elemento é constituído por um trinco de segurança associado a um elemento que o destranca apenas sob condições pré-determinadas. Possui ainda 4 botões de ação para desencadear tarefas pré-determinadas e uma botoneira de emergência. Para o caso de algum colaborador ficar preso dentro da célula, a PSENGate possui um sistema de abertura de emergência mecânico.



Figura 5.22: PSENGate da PILZ.

Por fim, a torre luminosa corresponde a uma torre com 4 luzes LED de cor diferente, com significados diferentes consoante os estados do sistema. Esta torre,

presente na Figura 5.23, tem as suas luzes ativadas através de sinais físicos que podem ser contínuos ou pulsados.



Figura 5.23: PSENGate da PILZ.

5.4 Software

Ao longo de todo o projeto são necessários vários *softwares* para diversas tarefas. Estes *softwares* são utilizados quer na fase de simulação e validação, quer na fase de programação.

Para a simulação e validação dos cenários utilizaram-se o Visual Components e o Roboguide, sendo este último o *software* de simulação da FANUC. Para programação foram utilizados o Sysmac Studio, no caso do PLC, e o Roboguide, no caso do robô.

5.4.1 Visual Components

O Visual Components é um *software* de simulação industrial 3D. A distribuição utilizada é a versão PREMIUM do Visual Components. Este *software* permite a configuração de *layouts*, característica que associada à capacidade da leitura de ficheiros CAD permite aumentar a semelhança ao cenário real, criando uma réplica fiel e permite também alterar o próprio CAD.

Associada à capacidade gráfica do Visual Components, existem duas funcionalidades: a modelação de componentes e a simplificação de geometrias. A modelação de componentes permite criar os componentes no próprio Visual Components e a simplificação de geometrias permite retirar detalhes desnecessários do componente.

Através das funcionalidades robóticas, permite fazer um estudo de uma célula criando padrões de movimento do robô, analisando colisões e alcances, e criando lógica nos programas dos robôs através de condições de programação. Existe ainda uma extensão que permite exportar os pontos criados no Visual Components para várias linguagens de robôs, incluindo RAPID e KAREL.

Uma das características da versão PREMIUM é a capacidade de fazer o ensino de movimentos do robô através de um movimento em curva da ferramenta ou criar uma trajetória através de uma marcação rápida e facilitada de pontos.

Existe também a possibilidade de, através do Visual Components, exportar desenhos 2D dos ambientes simulados ou, mesmo, criar vídeos ou PDF animados da simulação.

O Visual Components, para além das ferramentas de simulação, possui ferramentas estatísticas. Permite a criação e exportação de gráficos em Excel ou PDF de vários elementos da simulação, como tempos de ciclo ou mesmo peças produzidas por cada estação.

Na Figura 5.24 está exemplificado um cenário de *pick and place* com três robôs paralelos e um robô SCARA, fornecido como exemplo pelo Visual Components.

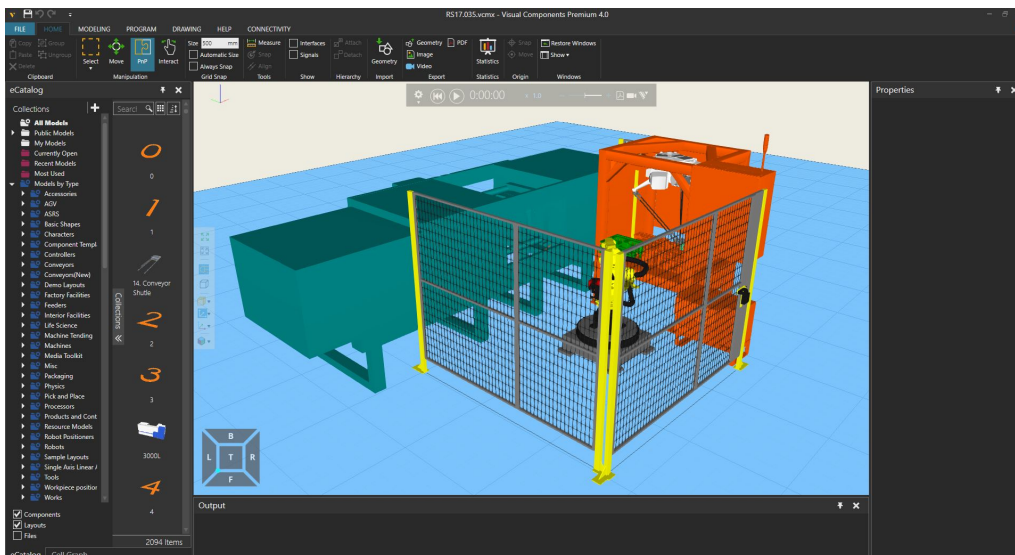


Figura 5.24: Exemplo de uma aplicação de *pick and place* no Visual Components.

5.4.2 Roboguide

O Roboguide é o *software* desenvolvido pela Fanuc para simulação e programação *offline* de robôs e soluções robóticas. A distribuição do Roboguide utilizada é o Roboguide HandlingPRO. Este *software* permite a programação via simulação de consola de programação e a programação através de texto. A programação através de texto pode ser feita de acordo com a linguagem da consola de programação ou na própria linguagem de programação Karel, desenvolvida pela Fanuc.

Através da capacidade de importar ficheiros CAD é possível simular um ambiente fiel ao projetado e a ser implementado. Após a criação da célula robotizada específica, o Roboguide permite que sejam avaliados alcances dos robôs, programação dos movimentos dos mesmos, marcação e calibração das *user* e *tool frames*, deteção de colisões e também permite avaliar os tempos de ciclo. O Roboguide possibilita também a gravação em vídeo do funcionamento programado da célula para análise.

A Figura 5.25 apresenta a simulação, do cenário a implementar, efetuada no Roboguide.

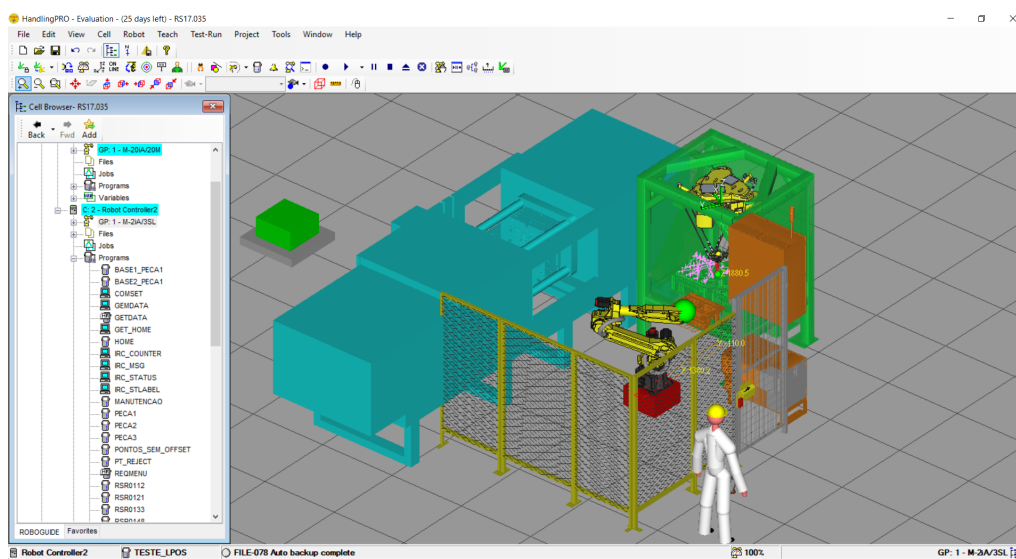


Figura 5.25: Exemplo de modelação e simulação do cenário a implementar.

5.4.3 Sysmac Studio

O Sysmac Studio é a ferramenta disponibilizada pela OMRON para programação de PLC. Este *software* permite a programação dos PLC OMRON mais recentes através das linguagens de programação vulgarmente utilizadas, permite a criação de blocos de funções e funções e ainda efetuar configurações de rede para ligação entre PLC e outros dispositivos. Neste *software* todos os programas se encontram organizados em forma de árvore e para que o autómato os execute é necessário defini-los na *task*. Uma ferramenta útil deste programa é o endereçamento automático de variáveis caso não seja necessário atribuir um endereço específico. Permite ainda a simulação do código produzido para fins de *debug*.

5.4.4 NB Designer

O NB Designer é um *software* disponibilizado pela OMRON para a programação de consolas da mesma marca. Este programa funciona através da criação de janelas e da utilização de blocos tornando uma programação bastante gráfica do pretendido. Uma das vantagens é a possibilidade de simulação da consola em questão e de todos os objetos colocados e programados.

5.5 Protocolo Ethernet/IP

O protocolo Ethernet/IP é protocolo utilizado para comunicação entre o PLC, consola e os robôs da solução. Neste caso caracteriza-se pela criação de uma rede, atribuindo a cada dispositivo um endereço dessa rede. A comunicação é feita através do envio e receção de pacotes de dados que são transmitidos fisicamente através de cabos Ethernet. Esta comunicação deve ser definida para que cada elemento saiba a quantidade de dados que espera receber e enviar, sendo que entre dispositivos OMRON essa definição já se encontra feita de fábrica.

5.6 Cenários analisados

5.6.1 Organização base do cenário

O cenário base corresponde ao visionado na Figura 5.26. A zona frontal da célula define-se pelas portas da mesa de montagem, ficando ao lado esquerdo da mesma o quadro elétrico e do lado direito a máquina de injeção de plástico. A porta de acesso ao interior da célula encontra-se localizada à esquerda da máquina. Na mesma linha da porta de acesso à célula encontra-se o robô responsável pela descarga da máquina, estando esta localizada frontalmente em relação a ele e com a mesa à sua esquerda. A torre luminosa que informa o estado da célula, através de um guia de luzes coloridas, localiza-se no topo do quadro elétrico.

O robô paralelo encontra-se montado na estrutura de forma a que o seu atuador final se encontre paralelo às mesas de montagem das pinças, como se verifica na Figura 5.26. Localiza-se centrado com a estrutura por forma a que o alcance a cada mesa seja o mesmo. O robô de descarga encontra-se a 1200 mm da máquina de injeção e a 800 mm da mesa de montagem por forma a melhorar o ciclo de movimento, sem que este seja impossibilitado por singularidades.

No *layout* ilustrado pela Figura 5.26 a zona traseira da célula está limitada por dois painéis de rede, um de 1500 mm e outro de 1200 mm, e a zona lateral limitada por um painel de 1500 mm e uma porta com 800 mm de comprimento.

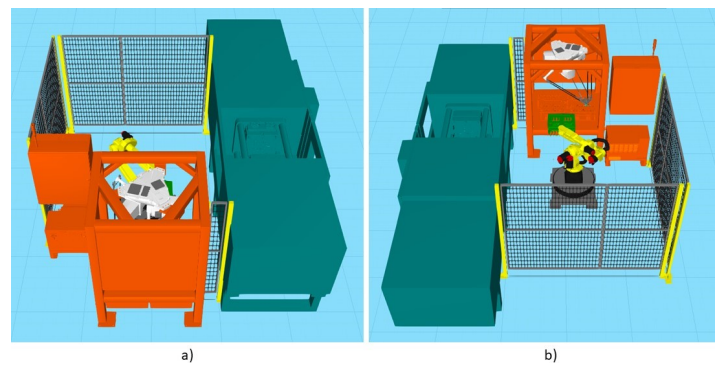


Figura 5.26: a) Vista frontal superior da célula. b) Vista traseira superior da célula.

5.6.2 Peça de 13 cm de comprimento

Na mesa de montagem existe uma parte que pode ser trocada rapidamente caso se mude a peça a produzir: as bases de montagem. Cada base é específica para cada tipo de peça, bem como os dedos da garra utilizada. Na Figura 5.27 estão representadas as bases já com a disposição das peças para montagem, conforme serão descarregadas.

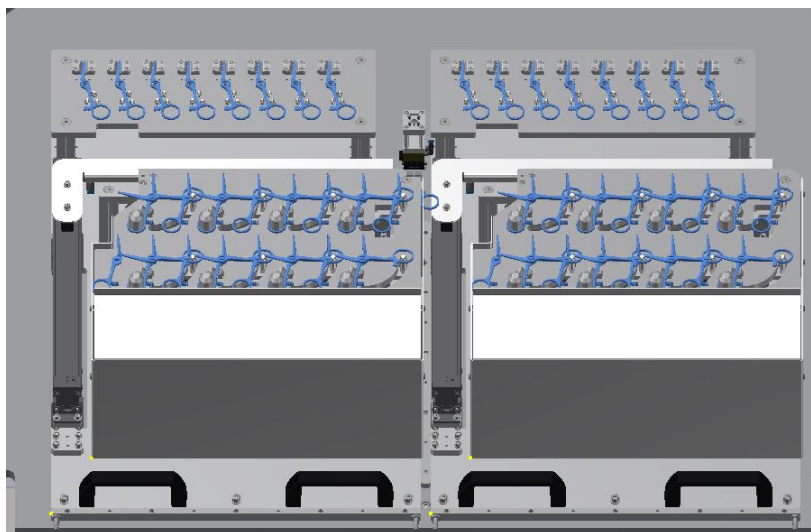


Figura 5.27: Bases para as peças de 13 cm de comprimento.

Estas bases são compostas por 16 sensores LTK-1050-303-505, um por cada fêmea e macho, para a sua detecção. Cada um destes sensores encontra-se colocado na placa. No caso das fêmeas localiza-se entre as cavilhas, onde se localiza

o orifício central da peça após a sua descarga e a guia do extremo da peça. Em relação aos machos localiza-se no espaço aberto entre a guia para o orifício da peça e a guia para o corpo da peça.

Na extremidade da placa onde assenta a base são colocados dois sensores DW-AS-623-M8-001 para verificar se a base está montada na mesa. No empurrador estão localizados quatro sensores SIES-8M-PO-24V-K-0,3-M8D para a deteção dos quatro estados do mesmo, e dois por cada pino de fecho, bem como dois por cada guia. Uma guia corresponde a cada quatro pinos, responsáveis por guiar as peças macho, colocados horizontalmente.

No total cada base integra 42 sensores.

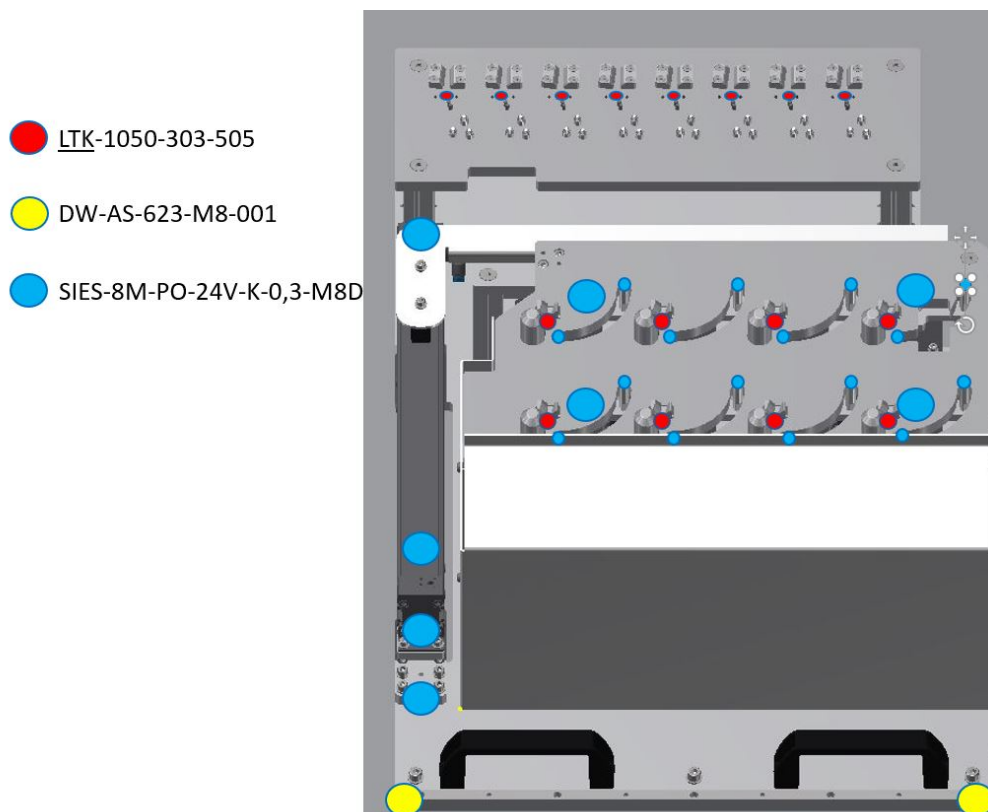


Figura 5.28: Localização dos sensores na base.

Na Figura 5.29 está representado um exemplo da peça final a ser produzida através deste sistema, composta por duas peças plásticas já com o corante adicionado na sua composição.



Figura 5.29: Exemplo da peça de 13 cm.

5.7 Conclusão

Em suma, neste capítulo foram apresentados os componentes a serem utilizados para a implementação de uma solução para dar resposta ao problema enunciado no início deste capítulo. Foi exposta uma breve análise à composição do sistema e a interação entre elementos, bem como foi efetuada uma abordagem dos utensílios tangíveis e intangíveis da composição do sistema.

No capítulo seguinte é efetuada uma abordagem à forma como foi implementada a solução, nomeadamente de gestão do sistema e de controlo da mesa de montagem e do robô paralelo, bem como a sua programação para a execução da tarefa proposta.

Capítulo 6

Implementação Prática

Após a análise da arquitetura do sistema global e da análise da arquitetura do sistema implementado e documentado nesta dissertação, segue-se o capítulo no qual é exposta a implementação do sistema. Este capítulo encontra-se dividido em três secções. A primeira secção denomina-se de modelação e simulação, na qual é exposto um conjunto de programas nos quais foi simulada a célula e como o fazer. A segunda secção denomina-se de programação do robô, na qual é exposto o raciocínio inerente ao programa desenvolvido para o robô. Na terceira secção, denominada de controlo da célula, é exposto o raciocínio que regeu a programação do controlo da célula.

6.1 Modelação e simulação

Nesta secção é apresentado, inicialmente, o conjunto de passos necessários para exportar o conjunto de ficheiros CAD necessários para a modelação. De seguida é exposto o conjunto de etapas para a modelação e simulação do sistema.

6.1.1 Exportar ficheiros CAD do projeto em Inventor

Para a exportação dos ficheiros em formato CAD para a modelação nos *softwares* de modelação e simulação foi necessário acesso ao projeto desenvolvido pelo departamento de projeto mecânico da RobotSol. O projeto é desenvolvido em Inventor, *software* da Autodesk específico para a tarefa. Para exportar cada um dos constituintes da célula necessários à modelação é necessário abrir o projeto de cada componente a exportar, selecionar a aba formato CAD e selecionar o tipo de ficheiro, como está ilustrado na Figura 6.1.

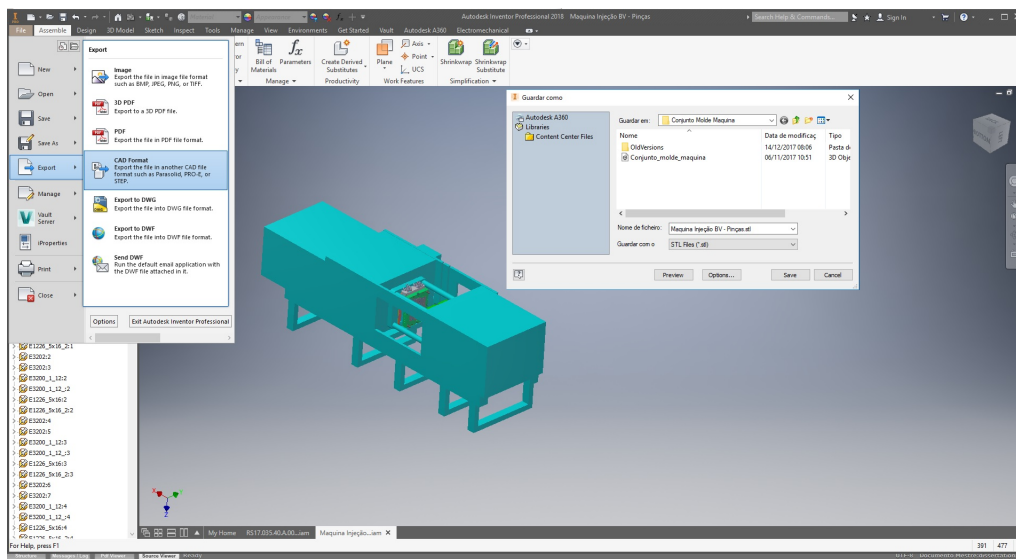


Figura 6.1: Exportar ficheiros CAD para modelação.

Os ficheiros foram exportados em formato *Standard Triangle Language*, STL, formato esse suportado por ambos os *softwares* de simulação. Os componentes exportados por este método foram: a máquina de injeção, a mesa de montagem, os atuadores finais de cada robô e o quadro elétrico. Os restantes componentes não foram exportados do projeto realizado pelo departamento de projeto mecânico, uma vez que existiam ficheiros de CAD desses componentes nas bibliotecas dos *softwares* utilizados.

É necessário, ao exportar um ficheiro através do Inventor, ter em atenção as unidades de medida utilizada no projeto e as unidades em que se está a exportar o ficheiro, pois pode conduzir a uma exportação com dimensões erradas. A unidade a escolher durante a exportação do ficheiro depende da unidade utilizada durante a fase de projeto no Inventor.

6.1.2 Esquema da célula

O esquema inicial da célula a implementar, contemplado na Figura 6.2, coloca o robô de descarga a 1200 mm da máquina de injeção e a 830 mm da mesa de montagem. A mesa de montagem encontra-se a 300 mm da máquina e a porta de acesso à célula a 850 mm do robô de descarga.

Para limitar a célula foram utilizados dois painéis de rede de 1500 mm de comprimento, um painel de 1200 mm de comprimento e uma porta de 800 mm de comprimento. Na zona traseira da célula encontram-se posicionados lado a lado um painel de 1500 mm e um de 1200mm alinhados perpendicularmente com a máquina de injeção de plástico. Paralelamente à máquina de injeção e

perpendicular a este conjunto de painéis encontram-se alinhados um painel de 1500 mm e a porta de acesso à célula. Nesta disposição, o controlador do robô de descarga encontra-se por baixo do quadro elétrico principal e o do robô delta encontra-se localizado por baixo da mesa, na zona das portas de acesso à mesa.

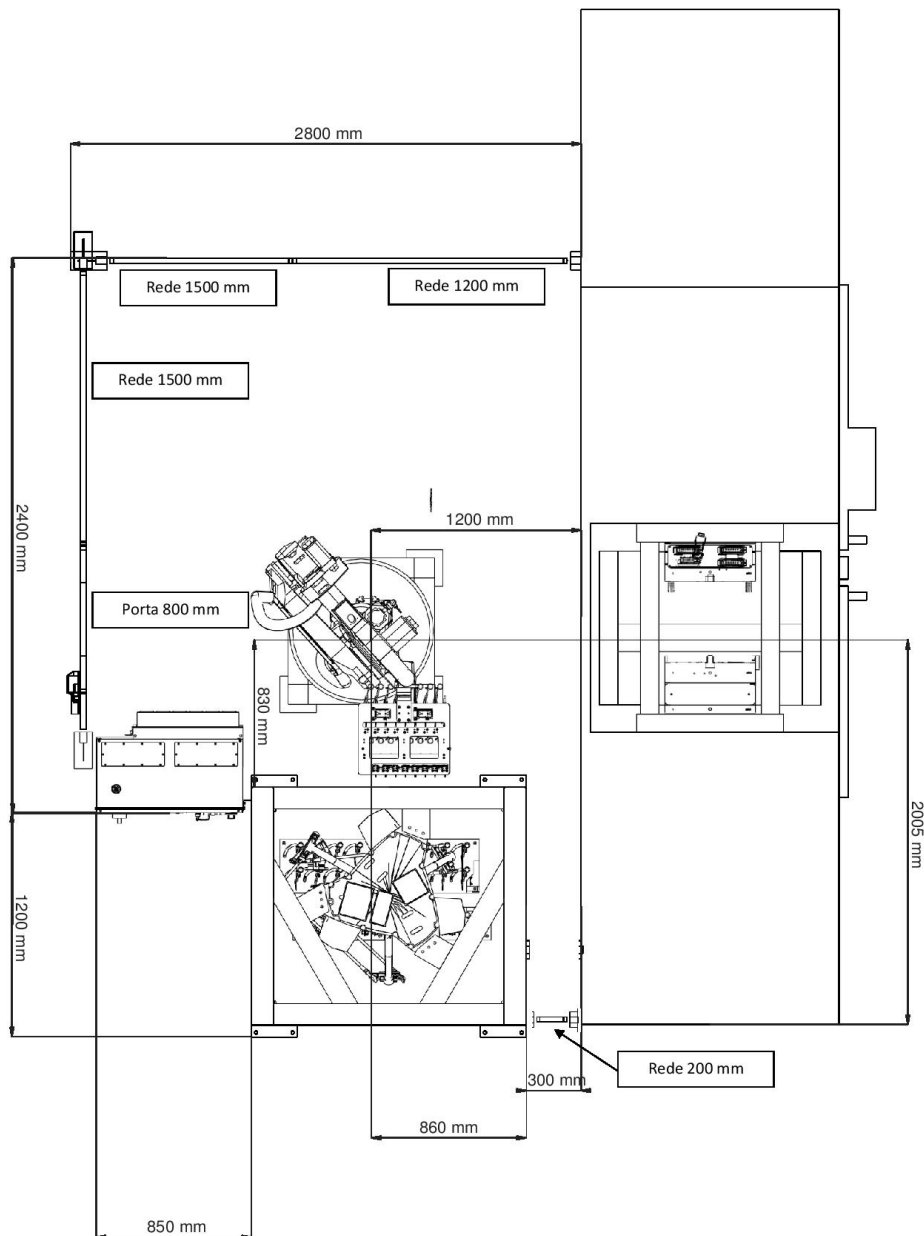


Figura 6.2: Esquema da célula e medidas das redes.

6.1.3 Modelação e simulação em Visual Components

6.1.3.1 Modelação

Modelar uma célula no Visual Components é bastante simples e rápido. Para criar um novo ficheiro basta abrir o programa que já inicia com um ambiente vazio. De seguida é necessário importar todos os ficheiros CAD dos elementos que não existam na biblioteca do Visual Components.

Para importar um ficheiro CAD é necessário clicar em "Geometry Import" no menu superior. De seguida abre-se uma janela de busca através da qual se localiza o ficheiro a importar. Após o ficheiro selecionado abre-se uma janela do lado direito com as características do ficheiro e nessa janela clica-se no botão *import* o que terminará com a importação e o aparecimento do objeto na janela. Estes passos verificam-se na Figura 6.3. Neste caso os elementos importados foram: a máquina de injeção, a garra para descarga da máquina, a garra do robô delta, a mesa de montagem com as bases montadas e o quadro elétrico.

Os robôs são introduzidos através da própria biblioteca do Visual Components. Para se seleccionar o robô é necessário aceder ao menu presente na parte lateral da janela do Visual Components, como está ilustrado na Figura 6.3. De seguida, na árvore de pastas, seleccionar "models by type" e, dentro desta pasta, a pasta "robots". No interior da pasta "robots" encontram-se divididos os robôs por marca, em pastas com o nome de cada marca. De seguida abre-se a pasta da Fanuc e surge uma janela ao lado com os robôs existentes na biblioteca e na qual são seleccionados os necessários.

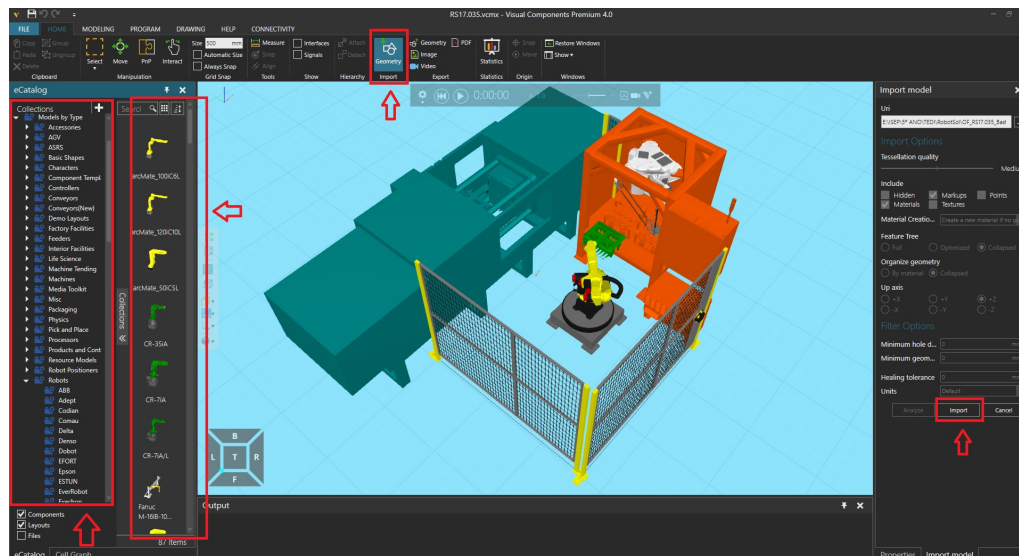


Figura 6.3: Importar ficheiros CAD para modelação.

Para introduzir o robô no ambiente de modelação e simulação basta arrastar o mesmo para a janela. O Visual Components possui na sua biblioteca não só robôs como outros elementos, sendo que neste caso recorreu-se à biblioteca do Visual Components para a colocação das redes na modelação da célula. As redes e os postes de suporte encontram-se na pasta "Accessories", em "models by type". As bibliotecas do Visual Components possuem também garras, que neste caso não foram utilizadas. No caso do robô de descarga a razão prende-se com a especificidade da garra, tendo sido desenhada pelo departamento de projeto mecânico e no caso do robô delta a garra não existe na biblioteca, sendo que a única parte desenvolvida pelo departamento de projeto mecânico foram os dedos da garra.

6.1.3.2 Simulação

O Visual Components possui um modo bastante simples para simular rapidamente um sistema e para ter noção do seu tempo de ciclo. Para proceder à simulação deste sistema é necessário efetuar a marcação dos pontos que o robô deve percorrer durante a execução do seu ciclo.

A criação de um programa, a incluir um conjunto de pontos, é feita através do menu "Program", localizado na aba superior da janela, como se visualiza na Figura 6.4. Para a marcação desses pontos é necessário mover o atuador final do robô para a posição específica onde se pretende marcar o ponto. A mobilização pode ser efetuada através de ferramentas de alinhamento e com determinados pontos físicos, recorrendo, por exemplo, à opção *align* e alinhar a garra do robô paralelo com um ponto numa das bases da mesa de montagem. É possível mover o robô manualmente nas suas juntas, selecionando *jog* no menu superior *manipulation*, ou linearmente, selecionando *move* no menu *manipulation*.

A marcação dos pontos é efetuada na janela que surge do lado esquerdo. Nessa janela é possível criarem-se vários ficheiros de programa com nomes diferentes. A marcação dos pontos pode ser efetuada com três instruções diferentes: pontos cujo movimento será efetuado ponto a ponto, pontos cujo movimento será efetuado de forma linear e pontos que definam uma trajetória. Estas instruções estão representadas na Figura 6.4 pelos números 1, 2 e 3, respetivamente.

Na mesa, os pontos foram marcados com o auxílio da ferramenta *align*, alinhando a peça macho colocada no CAD da garra e as peças fêmea colocadas no CAD da mesa, em cada base. Através deste método foi marcado um ponto para cada encaixe e um ponto de aproximação, alinhado com o de encaixe mas com uma cota diferente. Cada ponto pode ser configurado através da alteração das suas coordenadas e velocidade.

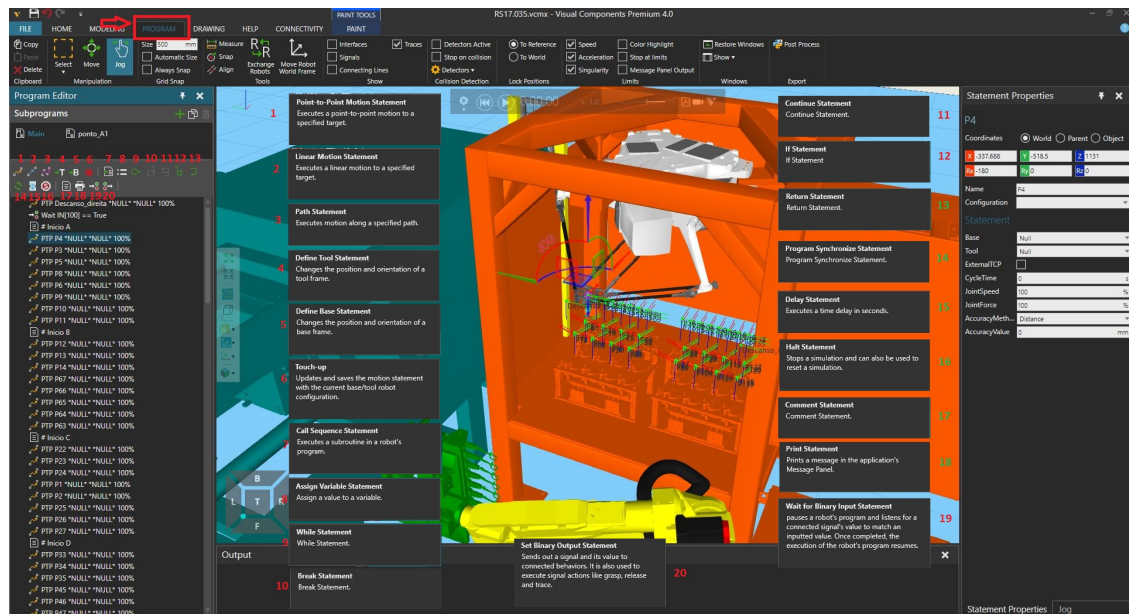


Figura 6.4: Criação de um programa no Visual Components.

