

Célula robotizada de paletização de caixas para a indústria alimentar

RICARDO DANIEL DE OLIVEIRA MARTINS

setembro de 2018

Célula robotizada de paletização de caixas para a indústria alimentar

Ricardo Daniel de Oliveira Martins



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: Ricardo Daniel de Oliveira Martins, N° 1130337, 1130337@isep.ipp.pt

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

Empresa: RobotSol

Supervisão: Rui Manuel Cardoso da Silva Rodrigues, rui.rodrigues@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

Agosto 2018

Agradecimentos

Gostaria de agradecer ao Professor Manuel Silva, pela sua disponibilidade e orientação dada durante a realização deste trabalho.

Um agradecimento a todas as pessoas com quem tive o prazer de trabalhar na RobotSol, em especial à equipa de automação na qual fui integrado, mais concretamente ao coordenador de equipa Rui Rodrigues e aos restantes elementos, João Fontes e José Nunes, por toda disponibilidade, amizade e conhecimentos partilhados.

A todos os meus amigos que sempre me acompanharam e partilharam comigo momentos e experiências únicas durante o percurso académico.

À minha namorada, Cláudia, por ter acompanhado e caminhado ao meu lado durante este processo. O teu amor, apoio incondicional, paciência, preocupação e conselhos foram muito importantes ao longo deste percurso. Foste o pilar que me ajudou a ultrapassar os momentos mais complicados e que tornou ainda melhores os momentos de felicidade.

Por fim, deixar um enorme agradecimento aos meus pais por estarem sempre ao meu lado durante o meu percurso académico e pelos votos de confiança depositados em mim, por me darem sempre a possibilidade de escolher o meu caminho e apoiarem as minhas decisões.

A todos, o meu obrigado!

Resumo

Nos dias de hoje, a procura por produtos de qualidade e em quantidades elevadas é cada vez maior, levando a que as organizações tenham a necessidade de se adaptarem e melhorarem os seus processos produtivos, tornando estes mais flexíveis e eficazes, de forma a responderem com sucesso às exigências dos mercados.

Por esta razão, a presente dissertação visa analisar o uso de manipuladores robóticos a nível industrial, e através dos conhecimentos adquiridos, realizar a implementação de uma célula robótica para uma linha de paletização de caixas. Este processo será alcançado através da programação do robô, e de todos os outros equipamentos necessários para realizar o controlo da célula, como por exemplo, a consola de interface homem-máquina e a unidade de controlo, na forma de um controlador lógico programável.

Para tal, foi necessário analisar o estado da arte dos manipuladores industriais em relação às suas características e métodos de programação, bem como o seu enquadramento industrial e a possibilidade de integração dos mesmos em processos de paletização. Após a análise teórica, seguiu-se a implementação prática da célula tendo sido realizados os testes necessários para validar o correto funcionamento do sistema.

Com a realização deste projeto, adquiriu-se conhecimentos na programação de manipuladores robóticos para a execução de tarefas de paletização, na programação de controladores lógicos programáveis e na programação de interfaces homem-máquina. Visto que também foi realizada a implementação prática da célula no cliente, foi possível obter competências e experiência no contexto do trabalho e mercado a nível industrial.

Palavras-Chave

Automação, Robótica Industrial, Paletização, Controladores Lógicos Programáveis, Interface Homem-Máquina, FANUC, Omron.

Abstract

Nowadays, the demand for quality products and in high quantities is increasing, leading organizations to have the need to adapt and improve their production processes, making them more flexible and efficient, in order to respond to the market demands.

For this reason, the present dissertation aims to analyze the use of robotic manipulators at an industrial level, and through the acquired knowledge, do the implementation of a robotic cell to perform a palletizing task of boxes. This process will be achieved through programming the robot, and all the other necessary equipment to perform cell control, such as the human-machine interface console and the control unit, in the form of a programmable logic controller.

For this, it was necessary to analyze the state of the art of industrial manipulators in relation to their characteristics and programming methods, as well as their industrial environment and the possibility of their integration in palletizing processes. After the theoretical analysis, the practical implementation of the cell was followed and the necessary tests were carried out to validate the correct functioning of the system.

With the realization of this project, knowledge was acquired in the programming of robotic manipulators for performing palletizing tasks, programming programmable logic controllers and programming human-machine interfaces. Since the practical implementation of the cell in the client was also carried out, it was possible to obtain skills and experience in the context of industrial work and market.