Para a simulação da montagem de cada peça foram marcados 3 pontos, quer na base de fêmeas, quer na base dos machos. Dois desses três pontos são iguais, uma vez que correspondem ao ponto de aproximação. A marcação desses pontos simula o *pick* da peça fêmea e o *place* (e conseqüente montagem) na peça macho. Para a simulação de sinais provenientes do PLC, ou do robô de descarga, foi utilizada a instrução *wait for binary input statement*. Esta instrução tem de estar ligada a uma instrução de *Set binary output statement* proveniente do outro robô, ou a um botão binário, para simulação do sinal. Cada uma destas simula uma entrada digital ou saída digital, respetivamente.

O Visual Components permite também, através da ativação no menu superior, avaliar colisões e suspender o movimento quando ocorrem as mesmas. É também possível limitar velocidades, acelerações e singularidades. Através da condição *if* foi possível testar a possibilidade de falhar alguma peça, uma vez que só produziria o código dos pontos quando a condição fosse satisfeita. A condição *while* permitiu colocar o código em *loop* até terminar o ciclo. Para simular o tempo de operação de cada base foi utilizada a função *delay statement*.

Este *software* permitiu a exportação dos pontos de simulação, porém não permitiu a exportação de condições, uma vez que o pós-processador não possui essa capacidade. Para a exportação dos pontos utiliza-se uma extensão ao *software*, conhecida como pós-processador, que traduz esses pontos para as linguagens de programação de robôs, nomeadamente para a linguagem KAREL utilizada pela Fanuc.

6.1.4 Modelação e simulação em Roboguide

6.1.4.1 Modelação

A modelação no Roboguide não é tão fácil como no Visual Components. Uma das desvantagens é o facto da biblioteca de CAD ser bastante mais limitada, porém possui os robôs necessários pois os utilizados são da marca Fanuc. Apesar da máquina de injeção de plástico ser da marca Fanuc, o modelo não está disponível na biblioteca de CAD. No Roboguide os únicos elementos cujo CAD (utilizado) existia na biblioteca, foram os robôs e as barreiras.

Para a criação de um projeto no Roboguide é necessário criar uma nova célula através de "New Cell", em "File", assinalado na Figura 6.5. A criação do novo projeto contempla 8 passos. No 1º passo é definido o nome do projeto, no 2º passo o método de criação do projeto, no 3º passo a versão de *software* a utilizar, no 4º passo o pacote de aplicações a ser utilizado, no 5º passo o robô a utilizar, no 6º passo a possibilidade de adição de grupos de movimento, no 7º passo as opções extra do robô e, finalmente, no 8º passo é mostrado um sumário das opções anteriores.

Na escolha da aplicação a utilizar, como a aplicação a simular é de *pick and place* e montagem, seleciona-se a aplicação HandlingTool e na escolha do robô seleciona-se o modelo M2iA/3SL. Nos restantes passos são utilizadas as definições por omissão, uma vez que não é necessário adicionar grupos de movimento ou alguma opção extra.

Após a criação do projeto aparece para simulação a célula já com o robô selecionado na posição de origem do mundo. Na janela da esquerda, destacada na Figura 6.5 e denominada de *cell browser*, na aba *Robot Controllers* é possível adicionarem-se mais controladores e robôs à simulação inicialmente criada. Cada controlador pode possuir, na sua dependência, mais do que um robô. Na realidade o mesmo é possível, vários robôs serem controlados por um único controlador. O *cell browser* está dividido em várias categorias, tais como: *fixtures*, *parts*, *machines*, *robot controllers*, *obstacles*, *workers*, *profiles*, entre outras. Estas categorias permitem dividir os objetos constituintes da simulação por grupos, de acordo com o seu objetivo.

Na categoria *robot controllers* inserem-se todos os controladores individuais de robôs utilizados. Todos os possíveis obstáculos utilizados para limitar a célula, como as barreiras e a porta, foram colocados na categoria *obstacles*. Também o quadro elétrico foi inserido nesta categoria. A máquina de injeção e a mesa de montagem foram inseridas na categoria *machines*, pois executam um processo cada. As barreiras podem ser desenhadas através de uma opção em obstáculos que permite desenhar no mundo o comprimento da fila de barreiras.

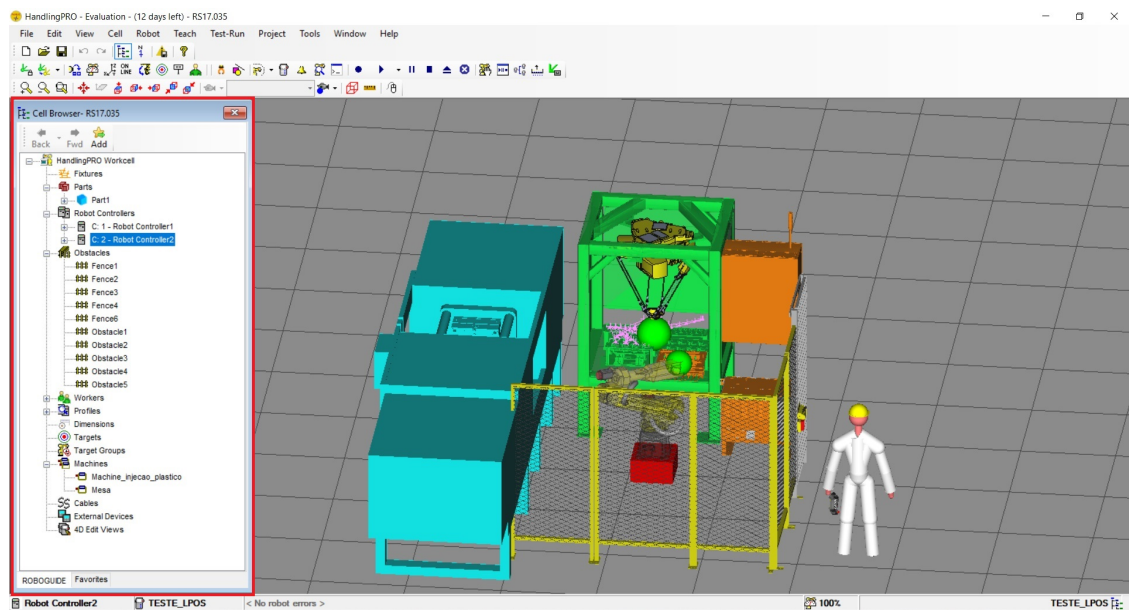


Figura 6.5: Modelação da célula em Roboguide.

6.1.4.2 Simulação

A simulação no Roboguide tem por base a simulação do ambiente real de programação e de ação robô. A programação do robô, explicitada na Secção 6.2, é parte fundamental da simulação pois através da mesma serão especificados todos os pontos, movimentos e condições que o mesmo tem de executar.

A marcação dos pontos foi efetuada da mesma forma que em ambiente de programação *online*, através da consola de programação presente na Figura 6.6. No ambiente de simulação é possível criar ciclos de pontos por forma a simular uma trajetória.

O Roboguide permite também a simulação de sinais de entrada através do *I/O panel*. Através deste painel, presente na Figura 6.6, é possível adicionar-se as entradas a simular, independentemente do seu tipo, e as saídas a monitorizar.

Através do seu *run panel*, o Roboguide permite simular os programas presentes nos robôs e apresenta informação sobre o tempo total de simulação, tempo de movimento, tempo da aplicação, tempo de espera e tempo de *delay*, através de uma janela de perfil. É possível criar até 9 perfis de simulação nos quais é detalhado o tipo de arranque do robô utilizado.

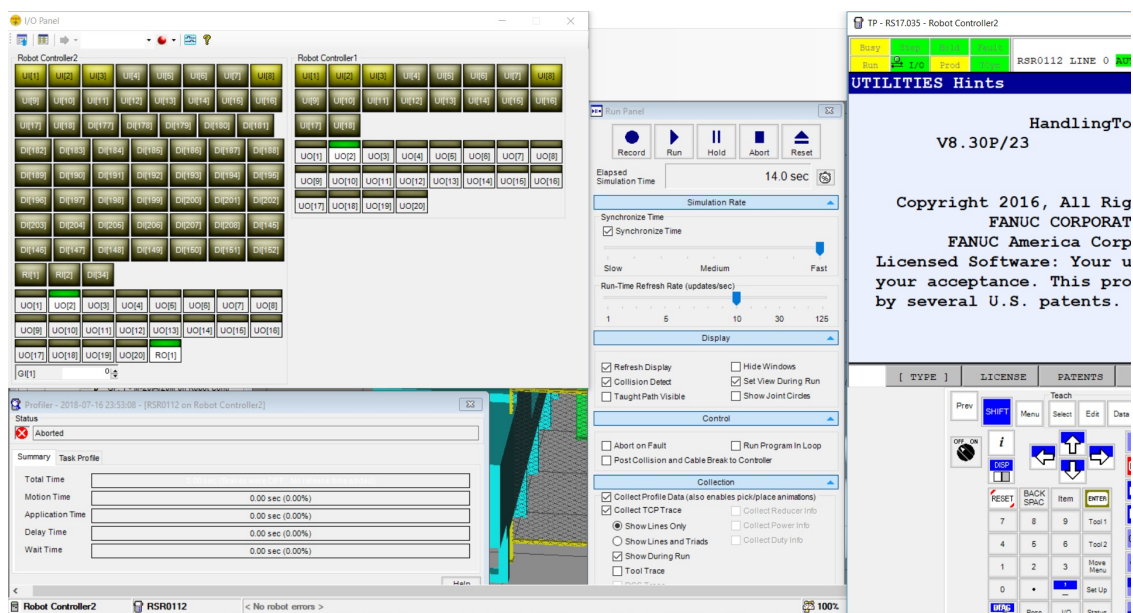


Figura 6.6: Simulação da célula em Roboguide.

É possível mover o robô manualmente para cada ponto através da esfera verde presente no punho do robô ou atribuir diretamente (como se verifica na Figura 6.7), através da janela de simulação da consola, as coordenadas do ponto para o qual se pretende mover o robô. O Roboguide possui também a capacidade de visualizar em simulação o volume de trabalho, trajetória e colisões. Está exemplificado na Figura 6.7 a selecção de visualização do volume de trabalho e ainda a opção de ativar uma saída ou entrada caso exista uma colisão detetada.

A simulação dos movimentos do robô, comportamento em relação a sinais externos e testes ao alcance do robô têm por base o *software* desenvolvido através da programação do robô e que é posteriormente utilizado na implementação prática e é exposto na Secção 6.2.

A criação de programas, através do *teach pendant*, e sua alteração pode ser feita em dois formatos: LS e TP. O formato LS é o mais próximo de edição através de editor de texto que o próprio Roboguide disponibiliza e o formato TP pode ser alterado através do emulador de *teach pendant* que o Roboguide possui.

O Roboguide permite exportar os programas desenvolvidos em formato TP por forma a serem introduzidos no controlador real do robô, bem como todas as alterações feitas em ambiente de simulação, como a criação de comentários a entradas e saídas. No caso de um programa, apenas ficheiros do tipo TP são lidos pelo controlador, ao contrário de ficheiros do tipo LS. Desta forma, tudo o que é produzido em ambiente de simulação pode ser facilmente implementado

no controlador real sendo também possível o processo inverso, alterar um *backup* do que está inserido no controlador real.

Este ambiente de simulação é bastante útil para teste de pontos criados através da programação *online* e para desenvolvimento de toda a lógica a implementar, de forma *offline*. Através deste ambiente de simulação é possível desenvolver todo o código sem necessitar do sistema, sendo este apenas necessário para marcação de sistemas de coordenadas e pontos.

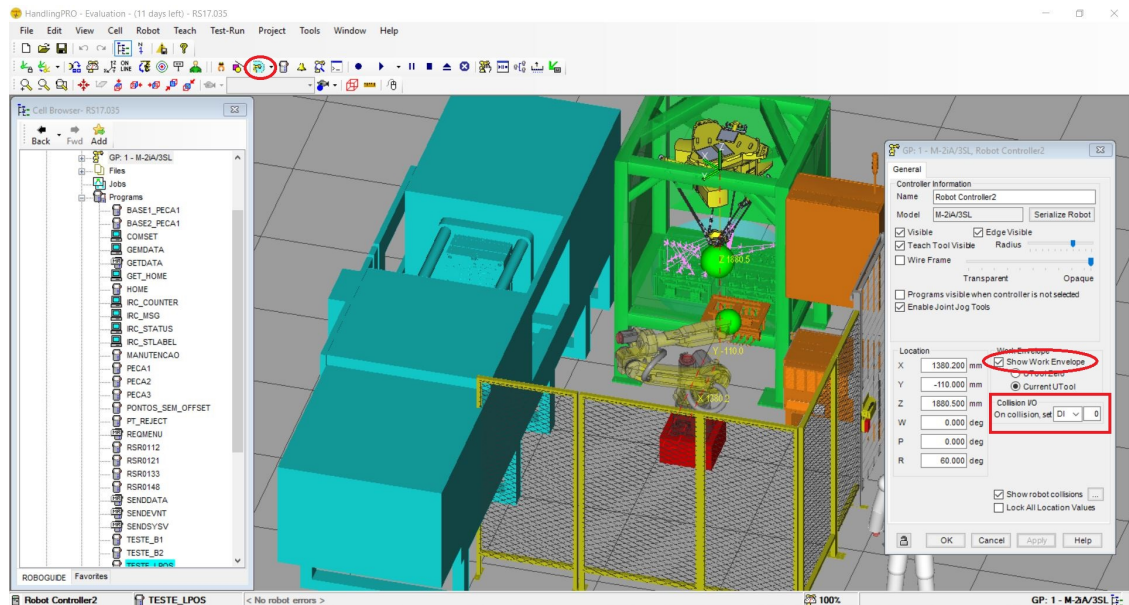


Figura 6.7: Simulação da célula em Roboguide - *work envelope* e configuração de sinais de colisão.

6.2 Programação do robô

Para a programação do robô é necessário ter em conta vários aspetos do funcionamento do mesmo. Os aspetos mais importantes são a configuração dos sinais utilizados pelo robô, o arranque remoto do robô, a configuração do atuador final e da carga a movimentar (*payload*), a configuração dos sistemas de coordenadas (*user frame*), a configuração do espaço partilhado por ambos os robôs (*space function*) e a criação do conjunto de programas a executar.

6.2.1 Configuração dos sinais utilizados pelo robô

O controlador do robô possui seis tipos de sinais de entrada e saída: digitais, analógicos, grupo, robô, *user operator panel* (UOP) e *system operator panel* SOP.

Estes sinais, cujo menu de acesso está representado na Figura 6.8, são utilizados para o controlo e sincronização de todas as tarefas do processo.

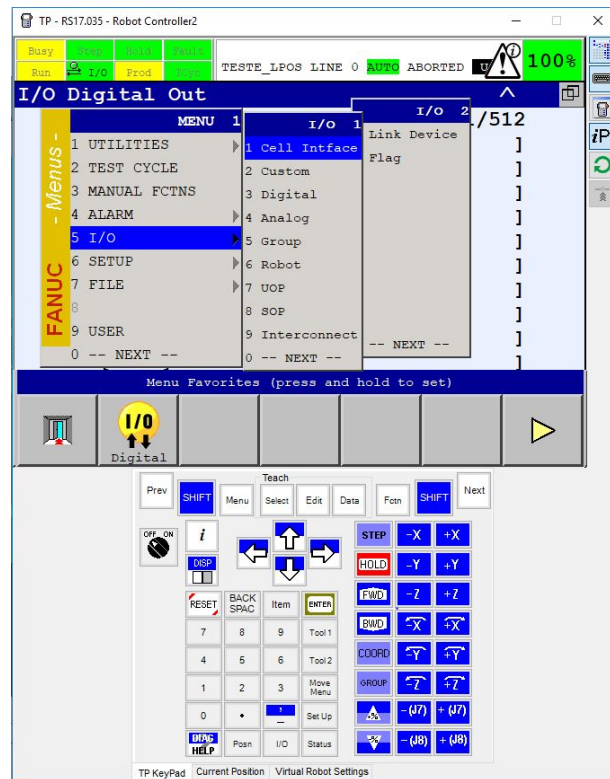


Figura 6.8: Menu na consola dos sinais de I/O.

Os sinais digitais correspondem a saídas e entradas digitais e são utilizados para a transmissão de informação binária, podendo possuir cartas físicas ou não.

Os sinais analógicos são utilizados para interface com sensores analógicos.

Os sinais de grupo são utilizados para transmitir informação não binária ou quando é necessário aglutinar um grande número de sinais a transmitir.

Os sinais do robô são um grupo de sinais específicos transmitidos por um cabo específico, denominado de *end effector*, com o objetivo de controlo o atuador final. Os sinais UOP são sinais digitais utilizados para interface com o PLC. Os sinais SOP são sinais que indicam o estado do robô e não possuem saída física. Apesar de não possuírem saída física podem ser consultados e processados nos programas.

Os sinais utilizados foram os sinais digitais, de grupo, do robô e os UOP. Apenas a transmissão dos sinais do robô foi feita através da porta física para

o efeito. Os sinais digitais, de grupo e UOP foram transmitidos via protocolo Ethernet.

Para associar os sinais à transmissão via Ethernet é necessário proceder à sua configuração. Esta configuração tem de ser feita para cada tipo de sinal individualmente e para entradas e saídas também individualmente; não basta configurar as entradas digitais ou UI, mas também é necessário a mesma configuração para as saídas digitais e UO.

A configuração é feita no menu do *teach pendant*, escolhendo o tipo de I/O na aba de 5 I/O. Após selecionado o tipo de I/O, como exemplifica a Figura 6.9, acede-se ao menu de configuração carregando no botão de *config*. Nesse mesmo menu são configuradas as portas associadas a cada intervalo de variáveis.

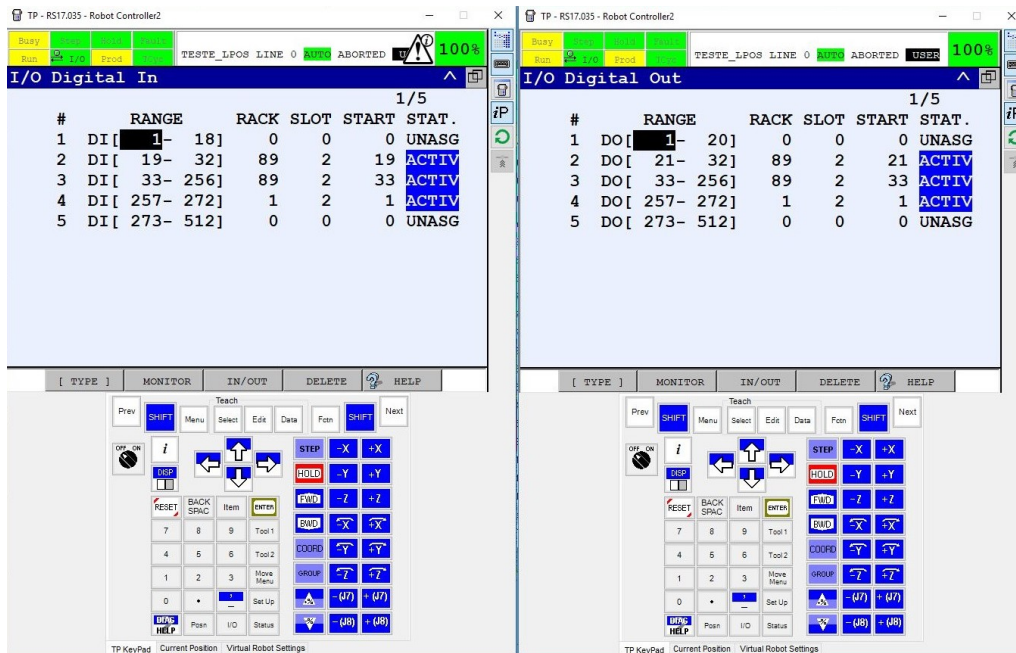


Figura 6.9: Configuração dos sinais de I/O digitais.

Os sinais de entrada digitais estão mapeados a partir da DI 19 para evitar problemas de comunicação na transmissão dos sinais para o PLC, uma vez que nas entradas 1 a 18 está mapeado outro tipo de sinais. Este mapeamento está associado a uma *rack* e a uma *slot*.

A *rack* corresponde a um código que indica o protocolo de interface entre o controlador do robô e os restantes periféricos. A *rack* 89 corresponde ao protocolo Ethernet/IP. No caso da *slot* corresponde ao número de módulos de entradas e saídas que compõe a *rack*. Neste caso cada controlador do robô necessita

de possuir um número de *slot* diferente por estarem ligados na mesma *rack*. A Figura 6.9 é proveniente do simulador, no qual não há problema em serem ambos configurados nas mesmas *rack* e *slot*, uma vez que os sinais estão a ser simulados e não há ligação entre ambos nem a nenhum PLC. Na realidade a *slot* 1 está configurada no robô 1, o robô de descarga, e a *slot* 2 configurada no robô 2, o robô delta.

Os sinais de saída digital começam a ser mapeados a partir da DO 21, pela mesma razão que os sinais de entrada e com as mesmas configurações. Em ambos os sinais, de entrada e de saída, o intervalo compreendido entre a posição 257 e a posição 272 corresponde às cartas físicas inseridas no controlador.

Os sinais de grupo, cuja configuração está exemplificada na Figura 6.10, seguem as mesmas configurações de *rack* e *slot* pois são transmitidos através do protocolo Ethernet/IP. Cada uma das variáveis de grupo está mapeada com o tamanho de 16 bits, o que corresponde a uma *word* do lado do PLC. A primeira variável de entrada de grupo mapeada começa no bit 49, o que corresponde ao início da variável *code program*, como se verifica no esquema de variáveis exposto no Anexo A.3. As restantes variáveis de grupo, quer de entrada, quer de saída, seguem a mesma lógica concordante com o Anexo A.

Figura 6.10: Configuração dos sinais de I/O de grupo.

Os sinais do robô não padecem de qualquer configuração em específico, uma vez que correspondem à atuação direta de saídas específicas do robô associadas

ao terminal de *end effector*. Estes sinais são utilizados para atuar e ler o estado do atuador final do robô.

O sinal de entrada de garra aberta está mapeado no pino 1 e o sinal de garra fechada está mapeado no pino 2. O sinal de saída para abertura da garra está mapeado no pino 1. Não existe sinal para fecho porque a garra fecha na ausência do sinal para abertura através de uma mola. Estes sinais estão ilustrados na Figura 6.11.

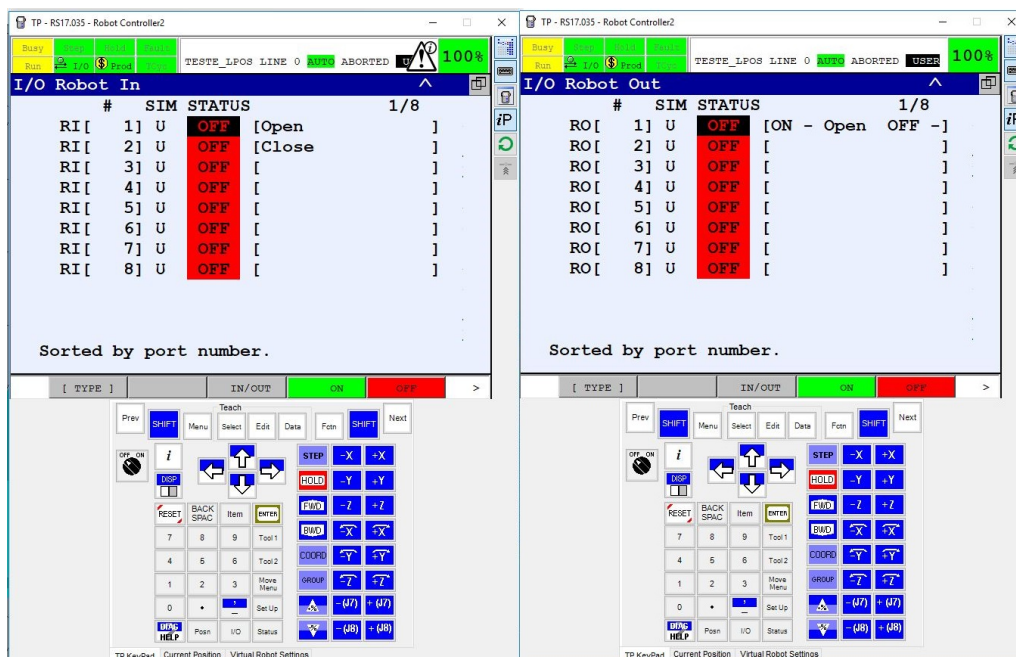


Figura 6.11: Sinais de I/O do robô.

Da mesma forma que se procede à configuração dos sinais digitais e de grupo, os sinais de interface com o PLC precisam de ser configurados no mesmo *rack* e *slot* dos anteriores, como ilustra a Figura 6.12. O número de sinais de entrada difere do número de sinais de saída em 2 bit. Estes sinais são definidos pela Fanuc e encontram-se expostos no Anexo A.

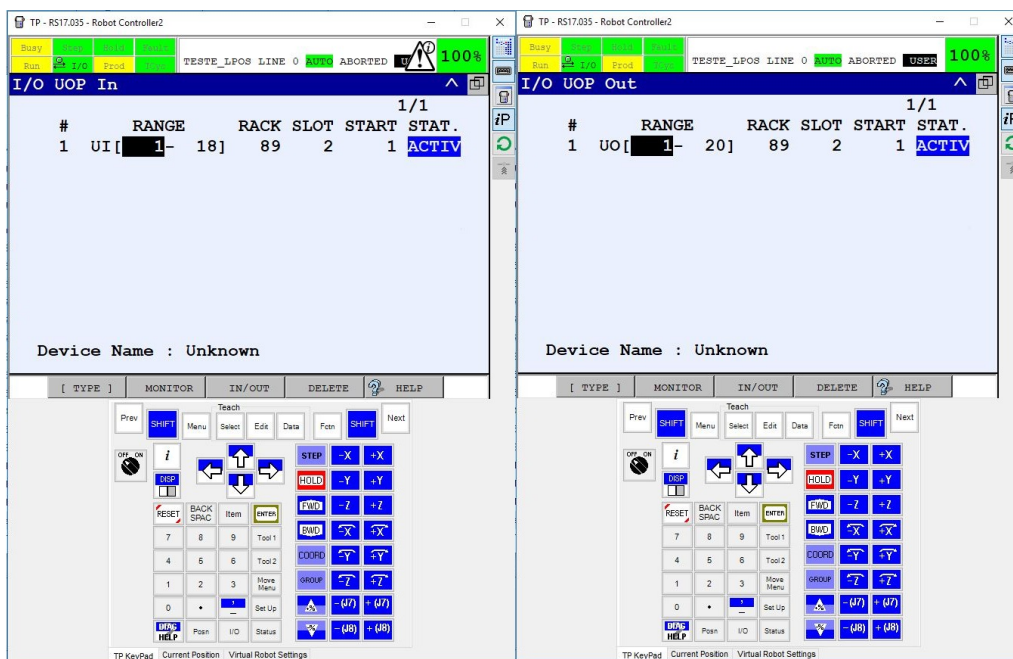


Figura 6.12: Sinais de I/O do robô de interface com PLC.

Os sinais de entrada para interface que a Fanuc disponibiliza são: IMSTP, HOLD, SFSPD, FAULT RESET, START, CSTOPI, HOME, ENBL, RSR/PNS, PNS-TROBE e PROD_START. O esquema que representa a troca destes sinais está ilustrado na Figura 6.13.

O IMSTP, para que o robô não se encontre em paragem forçada, deve estar sempre ativo. Quando ocorre um flanco descendente deste sinal o controlador aciona os travões, desliga a alimentação dos servos e exibe uma mensagem de erro.

O sinal de HOLD deve também estar sempre ativo. No seu flanco descendente, ou seja, quando é desativado interrompe o programa em execução e realiza uma paragem elétrica.

No caso de não ser pretendido executar uma paragem mas sim uma redução de velocidade utiliza-se o SFSPD. Deve também estar sempre ativo. Após a ocorrência de um flanco descendente deste sinal o controlador coloca o robô numa velocidade de segurança, interrompe o programa em execução e exibe uma mensagem de erro. O sinal para executar o *reset* e uma aceitação de erros é o FAULT RESET. O sinal de START é utilizado para arranque de forma remota. Por forma a executar uma paragem de um ciclo é ativado o sinal de CSTOPI.

É possível definir uma posição de casa para o robô, dentro de um programa do tipo macro, movê-lo para essa posição utilizando o sinal de HOME. A posição

casa corresponde a uma posição segura para o robô, pré-definida e marcada pelo programador ou operador.

O sinal que permite o movimento do robô é o sinal de ENBL que necessita de estar obrigatoriamente ativo.

Os sinais de RSR/PNS são sinais para seleção do programa a executar. Estes sinais ocupam as mesmas UO, desta forma é necessário selecionar se o arranque será feito via RSR ou PNS por forma a definir a utilização destas UO e a que sinais correspondem. Se o tipo de programa for PNS existe um sinal que valida o código do programa executado que é o PNSTROBE.

A Fanuc disponibiliza ainda um sinal para lançamento de produção que é o PROD_START.

Os sinais de saída para interface com PLC funcionam como *feedback* do estado do robô. Os sinais de saída disponibilizados pela Fanuc são: CMDENBL, SYSRDY, PROGRUN, PAUSED, HELD, FAULT, ATPMERCH, TPENBL, BATALM, BUSY, ACK/SNO e SNACK.

O CMDENBL é o sinal enviado pelo robô quando o seletor da consola está em OFF, o robô se encontra em modo REMOTE, os sinais de SFSPD e ENBL estão ativos, a variável do sistema \$RMT_MASTER a 0, o modo step desativo e o seletor de modo em AUTO. Este sinal prova que o controlador já se encontra preparado para o arranque remoto.

O sinal que indica que o sistema está preparado e os servos possuem tensão é o SYSTRDY e para indicar que um programa se encontra em execução no sistema é o sinal PROGRUN.

Quando um programa se encontra parado e à espera de ser reiniciado é ativado o sinal PAUSED e quando é pressionado o botão HOLD ativa-se o sinal HELD.

O sinal de FAULT é ativo quando o robô se encontra em erro e o sinal ATPERCH é ativo quando o robô se encontra numa posição de referência definida.

Quando o seletor da consola se encontra a ON é ativo o sinal de TPENBL e o sinal BATALAM é ativo quando as baterias se encontram num nível baixo de energia.

Os sinais de ACK são utilizados no arranque por RSR e confirmam a receção do programa RSR, os sinais de SNO executam a mesma tarefa mas no caso de programas por PNS e o sinal de SNACK confirma a receção do programa PNS e a sua aceitação.

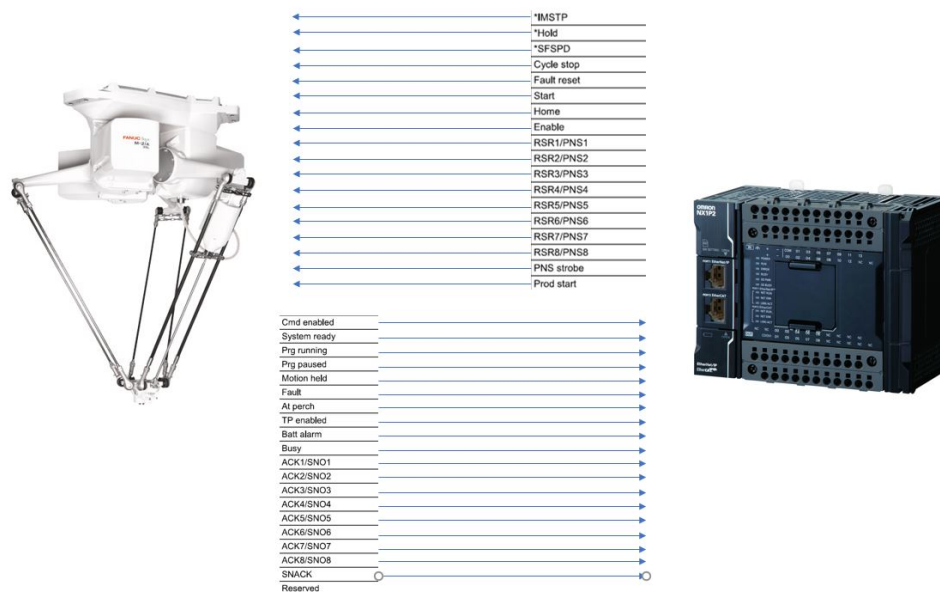


Figura 6.13: Sinais específicos para interface PLC e robô..

O robô 1, robô de descarga, necessita do mesmo tipo de configurações, variando apenas na definição de alguns sinais posteriormente. A configuração dos sinais é igual, apenas varia a utilidade de cada um que é definida pelo programador e pela tarefa a executar. Os sinais utilizados estão expostos no Anexo B.

6.2.2 Arranque remoto do robô

O arranque do robô pode ser ou não remoto. Se o arranque for local implica alguém a operar o *teach pendant* do robô. O objetivo do sistema é ser independente de um operador e controlada por um PLC - desta forma o método de arranque utilizado é o arranque remoto.

Os tipos de arranque remoto possíveis são: o arranque remoto de programa através de sinais UI, arranque remoto de programa via RSR, arranque remoto de programa via PNS, arranque através de STYLE e arranque através da opção OTHER.