Keywords

Automation, Industrial Robotics, Palletizing, Programmable Logic Controllers, Human Machine Interface, FANUC, Omron

Conteúdo

| | |
|--|-----------|
| Agradecimentos | v |
| Conteúdo | i |
| Lista de Figuras | v |
| Lista de Tabelas | ix |
| Acrónimos | xi |
| 1 Introdução | 1 |
| 1.1 Contextualização | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Calendarização | 2 |
| 1.4 Organização do Relatório | 3 |
| 2 Robótica Industrial | 5 |
| 2.1 Evolução Histórica | 6 |
| 2.2 Sistema Robótico | 9 |
| 2.2.1 Manipulador | 10 |
| 2.2.2 Controlador e Consola de Programação | 14 |
| 2.2.3 Atuador | 15 |
| 2.2.4 Atuador Final | 16 |
| 2.2.5 Sensores | 17 |
| 2.3 Características de um Manipulador | 18 |
| 2.3.1 Velocidade | 18 |
| 2.3.2 Graus de Liberdade | 18 |
| 2.3.3 Volume de Trabalho e Alcance | 18 |
| 2.3.4 Capacidade de Carga | 19 |
| 2.3.5 Resolução, Precisão e Repetibilidade | 20 |
| 2.4 Métodos de Programação | 20 |

| | | |
|----------|---|-----------|
| 2.5 | Conclusão | 22 |
| 3 | Enquadramento Industrial | 23 |
| 3.1 | Programmable Logic Controllers | 23 |
| 3.1.1 | Definição Geral | 24 |
| 3.1.2 | Tipos de PLC | 25 |
| 3.1.3 | Linguagens de Programação | 26 |
| 3.2 | Redes de Comunicação | 28 |
| 3.2.1 | Ethernet for Control Automation Technology - EtherCAT | 29 |
| 3.2.2 | IO-Link | 30 |
| 3.3 | Segurança | 31 |
| 3.4 | Conclusão | 32 |
| 4 | Aplicações de Paletização | 35 |
| 4.1 | O Processo de Paletização | 35 |
| 4.2 | Sistemas de Paletização Convencionais | 38 |
| 4.3 | Robótica e Paletização | 40 |
| 4.3.1 | Manipuladores Robóticos de Paletização | 41 |
| 4.3.2 | Atuadores Finais para Aplicações de Paletização | 42 |
| 4.3.3 | <i>Softwares</i> Dedicados à Paletização | 45 |
| 4.4 | Casos de Estudo | 48 |
| 4.4.1 | Forbes Chocolate | 48 |
| 4.4.2 | Y&S Candies | 50 |
| 4.4.3 | <i>Voit</i> | 52 |
| 4.5 | Conclusão | 53 |
| 5 | Célula de Paletização Robotizada | 55 |
| 5.1 | Definição do Problema | 55 |
| 5.2 | Funcionamento da Célula | 58 |
| 5.3 | Arquitetura do Sistema | 59 |
| 5.3.1 | Unidade de Controlo | 60 |
| 5.3.2 | Manipulador e Atuador Final | 61 |
| 5.3.3 | Interface Homem-Máquina | 65 |
| 5.3.4 | Transportadores | 66 |
| 5.3.5 | Mesas Formadoras de Mosaicos | 67 |
| 5.3.6 | Sistema de Transporte de Paletes - charriot | 69 |
| 5.3.7 | Dispositivos de Segurança | 70 |
| 5.4 | Conclusão | 76 |
| 6 | Implementação Prática | 77 |
| 6.1 | Simulação <i>Offline</i> | 77 |
| 6.2 | Configuração do Manipulador Robótico | 79 |

| | | |
|----------|--|------------|
| 6.2.1 | Sistemas de Coordenadas | 79 |
| 6.2.2 | Carga de Trabalho | 80 |
| 6.2.3 | Sinais de Entradas e Saídas | 81 |
| 6.2.4 | Registos e Registos de Posições | 83 |
| 6.2.5 | Background Logic | 84 |
| 6.2.6 | Posições de Referência | 84 |
| 6.2.7 | Configuração de Rede | 85 |
| 6.2.8 | Limitação dos eixos | 86 |
| 6.2.9 | <i>Space Functions</i> | 86 |
| 6.3 | Programação do Manipulador | 87 |
| 6.4 | Configuração e Programação do PLC | 93 |
| 6.4.1 | Configuração da Rede EtherCAT | 93 |
| 6.4.2 | Configuração da Rede EthernetIP | 95 |
| 6.4.3 | Programação do PLC e Controlo da Célula | 96 |
| 6.5 | Interface Homem-Máquina | 102 |
| 6.6 | Segurança | 106 |
| 6.6.1 | Barreira de Segurança | 106 |
| 6.6.2 | Scanners de Segurança | 110 |
| 6.6.3 | Programação do PLC de Segurança | 112 |
| 6.7 | Conclusão | 113 |
| 7 | Testes e Resultados | 115 |
| 7.1 | Simulação <i>Offline</i> do Código do Robô | 115 |
| 7.2 | Comunicação Robô-PLC | 116 |
| 7.3 | <i>I/O Check</i> | 117 |
| 7.4 | Sequência de Formação de Mosaicos e Transferência de Paletes | 118 |
| 7.5 | Movimentação do Robô e Manipulação de Objetos | 118 |
| 7.6 | Seguranças | 120 |
| 7.7 | Limitações de movimentos | 120 |
| 7.8 | Arranque do Sistema e Controlo de Funcionamento | 121 |
| 7.9 | Conclusões | 122 |
| 8 | Conclusões | 123 |
| 8.1 | Solução Adotada | 123 |
| 8.2 | Perspetivas Futuras | 125 |
| 8.3 | Conclusão | 126 |
| | Bibliografia | 127 |
| A | Sinais Trocados com o Manipulador FANUC | 133 |
| B | Fluxogramas Rotinas FANUC | 141 |