Para o arranque através dos sinais UI é necessário configurar os mesmos. É necessário fazer a ativação dos sinais UI no menu CONFIG, que se encontra no menu SYSTEM, como ilustra a Figura 6.14. Para ser possível o arranque é necessário que os sinais CMDENBL, IMSTP, HOLD, SFSPD e ENABLE estejam a nível alto. O controlador tem de se encontrar em modo remoto, ativando essa opção em CONFIG, dentro do menu SYSTEM. O seletor do modo do controlador

deve encontrar-se em AUTO. No menu VARIABLES, do menu SYSTEM, deve-se colocar a variável \$RMT_MASTER a 0 e ainda no menu SYSTEM deve-se colocar a opção "Start For Continue Only" a FALSE, em CONFIG. No caso do TP este deve estar desabilitado, bem como a opção STEP, e caso não haja falhas o sinal SYSRDY estará a nível alto. A seleção do programa deve ser feito através do botão select do TP ou através das entradas RSR/PNS. Por fim, para executar o programa deve ser enviado o sinal START em flanco descendente.

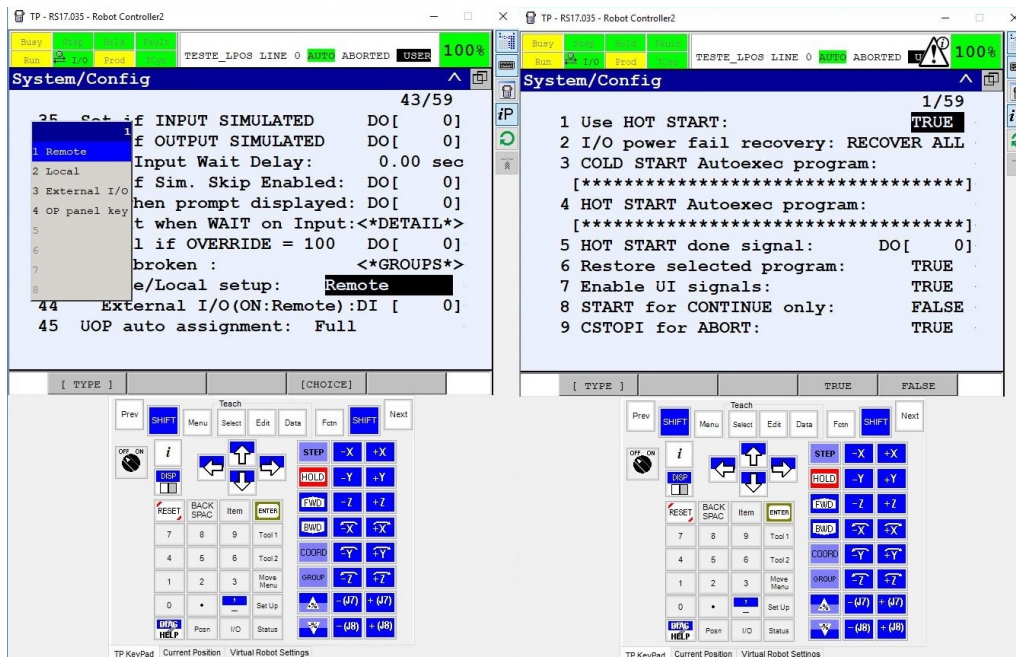


Figura 6.14: Configurações para arranque através dos sinais UI.

O arranque remoto de um programa via RSR baseia-se num pedido de serviço ao robô. Este pedido é feito através do envio de um sinal digital, sendo possíveis até oito pedidos. No momento em que o controlador recebe o sinal atesta a sua validade, e caso a aceite, determina o programa a ser executado e lança-o caso não exista nenhum em funcionamento; se existir, este fica em espera. Quando o robô recebe o sinal de RSR envia um sinal de validação através dos sinais de ACK. O nome do programa deve ser RSR[xxxx], em que [xxxx] é um número compreendido entre 0000 e 9999.

Para a utilização deste tipo de arranque é necessário configurá-lo. Cada entrada de RSR tem de ser ativada e tem de lhe ser atribuído um número. Para fazer essa ativação é necessário aceder a MENU, de seguida SETUP, e seleccionar a opção "Prog Select" e surgirá uma janela semelhante à da esquerda presente na Figura 6.15. Após a opção seleccionada é necessário seleccionar RSR e apa-

receberá uma janela semelhante à janela da direita da Figura 6.15. Os primeiros quatro valores foram atribuídos copiando os do manual da Fanuc e os restantes sem qualquer critério. Apenas se encontra ativo o primeiro RSR sendo o único necessário. Nesta janela é também necessário atribuir um número de base que estará no cálculo do valor a colocar no nome do programa RSR e o prefixo de trabalho que corresponde às letras RSR.

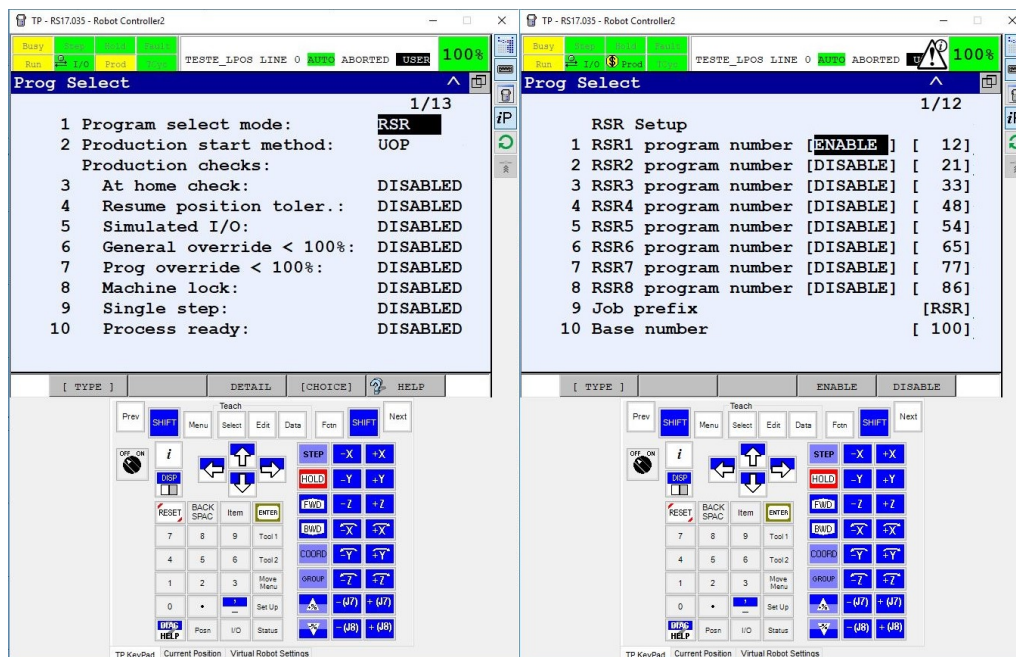


Figura 6.15: Configurações para arranque através dos sinais RSR.

O nome do programa a iniciar através do sinal RSR1 é RSR0112. Os dois primeiros algarismos correspondem à base (que é 0100) e os seguintes correspondem a este valor mais o definido anteriormente, ou seja 0100 + 12.

Após o estudo dos métodos de arranque (arranque via PNS, RSR, sinais UI, através de STYLE e através de OTHER) este método mostrou-se o mais indicado para a implementação. O método de UI surge associado ao arranque através de RSR uma vez que utilizando apenas os sinais UI levava à necessidade de selecionar o programa na consola e o objetivo é que o PLC lance o programa principal que faz a chamada dos restantes. Desta forma, o método de arranque utilizado foi o de RSR com recurso aos sinais de UI.

6.2.3 Configuração da ferramenta e da carga a movimentar

A ferramenta utilizada foi uma garra da Schunk, com um aumento face ao punho do robô, e dedos desenhados pelo departamento de mecânica da Robot-

Sol. Para a marcação das bases foi utilizada uma ponteira colocada num dos dedos da garra. Deste modo é necessário a configuração de duas ferramentas, como se verifica na Figura 6.16: uma garra com ponteira e uma garra. No caso da garra com ponteira a *utool* está definida na ponta da ponteira e no caso da garra sem ponteira está definida no centro da garra.

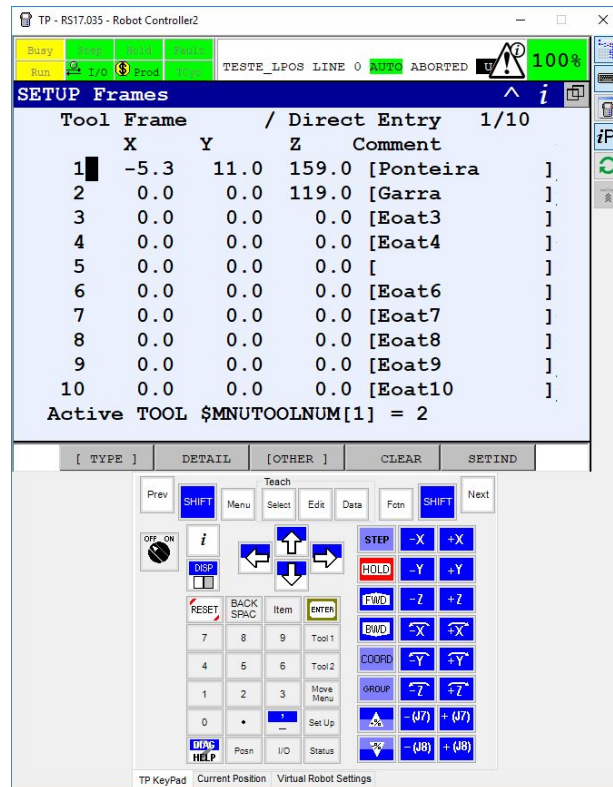


Figura 6.16: Menu de definição das ferramentas a utilizar.

A definição das ferramentas é feita acedendo a MENU, SETUP e de seguida selecionando a opção FRAMES. De seguida aparecerá uma janela semelhante à da Figura 6.16. Para definir uma ferramenta é necessário carregar na tecla DETAIL para abrir a janela de configuração da ferramenta.

A primeira ferramenta a configurar é a ponteira, por forma a marcar os sistemas de coordenadas. Essa ferramenta encontra-se rodada em relação ao sistema de coordenadas do punho do robô em $-57,976^\circ$. O limite da ponteira encontra-se distante da origem de coordenadas do punho do robô $-5,250$ mm no eixo dos XX, 11 mm no eixo dos YY e 159 mm no eixo dos ZZ. Estas configurações estão ilustradas na janela da esquerda na Figura 6.17.

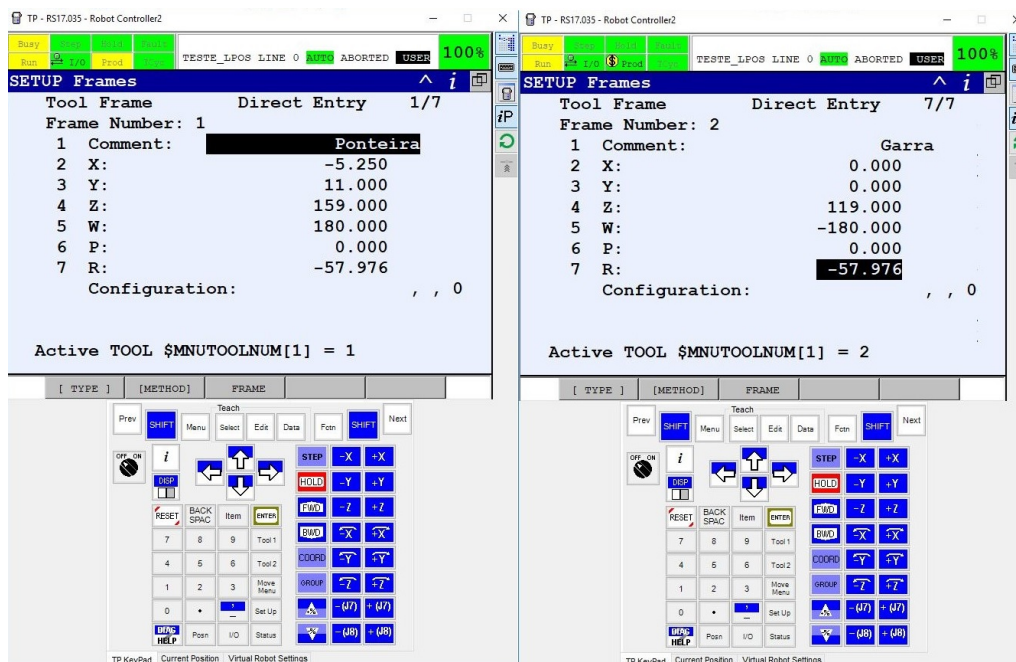


Figura 6.17: Configuração das ferramentas a utilizar.

A segunda ferramenta a configurar é a garra sem ponteira. Esta ferramenta possui o eixo de coordenadas rodado em relação ao do punho do robô na mesma grandeza que a ferramenta anterior, porém o seu centro é paralelo ao do punho. Este centro do eixo de coordenadas apenas se encontra distanciado 119 mm do punho. Estas configurações estão ilustradas na janela da direita na Figura 6.17.

6.2.4 Configuração dos sistemas de coordenadas

O robô possui 5 tipos base de sistemas de coordenadas, sendo um no domínio das juntas e quatro no domínio cartesiano. O sistema no domínio das juntas é o sistema JOINT e os sistemas no domínio cartesiano são: o WORLD, o TOOL, o USER e o JOGFRM.

O sistema de coordenadas JOINT especifica a rotação em graus de cada junta. No caso deste robô não é útil este sistema de coordenadas associado às 3 primeiras juntas, pois o movimento no espaço do robô é resultante das posições conjuntas destas juntas.

O sistema de coordenadas WORLD é um conjunto de coordenadas cartesianas absolutas cuja origem é definida pela Fanuc e localiza-se num ponto conceptual e não físico. Este sistema de coordenadas não pode ser alterado.

O sistema de coordenadas TOOL corresponde ao sistema de coordenadas referente à ferramenta que é configurado quando é feita a configuração da ferramenta.

O sistema de coordenadas USER é um sistema de coordenadas cartesiano relativo cuja origem e orientação são definidos pelo utilizador. É possível definir-se até 9 destes sistemas de coordenadas. Para configurar uma *user frame* é necessário aceder ao MENU, abrir o menu SETUP e seleccionar FRAMES. Aparecerá a janela presente na Figura 6.18 e é necessário pressionar o botão F3 abrindo a opção [OTHER]. Após seleccionar essa opção surge uma janela e nessa janela selecciona-se USER. Surge então a janela correspondente às *user frames*.

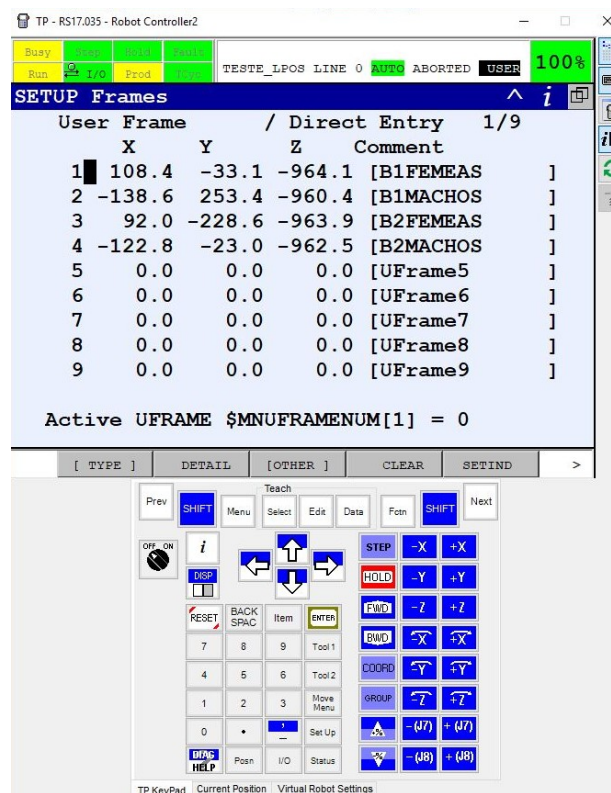


Figura 6.18: Menu dos sistemas de coordenadas de utilizador.

Para configurar uma *user frame* é necessário abrir a mesma pressionando o botão ENTER. Após seleccionada a *user frame* a configurar pressiona-se o botão F2 para seleccionar [METHOD] e surgem três métodos: três pontos, quatro pontos e método de entrada direta, como exemplifica a janela da esquerda da Figura 6.19. O método utilizado, quer na simulação, quer na prática, foi o método dos três pontos, que ao ser seleccionado abre a janela da direita da Figura 6.19.

O método dos três pontos consiste na marcação do ponto de origem, do ponto do sentido do eixo X e do ponto do sentido do eixo Y. Com estes três pontos o controlador calcula o sentido do eixo do Z e dá origem ao novo sistema de coordenadas.

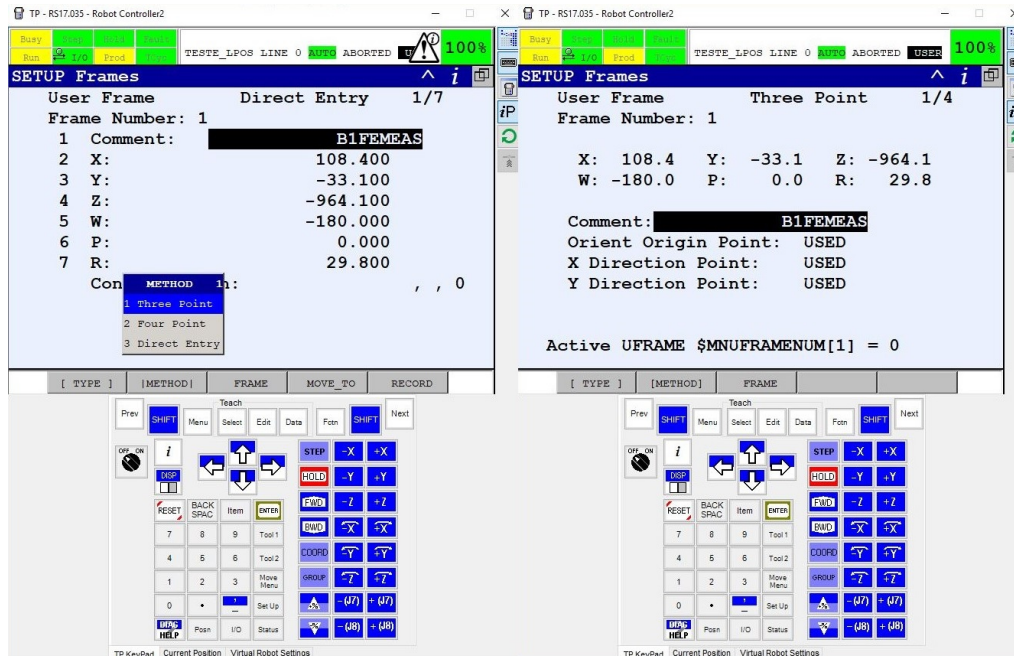


Figura 6.19: Configuração dos sistemas de coordenadas.

Foram marcadas quatro *user frames*, duas por cada base, como ilustra a Figura 6.20. Em cada base é marcada uma *user frame* para a zona onde são colocadas as peças macho e uma para a zona onde são colocadas as peças fêmeas.

Em todas as *user frames* marcadas, o eixo dos ZZ aponta no sentido descendente. Em cada uma das bases a orientação do eixo dos XX e do eixo dos YY é a mesma para a zona das peças macho e para a zona das peças fêmea. Entre bases as orientações dos eixos variam.

A marcação dos três pontos foi feita com o recurso à ponteira de marcação e com o auxílio dos parafusos de aperto das bases das fêmeas e dos machos à base geral. Esses parafusos serviram com pontos de guia para a marcação das bases. Cada ponto foi marcado colocando a ponteira no interior do parafuso, num ponto semelhante nos três parafusos.

No caso da base 1 a origem foi marcada no parafuso superior esquerdo para cada uma das bases de peça, como se verifica na Figura 6.20. O ponto que define as abcissas está marcado em paralelo com a origem no parafuso superior direito

e o ponto que define as ordenadas é marcado em paralelo com a origem no parafuso inferior esquerdo.

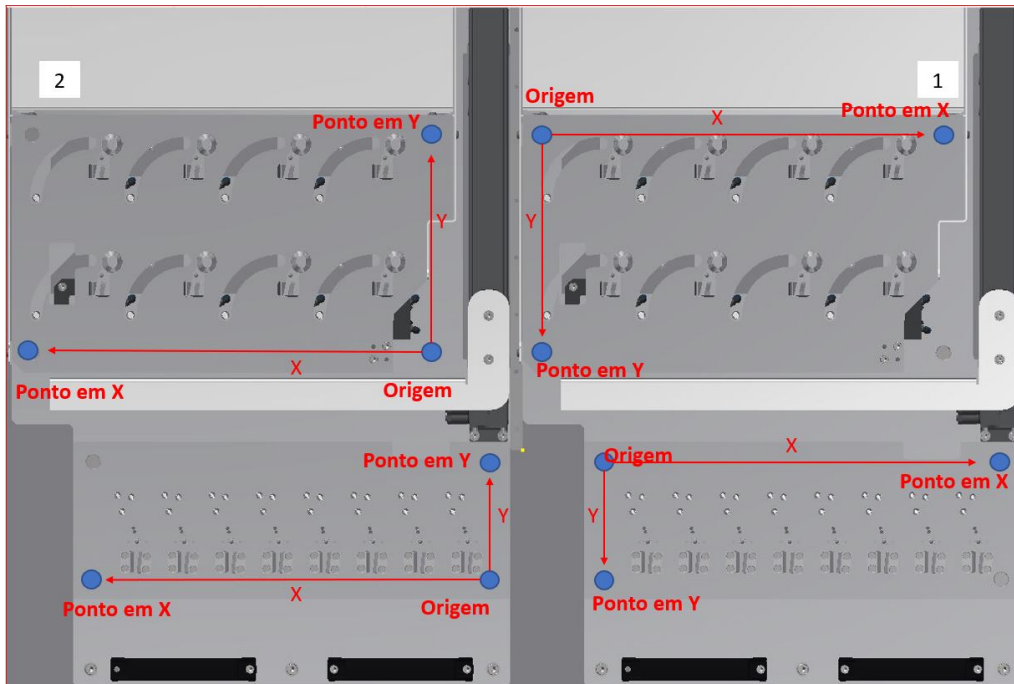


Figura 6.20: *User frames* marcadas em simulação e na prática.

No caso da base 2 a origem foi marcada no parafuso inferior direito, como está representado na Figura 6.20. O ponto que define as abcissas está marcado no parafuso inferior esquerdo e o ponto que define as ordenadas está marcado no parafuso superior direito.

É necessário que o ponto que define as abcissas esteja alinhado com o ponto da origem porque caso não seja marcado alinhado altera completamente o sistema de coordenadas. O controlador calcula como base o eixo das abcissas sempre alinhado com a origem e cria o restante referencial com base neste eixo.

6.2.5 Configuração do espaço partilhado entre robôs

Devido ao facto de existirem espaços de trabalho partilhados por ambos os robôs, existiu a necessidade de se criarem dois volumes de trabalho no espaço partilhado. Cada volume corresponde a uma das bases. Estes volumes são independentes, uma vez que os robôs devem poder trabalhar paralelamente em bases diferentes. Devido à possibilidade da ocorrência de uma colisão entre ambos, por partilha do espaço de trabalho, a Fanuc disponibiliza uma opção chamada *space function* para criação dos volumes de trabalho e evitar assim co-

lisões. Esta função define um volume de trabalho que quando um dos robôs se encontra no seu interior leva a que o outro pare imediatamente ao tentar entrar nesse volume. A cada volume de trabalho estão associados um sinal digital entrada e de saída. Quando um dos robôs se encontra com o TCP no interior do volume de trabalho o sinal de saída desse robô, referente ao volume de trabalho que ocupa, passa a nível baixo. Quando o sinal de entrada o outro robô, referente ao mesmo volume de trabalho e interligado com o de saída do outro robô, se encontra em nível baixo e este tenta aceder ao volume trabalho pára imediatamente. É necessário que os volumes de trabalho sejam coincidentes e possuam dimensão suficiente para evitar colisões entre atuadores finais, quando ambos os robôs trabalham paralelamente.

Para aceder à função *space function* é necessário pressionar o botão MENU, aceder ao menu SETUP e na segunda janela selecionar a opção SPACE FCBT. Após selecionar essa opção aparecerá uma janela semelhante à da Figura 6.21.

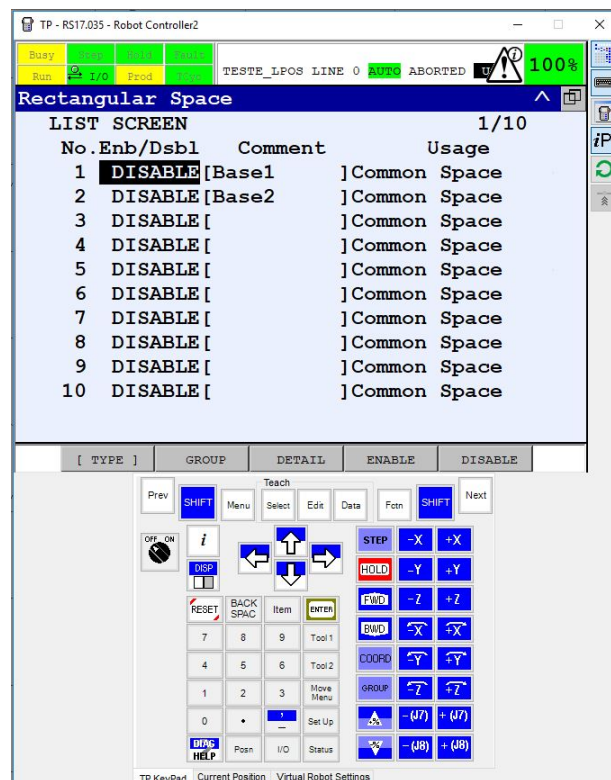


Figura 6.21: Menu de *space function*.

Podem ser configurados até 10 volumes de trabalho neste menu. Para a configuração de cada um é necessário clicar na tecla F3 por forma a abrir a janela DETAIL. Nessa janela, presente à esquerda na Figura 6.22, é possível ativar ou

desativar o volume de trabalho, definir os sinais, a prioridade e o método de ativação, que pode ser no interior ou exterior do volume. Ao selecionar a tecla F2 abre-se a janela à direita da Figura 6.22 e nessa janela é configurado o volume de trabalho. Para a configuração desse volume marca-se um primeiro vértice e depois pode o restante ser definido pela marcação de um segundo vértice ou através da definição do tamanho manualmente. O método utilizado foi o de marcação de dois vértices com o auxílio da ferramenta "Ponteira".

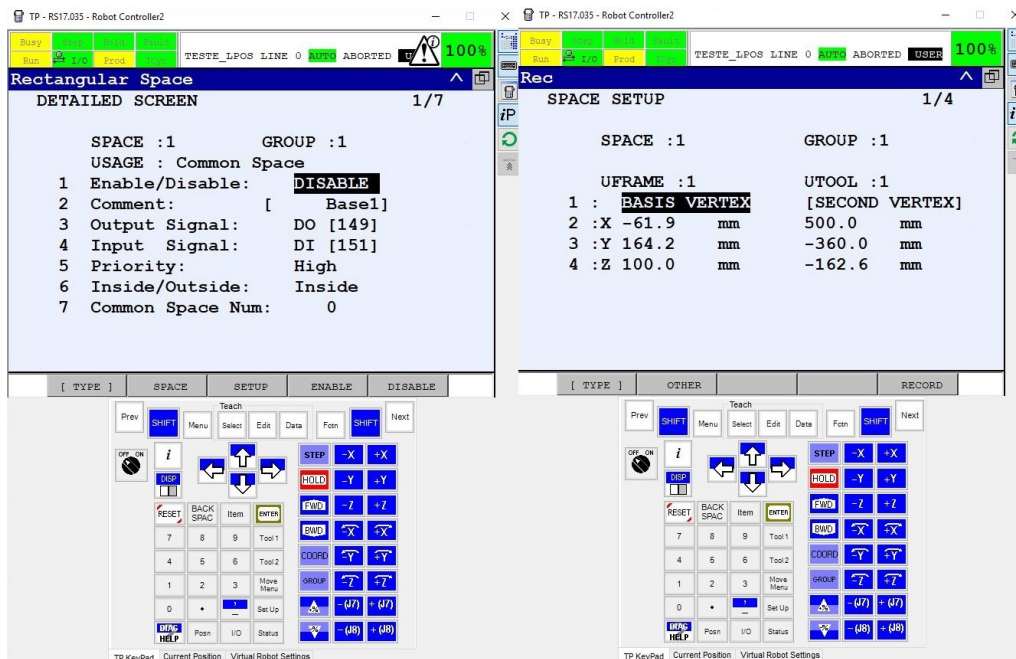


Figura 6.22: Menu de *space function*.

O sinal de saída configurado para o volume de trabalho da base 1 foi o DO[151] e o sinal de entrada foi o DI[151]. Para a base 2 o sinal de saída configurado foi o DO[152] e o sinal de entrada foi o DI[152]

6.2.6 Configuração da rede

Para que a célula trabalhe de forma autónoma é necessário que os robôs comuniquem com o centro de controlo da célula e também entre si. Essa comunicação é feita através do protocolo Ethernet/IP. Para utilização deste protocolo é necessário configurar alguns parâmetros. As configurações no robô são efetuadas pressionando o botão MENU, acedendo ao menu I/O e, na segunda página, escolher a opção Ethernet/IP. Após selecionar essa opção, abre-se uma janela semelhante à da esquerda da Figura 6.23 na qual aparecem as SLOT de comunicação. O robô 2 comunica com o PLC através da SLOT 2 enquanto que o robô 1

comunica através da SLOT 1. Clicando ENTER na SLOT que se pretende configurar abre-se a janela da direita na qual é definido o número de *words* a enviar e receber, que no caso são 16 *words*. A comunicação com o PLC através de *words* facilita na leitura e escrita de dados pois é comum à lógica do PLC utilizado.

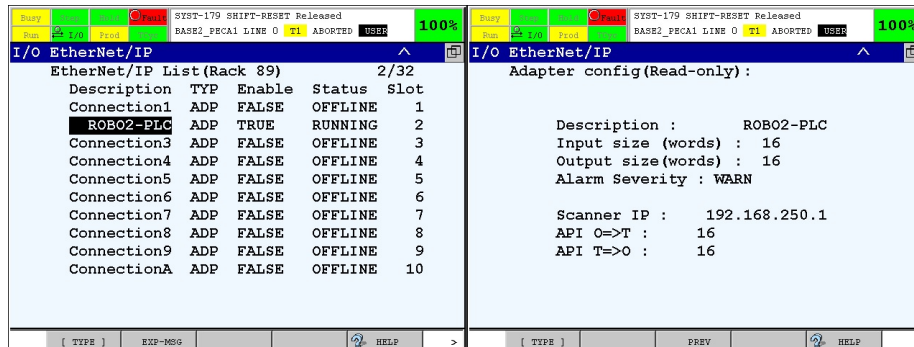


Figura 6.23: Configuração dos dados a transmitir.

Para que a comunicação funcione e o estado apareça *running* na janela de configuração de dados a transmitir, o que significa que a comunicação foi estabelecida, é necessário configurar os endereços dos componentes ligados em rede. Para configurar os endereços no robô é necessário pressionar a tecla MENU, aceder ao menu SETUP, e na segunda janela carregar em HOST COM. Na janela que se abre seleciona-se TCP/IP e abrirá uma janela igual à da Figura 6.24. Nessa janela é atribuído o endereço ao robô e mapeados os restantes endereços importantes. No caso apenas foi mapeado o PLC, uma vez que os robôs não comunicam diretamente entre si, mas sim através do PLC.

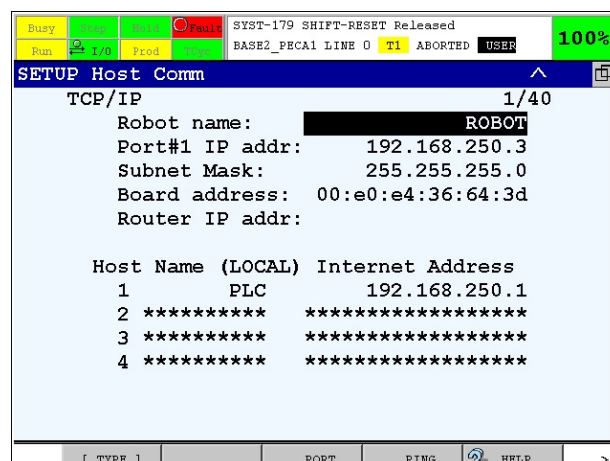


Figura 6.24: Configuração dos endereços.

6.2.7 Criação do conjunto de programas a executar

A lógica do movimento do robô está dividida por seis rotinas de código. Esta estrutura pode ser ampliada caso existam mais peças a serem produzidas. O diagrama, presente na Figura 6.25, explicita a chamada de programas por parte de cada uma das rotinas principais. O programa principal desencadeia as rotinas de manutenção e de peça, de acordo com determinadas condições. O programa de peça, mediante a leitura de sinais, produz a gestão de qual tipo de base deve estar inserida para executar o processo. Para além destas seis rotinas, existe também uma rotina de rejeição de peça no caso de esta não possuir par para a montagem, sendo essa rotina chamada por outras rotinas.