C Fluxogramas Programas PLC

151

Lista de Figuras

| | | |
|------|---|----|
| 1.1 | Calendarização do projeto | 2 |
| 2.1 | <i>Robot Knight</i> de Leonardo Da Vinci, criado por Mark Rosheim | 7 |
| 2.2 | Unimate a realizar uma operação de manipulação | 8 |
| 2.3 | Robô colaborativo YuMi – IRB 14000 da ABB | 9 |
| 2.4 | Arquitetura geral de um sistema robótico | 9 |
| 2.5 | Tipos de juntas de um manipulador robótico | 11 |
| 2.6 | Manipulador com configuração cartesiana e respetivo volume de trabalho. | 12 |
| 2.7 | Manipulador com configuração cilíndrica e respetivo volume de trabalho. | 12 |
| 2.8 | Manipulador com configuração esférica e respetivo volume de trabalho. | 13 |
| 2.9 | Manipulador com configuração SCARA e respetivo volume de trabalho. | 13 |
| 2.10 | Manipulador com configuração articulada e respetivo volume de trabalho. | 14 |
| 2.11 | Manipulador com configuração paralela e respetivo volume de trabalho. | 14 |
| 2.12 | Controlador IRC5 da ABB e respetivo <i>teach pendant</i> | 15 |
| 2.13 | Garra mecânica Schunk SWG e garra magnética Schunk GSW-M | 17 |
| 2.14 | Volume de trabalho e alcance do robô KUKA KR 1000 L750 Titan | 19 |
| 2.15 | Capacidade de carga do robô KUKA KR 1000 L750 Titan | 19 |
| 2.16 | Comportamento de um manipulador com diferentes capacidades de precisão e repetibilidade | 20 |
| 3.1 | Princípio de funcionamento de um PLC | 24 |
| 3.2 | PLC Omron modular da série CJ2M e PLC Omron compacto da série CP1E | 25 |
| 3.3 | Linguagens de programação de PLC de acordo com a norma IEC 61131 | 26 |
| 3.4 | Arquitetura de um sistema com comunicação EtherCAT | 29 |
| 3.5 | Arquitetura de um sistema com comunicação IO-Link | 30 |
| 4.1 | Exemplo de mosaico entrelaçado e em coluna | 36 |