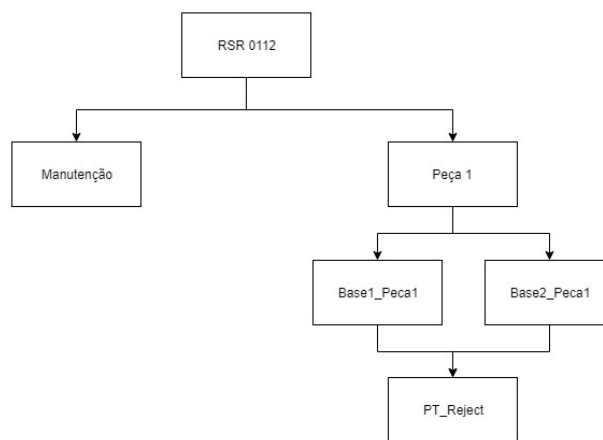


Figura 6.25: Diagrama de árvore das rotinas.

A rotina RSR é lançada através da ativação do sinal UI, proveniente do PLC, RSR1. Analisando o fluxograma presente no Anexo C.1, o robô executa a rotina de manutenção e finda essa rotina procede às inicializações dos sinais que informam que foram executados os trabalhos nas bases, colocando-os a OFF.

Após as inicializações procede-se à verificação de duas possíveis condições. A primeira condição consiste na receção do sinal de produção e na ausência de receção do sinal de pedido de acesso à célula. A segunda condição consiste na receção de um pedido de paragem controlada com ausência do sinal de pedido de acesso. Caso se verifique uma destas condições procede-se a uma verificação sequencial do código do programa. O código do programa consiste na atribuição de um valor a cada tarefa independente a executar, ou seja, cada programa a executar depende do valor enviado pelo PLC ao robô. Este código do programa possui valores compreendidos entre o valor 1 até ao valor 4. Quando este código de programa possui o valor 1 indica que o utilizador fez um pedido para que o robô se localize na posição HOME. No caso de o valor ser 2 o utilizador indicou

que pretende produzir a peça número 1, ou seja, a peça de 13 cm. Esta rotina está ainda preparada para a expansão prevista da produção de mais dois tipos de peças que se encontram com os valores 3 e 4, como se verifica na Tabela 6.1. Após o término destas rotinas é novamente verificada a condição anterior. Isto acontece em ciclo.

Tabela 6.1: Significado do valor recebido através do código do programa

Code Program GI[1]	Rotina a lançar
1	Manutencao
2	Peca1
3	Peca2
4	Peca3

O ciclo termina caso a condição anterior não se verifique e seja enviado um sinal de pedido de acesso. Quando esta condição se verifica o robô executa a rotina de manutenção e fica em espera que deixe de ser enviado o sinal de pedido de acesso e seja enviado um sinal de pedido de produção, regressando ao ciclo anterior. Para terminar a execução desta rotina RSR é necessário o envio de um sinal de CSTOPI ao robô.

O RSR no primeiro ciclo executa inicialmente a chamada à rotina de manutenção. Nesta rotina, como explicita o fluxograma do Anexo C.2, é atribuído o valor 0 à variável USERFRAME_NUM o que significa que o sistema de coordenadas a ser utilizado é o sistema de coordenadas WORLD do robô. É lida e gravada a posição atual linear do robô, ou seja em coordenadas cartesianas. Após a gravação da posição atual, esta é comparada com uma posição HOME previamente definida e caso as coordenadas sejam diferentes procede-se à abertura da garra, colocação da mesma numa altura de segurança e movimento para a posição de HOME. No caso de o robô já se encontrar nessa posição a rotina termina automaticamente.

Os registos nos quais são guardadas estas posições denominam-se de *position registers*, vulgarmente PR[X] ou PR[X, Y], em que X designa o número do PR e o Y designa a coordenada a alterar.

Após ser executada a rotina manutenção, a rotina seguinte a ser chamada é a rotina de peça. A rotina de peça em funcionamento é a Peca1. Nesta rotina, como exemplifica o fluxograma do Anexo C.3, a rotina espera por um sinal para operar na base 1 ou na base 2. Depois de receber esse sinal verifica se corresponde à base 1 e se corresponder lança o programa Base1_Peca1. Caso o sinal não seja correspondente à base 1 verifica se é o sinal para a base 2 e se o for lança a rotina Base2_Peca1. Após executar a tarefa esta rotina termina e regressa à

rotina principal. Esta rotina possui um tempo limite para receber cada um dos sinais, caso não aconteça volta à rotina principal.

Em resumo, a base do sistema segue a lógica presente no fluxograma do Anexo C.4.

Em cada base foram designados dois conjuntos: conjunto A e conjunto B. Cada um destes conjuntos engloba 4 peças macho e as fêmeas correspondentes. Estes conjuntos foram criados devido ao espaçamento entre as peças macho em cada base. Este espaçamento não permite a montagem de todas as peças fêmea e fecho das pinças, uma vez que levaria à colisão das peças. Esta colisão deve-se ao facto de na posição após a montagem a pinça aberta ocupa o espaço da montagem da peça fêmea no macho ao seu lado. Como ilustra a Figura 6.26, as peças que constituem cada conjunto de machos são intercaladas para evitar as colisões durante a montagem.

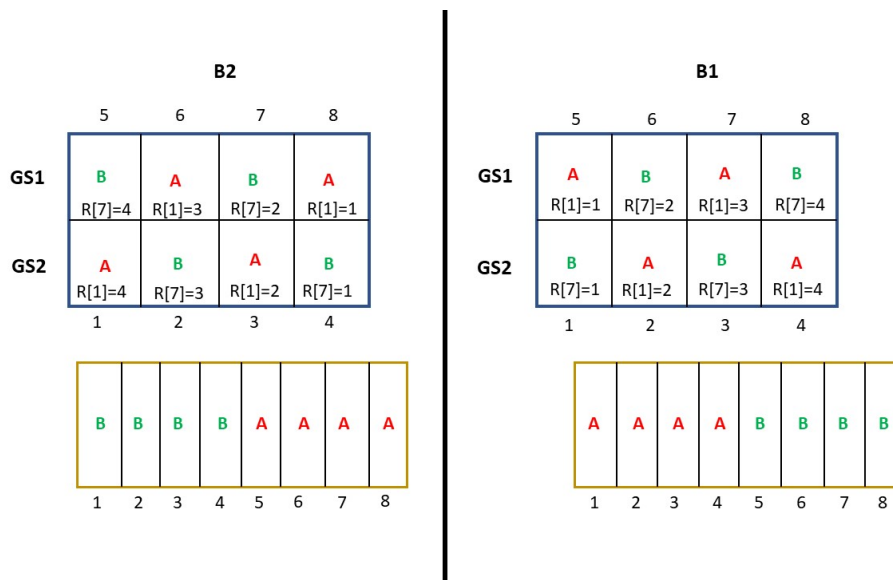


Figura 6.26: Esquema das bases para montagem das pinças.

Para cada base existe um programa diferenciado porém a lógica é mesma, como se verifica nos fluxogramas presentes nos Anexos C.5 e C.6. No início do programa de cada base faz-se a inicialização de todos os registos de posição a utilizar e registos auxiliares. São utilizados três registos de posição por cada base, um por cada conjunto de peças (conjunto de peças fêmea, conjuntos de machos A e conjunto de machos B). São utilizados quatro registos auxiliares, um deles para a contagem de ciclos e dos machos do conjunto A, outro para a contagem dos macho do conjunto B, e os restantes para a posição dos sensores, quer para fêmeas, quer para machos.

Após a inicialização verifica-se em que conjunto se está a trabalhar. De seguida verifica-se a existência de peça fêmea, que caso existam, leva à verificação da peça macho. No caso da existência das duas é montada a pinça; porém, se não existir peça macho é feita a rejeição da peça fêmea. No caso de não existir peça fêmea é feito o teste da existência da peça macho e caso exista esta é rejeitada. Esta verificação é feita para todas as peças dos dois conjuntos de montagem ilustrados na Figura 6.26. Após a montagem de ambos os conjuntos é terminada esta rotina e regressa à rotina de peça.

A dificuldade desta rotina prende-se com a necessidade de efetuar a mesma com condições IF consecutivas no interior de condições de ciclo FOR e de IF que representam o ciclo de montagem e a condição do conjunto. Para cada uma das bases está marcado um ponto para peças fêmea e oito pontos para peças macho. No caso das peças fêmea é utilizada a técnica de *offset*, quer no movimento lateral, quer no movimento vertical. No caso dos machos o *offset* é utilizado apenas no movimento vertical, porque as peças macho não se encontram equidistantes o que leva a erro na montagem.

6.3 Controlo da célula

O controlo da célula é feito através de um PLC que está ligado em rede com os dois robôs e a HMI. O PLC, para além de controlar a mesa de montagem e de permitir que os robôs executem os seus trabalhos, também faz a interface entre os robôs.

6.3.1 Configurações no PLC

Antes de se proceder à configuração da conexão EtherCat e das cartas a utilizar, bem como a configuração da rede, é necessário criar o projeto de programação do controlo e definir o PLC a utilizar. No *software* Sysmac Studio cria-se o projeto no qual é configurada a conexão EtherCat e a rede Ethernet/IP. Também através deste *software* é programado o PLC. O seu ambiente gráfico está ilustrado na Figura 6.27.

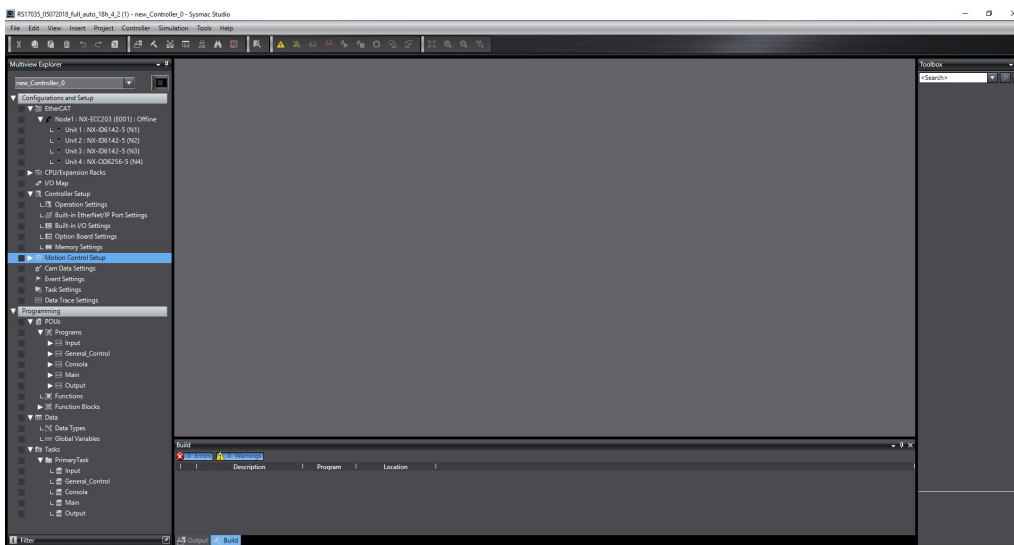


Figura 6.27: Ambiente de trabalho do Sysmac Studio.

Na zona lateral esquerda da janela existe um menu explorador no qual são efetuadas grande parte das configurações. Numa primeira instância procedeu-se à configuração do módulo EtherCat que faz a comunicação entre o PLC e as cartas de entradas e saídas. Para configurar este módulo é necessário abrir o menu EtherCat, clicando no explorador da esquerda onde diz "EtherCAT". De seguida abre-se uma janela na qual se colocam todos os módulos EtherCAT presentes na rede, no caso o módulo utilizado é o NX-ECC203. Clicando sobre o módulo surge o nó onde se acrescentam as cartas a que estão ligadas, neste caso as cartas de entradas e de saída. No Anexo D.1 estão presentes estas configurações.

Após configurada a rede EtherCAT procede-se ao mapeamento das entradas e saídas. Para isso é necessário criar um conjunto de variáveis para atribuição dessas entradas e saídas. O mapa completo de variáveis encontra-se no Anexo D.2. No menu explorador seleciona-se a opção "I/O Map" e abre-se uma janela semelhante à do Anexo D.2. Nesta janela são mapeadas todas as variáveis referentes aos sinais de entrada e saída, quer das cartas de I/O, quer das entradas e saídas que o próprio PLC possui.

No menu de explorador existe um menu chamado DATA, composto por dois sub-menus: "Global Variables" e "Data Types". Nas "Global Variables" são definidas as variáveis globais e no "Data Types" criados os tipos de dados.

Para a criação de programas abre-se o menu "Programming" e no separador "POUs" acrescentam-se os programas que podem ser compostos por várias secções.

Para que o PLC execute estes programas é necessário adicioná-los no menu de "Task Settings" a tarefas a executar. Abrindo este menu seleciona-se o icon de "Program Assignment Settings" e nessa aba acrescentam-se os programas a executar, como mostra Anexo D.3.

Por fim é necessário configurar a rede de comunicação entre o PLC e os robôs. Para configurar esta rede do lado do PLC utilizaram-se às configurações no Sysmac e no Network configurator. Utilizou-se o Network configurator uma vez que através do Sysmac não estava a ser possível configurar a rede de forma funcional. Para configurar no Sysmac é necessário pressionar Tools no topo da janela e selecionar "Ethernet/IP Connection Settings". Abre-se uma janela em que surge o nó de conexão referente ao PLC e abre-se esse nó. Na janela que abrirá acrescentam-se os dois robôs, clicando no botão mais no lado direito da janela, atribuindo o endereço de cada um e associando o ficheiro de dados, o *Electronic Device Data Sheet* (EDS), de cada um que contém todas as informações necessárias ao PLC sobre o controlador do robô. Ainda nesta janela, declaram-se as variáveis de comunicação de entrada e saída e o seu tamanho e do lado esquerdo clica-se no ícone abaixo do "Tag Set" e abre-se a janela "Connections" na qual são definidos os tamanhos dos dados a transmitir e são atribuídas as variáveis à transmissão. Este procedimento está ilustrado no Anexo D.4.

É necessário proceder à mesma configuração no *software* Network configurator, acrescentando a informação do número de *words* a transmitir. No Network configurator é feito o *scan* dos dispositivos associados na rede. De seguida atribui-se o EDS a cada um dos robôs e após possuírem as informações contidas no EDS e ao se arrastarem os ícones dos mesmos para cima do ícone do PLC procede-se às configurações acima descritas. Esta janela está ilustrada também no Anexo D.4.

Após a configuração da rede já é possível a troca de sinais entre o PLC e os robôs.

6.3.2 Lógica implementada no controlo

A lógica de controlo está presente nos programas criados para o controlo da célula e interface com os robôs.

O controlo está organizado em cinco partes constituintes: Input, General_Control, Consola, Main e Output. Esta organização está ilustrada na Figura 6.28. No auxílio deste controlo recorreu-se à criação de *function blocks* para determinadas tarefas.

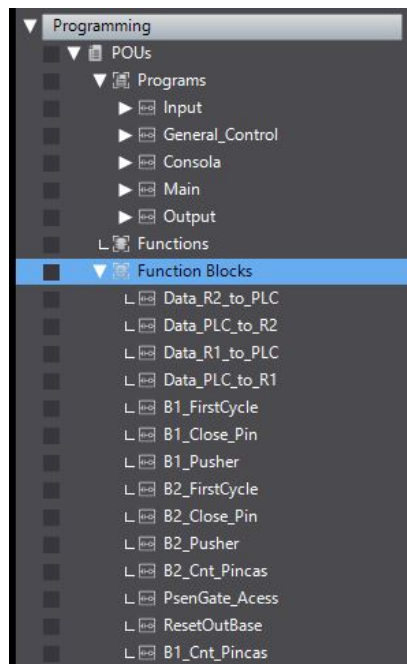


Figura 6.28: Organização da programação do controlo.

O programa Input é composto por quatro secções: B1, B2, Others e Robot. Neste programa são feitas as leituras de todas as entradas e copiadas para variáveis de trabalho.

Na hierarquia segue-se o programa General_Control, que é composto por oito secções: StartControl, AccessRequest, AirControl, Recipes, StartR1, StartR2, RobotWorkVolume e RobotR2Sensors. Este programa é responsável pelo arranque e controlo geral da célula.

Também foi implementado um programa para interface com a consola e gestão de alarmes, denominado de Consola. Este programa é composto pelas seguintes secções: PLC_to_HMI, HMI_to_PLC e Alarms.

Para o controlo das ações de cada base, contagem de peças, controlo do ciclo da mesa, controlo da torre luminosa, deteção de problemas e contagem de tempos improdutivos foi criado um programa com o nome Main. Na composição deste programa estão as seguintes secções: First_Cycle_Reset, R1_PLC_Interface, Counter_Pieces, R2_Mount_Close_pieces, Push, Act_Table_B1, Act_Table_B2, Table_Conditions, Luminous_Tower, Stop_Times_Simple e Debbug.

O último programa da hierarquia é denominado de Output e é semelhante ao Input porém com o objetivo de fazer a escrita nas saídas dos valores provenientes de variáveis de trabalho. As secções que o compõem são: B1, B2, Others e Robot.

A árvore dos programas enunciados encontra-se ilustrada na Figura 6.29.

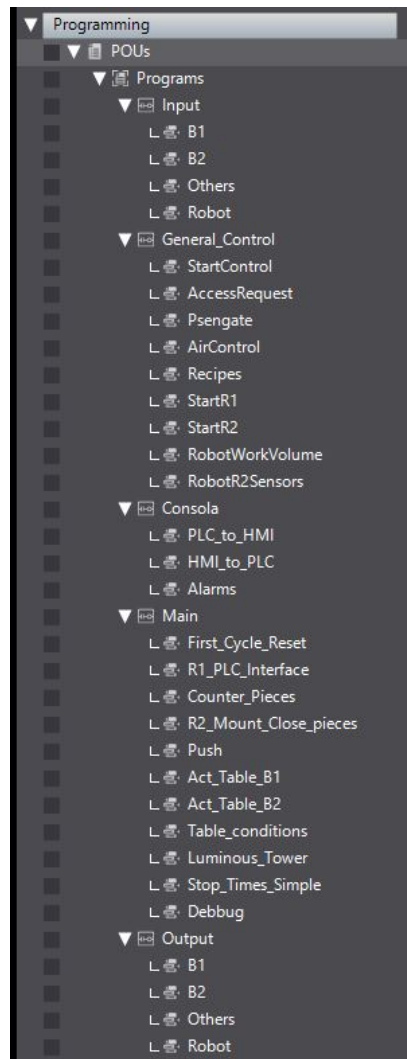


Figura 6.29: Organização de cada programa do controlo.

No programa Input são lidas variáveis mapeadas nas portas referentes aos sensores e através de um código em ladder colocam-se os valores em variáveis de trabalho. Esta leitura é efetuada para todos os sensores de ambas as bases associados às partes móveis de cada uma e sensores de peça. No caso da secção Others é efetuada a leitura das portas do PLC, bem como dos sensores de montagem da base, da posição de gaveta e da existência de caixas. Nesta rotina é também efetuada a leitura das 16 *words*, previamente definidas durante a configuração da rede, provenientes do robô através de dois *function blocks* semelhantes, um para cada robô, denominados de Data_R1_to_PLC e Data_R2_to_PLC.

Neste *function block* é feita uma cópia dos valores presentes em cada *word* do vetor de transmissão para variáveis de trabalho. O mapeamento das variáveis e a ilustração da cópia dos valores de entrada estão mapeados no Anexo E.1, bem como o diagrama de blocos que ilustra o processo.

Devido à velocidade de processamento dos programas, o PLC oferece a noção de que as secções de programa são processadas em paralelo, apesar de ser sequencial. No programa *General_Control* ocorre a execução das secções *StartControl*, *AccessRequest*, *AirControl*, *Recipes*, *RobotWorkVolume* e *RobotR2Sensors* praticamente em paralelo. As secções *StartR1* e *StartR2* arrancam apenas quando a rotina de *StartControl* se encontra no *step* referente ao arranque dos robôs (após verificação das emergências), como ilustra a Figura 6.30.

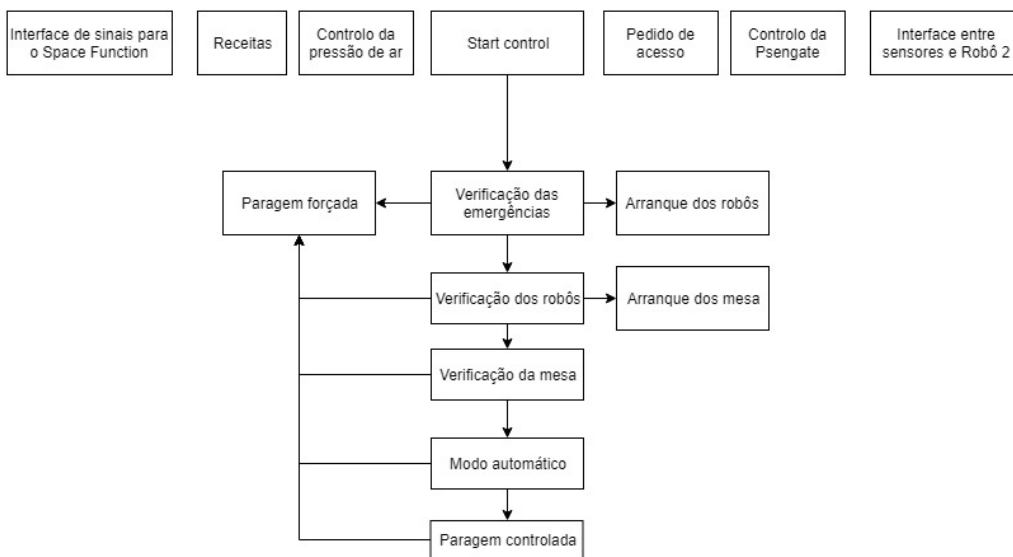


Figura 6.30: Esquema do programa *General_Control*.

Na secção *StartControl*, ilustrada pelo fluxograma do Anexo E.2, é feito o controlo do arranque da célula, inicialmente através da verificação do estado das emergências, de seguida verificação do estado dos robôs, e por último, a verificação do estado da mesa. Após estas verificações o sistema é colocado em automático. Quando o botão de *start* é pressionado inicia-se o arranque do sistema. Durante o arranque é efetuada a verificação do estado das emergências, e no caso de não existir nenhuma emergência, efetua-se o arranque dos robôs através das secções *StartR1* e *StartR2*. Terminado o arranque dos robôs procede-se ao reinício das bases para o estado de produção, no caso de estas estarem livres de peças. Caso o sistema esteja em modo automático é possível efetuar uma paragem controlada ao pressionar o botão *Stop*, sendo que este modo só pára o sistema após o fim do ciclo com os robôs em posição de segurança. Se este

botão for pressionado durante um período superior a 4 segundos desencadeia-se uma paragem forçada que pára o sistema.

Nas rotinas de StartR1 e StartR2, como ilustram os fluxogramas dos Anexos E.5 e E.6, são enviados os sinais IMSTP, HOLD, SFSPS e ENBL que o robô necessita para iniciar o seu movimento. De seguida é efetuado o reinício das falhas que se repete a cada vez em que se pressiona no botão Start. Após se verificar que o robô está pronto para efetuar uma tarefa, e que não possui tarefa nenhuma atribuída, é feito o envio do RSR a executar. Por fim é enviado o código do programa a executar. Também nesta rotina é feito o envio do código do programa a executar numa paragem controlada e definida a opção de saída do modo pausa. Para além disto, na rotina do Robô 1 existe o envio de um sinal para parar a máquina de injeção quando está próximo o fim de produção.

O fluxograma do Anexo E.3 representa as restantes rotinas para o controlo geral da célula. Na rotina AccessRequest é verificado se o sistema se encontra em modo automático, e caso se encontre e seja efetuado um pedido de acesso, é concedido o acesso à célula quando esta se encontrar em segurança, parada com os robôs em posições seguras. Na secção Psengate é efetuada a gestão dos sinais necessários a trancar e destrancar a Psengate, bem como a interface entre a psengate e o sistema. O controlo da pressão de ar é efetuado através da verificação do sinal erro na secção AirControl. Por fim, na secção Recipes é verificada a receita proveniente da HMI e feito o carregamento da mesma para o sistema.

O programa seguinte é o de interface com a consola. Na secção PLC_to_HMI são enviados o número de peças produzidas e o número de peças perdidas. É enviado também cada um destes dados mas por minuto, bem como a produção por base. Nesta secção são enviados também os estados dos sensores à HMI. Na secção HMI_to_PLC são lidos sinais para reiniciar contadores, número de peças a produzir por base, tipo de peça, sinal para carregar a receita e ativar o sistema. Nesta secção também é possível ativar-se um modo manual que permite ativar cada uma das partes da mesa através do HMI. Este modo está apenas disponível para o administrador do sistema. Na secção Alarms são enviados sinais à HMI de que alguma parte de cada base não executou o movimento pretendido, ou seja, se por exemplo as guias possuem sinal para avançar e não se encontram avançadas ao fim de um período de tempo. Esta rotina permite verificar o que está a bloquear o sistema no caso das bases da mesa. No Anexo E.7 está ilustrado o funcionamento deste programa.

No programa Main é efetuado e controlado o ciclo de montagem das pinças e da produção. A atuação das saídas é efetuada nas secções Act_Table_B1 e Act_Table_B2, nas quais são verificadas as condições para atuação das guias, pinos rotativos e empurrador. Consoante condições satisfeitas em outras secções de código, as guias avançam ou recuam, os pinos de fecho avançam ou recuam,

o empurrador sobe ou desce e avança ou recua. Uma parte constituinte do programa Main é o ciclo de montagem, representado pelo fluxograma do Anexo E.8. Neste ciclo de montagem, em cada secção é feita a chamada de blocos de funções nos quais é avaliado o estado do sistema e atuadas saídas, nas secções Act_Table_B1 e Act_Table_B2, de acordo com as condições destes blocos de funções.

Na secção First_Cycle_Reset são chamados os blocos de funções responsáveis pela inicialização das bases, caso não se encontrem em posição de trabalho, denominados de B1_FirstCycle e B2_FirstCycle. Neste ciclo verifica-se se o empurrador está recuado e em baixo, se apenas a guia 1 está elevada e se os pinos de fecho se encontram recuados. Se não acontecer, como ilustrado pelo fluxograma do Anexo E.9, são baixadas ambas as guias, de seguida elevado o empurrador, recuado o empurrador, subida a guia 1 e recuados os pinos de fecho. Os eventos ocorrem por esta ordem por forma a evitar colisões entre os componentes. Ainda no fim deste ciclo é enviado ao robô de descarga o sinal da base em que irá fazer a descarga.

Após a mesa estar pronta para trabalho procede-se ao ciclo de interface com o robô 1, por forma a que este deposite as peças na mesa. Esta interface é efetuada através de pedidos de acesso à área por parte do mesmo, que são concedidos quando a mesa se encontra desimpedida. Durante o ciclo é elevado o segundo conjunto de guias, após o robô descarregar as 8 fêmeas e os primeiros 4 machos, por forma a descarregar os restantes. No fim do ciclo o robô envia um sinal de área livre para que possa continuar o ciclo de montagem.

A secção que se segue é a de contagem de peças na qual se recorre a blocos de funções que fazem a verificação dos sinais e incrementam as variáveis de contagem. Esta contagem é feita uma vez em cada ciclo, analisando todos os sensores de peça. Os blocos utilizados denominam-se de B1_Cnt_pincas e B2_Cnt_pincas. Nestes blocos são verificados os sensores ativos e inativos e os valores registados são contados e acrescentados às peças produzidas e perdidas. Caso as peças apareçam sem par são tidas como perdidas, aquelas que possuem par são tidas como produzidas. Deste bloco de funções resultam os valores de peças produzidas e perdidas por ciclo e o total das mesmas.

Terminada a contagem procede-se à montagem das peças por parte do robô e fecho das peças por parte da mesa. Para tal, como ilustra o fluxograma do Anexo E.10, inicialmente envia-se ao robô de montagem a base a operar e antes de enviar o conjunto de operação ordena-se o avanço dos pinos do primeiro conjunto. Quando estes já se encontram avançados, já se podem montar as fêmeas e no fim ordenar o seu recuo para finalizar a montagem. No momento em que os pinos do primeiro conjunto se encontram recuados, efetua-se o avanço do segundo conjunto e ordena-se ao robô a sua montagem. No final da montagem do

segundo conjunto efetua-se o mesmo recuo de pinos. Terminada a montagem e fecho das peças envia-se o sinal para recuar as guias.

Por fim, o ciclo do empurrador verifica se as guias estão em baixo, manda subir o empurrador e empurra peças para a caixa colocada por baixo da base.

As restantes secções têm por objetivo recolher informação ou controlar outras partes do sistema. Na secção `Table_conditions` é verificado se a mesa tem peças ou está livre, se já atingiu a produção definida e se se encontra livre para a descarga. A secção `Luminous_Tower` ativa as luzes da torre luminosa, de acordo com um esquema pré-definido e representado na Tabela 6.2. A rotina `Stop_Times_Simple` efetua a contagem e memória dos tempos de paragem por emergência, fim de produção e porta da célula aberta. Na secção `Debbug` é efetuada a análise do número de falhas de uma mesma peça e dado um alerta quando esta falha mais de três vezes consecutivas.

Tabela 6.2: Esquema de cores e significados da torre luminosa

Cor	Significado
Vermelho fixo	Linha parada
Vermelho intermitente	Linha em emergência
Amarelo fixo	Necessita da atenção do utilizador (não bloqueia o processo)
Amarelo intermitente	Necessita da atenção do utilizador (bloqueia o processo)
Verde fixo	Linha em automático
Verde intermitente	Linha a arrancar
Azul	Acesso autorizado à linha

Por fim, o programa `Output`, através do qual é copiado o valor de todas as variáveis de trabalho utilizadas ao longo das rotinas anteriores para as variáveis de saída, por forma a atuar as saídas físicas, como está representado no fluxograma do Anexo E.11. Este programa possui secções semelhantes ao de `Input`.

6.4 Conclusão

Em suma, este capítulo tem como finalidade englobar todos os passos tomados ao longo da montagem e programação do sistema até ao mesmo começar a produzir. É perceptível que desde a simulação à implementação é um processo que demora algum tempo, por forma a evitar erros que mais tarde possam inviabilizar o projeto. Desta forma, é necessário verificar se cada um dos componentes constituintes cumpre as tarefas em ambiente de simulação por forma a escolher o mais adequado. Na parte de implementação prática é necessário tratar de cada componente individualmente, tendo em conta a tarefa que vai executar e a sua prioridade e interface com os restantes componentes. No caso da gestão da célula, é necessária máxima atenção às condições de atuação de cada componente por forma a evitar acidentes que possam danificar o sistema.

Capítulo 7

Testes e Resultados

Após o estudo de cada componente e das soluções, no presente capítulo é estudada a arquitetura do sistema e a criação de um algoritmo de implementação. Ao longo de cada uma das fases já expostas anteriormente é necessário ir procedendo a testes por forma a validar cada fase do projeto. Desde a fase de simulação, passando pela fase de montagem e comunicação, até à fase de produção, foi necessário procederem-se a testes por forma a perceber a validade da solução. Os resultados são consequência dos testes feitos ao longo da vida do projeto, resultando da sua análise a conclusão da conformidade do protótipo final com os objetivos propostos cumprir.

7.1 Testes em ambiente simulação

7.1.1 Teste de validação da mesa, bases e garra

Para se proceder à encomenda de material para a montagem do sistema foi necessário proceder a testes de alcance. Estes testes serviram para validar o posicionamento das bases dentro do volume de trabalho do robô e do comprimento da garra para a montagem. O comprimento da garra é um fator vital porque um comprimento excessivo pode colocar em risco o movimento do robô e aumentar o perigo de colisão, e o comprimento reduzido pode conduzir a que o robô não alcance o ponto de montagem e de pegar peça.

Tendo em conta o alcance do robô e a capacidade do simulador de detetar colisões procedeu-se à validação do *layout*. Foi possível verificar o alcance por parte do robô de todos os pontos necessários para pegar peças e montar pinça. A verificação do ponto de montagem foi efetuado através do estudo de colisão entre a peça, colocada no atuador final, e a base de montagem. Esta colisão define o ponto limite que o robô pode alcançar, uma vez que pode representar dano peça. Este processo está ilustrado na Figura 7.1.

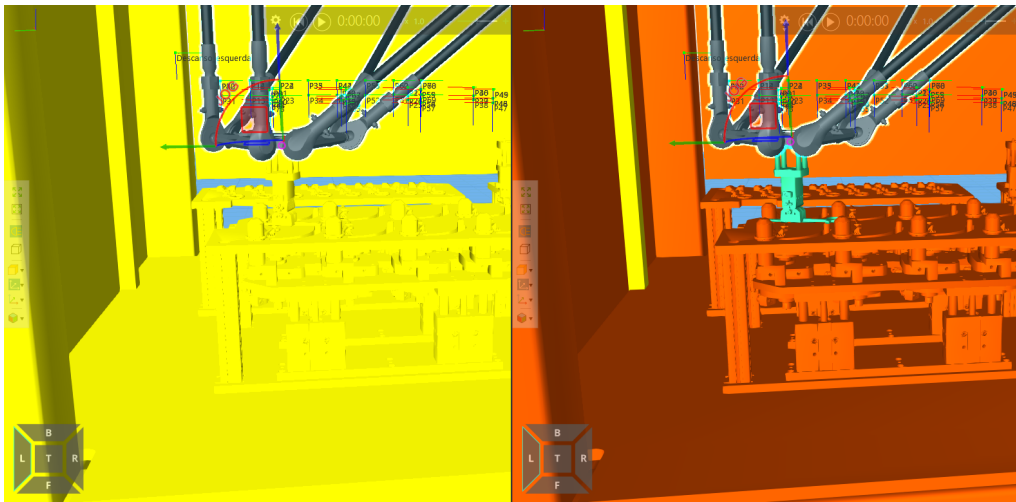


Figura 7.1: Simulação de colisão no Visual Components.

7.1.2 Teste do arranque do robô

O arranque do robô é feito através de um conjunto de sinais expostos no Capítulo 6. Estes sinais são simulados através uma ferramenta que o Roboguide disponibiliza, denominada de I/O pane. Através deste painel é possível ativar-se e desativar-se qualquer entrada que se deseje e verificar o estado de qualquer saída. Desta forma, através deste painel foram simulados todos os sinais necessários para o arranque da rotina RSR. Esta rotina, em analogia, corresponde à *main* do robô.

Para o teste de arranque é necessário selecionar os sinais necessários ao funcionamento do robô em automático, especificados no Capítulo 6 (IMSTP, HOLD, entre outros), e enviado o sinal de RSR. O sinal de RSR a ser enviado para testar o arranque é o UI[9] que corresponde ao RSR1. O ambiente de teste está ilustrado na Figura 7.2.

O resultado obtido em ambiente de simulação foi o lançamento do programa do robô RSR112. Ainda durante este teste foram testadas as paragens através de sinais do sistema, como o sinal de Cycle Stop em flanco descendente, e através do retiro de sinais necessários ao funcionamento, como o IMSRP e o Hold.

O simulador Roboguide permitiu efetuar, com resultados satisfatórios, os testes de controlo de lançamento, paragem e término do programa RSR que será utilizado como *main* do robô. Porém, o simulador não permite a simulação de pausar programa e retomá-lo, ficando preso num ciclo em que o sinal de Hold System não é retomado, levando a que seja necessário reiniciar o lançamento do programa e o arranque do sistema.

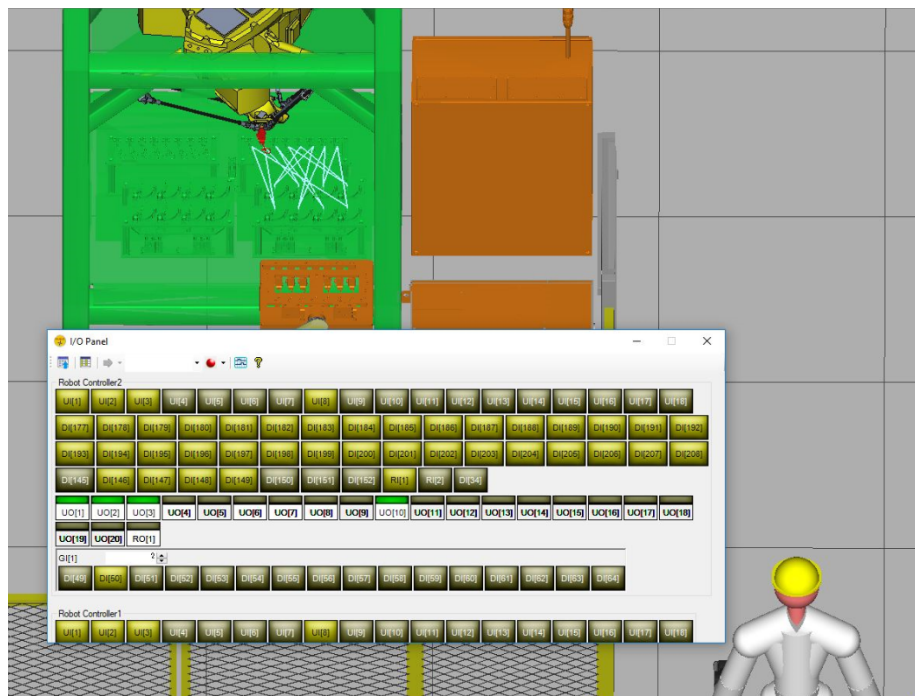


Figura 7.2: Simulação de arranque do robô via RSR no roboguide.

7.1.3 Testes de movimento controlado por sinais

Através do I/O panel simulam-se os sinais necessários para cumprir as condições de movimento. É feita a simulação do envio de tarefa e sinais de base a operar. Após a receção desses sinais por parte do robô são executadas as respectivas rotinas de trabalho.

Os movimentos produzidos em cada base estão representados pelas linhas desenhadas nas mesas da Figura 7.3. Estas linhas representam os movimentos que o robô faz entre as fêmeas e os machos, por cada ciclo de operação da base.

Visto que o cenário foi previamente validado, não existiu nenhum problema com os movimentos do robô, não existindo quaisquer colisões com alguma parte da estrutura. Os movimentos são controlados por sinais que indicam ao robô qual a base a operar e quais as peças que se encontram para montagem, ou seja, os sinais recebidos pelos sensores.

Os resultados foram satisfatórios, permitindo validar as rotinas de *pick and place* das peças com e sem rotina de rejeição de peças, bem como o seu controlo através de sinais provenientes do PLC. Deste modo, foi validado o movimento e ciclo de trabalho do robô controlado por sinais externos, neste caso simulados.

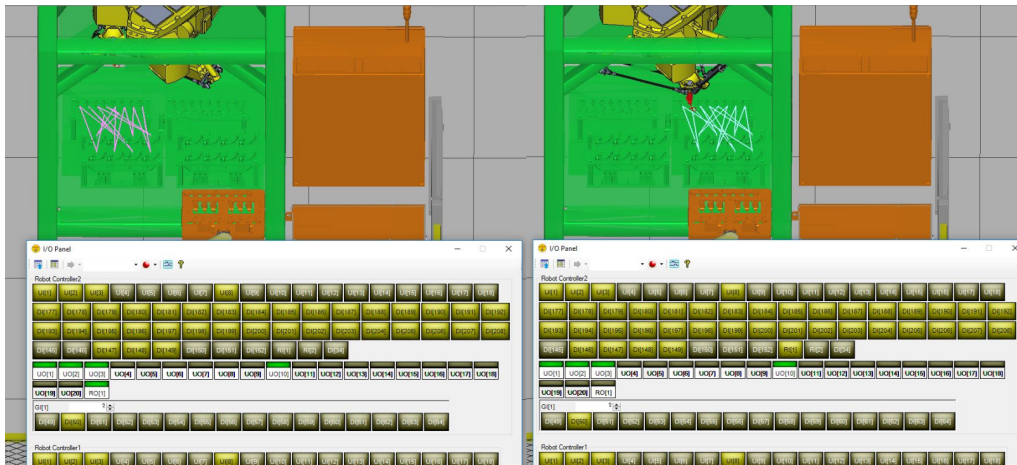


Figura 7.3: Simulação dos movimentos do robô no roboguide.

7.1.4 Teste do tempo de movimento do robô

Com recurso ao menu de simulação, é efetuado o teste dos movimentos do robô em modo automático e conseguida a perceção dos tempos através do menu profile. Este teste tem por objetivos perceber o tempo de movimento do robô ao efetuar o ciclo pretendido, ou seja, o tempo que demora a percorrer os pontos programados de acordo com a lógica do processo. Este tempo corresponde aproximadamente ao tempo de ciclo deste robô, quando no limite é desprezado o tempo de abertura e fecho do atuador final.

Este teste permite também ter a perceção da trajetória percorrida pelo robô durante a sua rotina, conduzindo à validação da possibilidade de ambos os robôs produzirem trabalho na mesma mesa, desde que em bases diferentes. Para testar esta possibilidade testou-se também a função *space function*.

Na Figura 7.4 percebe-se que, em média, o robô demora cerca de 8 a 9 segundos, por base, no seu movimento de *pick and place* e montagem das pinças. Nos testes praticados em simulação percebeu-se que o tempo em cada base é aproximadamente o mesmo. Na Figura 7.4 a base 2 apresenta um valor superior de tempo de movimento pois no código simulado contém uma instrução para o robô se mover para a posição *home* que não está contemplada no código simulado para a base 1.

Através desta simulação foi perceptível que o robô apresenta uma cadencia de montagem superior à cadência de entrada de peças na mesa.

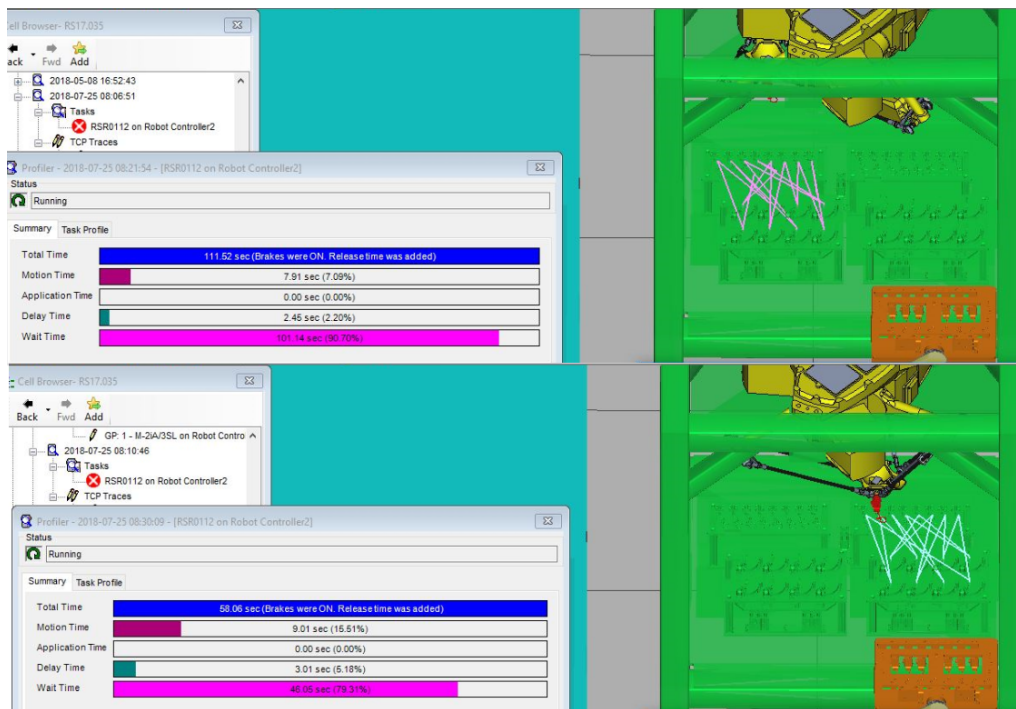


Figura 7.4: Simulação do tempo de movimento do robô no roboguide.

7.2 Testes da comunicação e da componente física

Antes de proceder aos testes do código produzido para o sistema é necessário proceder ao teste da comunicação entre os robôs, o centro de controlo e a interface com o utilizador. É também necessário proceder ao teste de todas as entradas e saídas que compõem o sistema por forma a perceber se as ligações foram bem efetuadas aquando da montagem e se cumprem o esquema elétrico e de montagem. Associado a este teste é possível verificar a conformidade do material físico produzido.

7.2.1 Teste de comunicação entre PLC e robôs

O teste efetuado para a verificação de conformidade da comunicação entre os robôs e o PLC consistiu na simulação e ativação de sinais em ambos os componentes e a verificação da receção do sinal no lado contrário.

Procedeu-se ao teste de todos os grupos de variáveis a trocar e das configurações. O resultado deste teste residiu numa comunicação bem efetuada entre o robô 1 e o PLC e cuja troca de sinais se mostrou bem efetuada após todas as configurações feitas em termos de rede. Antes de efetuar o envio de qualquer

sinal procedeu-se ao teste de *ping* por forma a garantir que a configuração de rede está bem efetuada.

No caso do robô 2 a ligação inicialmente falhou, devido a uma falha na configuração da SLOT que não pode ser igual à do robô 1. Após a correção desse erro, e da configuração dos endereços e quantidade de dados a trocar em ambos os extremos da ligação, a ligação foi bem sucedida.

Para testar a validação da rede procedeu-se à utilização do protocolo texttipping após a configuração previamente abordada. Esta validação e definição dos pacotes de dados trocados entre robô e PLC estão ilustrados na Figura 7.5.

The screenshot displays a PLC HMI interface with a status bar at the top showing 'SRVO-003 Deadman switch released' and a green '60%' indicator. Below the status bar, there are two main panels. The left panel, titled 'I/O EtherNet/IP', shows a table of connections for Rack 89. The right panel, titled 'I/O EtherNet/IP Adapter config(Read-only):', shows configuration details for the 'ROB02-PLC' connection, including IP address and API settings. At the bottom, a 'PING' button is highlighted, and a message indicates 'Ping 192.168.250.1 succeeded'.

Description	TYP	Enable	Status	Slot
Connection1	ADP	FALSE	OFFLINE	1
ROB02-PLC	ADP	TRUE	RUNNING	2
Connection3	ADP	FALSE	OFFLINE	3
Connection4	ADP	FALSE	OFFLINE	4
Connection5	ADP	FALSE	OFFLINE	5
Connection6	ADP	FALSE	OFFLINE	6
Connection7	ADP	FALSE	OFFLINE	7
Connection8	ADP	FALSE	OFFLINE	8
Connection9	ADP	FALSE	OFFLINE	9
ConnectionA	ADP	FALSE	OFFLINE	10
ConnectionB	ADP	FALSE	OFFLINE	11
ConnectionC	ADP	FALSE	OFFLINE	12
ConnectionD	ADP	FALSE	OFFLINE	13
ConnectionE	ADP	FALSE	OFFLINE	14
ConnectionF	ADP	FALSE	OFFLINE	15
ConnectionG	ADP	FALSE	OFFLINE	16
ConnectionH	ADP	FALSE	OFFLINE	17
ConnectionI	ADP	FALSE	OFFLINE	18
ConnectionJ	ADP	FALSE	OFFLINE	19
ConnectionK	ADP	FALSE	OFFLINE	20

Adapter config(Read-only):

Description : ROB02-PLC
 Input size (words) : 16
 Output size(words) : 16
 Alarm Severity : WARN
 Scanner IP : 192.168.250.1
 API 0=>T : 16
 API T=>0 : 16

Ping 192.168.250.1 succeeded

Figura 7.5: Teste de comunicação entre R2 e PLC.

7.2.2 Testes de comunicação entre PLC e HMI

O PLC envia dados à HMI o que leva à necessidade de estabelecer uma comunicação entre os mesmos. Esta comunicação é importante porque a HMI irá expor toda a informação relevante do sistema, processada pelo PLC.

Através da ligação da HMI à rede onde se encontram ligados os robôs e o PLC, foi possível testar a comunicação entre a HMI e o PLC. Esse teste foi

efetuado através do acesso da HMI a um conjunto de memórias de variáveis nas quais se encontra alocada a informação a expor na HMI.

Estes testes mostraram resultados satisfatórios: a HMI foi capaz de mostrar o estado dos sensores, dos atuadores e de algumas informações necessárias ao operador, através da leitura de variáveis mapeadas em locais de memória especificamente definidos para tal. A comunicação no sentido contrário foi testada através da leitura de informações como o tipo de peça a produzir e número de peças a produzir, cujo resultado foi o esperado, uma vez que o PLC recebia sem problemas esses dados.

7.2.3 Teste da leitura dos sinais da mesa

A troca de sinais com a mesa engloba a leitura de sinais através das entradas. Este é um dos testes mais importantes no arranque do projeto, uma vez que permite verificar se as ligações se encontram de acordo com o esquema elétrico. Através deste teste são perceptíveis erros de ligação ou até danos nos sensores.

Este teste foi efetuado após a montagem do sistema. Primeiro foram verificadas as entradas diretas do PLC que não apresentaram qualquer problema. Após o teste destas ligações foi testada a ligação entre o PLC e o módulo EtherCAT que ocorreu sem problemas. Terminados estes testes foram testadas as ligações entre os sensores e as cartas do PLC e neste teste foram detetados alguns erros de ligação. Os sensores das guias 1 e 2 encontravam-se trocados na carta, bem como os sensores dos pinos avançados e recuados, não se encontrando concordantes com o esquema elétrico cujo excerto está representado na Figura 7.6.

Ainda através deste teste, foi perceptível também a existência de sensores danificados que tiveram de ser substituídos, nomeadamente o sensor de peça 8 da base 2, o sensor de empurrador avançado da base 1 e os sensores dos pinos 5 e 7 avançados da base 1.

Já no final dos testes das entradas percebeu-se um erro de montagem nos sensores responsáveis pela deteção da montagem da base, estando estes bastante distantes do ponto de leitura e levando à necessidade de ajuste de posição. O sensor da gaveta necessitou do mesmo ajuste; pelo contrário, os sensores de caixa encontravam-se sem qualquer problema.

Após as alterações repetiu-se o processo por forma a despistar qualquer possível dano causado a algum sensor, desta vez sem erros registados e com resultados bastante satisfatórios.

do PLC. Este teste revelou alguns sinais trocados, o que foi facilmente resolvido trocando algumas linhas na montagem elétrica.



Figura 7.7: Exemplo da base na qual foram efetuados os testes.

7.2.5 Testes aos sinais de interface entre robô e atuador final

A interface entre o robô e o seu atuador final é caracterizada pela troca de sinais digitais. É enviado um sinal digital que quando se encontra a nível alto conduz à abertura da garra e a nível baixo ao seu fecho. Foi efetuado o teste à abertura e fecho da garra através do envio desse sinal, utilizando o *End Effector* do robô, e o teste finalizou-se sem quaisquer problemas e com o resultado esperado.

Após a verificação da abertura e fecho da garra, verificaram-se os sinais que informam a abertura total e fecho total desta. Este teste também ocorreu sem problema, com a leitura dos resultados satisfatória, estando de acordo com os esperados.

A garra na qual foram efetuados os testes encontra-se ilustrada na Figura 7.8.



Figura 7.8: Garra testada.

7.3 Testes ao sistema

Os testes ao sistema podem ser divididos em duas partes: os testes com recurso a amostras de peças enviadas pelo cliente e os testes em produção, já efetuados em cliente. Os primeiros testes destinaram-se a validar a resolução de anomalias encontradas em fábrica e o funcionamento básico do sistema e os segundos à validação do funcionamento do sistema em cliente, tendo por base a correção de possíveis anomalias que surgem quando é ativo o sistema global.

7.3.1 Testes com amostras de peças

Os testes com as amostras de peças foram efetuados em fábrica, sem recurso ao sistema de alimentação e descarga de peças. Estes testes eram efetuados colocando as peças nas bases, permitindo desta forma testar os ciclos de montagem do robô, os ciclos de fecho e colocação em caixas das pinças e o controlo dos mesmos via PLC. Através destes testes foi possível o ajuste dos pontos de montagem das pinças, os ajustes ao circuito pneumático e controlo do sistema da mesa.

Os testes expostos na Tabela 7.1 representam os testes efetuados em fábrica para ajuste dos pontos, verificando-se alguns erros nos pontos marcados até ao teste 16, no caso da base 1, e ao teste 14, no caso da base 2. Estes pontos, devido ao seu grau de precisão, conduziram à necessidade de remarcação e ajuste várias vezes.

Tabela 7.1: Testes de montagem por parte do robô e escoamento por parte da mesa

Teste	Base 1 peças colocadas	Base 2 peças colocadas	Base 1 peças montadas	Base 2 peças montadas	Porcentagem de peças montadas
1	8 fêmeas 8 machos	8 fêmeas 8 machos	5	4	Base 1 - 62,5% Base 2 - 50,0%
2	8 fêmeas 8 machos	8 fêmeas 8 machos	4	4	Base 1 - 50,0% Base 2 - 50,0%
3	8 fêmeas 8 machos	8 fêmeas 8 machos	5	5	Base 1 - 62,5% Base 2 - 62,5%
4	8 fêmeas 8 machos	8 fêmeas 8 machos	6	4	Base 1 - 75,0% Base 2 - 50,0%
5	8 fêmeas 8 machos	8 fêmeas 8 machos	5	5	Base 1 - 62,5% Base 2 - 62,5%
6	8 fêmeas 8 machos	8 fêmeas 8 machos	6	5	Base 1 - 75,0% Base 2 - 62,5%
7	8 fêmeas 8 machos	8 fêmeas 8 machos	7	7	Base 1 - 87,5% Base 2 - 87,5%
8	8 fêmeas 8 machos	8 fêmeas 8 machos	6	6	Base 1 - 75,0% Base 2 - 75,0%
9	8 fêmeas 8 machos	8 fêmeas 8 machos	6	6	Base 1 - 75,0% Base 2 - 75,0%
10	8 fêmeas 8 machos	8 fêmeas 8 machos	7	6	Base 1 - 87,5% Base 2 - 75,0%
11	8 fêmeas 8 machos	8 fêmeas 8 machos	8	7	Base 1 - 100% Base 2 - 87,5%
12	8 fêmeas 8 machos	8 fêmeas 8 machos	8	6	Base 1 - 100% Base 2 - 75,0%
13	8 fêmeas 8 machos	8 fêmeas 8 machos	8	6	Base 1 - 100% Base 2 - 75,0%
14	8 fêmeas 8 machos	8 fêmeas 8 machos	8	7	Base 1 - 100% Base 2 - 87,5%
15	8 fêmeas 8 machos	8 fêmeas 8 machos	7	8	Base 1 - 87,5% Base 2 - 100%
16	8 fêmeas 8 machos	8 fêmeas 8 machos	7	8	Base 1 - 87,5% Base 2 - 100%
17	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
18	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
19	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
20	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
21	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
22	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
23	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
24	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%
25	8 fêmeas 8 machos	8 fêmeas 8 machos	8	8	Base 1 - 100% Base 2 - 100%

Também foi efetuado o teste da rotina de rejeição. Através destes testes foi

perceptível a necessidade de remarcação das bases, uma vez que o ponto subia quando era efetuado o *offset* horizontal, o que provava que as bases tinham o referencial ligeiramente torto. Os resultados deste teste encontram-se expostos na Tabela 7.2.

Tabela 7.2: Teste da rejeição das peças

Teste	Base 1 peças colocadas	Base 2 peças colocadas	Base 1 peças rejeitadas	Base 2 peças rejeitadas	Percentagem de peças montadas
1	8 fêmeas 8 machos	8 fêmeas 8 machos	6 fêmeas 0 machos	7 fêmeas 0 machos	Base 1 fêmeas - 75,0% Base 1 machos - 0% Base 2 fêmeas- 87,5% Base 2 machos- 0%
2	8 fêmeas 8 machos	8 fêmeas 8 machos	6 fêmeas 0 machos	6 fêmeas 0 machos	Base 1 fêmeas - 75,0% Base 1 machos - 0% Base 2 fêmeas- 75,0% Base 2 machos- 0%
3	8 fêmeas 8 machos	8 fêmeas 8 machos	6 fêmeas 0 machos	7 fêmeas 0 machos	Base 1 fêmeas - 75,0% Base 1 machos - 0% Base 2 fêmeas- 87,5% Base 2 machos- 0%
4	8 fêmeas 8 machos	8 fêmeas 8 machos	7 fêmeas 0 machos	7 fêmeas 0 machos	Base 1 fêmeas - 87,5% Base 1 machos - 0% Base 2 fêmeas- 87,5% Base 2 machos- 0%
5	8 fêmeas 8 machos	8 fêmeas 8 machos	7 fêmeas 0 machos	7 fêmeas 0 machos	Base 1 fêmeas - 87,5% Base 1 machos - 0% Base 2 fêmeas- 87,5% Base 2 machos- 0%
6	8 fêmeas 8 machos	8 fêmeas 8 machos	6 fêmeas 0 machos	8 fêmeas 0 machos	Base 1 fêmeas - 75,0% Base 1 machos - 0% Base 2 fêmeas- 100% Base 2 machos- 0%
7	8 fêmeas 8 machos	8 fêmeas 8 machos	8 fêmeas 0 machos	8 fêmeas 0 machos	Base 1 fêmeas - 100% Base 1 machos - 0% Base 2 fêmeas- 100% Base 2 machos- 0%
8	8 fêmeas 8 machos	8 fêmeas 8 machos	8 fêmeas 0 machos	8 fêmeas 0 machos	Base 1 fêmeas - 100% Base 1 machos - 0% Base 2 fêmeas- 100% Base 2 machos- 0%
9	8 fêmeas 8 machos	8 fêmeas 8 machos	8 fêmeas 0 machos	8 fêmeas 0 machos	Base 1 fêmeas - 100% Base 1 machos - 0% Base 2 fêmeas- 100% Base 2 machos- 0%
10	8 fêmeas 8 machos	8 fêmeas 8 machos	8 fêmeas 0 machos	8 fêmeas 0 machos	Base 1 fêmeas - 100% Base 1 machos - 0% Base 2 fêmeas- 100% Base 2 machos- 0%

Com base nos resultados destes testes chegou-se à conclusão que os dedos da garra teriam de sofrer algumas alterações. Inicialmente foi necessário proceder ao seu corte lateral, por forma a permitir a montagem das peças, pois os dedos iniciais da garra levavam à quebra da peça macho. Após esta alteração chegou-se à conclusão da necessidade da criação de um rasgo central por forma a conseguir pegar nas peças macho para a rejeição das mesmas.

7.3.2 Testes em produção

Já em cliente começaram os testes do sistema em modo global. A descarga é efetuada por um robô e, após algumas descargas, foi perceptível a necessidade de alterar as dimensões das guias para os machos porque devido ao centro de massa da peça se encontrar deslocado esta ficava presa nas guias. Também através do teste da descarga foi acrescentada uma eltroválvula para permitir a subida e descida individual dos cilindros das guias, evitando a colisão com o robô de descarga.

Durante a fase de testes o cliente procedeu à alteração da matéria prima utilizada para a produção das pinças.

Os primeiros testes, cujos dados estão agrupados na Tabela 7.3, foram efetuados com a matéria prima antiga, cujas propriedades são diferentes da nova matéria prima utilizada. A alteração da matéria prima conduz à necessidade de alterar a marcação dos pontos e bases. Durante estes testes foram necessários ajustes às bases, inclusive aos pinos de fecho, por forma a evitar colisões durante a montagem das peças.

Nestes testes foi verificada uma elevada taxa de montagem e baixa taxa de falha. Porém, este material, devido à sua resistência, mostrou-se muito sensível deformando facilmente e riscando, o que causa inutilidade na pinça final devido ao controlo de qualidade do produto.

Os testes ilustrados na Tabela 7.4 apresentam resultados do sistema com a nova matéria prima, que colmata alguns problemas da anterior. A deformação e risco já são menores com esta matéria prima. Os resultados apresentam-se ainda com algumas falhas, devido à alteração das propriedades da peça para montagem de acordo com a temperatura, uma vez que quanto mais quente, mais deformável se torna o material e dificulta desta forma a marcação do ponto ideal para cada montagem.

Através destes testes e dos seus resultados conseguiu-se melhorar os resultados em termos de produção e também em termos de funcionamento do sistema. Permitiram um melhoramento em termos de pontos e tempos de ação da mesa.

Os testes em produção e as suas estatísticas são também calculados pelo PLC permitindo assim ao operador perceber de que forma se está a comportar o sistema ao longo do tempo de trabalho

Tabela 7.3: Teste em produção com matéria prima antiga

Teste	Base 1 peças colocadas	Base 2 peças colocadas	Base 1 peças montadas	Base 2 peças montadas	Erros de montagem	Erros na rejeição
1	8 fêmeas 8 machos	7 fêmeas 8 machos	7 em 9,7 s	7 em 9,7 s	Base 1 - 1 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
2	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,0 s	8 em 10,1 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
3	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 9,90 s	8 em 10,1 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
4	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,2 s	8 em 9,90 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
5	8 fêmeas 8 machos	7 fêmeas 8 machos	6 em 10,2 s	6 em 10,1 s	Base 1 - 2 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
6	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,0 s	8 em 10,1 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
7	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,1 s	8 em 10,0 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
8	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,0 s	8 em 10,2 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
9	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 9,80 s	8 em 10,2 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
10	8 fêmeas 8 machos	8 fêmeas 8 machos	7 em 10,1 s	7 em 9,90 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
11	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 9,83 s	7 em 9,91 s	Base 1 - 0 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
12	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,0 s	8 em 9,79 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
13	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 9,85 s	8 em 9,98 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
14	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,0 s	8 em 10,1 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
15	8 fêmeas 8 machos	8 fêmeas 8 machos	8 em 10,2 s	8 em 9,94 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0

Tabela 7.4: Testes em produção com a nova matéria prima

Teste	Base 1 peças colocadas	Base 2 peças colocadas	Base 1 peças montadas	Base 2 peças montadas	Erros de montagem	Erros na rejeição
1	7 fêmeas 8 machos	7 fêmeas 8 machos	7 em 10,0 s	6 em 10,1 s	Base 1 - 0 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
2	8 fêmeas 8 machos	7 fêmeas 8 machos	8 em 10,1 s	7 em 10,1 s	Base 1 - 0 Base 2 - 0	Base 1 fêmeas - 0 Base 1 machos - 1 Base 2 fêmeas - 0 Base 2 machos - 0
3	7 fêmeas 8 machos	8 fêmeas 8 machos	5 em 9,68 s	7 em 9,97 s	Base 1 - 2 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
4	7 fêmeas 8 machos	8 fêmeas 8 machos	5 em 10,2 s	7 em 10,0 s	Base 1 - 2 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
5	8 fêmeas 8 machos	7 fêmeas 8 machos	7 em 9,99 s	6 em 9,86 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 1 Base 2 fêmeas - 0 Base 2 machos - 0
6	7 fêmeas 8 machos	7 fêmeas 8 machos	6 em 10,1 s	6 em 9,83 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
7	8 fêmeas 8 machos	8 fêmeas 8 machos	7 em 9,90 s	7 em 9,87 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
8	7 fêmeas 8 machos	7 fêmeas 8 machos	8 em 10,2 s	8 em 9,98 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 0 Base 1 machos - 0 Base 2 fêmeas - 0 Base 2 machos - 0
9	7 fêmeas 8 machos	8 fêmeas 8 machos	6 em 10,2 s	7 em 10,0 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 1 Base 1 machos - 1 Base 2 fêmeas - 0 Base 2 machos - 0
10	8 fêmeas 8 machos	7 fêmeas 8 machos	7 em 10,1 s	6 em 10,2 s	Base 1 - 1 Base 2 - 1	Base 1 fêmeas - 1 Base 1 machos - 1 Base 2 fêmeas - 0 Base 2 machos - 0

Como se verifica através da Tabela 7.3 o robô paralelo demora, em média, 9,992 s a montar as peças na base 1 e 10,001 s a montar na base 2, utilizando a matéria prima antiga. Com base na Tabela 7.4 verifica-se que o robô paralelo demora, em média, 10,047 s a montar as pinças na base 1 e 9,991 s a montar na base 2. Tendo em conta um tempo de 2,0 s por parte da mesa para fecho e colocação em caixas das pinças, o tempo de ciclo aproxima-se dos 12 s estabelecidos como meta a alcançar.

7.4 Conclusão

Em suma, este capítulo permitiu testar cada componente constituinte do sistema e perceber se este se encontra de acordo com os seus requisitos em termos da necessidade do sistema. Através do teste dos elementos mecânicos foi possível um melhoramento do sistema em termos de construção e conceção. Através dos resultados do sistema foi possível perceber se a solução é a mais indicada para responder ao problema colocado.

Apesar de ser necessário alterar-se a temperatura de montagem da peça, a

superfície de contacto com a base por parte do empurrador e da otimização das bases a nível pneumático e mecânico, o sistema encontra-se bastante próximo da sua versão final e indicada para cumprir o objetivo da aplicação.

Capítulo 8

Conclusão

A presente tese teve como principal objetivo o desenvolvimento de uma solução de descarga e montagem de pinças, bem como a sua programação e controlo. Este projeto foca-se em duas vertentes distintas que coexistem no mesmo: montagem de pinças através da utilização de um robô e uma mesa automatizada e controlo de todo o sistema com base num controlador lógico programável. Estas componentes do projeto motivaram o estudo de novas tecnologias associadas a aplicações industriais e o estudo dos componentes basilares do sistema, como o controlador e o robô. O estudo efetuado incidiu também na interligação destes sistemas para posterior utilização.

Tendo estes conceitos por base, neste capítulo está descrita uma análise aos objetivos iniciais e o sucesso ou insucesso associado na tentativa de os alcançar, bem como os obstáculos encontrados e também as possíveis melhorias a serem efetuadas ao sistema.

8.1 Solução implementada

A solução implementada divide-se em duas componentes. Em relação ao sistema de montagem foi desenvolvida uma solução que engloba um robô do tipo paralelo, de 4 eixos, e um sistema de fecho e escoamento automatizado de pinças. Este sistema é composto por duas bases removíveis para troca rápida, no caso de se pretender a produção de um tipo diferente de peça sem a necessidade de um novo sistema.

No global o sistema é composto por mais um robô e um dispositivo de interface humana, cuja implementação não é desenvolvida nesta tese. O sistema de controlo utilizado foi um PLC, devido à complexidade associada ao controlo do sistema global e da sua rapidez no processo de controlo, bem como à sua flexibilidade. Este componente permite uma flexibilidade de expansão no caso de necessidade.

O sistema comunica utilizando dois protocolos: EtherCAT e Ethernet/IP. A utilização do protocolo EtherCAT permite a alocação das cartas de entradas e saídas mais próximas dos componentes a estas conectados e evita a ocupação, por parte das mesmas, do quadro principal ligadas diretamente ao PLC bem como diminui a quantidade de linhas que necessitavam de ser ligadas ao quadro principal caso se neste se encontrassem as cartas de entradas e saídas. Cada módulo EtherCAT possui elevadas velocidades de transferência e permite a ligação de até 63 cartas.

O protocolo Ethernet/IP permite a comunicação entre o sistema de controlo, sistema de interface humana e robôs, através da troca de pacotes de dados entre endereços previamente atribuídos a cada componente. Permite criar um canal de comunicação entre qualquer componente ligado numa mesma rede.

A ligação entre o quadro secundário e o principal, entre os componentes das bases e o quadro secundário, é feita utilizando fichas Harting o que permite a conexão e desconexão rápida das mesmas sem provocar dano a qualquer componente.

O robô possui 4 eixos, sendo 3 paralelos e um rotativo. Esta configuração limita o volume de trabalho e a sua capacidade de carga, porém permite velocidades mais elevadas dentro do volume de trabalho. Visto que o tamanho das bases não excede o volume de trabalho do robô, este pode executar a tarefa de montagem das pinças sem qualquer limitação a nível do manuseamento das peças. Em relação à programação e configuração do robô, o processo é de simples compreensão, porém pouco intuitivo e bastante antiquado em comparação com as interfaces de outras marcas. Toda a configuração e programação pode ser efetuada através de um simulador e posteriormente descarregada para o robô. O simulador emula o processo de programação *online* através da emulação do *teach pendant*, no qual é possível efetuar-se a marcação dos pontos necessários e o desenvolvimento da lógica associada ao processo de programação.

Em relação ao atuador final optou-se por uma garra de dois dedos paralelos de atuação simples. Inicialmente surgiram problemas associados ao desenho dos dedos da garra que, na fase de montagem, provocava o dano das peças macho. Após as alterações dos dedos, criando um relevo lateral, foi possível o encaixe das peças, ficando a peça macho localizada nesse relevo. Porém esta alteração não foi suficiente, uma vez que a garra não conseguia pegar nas peças macho para a rejeição. Para solucionar esse problema criou-se um relevo interior com a forma do pino central das peças macho. A alternativa de atuador final em aplicações de *pick and place*, na qual é baseada a manipulação das peças, é o de vácuo o que não é o ideal devido à reduzida área de contacto e à necessidade de efetuar forças consideráveis na fase de montagem.

A mesa de montagem, apesar de efetuar o objetivo para o qual foi cons-

truída, possui ainda alguns problemas mecânicos associados à produção das peças, o que conduz a problemas no movimento de peças como as guias, pinos de fecho das pinças e empurrador. Também os pequenos desvios associados ao erro de medida dos perfis e da montagem dos pinos das guias conduzem à impossibilidade de recurso a funções de *offset* na zona de montagem uma vez que é processo que exige precisão e há desvios superiores a 1 mm. Deste modo, o sistema obriga à marcação de 9 pontos por base, com bastante precisão, o que inviabiliza um pouco a troca rápida de bases. A exatidão nas medidas conduz a que ao mínimo erro de montagem ou cedência do material possa conduzir ao mau funcionamento da mesa de montagem.

Em relação à segurança do sistema esta recorre a relés de segurança que, no caso de emergência ou abertura de portas, executam uma paragem quase instantânea do sistema. A utilização de uma PSENGate permite o acesso à célula de forma controlada, sem existir a necessidade de parar forçadamente o sistema e a necessidade de o reiniciar. As seguranças da célula e dos robôs encontram-se interligadas por forma a reduzir os riscos ao operador no acesso à célula. A gestão da segurança poderia ter sido feita através de um PLC de segurança que nesta aplicação, devido à simplicidade de segurança, não justificava o investimento.

Em suma, apesar de alguns inconvenientes ultrapassados e algumas imperfeições, o sistema cumpre os objetivos principais de gestão da célula e montagem das pinças. O sistema encontra-se próximo das exigências, porém necessita de algumas afinações.

8.2 Melhorias futuras

Após a implementação, teste do sistema e de uma análise dos resultados obtidos, é possível encontrar algumas alterações necessárias à melhoria do funcionamento do sistema.

1. Melhoria construtiva das bases, garantindo distâncias entre pinos e o nivelamento da placa na qual são montadas as peças.
2. Melhoria na marcação das bases e pontos de montagem por forma a melhorar a precisão e evitar a possibilidade de qualquer dano na peça final.
3. Garantir o arrefecimento das peças por forma a evitar a alteração das suas propriedades que alteram o ponto de montagem.
4. Criação de um método, ou sistema, para auxiliar a marcação dos pontos de montagem devido à sua elevada precisão.
5. Expansão do sistema para outros tipos de peça.
6. Melhoria nos movimentos do robô.

7. Melhoria no tempo de ciclo da mesa e garantir a localização das peças enquanto descem as guias, por forma a eliminar a imobilização do empurrador.

8.3 Considerações finais

A possibilidade de efetuar o presente projeto com recurso ao material fornecido e em contexto industrial mostrou-se bastante enriquecedor. Contribuiu para o aumento de conhecimento em termos de aplicação destes componentes, bem como da sua programação. O ambiente industrial permitiu ganhar valências em termos de operar sobre pressão. Em suma, este projeto permitiu não só a aquisição de conhecimentos técnicos importantes para execução de um projeto industrial, bem como conhecimentos nas áreas de gestão de tempo e de raciocínio rápido para a resolução de problemas.

Referências Bibliográficas

- [1] FANUC, “FANUC Robot M2iA,” 2015. [citado na p. vii, xi, 51]
- [2] ABB, “IRB 140,” 2017. [citado na p. xi, 49, 50]
- [3] ABB, “IRB 360 FlexPicker,” 2017. [citado na p. xi, 49, 50]
- [4] ABB, “IRB 910SC SCARA.” [citado na p. xi, 49, 50]
- [5] KUKA, “KUKA Sensitive robotics...LBR iiwa,” 2017. [citado na p. xi, 50, 51]
- [6] KUKA, “The compact robot for low payloads KR5 sixx,” 2017. [citado na p. xi, 50, 51]
- [7] KUKA, “KR6 R900 Sixx,” 2018. [citado na p. xi, 50, 51]
- [8] FANUC, “FANUC Robot CR35iA,” 2015. [citado na p. xi, 51, 52]
- [9] FANUC, “FANUC Robot SR3iA/SR6iA,” 2015. [citado na p. xi, 51]
- [10] Capek, Karel, *Robots Universales Rossum*. Madrid Alianza Editor, 1966. [citado na p. 7, 9, 10]
- [11] Asimov, Isaac, *I, Robot*. Macmillan Readers, 1950. [citado na p. 8]
- [12] Merriam-Webster, “Definition of robot .” [citado na p. 8]
- [13] Oxford Dictionaries, “Robot Definition of robot in English by Oxford Dictionaries.” [citado na p. 8]
- [14] Priberam Dicionários, “Significado/definição de robô no Dicionário Priberam da Língua Portuguesa.” [citado na p. 8]
- [15] Robotic Industries Association, “Robot terms and definitions.” [citado na p. 8]
- [16] Robotic Industries Association, “RIA - Robotics Online - Defining The Industrial Robot Industry and All It Entails.” [citado na p. 8]

- [17] Groover, Mikell P., *Automation, Production System, and Computer Integrated Manufacturing*. Prentice Hall, 2007. [citado na p. 8, 17, 18, 23, 24, 25, 27, 30, 31, 32, 33, 34, 35, 41, 42]
- [18] Silva, Manuel, "Introdução à robótica," 2012. [citado na p. 9, 10]
- [19] Robotshop INC., "History of Robotics: Timeline," 2008. [citado na p. 9, 10]
- [20] Guinan, Paul, *The Steam man*. The Victorian Robots, 2000. [citado na p. 9]
- [21] Guinan, Paul, *Steam man mark II*. The Victorian Robots, 2000. [citado na p. 9]
- [22] Jazar, Reza N., *Theory of Applied Robotics: Kinematics, Dynamics, and Control, Second Edition*. Springer, 2010. [citado na p. 11, 14, 15, 18, 20]
- [23] J.D. La Fuente, J. Santiago, A. Roman, C. Dumitrache, and D. Casasanto, *Handbook of Robotics*. Springer, 2014. [citado na p. 11, 12, 13, 14, 16, 17, 18, 22]
- [24] NPTEL, "Lecture 5: Industrial Robotics." [citado na p. 11, 12]
- [25] Silva, Manuel, "Noções Básicas de Robótica," 2016. [citado na p. 11, 14, 15, 22]
- [26] S.Y. Nof, *Handbook of Industrial Robotics, Second*. John Wiley and Sons, 1999. [citado na p. 12, 13, 14, 16, 17, 18, 19, 20, 24, 25, 26, 27, 31, 33, 34, 48, 49]
- [27] FANUC, "FANUC M-20iA industrial robot." [citado na p. 15]
- [28] American Robot, "Absolute Accuracy for Advanced Robotics." [citado na p. 15]
- [29] KUKA, "KUKA KR C4: The Power Of Control." [citado na p. 21]
- [30] FANUC, "Acessórios para robôs." [citado na p. 21]
- [31] ABB, "Roboterprogrammierung für jedermann." [citado na p. 26]
- [32] Silva, Manuel, "Métodos de Programação de Robôs," 2017. [citado na p. 26, 27]
- [33] FANUC, "FANUC ROBOGUIDE Simulation Software." [citado na p. 28]
- [34] Jong-Oh Park, "South Korean private-public partnership to invest 2.6B dollar in robot industry by 2018." [citado na p. 29]
- [35] Silva, Manuel, "Aplicações Industriais da Robótica," 2012. [citado na p. 31, 32, 33, 34, 35]
- [36] JHRobotics, INC. Automation and Manufacturing, "Picking and Packaging." [citado na p. 32]
- [37] Petruzella, Frank, *Programmable Logic Controllers, Fifth Edition*. Mc Graw Hill Education, 2017. [citado na p. 36, 37, 39, 40, 43, 44, 45]

- [38] W. Bolton, *Programmable Logic Controllers, Fifth Edition*. Springer, 2009. [citado na p. 36, 37, 39, 40, 43]
- [39] RobotWorx, "Assembly Robots." [citado na p. 49]
- [40] RobotWorx, "KUKA Pick and Place Robots." [citado na p. 50]
- [41] ABB, "PickMaster Industrial software product," 2010. [citado na p. 52, 53]
- [42] KUKA, "KUKA Technology KUKASim," 2017. [citado na p. 53]
- [43] KUKA, "KUKA.FlexPal." [citado na p. 54]
- [44] FANUC, "Roboguide," 2008. [citado na p. 54]
- [45] FANUC, "FANUC Roboguide HandlingPRO," 2008. [citado na p. 54]
- [46] FANUC, "FANUC iRPickTool," 2008. [citado na p. 54]
- [47] Silva, Manuel, "Atuadores Finais para Robôs Industriais," 2016. [citado na p. 55, 57]
- [48] Schunk, "Pick and Place." [citado na p. 55]
- [49] Schunk, "Parallel gripper." [citado na p. 55]
- [50] Schunk, "Centric gripper." [citado na p. 56]
- [51] Schunk, "4-finger gripper." [citado na p. 56]
- [52] ISQ, *Passaporte de segurança*. ISQ, 2014. [citado na p. 57]
- [53] Silva, Manuel, "Segurança na Programação e Operação de Robôs Industriais," 2012. [citado na p. 57, 58, 59, 60, 61]
- [54] ISHN, "The industrial robot safety standard." [citado na p. 58]
- [55] RIA, "Industrial Robots and Robot Systems Safety Requirements," 2012. [citado na p. 58, 59]
- [56] ABB, "Picking pizza picker. Case study: Vortex Systems and Panidea," 2005. [citado na p. 62, 63]
- [57] OMRON adept, "A clean solution for a chocolate factory." [citado na p. 64, 65]
- [58] Mitsubishi Electric, "Programming and Installation Time with Easy-To-Use Robots," 2013. [citado na p. 65, 66]
- [59] ABB, "Packing and palletizing of coff," 2005. [citado na p. 67, 68]

- [60] KUKA Robotics Corporation, "KUKA Long Range Robot Outperforms Layering Palletizer." [citado na p. 68, 69]
- [61] Universal Robotics, "Food Industry palletizing cobot with vision system works in tight quarters." [citado na p. 69, 70]
- [62] FANUC, "Datasheet FANUC M20iA/20M," 2017. [citado na p. 74]
- [63] FANUC, "FANUC Robot M2iA," 2017. [citado na p. 75, 76, 77, 78]
- [64] FANUC, "Datasheet FANUC Roboshot S2000i," 2008. [citado na p. 78]
- [65] FESTO, "Solenoid valves VUVG/valve terminals VTUG," 2018. [citado na p. 79]
- [66] FESTO, "Proximity sensors SIES-8M, inductive for T-slot," 2018. [citado na p. 80]
- [67] Contrinex, "DW-AS-623-M8-001." [citado na p. 80, 81]
- [68] Contrinex, "LWR-C23PA-PMS-403." [citado na p. 81]
- [69] FESTO, "Pressure sensors SPAN," 2018. [citado na p. 81]
- [70] Contrinex, "LTK-1050-303-505." [citado na p. 82]

Anexo A

Variáveis de comunicação entre Robô e o PLC

Nesta secção de anexos estão expostos os padrões de mapeamento utilizados para comunicação entre o PLC e o robô paralelo.

O mapeamento das entradas e saídas do robô segue um padrão criado na RobotSol. Este padrão divide as entradas e saídas dos robô em grupos com diferentes finalidades.

No caso do robô 2 sofreu uma pequena alteração no caso das entradas reservadas para o atuador final, passando a ser utilizadas para leitura dos sinais provenientes dos sensores de peça colocados nas bases.

No seguimento deste anexo são expostas apenas as variáveis utilizadas e não a totalidade de variáveis mapeadas no robô e no PLC, uma vez que o conjunto de variáveis é bastante extenso.

A.1 Mapeamento padrão das entradas e saídas da Robot-Sol

Tabela A.1: Divisão das entradas e saídas do Robô 2 por grupos

Grupo de variáveis	Entradas/Saídas do robô
System	Entradas: UI[1] até UI[18] e DI[19] até DI[24]
	Saídas: UO[1] até UO[19] e DO[20] até DO[24]
General Signals	Entradas: DI[33] até DI[48]
	Saídas: DO[33] até DO[48]
Code Program	Entradas: GI[1] (DI[49] até DI[64])
	Saídas: GO[1] (DO[49] até DO[64])
Work Area	Entradas: Não endereçado DI[65] até DI[80] Area Ready DI[81] até DI[96]
	Saídas: Area Request DO[65] até DO[80] Area Release DO[81] até DO[96]
Colision	Entradas: DI[97] até DI[112]
	Saídas: DO[97] até DO[112]
Part Request/Ready	Entradas: GI[2] (DI[113] até DI[128])
	Saídas: GO[2] (DO[113] até DO[128])
Code Job Done	Entradas: GI[3] (DI[129] até DI[144])
	Saídas: GO[3] (DO[129] até DO[144])
Specific Signals	Entradas: DI[145] até DI[160]
	Saídas: DO[145] até DO[160]
Error Code	Entradas: DI[161] até DI[176]
	Saídas: DO[161] até DO[176]
CMD Gripper	Entradas: DI[177] até DI[208]
	Saídas: DO[177] até DO[192]

A.2 Sentido R2 -> PLC

Tabela A.2: Variáveis utilizadas para as saídas do Robô 2 e entradas do PLC

Variável	Type	Saída do robô	Entrada do PLC	Comentário
W_R2_CMDENBL	BOOL	UO1	%W100.00	Robot Remote Mode
W_R2_SYSTDY	BOOL	UO2	%W100.01	System Ready
W_R2_PROGRUN	BOOL	UO3	%W100.02	Program Running
W_R2_PAUSED	BOOL	UO4	%W100.03	Program Paused
W_R2_Hold_Fanuc	BOOL	UO5	%W100.04	Program Hold
W_R2_FAULT	BOOL	UO6	%W100.05	System Fail
W_R2_Home_check	BOOL	UO7	%W100.06	Reached Position
W_R2_TPENBL	BOOL	UO8	%W100.07	Teach Pendant Enabled
W_R2_BATALM	BOOL	UO9	%W100.08	Battery CMOS RAM failed
W_R2_BUSY	BOOL	U10	%W100.09	Robot Program Execution
W_R2_ACK1_SNO1	BOOL	U11	%W100.10	RSR1/PNS1
W_R2_Auto	BOOL	DOUT21	%W101.04	Robot in Auto Mode
W_R2_T1	BOOL	DOUT22	%W101.05	Robot in T1 Mode
W_R2_T2	BOOL	DOUT23	%W101.06	Robot in T2 Mode
W_R2_ESTOP	BOOL	DOUT24	%W101.07	ESTOP
W_R2_Prog_Request	BOOL	DOUT33	%W102.00	Robot Request number program
W_R2_Door_Access	BOOL	DOUT34	%W102.01	Door Access
W_R2_Maintenance_Pos	BOOL	DOUT35	%W102.02	In Maintenance Position
W_R2_Reserve_SS1_In	BOOL	DOUT145	%W109.00	Base1 SetA Job Done
W_R2_Reserve_SS2_In	BOOL	DOUT146	%W109.01	Base1 SetB Job Done
W_R2_Reserve_SS3_In	BOOL	DOUT2147	%W109.02	Base2 SetA Job Done
W_R2_Reserve_SS4_In	BOOL	DOUT148	%W109.03	Base2 SetB Job Done
W_R2_Reserve_SS5_In	BOOL	DOUT149	%W109.04	Base19cm Job Done
W_R2_Reserve_SS6_In	BOOL	DOUT150	%W109.05	Base24cm job Done
W_R2_Reserve_SS7_In	BOOL	DOUT151	%W109.06	Working Base1
W_R2_Reserve_SS8_In	BOOL	DOUT152	%W109.07	Working Base2

A.3 Sentido PLC -> R2

Tabela A.3: Variáveis utilizadas para as entradas do Robô 2 e saídas do PLC

Variável	Type	Entrada do robô	Saída do PLC	Comentário
W_R2_IMSTP	BOOL	UI1	%W120.00	Immediate Software Stop
W_R2_HOLD	BOOL	UI2	%W120.01	External Controlled Stop
W_R2_SFSPS	BOOL	UI3	%W120.02	Reduced Speed Habilitation
W_R2_CSTOPI	BOOL	UI4	%W120.03	Cycle Stop
W_R2_FAUL_RESET	BOOL	UI5	%W120.04	Fault Reset
W_R2_START	BOOL	UI6	%W120.05	Program Start (running) Falling Edge
W_R2_HOME	BOOL	UI7	%W120.06	Robot in home point
W_R2_ENBL	BOOL	UI8	%W120.07	Movement Enabled
W_R2_RSR1_PNS1	BOOL	UI9	%W120.08	RSR1/PNS1
W_R2_PROD_START	BOOL	UI18	%W121.01	Cycle Start
W_R2_do_Prog_Num_Valid	BOOL	DIN33	%W122.00	Program Number Valid (Prog=AckProg)
W_R2_do_Door_Access	BOOL	DIN34	%W122.01	Request Door Access
W_R2_do_End_Maintenance	BOOL	DIN35	%W122.02	End Maintenance Position
W_R2_Reserve_SS1_Out	BOOL	DIN145	%W129.00	Base1
W_R2_Reserve_SS2_Out	BOOL	DIN146	%W129.01	Base2
W_R2_Reserve_SS3_Out	BOOL	DIN147	%W129.02	SetA B1
W_R2_Reserve_SS4_Out	BOOL	DIN148	%W129.03	SetB B1
W_R2_Reserve_SS5_Out	BOOL	DIN149	%W129.04	Do Production
W_R2_Reserve_SS6_Out	BOOL	DIN150	%W129.05	Access Request
W_R2_Reserve_SS7_Out	BOOL	DIN151	%W129.06	Working Base1
W_R2_Reserve_SS8_Out	BOOL	DIN152	%W129.07	Working Base2
W_R2_Reserve_SS9_Out	BOOL	DIN153	%W129.08	Aux Paragem Controlada
W_R2_Reserve_SS10_Out	BOOL	DIN154	%W129.09	SetA B2
W_R2_Reserve_SS11_Out	BOOL	DIN155	%W129.10	SetB B2

Tabela A.4: Variáveis utilizadas para as entradas do Robô 2 e saídas do PLC - sensores

Variável	Type	Entrada do robô	Saída do PLC	Comentário
W_R2_Sensor1_Base1_Out	BOOL	DIN177	%W131.00	Base 1 - Female 1
W_R2_Sensor2_Base1_Out	BOOL	DIN178	%W131.01	Base 1 - Female 2
W_R2_Sensor3_Base1_Out	BOOL	DIN179	%W131.02	Base 1 - Femea 3
W_R2_Sensor4_Base1_Out	BOOL	DIN180	%W131.03	Base 1 - Female 4
W_R2_Sensor5_Base1_Out	BOOL	DIN181	%W131.04	Base 1 - Female 5
W_R2_Sensor6_Base1_Out	BOOL	DIN182	%W131.05	Base 1 - Female 6
W_R2_Sensor7_Base1_Out	BOOL	DIN183	%W131.06	Base 1 - Female 7
W_R2_Sensor8_Base1_Out	BOOL	DIN184	%W131.07	Base 1 - Female 8
W_R2_Sensor9_Base1_Out	BOOL	DIN185	%W131.08	Base 1 - Male 1
W_R2_Sensor10_Base1_Out	BOOL	DIN186	%W131.09	Base 1 - Male 6
W_R2_Sensor11_Base1_Out	BOOL	DIN187	%W131.10	Base 1 - Male 3
W_R2_Sensor12_Base1_Out	BOOL	DIN188	%W131.11	Base 1 - Male 8
W_R2_Sensor13_Base1_Out	BOOL	DIN189	%W131.12	Base 1 - Male 5
W_R2_Sensor14_Base1_Out	BOOL	DIN190	%W131.13	Base 1 - Male 2
W_R2_Sensor15_Base1_Out	BOOL	DIN191	%W131.14	Base 1 - Male 7
W_R2_Sensor16_Base1_Out	BOOL	DIN192	%W131.15	Base 1 - Male 4
W_R2_Sensor1_Base2_Out	BOOL	DIN193	%W132.00	Base 2 - Female 8
W_R2_Sensor2_Base2_Out	BOOL	DIN194	%W132.01	Base 2 - Female 7
W_R2_Sensor3_Base2_Out	BOOL	DIN195	%W132.02	Base 2 - Female 6
W_R2_Sensor4_Base2_Out	BOOL	DIN196	%W132.03	Base 2 - Female 5
W_R2_Sensor5_Base2_Out	BOOL	DIN197	%W132.04	Base 2 - Female 4
W_R2_Sensor6_Base2_Out	BOOL	DIN198	%W132.05	Base 2 - Female 3
W_R2_Sensor7_Base2_Out	BOOL	DIN199	%W132.06	Base 2 - Female 2
W_R2_Sensor8_Base2_Out	BOOL	DIN200	%W132.07	Base 2 - Female 1
W_R2_Sensor9_Base2_Out	BOOL	DIN201	%W132.08	Base 2 - Male 8
W_R2_Sensor10_Base2_Out	BOOL	DIN202	%W132.09	Base 2 - Male 3
W_R2_Sensor11_Base2_Out	BOOL	DIN203	%W132.10	Base 2 - Male 6
W_R2_Sensor12_Base2_Out	BOOL	DIN204	%W132.11	Base 2 - Male 1
W_R2_Sensor13_Base2_Out	BOOL	DIN205	%W132.12	Base 2 - Male 4
W_R2_Sensor14_Base2_Out	BOOL	DIN206	%W132.13	Base 2 - Male 7
W_R2_Sensor15_Base2_Out	BOOL	DIN207	%W132.14	Base 2 - Male 2
W_R2_Sensor16_Base2_Out	BOOL	DIN208	%W132.15	Base 2 - Male 5

Anexo B

Variáveis de comunicação entre R1 e PLC

B.1 Mapeamento padrão das entradas e saídas da Robot-Sol

Tabela B.1: Divisão das entradas e saídas do Robô 1 por grupos

Grupo de variáveis	Entradas/Saídas do robô
System	Entradas: UI[1] até UI[18] e DI[19] até DI[24]
	Saídas: UO[1] até UO[19] e DO[20] até DO[24]
General Signals	Entradas: DI[33] até DI[48]
	Saídas: DO[33] até DO[48]
Code Program	Entradas: GI[1] (DI[49] até DI[64])
	Saídas: GO[1] (DO[49] até DO[64])
Work Area	Entradas: Não endereçado DI[65] até DI[80] Area Ready DI[81] até DI[96]
	Saídas: Area Request DO[65] até DO[80] Area Release DO[81] até DO[96]
Colision	Entradas: DI[97] até DI[112]
	Saídas: DO[97] até DO[112]
Part Request/Ready	Entradas: GI[2] (DI[113] até DI[128])
	Saídas: GO[2] (DO[113] até DO[128])
Code Job Done	Entradas: GI[3] (DI[129] até DI[144])
	Saídas: GO[3] (DO[129] até DO[144])
Specific Signals	Entradas: DI[145] até DI[160]
	Saídas: DO[145] até DO[160]
Error Code	Entradas: DI[161] até DI[176]
	Saídas: DO[161] até DO[176]
CMD Gripper	Entradas: DI[177] até DI[192]
	Saídas: DO[177] até DO[192]

B.2 Sentido R1 -> PLC

Tabela B.2: Variáveis utilizadas para as saídas do Robô 1 e entradas do PLC

Variável	Type	Saída do robô	Entrada do PLC	Comentário
W_R1_CMDENBL	BOOL	UO1	%W140.00	Robot Remote Mode
W_R1_SYSTDY	BOOL	UO2	%W140.01	System Ready
W_R1_PROGRUN	BOOL	UO3	%W140.02	Program Running
W_R1_PAUSED	BOOL	UO4	%W140.03	Program Paused
W_R1_Hold_Fanuc	BOOL	UO5	%W140.04	Program Hold
W_R1_FAULT	BOOL	UO6	%W140.05	System Fail
W_R1_Home_check	BOOL	UO7	%W140.06	Reached Position
W_R1_TPENBL	BOOL	UO8	%W140.07	Teach Pendant Enabled
W_R1_BATALM	BOOL	UO9	%W140.08	Battery CMOS RAM failed
W_R1_BUSY	BOOL	U10	%W140.09	Robot Program Execution
W_R1_ACK1_SNO1	BOOL	U11	%W140.10	RSR1/PNS1
W_R1_Auto	BOOL	DOUT21	%W141.04	Robot in Auto Mode
W_R1_T1	BOOL	DOUT22	%W141.05	Robot in T1 Mode
W_R1_T2	BOOL	DOUT23	%W141.06	Robot in T2 Mode
W_R1_ESTOP	BOOL	DOUT24	%W141.07	ESTOP
W_R1_Prog_Request	BOOL	DOUT33	%W142.00	Robot Request number program
W_R1_Door_Access	BOOL	DOUT34	%W142.01	Door Access
W_R1_Maintenance_Pos	BOOL	DOUT35	%W142.02	In Maintenance Position
W_R1_di_AreaReq_1	BOOL	DOUT65	%W144.00	Base 1 request
W_R1_di_AreaReq_2	BOOL	DOUT66	%W144.01	Base 2 request
W_R1_di_AreaRel_1	BOOL	DOUT81	%W145.00	Base 1 release
W_R1_di_AreaRel_2	BOOL	DOUT82	%W145.01	Base 2 release
W_R1_Reserve_SS1_In	BOOL	DOUT145	%W149.00	HMI_Mould_Close
W_R1_Reserve_SS2_In	BOOL	DOUT146	%W149.01	HMI_Mould_Open
W_R1_Reserve_SS3_In	BOOL	DOUT2147	%W149.02	HMI_Ejec_Back
W_R1_Reserve_SS4_In	BOOL	DOUT148	%W149.03	HMI_Ejec_Forward
W_R1_Reserve_SS5_In	BOOL	DOUT149	%W149.04	HMI_OP_Mode
W_R1_Reserve_SS6_In	BOOL	DOUT150	%W149.05	Sinal para subir guia do conjunto2
W_R1_Reserve_SS7_In	BOOL	DOUT151	%W149.06	R1_FORA_BASE1 [Working Base 1 - DO[] Space Fnct (Output to R2)]
W_R1_Reserve_SS8_In	BOOL	DOUT152	%W149.07	R1_FORA_BASE2 [Working Base 2 - DO[] Space Fnct (Output to R2)]
W_R1_Reserve_SS11_In	BOOL	DOUT155	%W149.10	Check_Stop

B.3 Sentido PLC -> R1

Tabela B.3: Variáveis utilizadas para as entradas do Robô 1 e saídas do PLC

Variável	Type	Entrada do robô	Saída do PLC	Comentário
W_R2_IMSTP	BOOL	UI1	%W160.00	Immediate Software Stop
W_R2_HOLD	BOOL	UI2	%W160.01	External Controlled Stop
W_R2_SFSPS	BOOL	UI3	%W160.02	Reduced Speed Habilitation
W_R2_CSTOPI	BOOL	UI4	%W160.03	Cycle Stop
W_R2_FAUL_RESET	BOOL	UI5	%W160.04	Fault Reset
W_R2_START	BOOL	UI6	%W160.05	Program Start (running) Falling Edge
W_R2_HOME	BOOL	UI7	%W160.06	Robot in home point
W_R2_ENBL	BOOL	UI8	%W160.07	Movement Enabled
W_R2_RSR1_PNS1	BOOL	UI9	%W160.08	RSR1/PNS1
W_R2_PROD_START	BOOL	UI18	%W161.01	Cycle Start
W_R2_do_Prog_Num_Valid	BOOL	DIN33	%W162.00	Program Number Valid (Prog=AckProg)
W_R2_do_Door_Access	BOOL	DIN34	%W162.01	Request Door Access
W_R2_do_End_Maintenance	BOOL	DIN35	%W162.02	End Maintenance Position
W_R1_do_AreaReady_1	BOOL	DIN81	%W165.00	Base 1 Ready
W_R1_do_AreaReady_2	BOOL	DIN82	%W165.01	Base 2 Ready
W_R2_Reserve_SS1_Out	BOOL	DIN145	%W169.00	Base1 Producao
W_R2_Reserve_SS2_Out	BOOL	DIN146	%W169.01	Base2 Producao
W_R2_Reserve_SS3_Out	BOOL	DIN147	%W169.02	Access Request
W_R2_Reserve_SS7_Out	BOOL	DIN151	%W169.06	R2_FORA_BASE1 [Working Base1 - DI[] Space Fcnt (Input from R2)]
W_R2_Reserve_SS8_Out	BOOL	DIN152	%W169.07	R2_FORA_BASE2 [Working Base2 - DI[] Space Fcnt (Input from R2)]

Anexo C

Programação Robô2

C.1 Fluxograma do programa principal

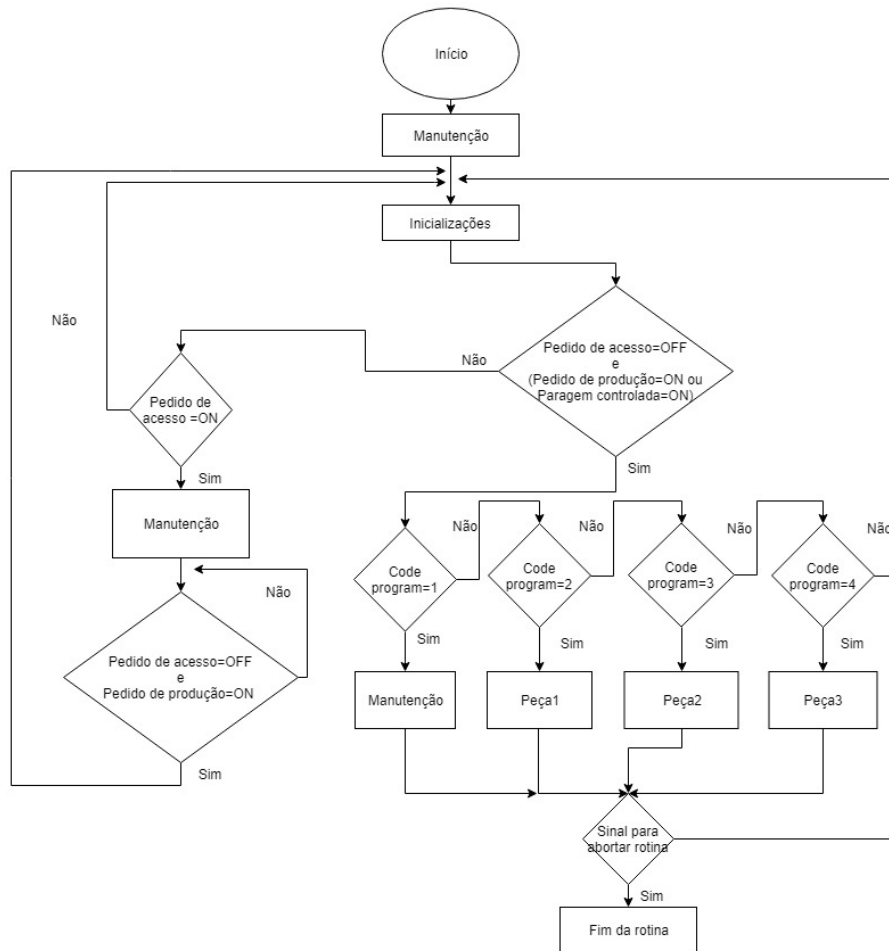


Figura C.1: Fluxograma do programa principal do robô.

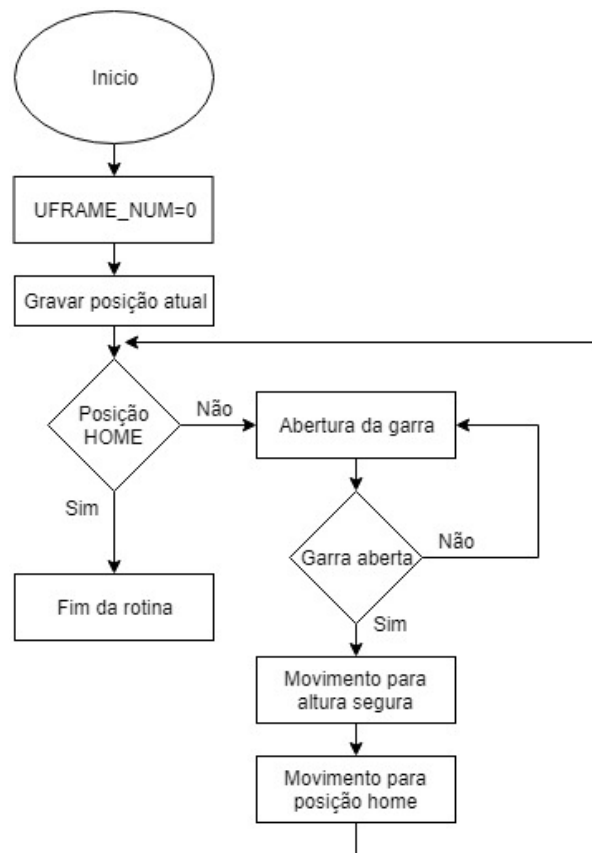
C.2 Fluxograma do programa de manutenção

Figura C.2: Fluxograma da rotina de manutenção do robô.

C.3 Fluxograma do programa da peça 13cm

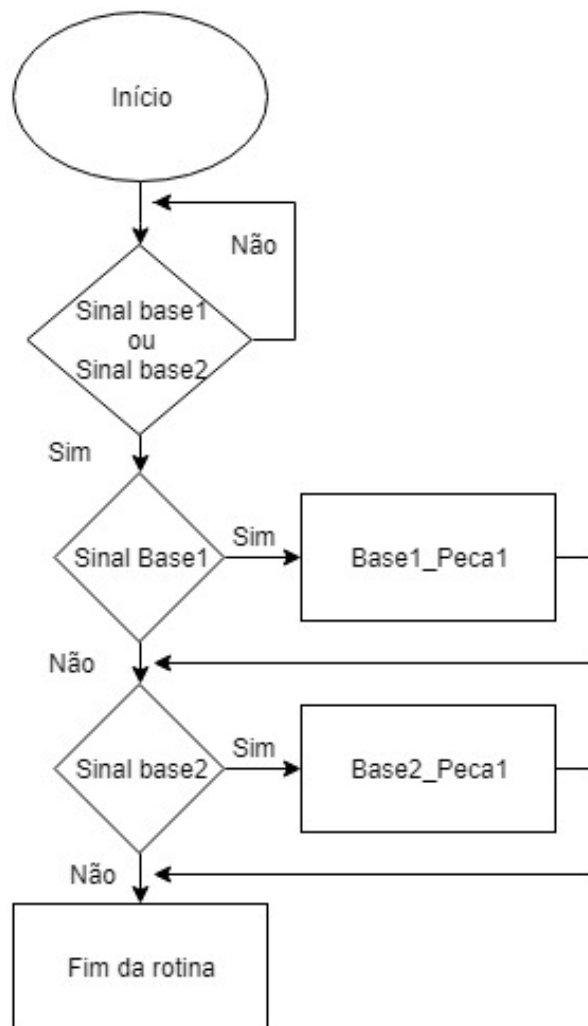


Figura C.3: Fluxograma da rotina da peça 1 do robô.

C.4 Fluxograma base da lógica seguida pelo robô

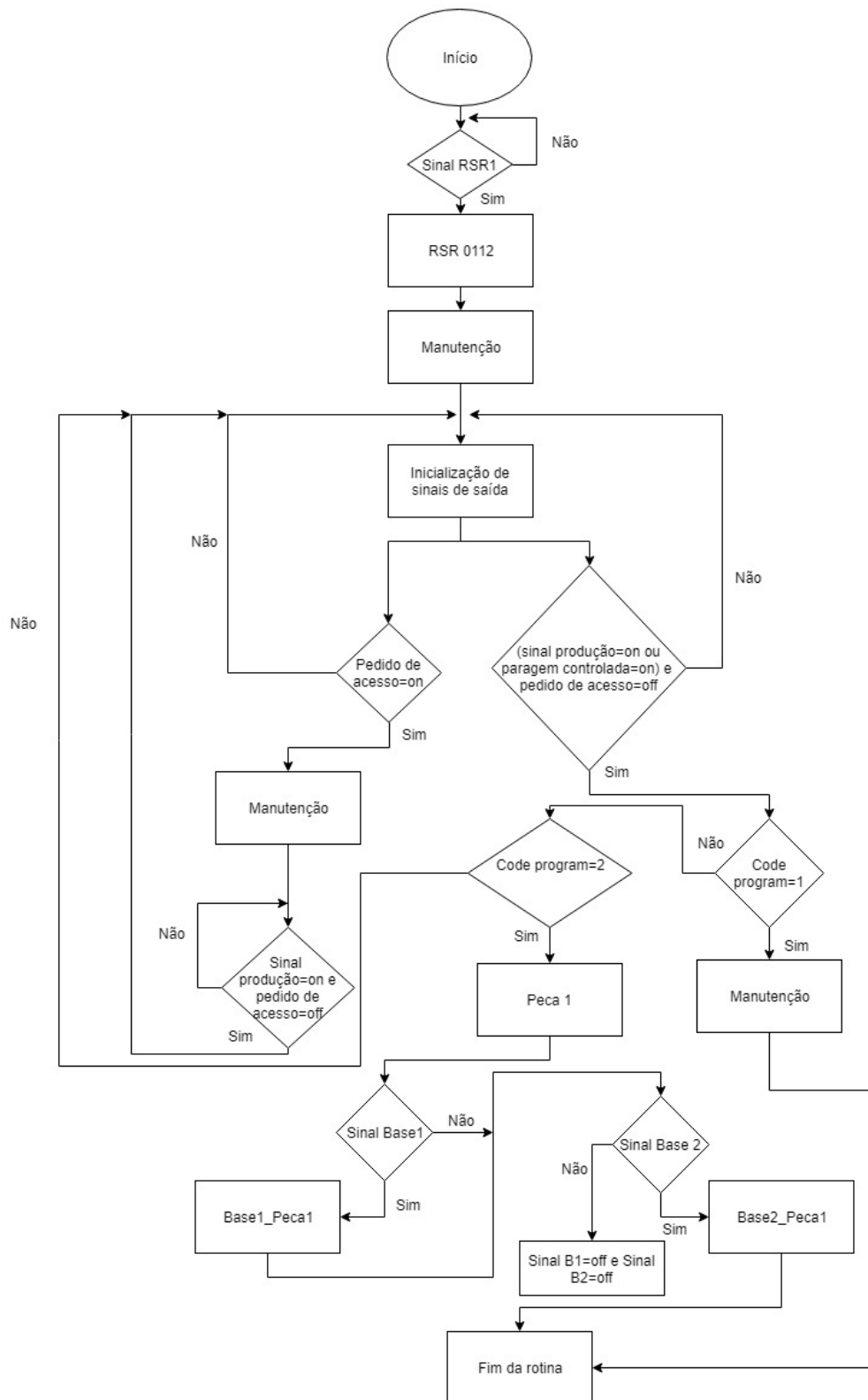


Figura C.4: Fluxograma base da lógica seguida pelo robô.

C.5 Fluxograma base da lógica seguida pelo robô na base 1

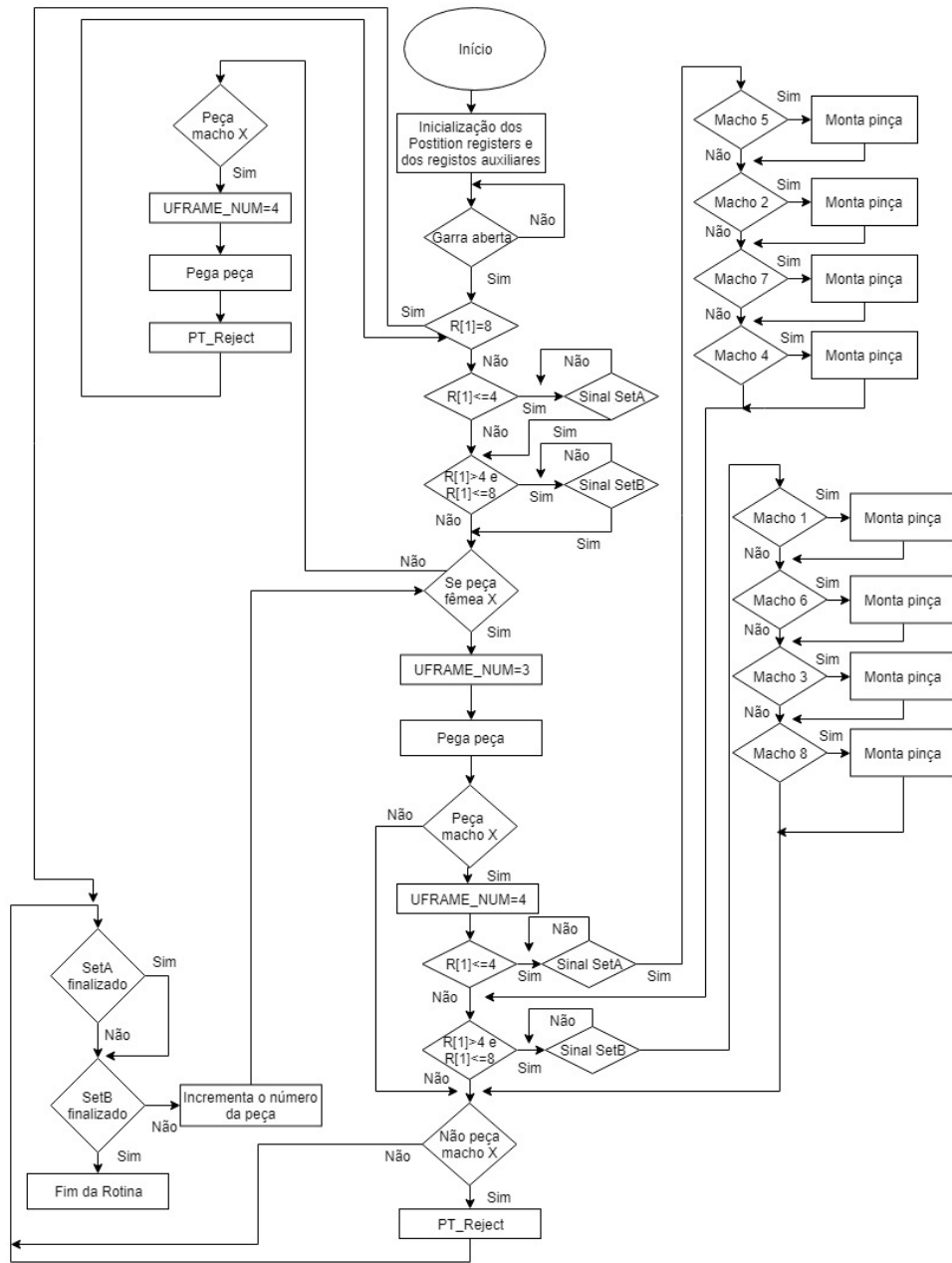


Figura C.5: Fluxograma base da lógica seguida pelo robô ao montar peças na base 1.

Anexo D

Configuração do PLC

D.1 Configuração da rede EtherCat

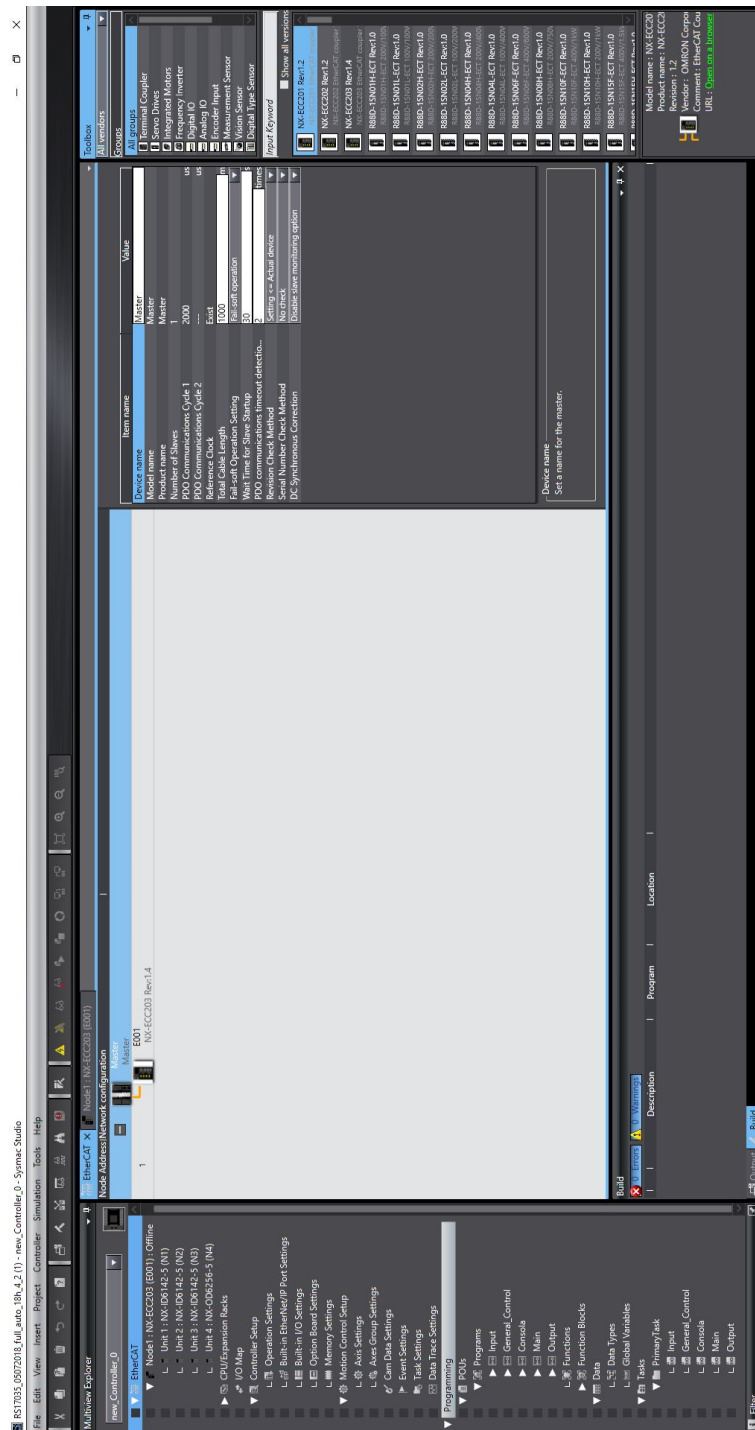


Figura D.1: Menu de configuração dos módulos EtherCAT.

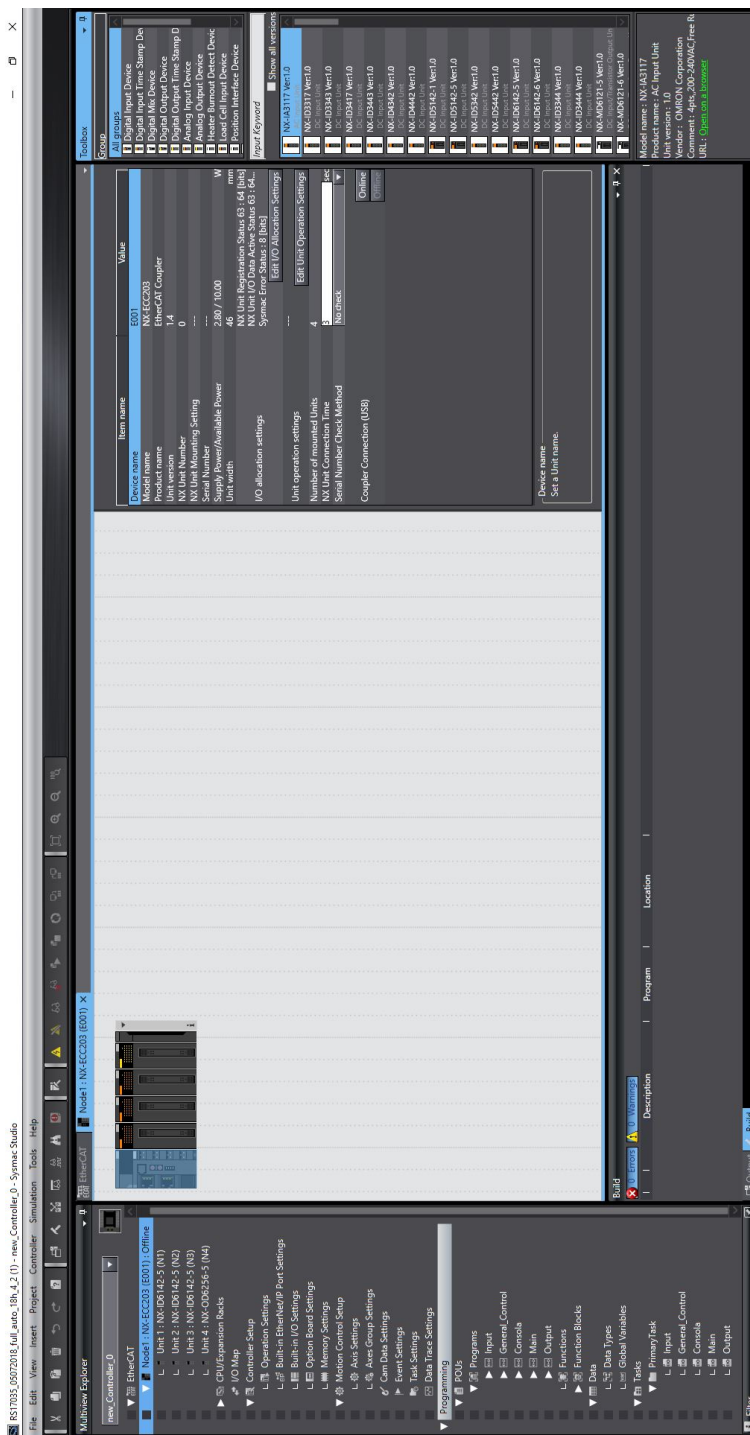


Figura D.2: Nó da rede EtherCAT para ligação das cartas de entradas e saídas.

D.2 Mapeamento das entradas e saídas

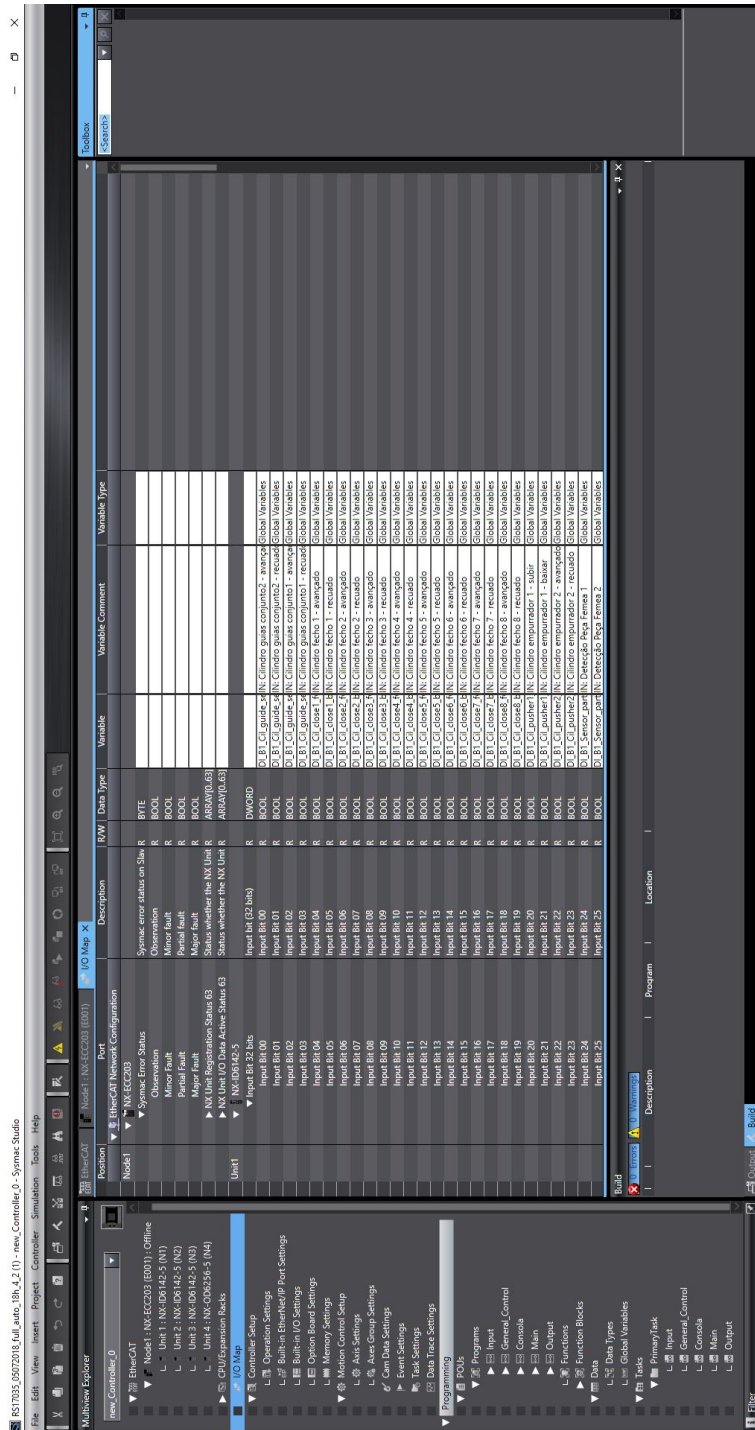


Figura D.3: Janela de mapeamento das entradas e saídas.

D.3 Seleção dos programas a executar

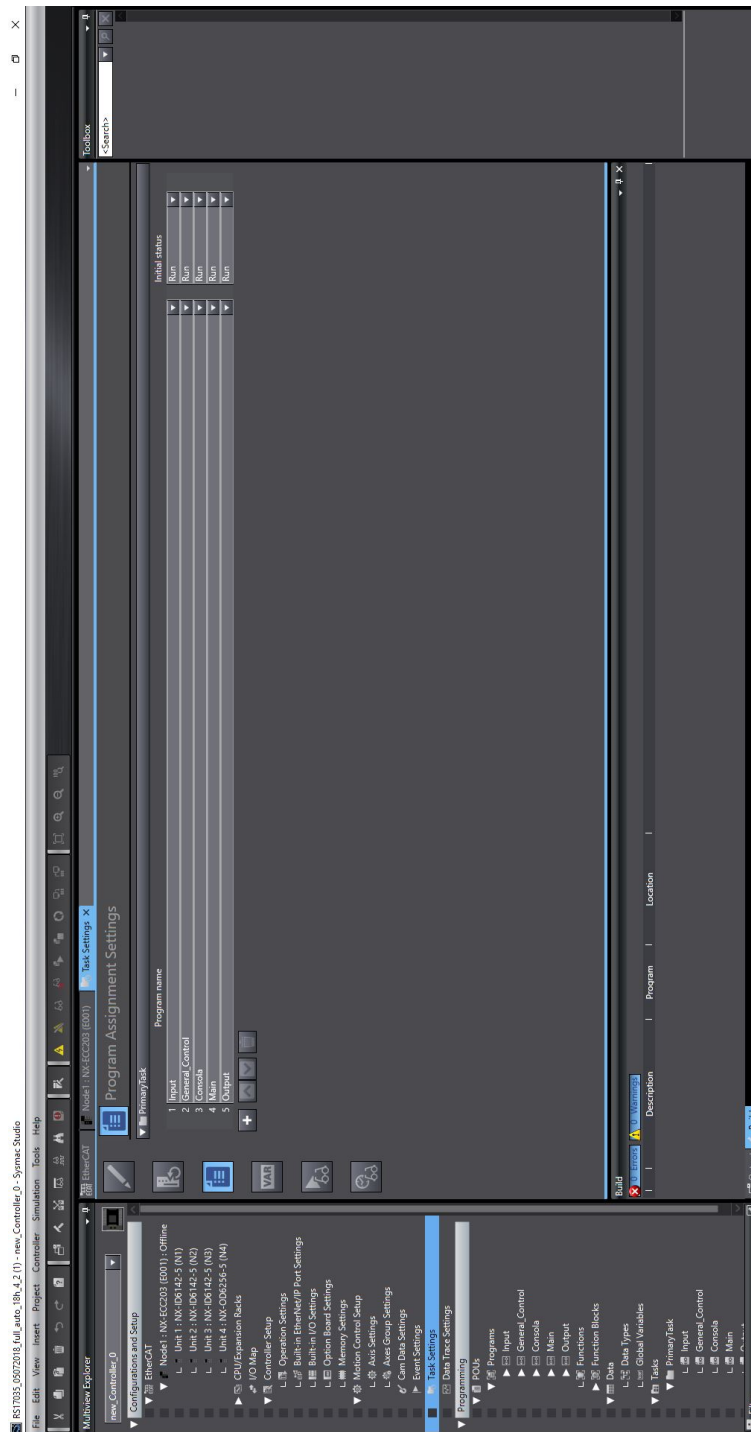


Figura D.4: Menu de seleção de tarefas a executar.

D.4 Configuração da rede de comunicação entre o PLC e os robôs

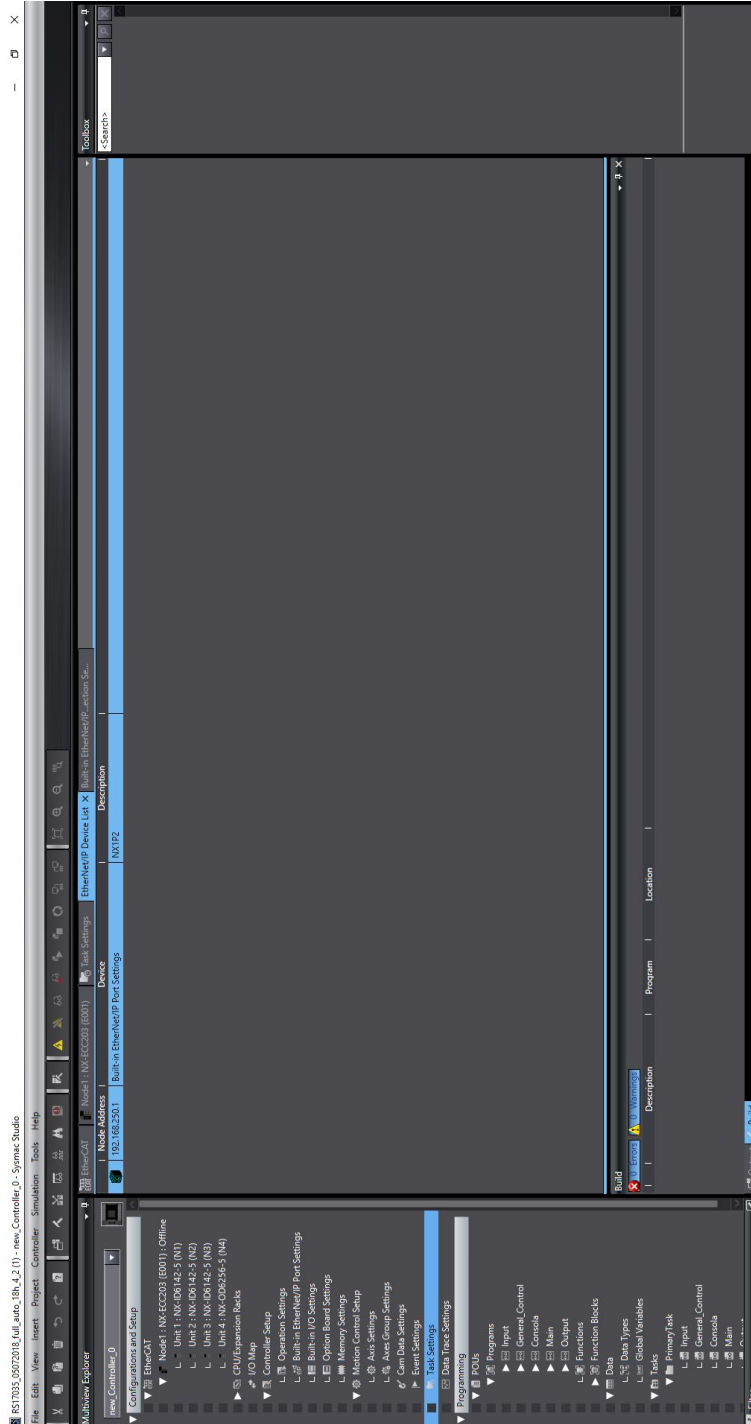


Figura D.5: Nó de comunicação entre o PLC e os robôs.

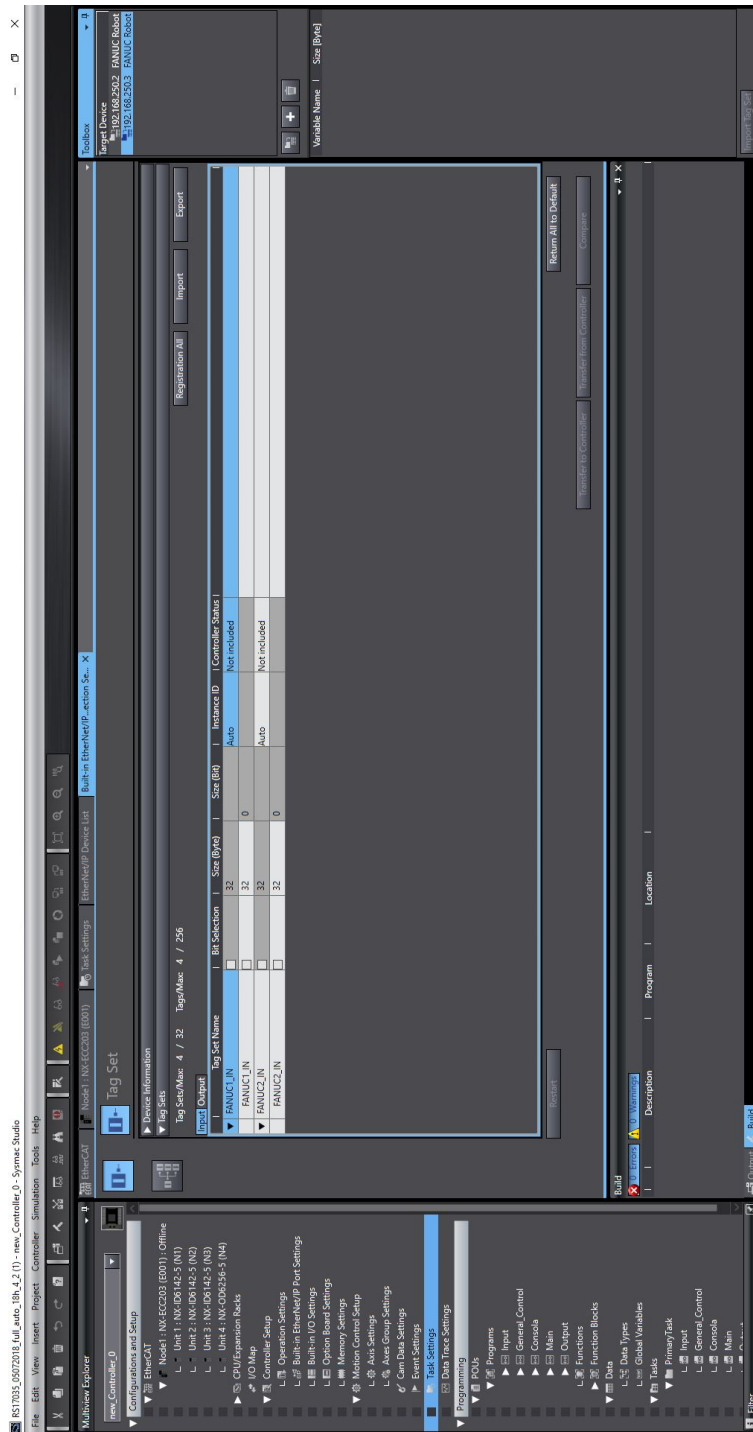


Figura D.6: Robôs ligados ao PLC.

D.4. CONFIGURAÇÃO DA REDE DE COMUNICAÇÃO ENTRE O PLC E OS ROBÔS

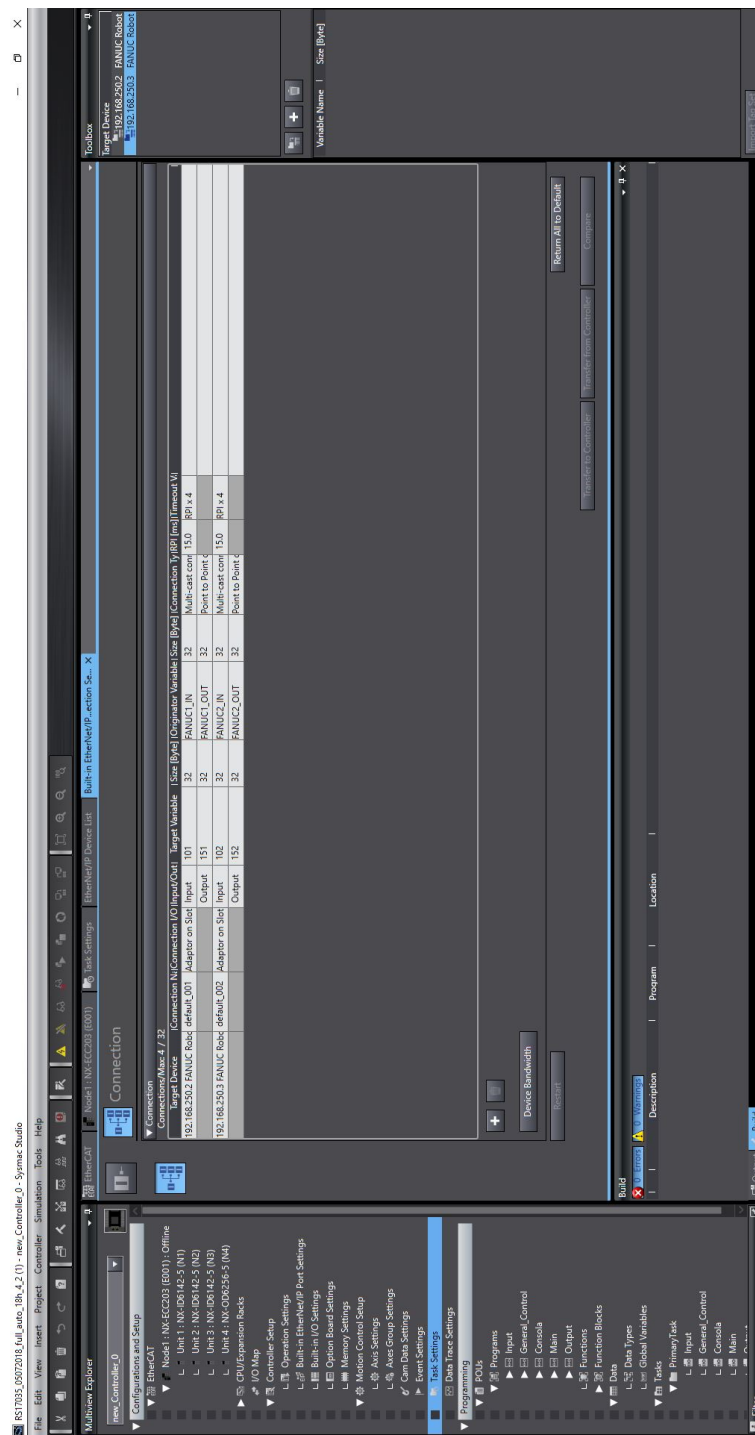


Figura D.7: Definição da comunicação.

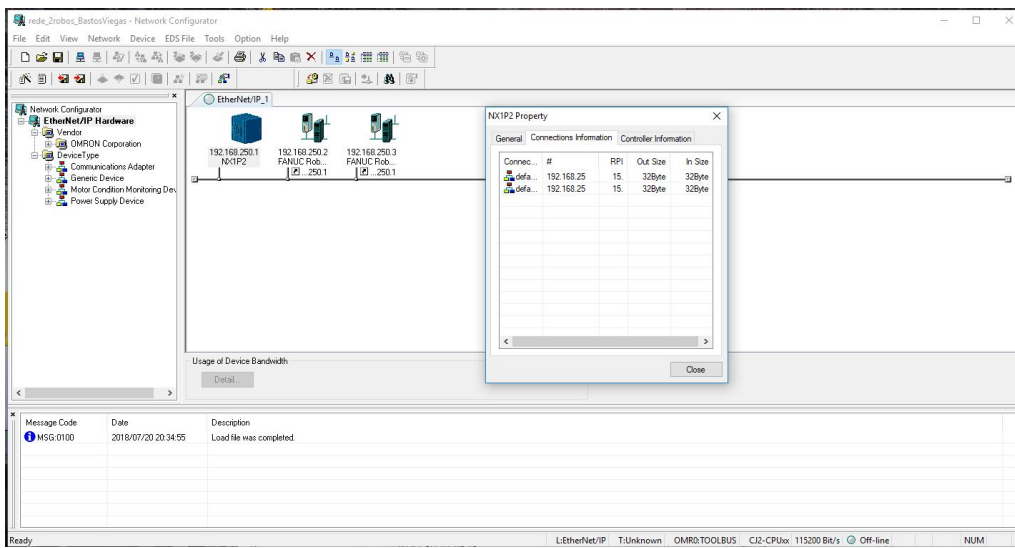


Figura D.8: Configuração da rede no Network Configurator.

Anexo E

Lógica de controlo

E.1 Leitura das entradas do PLC



Figura E.1: Leitura da informação proveniente do Robô.

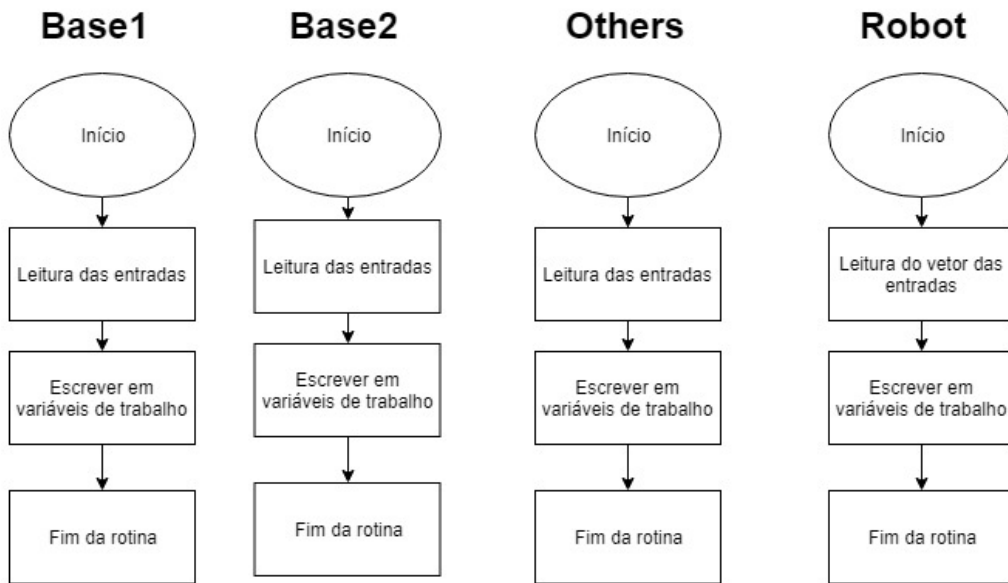


Figura E.2: Diagrama de blocos do programa Input.

E.2 Arranque do sistema

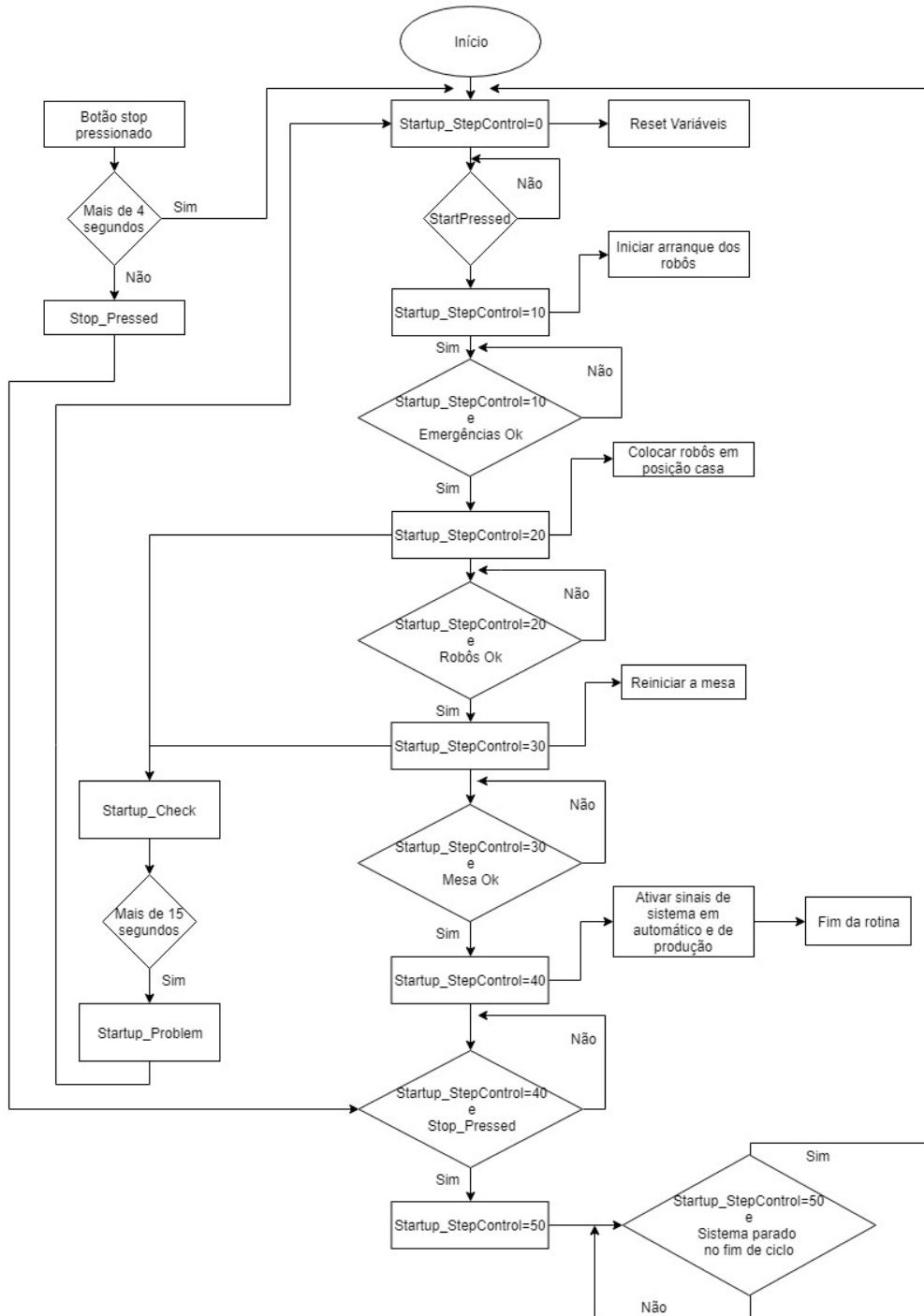


Figura E.3: Fluxograma do controle do arranque do sistema.

E.3 Rotinas de arranque

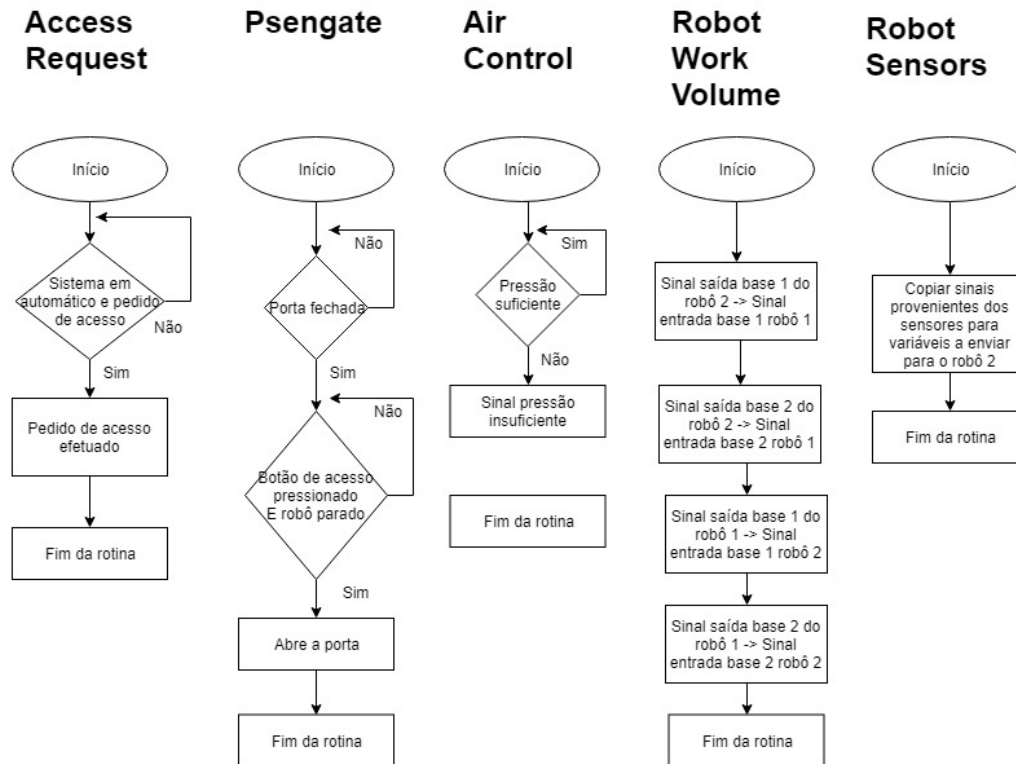


Figura E.4: Fluxograma das rotinas de controlo de pedido de acesso, controlo da psengate, controlo de ar, controlo do volume de trabalho, interface entre sensores e o robô delta.

E.4 Rotinas de carregar receita

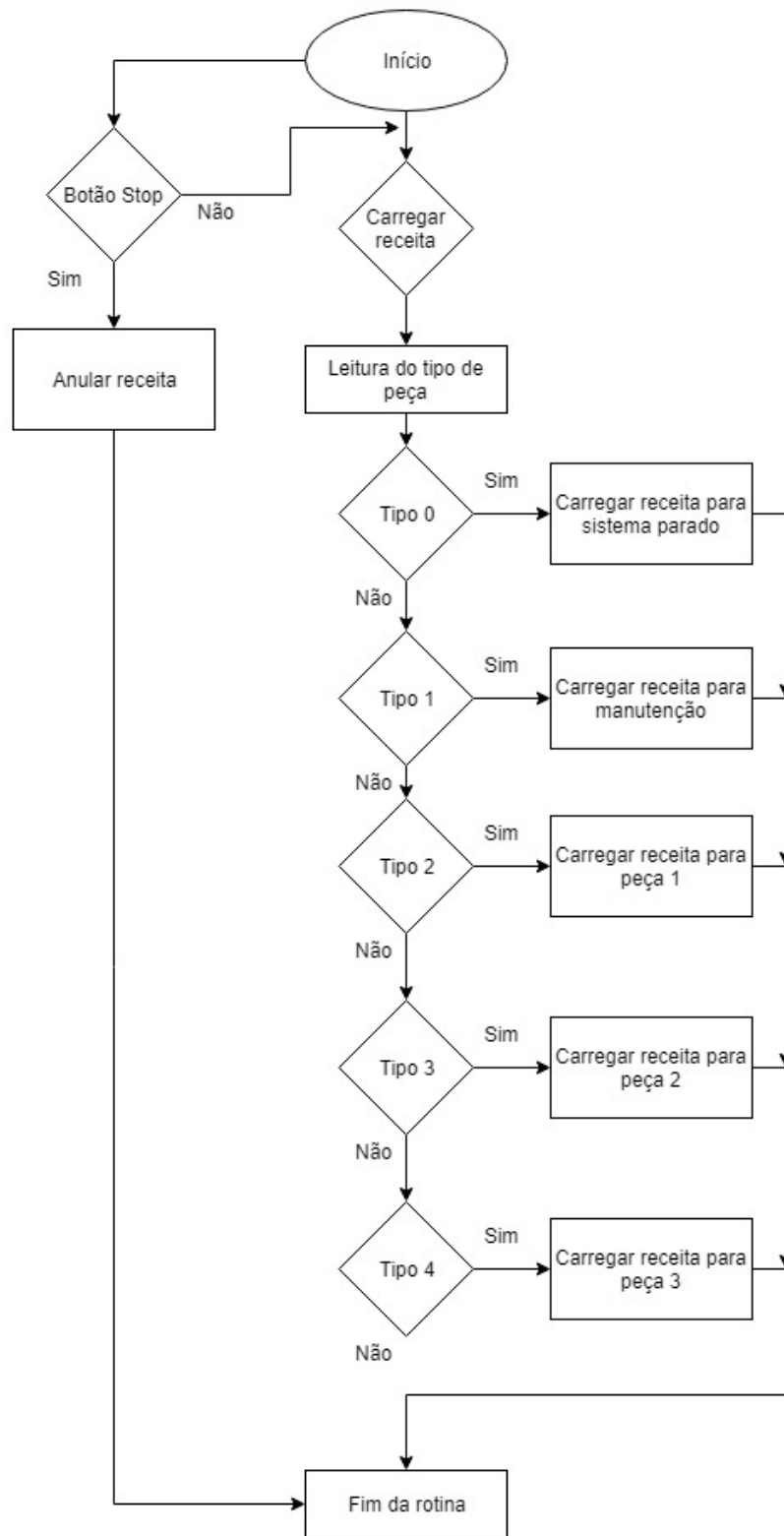


Figura E.5: Carregar receita proveniente do HMI.

E.5 Arranque do robô de descarga

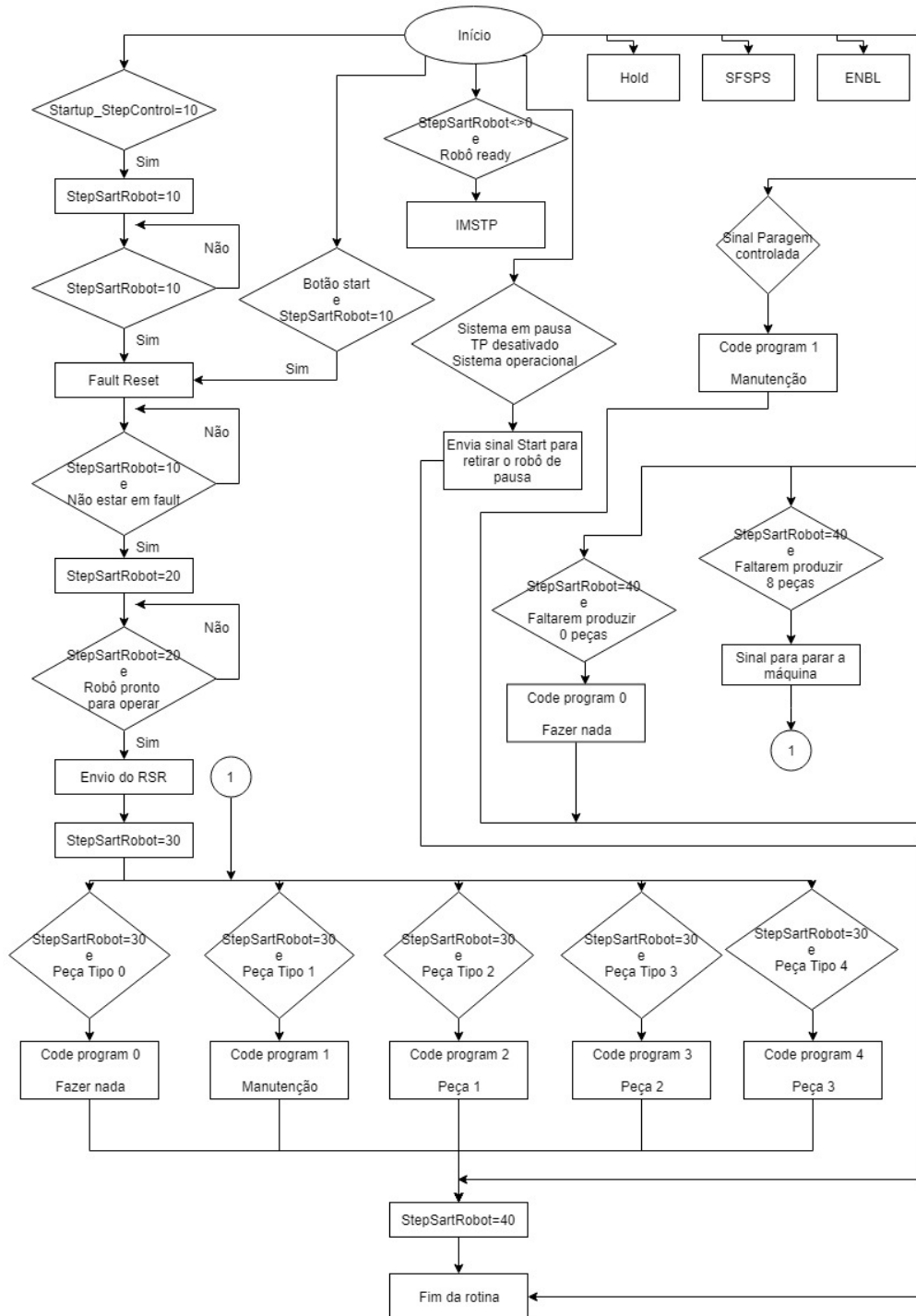


Figura E.6: Arranque do de descarga.

E.6 Arranque do robô delta

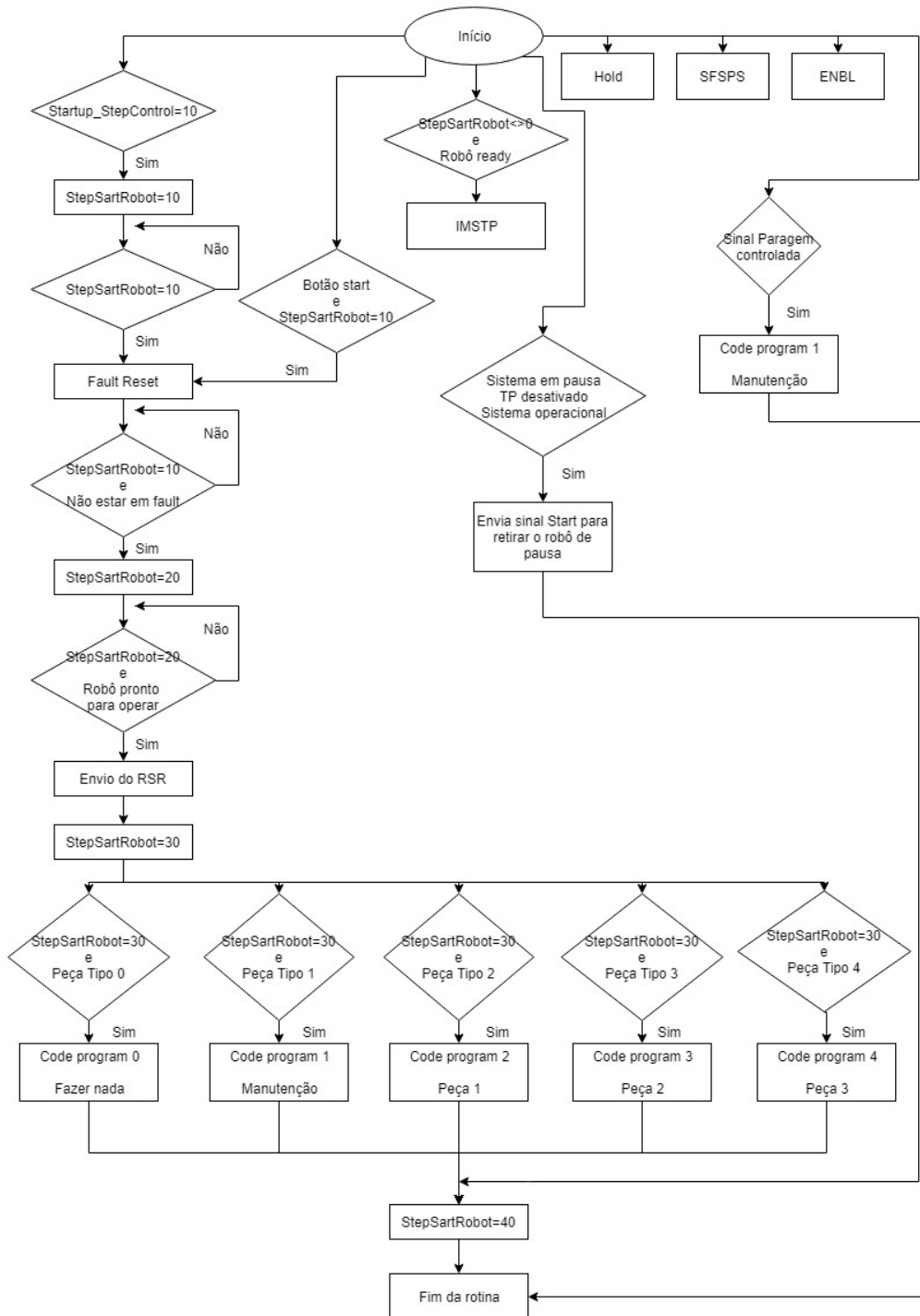


Figura E.7: Arranque do robô delta.

E.7 Comunicação com a HMI

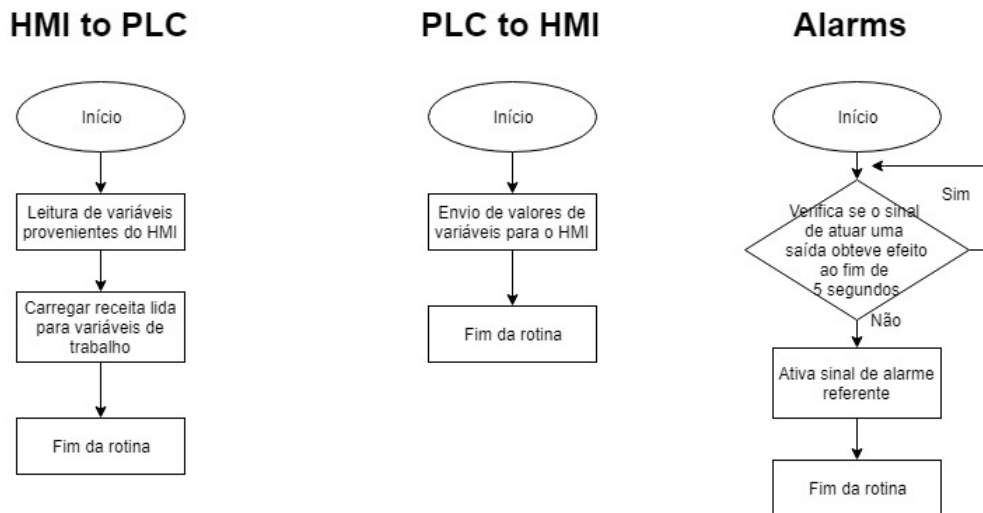


Figura E.8: Fluxograma da comunicação entre PLC e HMI.

E.8 Ciclo base do sistema



Figura E.9: Fluxograma do ciclo base do sistema.

E.9 Ciclo inicial da montagem

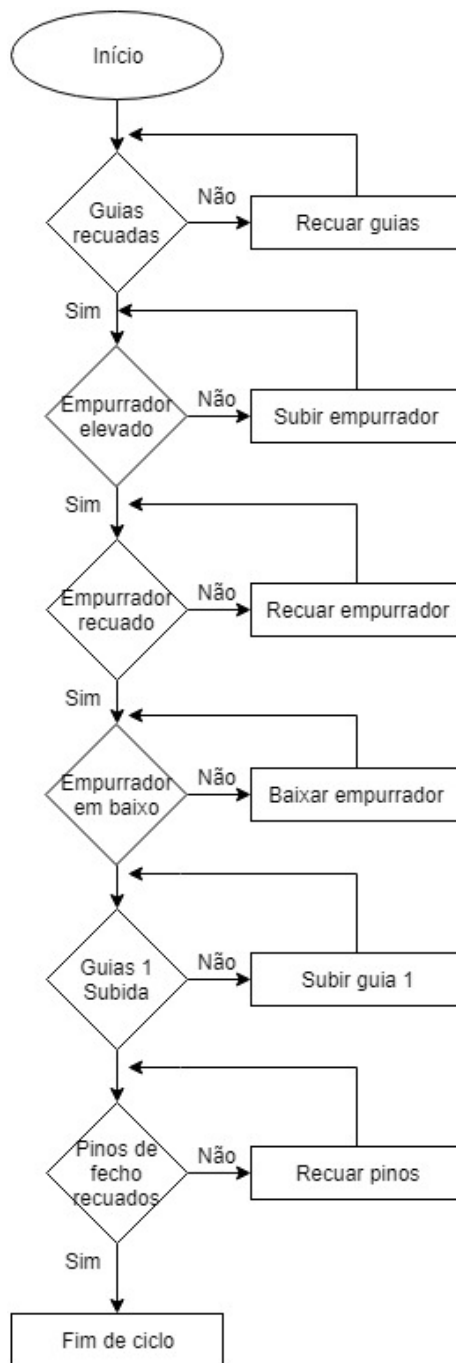


Figura E.10: Fluxograma do ciclo inicial da montagem.

E.10 Ciclo de montagem do robô 2

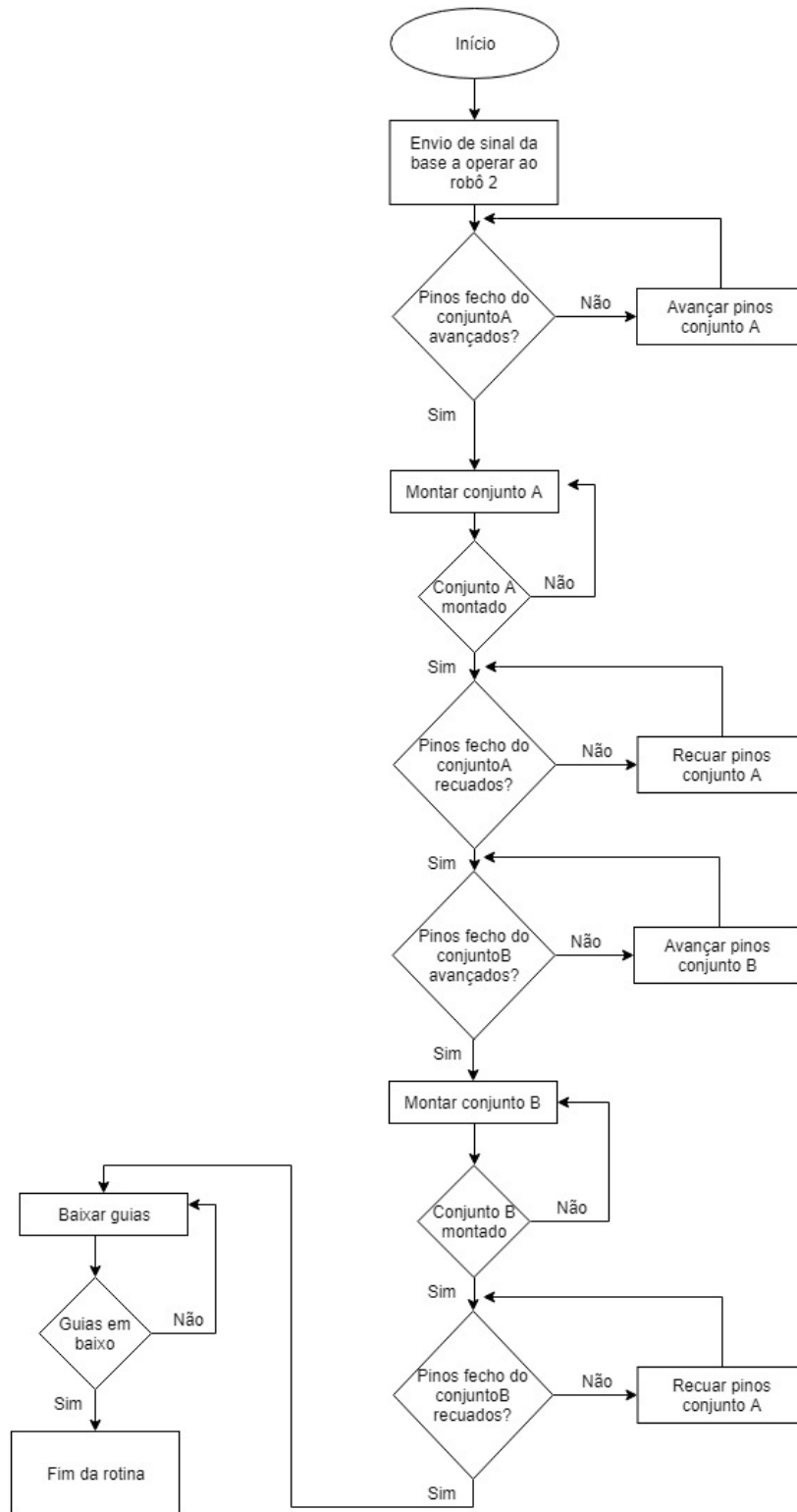


Figura E.11: Fluxograma do ciclo de montagem do robô 2.

E.11 Ciclo de escrita nas saídas do PLC

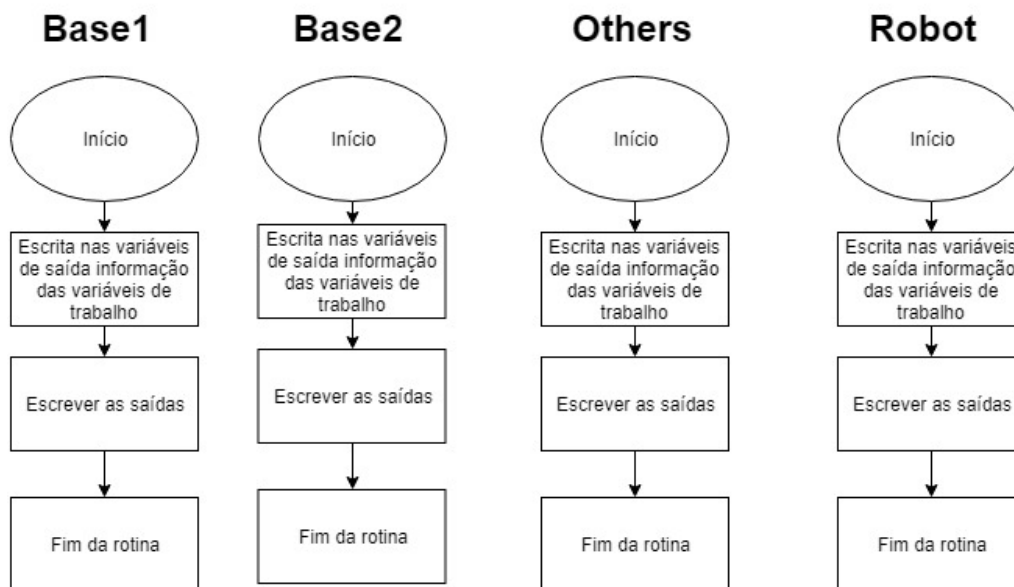


Figura E.12: Fluxograma do ciclo de escrita das saídas.