| | | |
|------|---|----|
| 4.2 | Exemplo de <i>display pallet</i> , <i>mixed layer pallet</i> e <i>true mixed pallet</i> | 37 |
| 4.3 | Exemplo de paletizador convencional <i>floor level</i> e <i>high level</i> | 38 |
| 4.4 | Componentes base de um sistema de paletização convencional | 39 |
| 4.5 | Componentes base dos sistemas de paletização robotizados | 40 |
| 4.6 | Trajectoria efetuada por um manipulador no processo de paletização | 42 |
| 4.7 | Tipos de atuadores finais mais comuns em aplicações de paletização | 44 |
| 4.8 | Atuadores finais com dois tipos de sistemas de manipulação | 44 |
| 4.9 | Atuador final da Brenton Engineering com sistema em cortina | 45 |
| 4.10 | Janela de configuração de parâmetros no <i>software</i> PalletPro | 47 |
| 4.11 | Sistema implementado na empresa Forbes Chocolate e garra mecânica utilizada no manipulador | 49 |
| 4.12 | Célula de trabalho instalada na Y&S Candies | 51 |
| 4.13 | Manipulador KUKA a paletizar as placas de madeira | 53 |
| 5.1 | Mosaicos definidos para a célula de paletização | 56 |
| 5.2 | Configuração da célula implementada | 57 |
| 5.3 | Arquitetura da célula implementada | 60 |
| 5.4 | PLC industrial Omron NX1P2-9024DT1 | 61 |
| 5.5 | Manipulador robótico industrial FANUC M410iC/315, Controlador R-30iB e Consola de Programação iPendant | 62 |
| 5.6 | Atuador final implementado | 63 |
| 5.7 | Componentes pneumáticos utilizados no atuador final | 64 |
| 5.8 | Sistema de controlo do atuador final | 64 |
| 5.9 | Painel de controlo | 65 |
| 5.10 | Tapetes de correntes implementados | 66 |
| 5.11 | Sensor de reflexão no objeto ML100 e Sensor de medição de distância VDM28 | 67 |
| 5.12 | Mesas de formação de mosaico implementadas | 68 |
| 5.13 | Rolo motorizado RollerDrive EC310 e Controlador DriveControl 20 | 69 |
| 5.14 | Desenho do sistema de transporte de paletes implementado | 69 |
| 5.15 | Controlador de segurança e respetivas cartas de entradas e saídas de segurança | 71 |
| 5.16 | Relé de segurança | 72 |
| 5.17 | Botoneira de emergência | 72 |
| 5.18 | Fechadura eletrónica de portas | 73 |
| 5.19 | <i>Scanner</i> de segurança Omron | 73 |
| 5.20 | Zonas de deteção do <i>scanner</i> | 74 |
| 5.21 | Interruptor de fim de curso de segurança | 75 |
| 5.22 | Barreira de segurança | 76 |
| 6.1 | Alcance vertical do manipulador sem pedestal e com pedestal | 78 |
| 6.2 | Análise da trajetória para a manipulação da paleta | 78 |

| | | |
|------|--|-----|
| 6.3 | Validação do volume de trabalho do manipulador robótico | 79 |
| 6.4 | Sistemas de coordenadas de um manipulador robótico | 80 |
| 6.5 | Cargas de trabalho definidas no manipulador robótico | 81 |
| 6.6 | Configuração dos sinais UOP através da consola de programação . . . | 81 |
| 6.7 | Configuração dos sinais de grupo através da consola de programação . | 83 |
| 6.8 | Configuração do programa a correr em <i>Background Logic</i> | 84 |
| 6.9 | Configuração das posições de referência | 85 |
| 6.10 | Configuração da comunicação EthernetIP do lado do controlador ro- bótico | 85 |
| 6.11 | Configuração da limitação dos eixos do manipulador | 86 |
| 6.12 | <i>Space Functions</i> configurados no manipulador | 87 |
| 6.13 | Diagrama de árvore dos programas do manipulador robótico | 88 |
| 6.14 | Fluxograma do programa RSR0100 | 89 |
| 6.15 | Fluxograma do programa AUTOSEQ | 91 |
| 6.16 | Configuração da rede EtherCAT na unidade de controlo | 94 |
| 6.17 | Configuração dos nós EtherCAT e da <i>rack</i> principal da unidade de controlo | 95 |
| 6.18 | Janela de criação das tramas de comunicação EthernetIP | 96 |
| 6.19 | Configuração da rede EthernetIP do sistema | 97 |
| 6.20 | Diagrama de árvore dos programas da unidade de controlo | 98 |
| 6.21 | Fluxograma da lógica de arranque do sistema | 99 |
| 6.22 | Janela HOME da consola | 103 |
| 6.23 | Janela Seleção de Produção da consola | 103 |
| 6.24 | Janela Estados Célula da consola | 104 |
| 6.25 | Janela de estado da mesa formadora de mosaicos | 104 |
| 6.26 | Janela de comando manual da mesa formadora de mosaicos | 105 |
| 6.27 | Janela de historial de alarmes | 105 |
| 6.28 | Janelas de gestão de utilizadores e adição/remoção de utilizadores . . | 106 |
| 6.29 | Método de <i>muting</i> implementado na barreira de segurança | 109 |
| 6.30 | Zonas de segurança configuradas nos <i>scanners</i> | 111 |
| 6.31 | Configuração das cartas de entradas e saídas de segurança | 112 |
| 6.32 | <i>Function Blocks</i> de segurança | 113 |
| 7.1 | Trajectoria de <i>pick</i> e <i>place</i> de palete | 116 |
| 7.2 | Teste de comunicação de sinais digitais e registos entre o robô e o PLC | 117 |
| 7.3 | Receção dos sinais dos sensores do atuador final na consola HMI . . . | 118 |
| 7.4 | Tela adicionada ao sistema de manipulação do atuador final | 119 |
| B.1 | Fluxograma de funcionamento da rotina <i>CLOSE_GRIPPER</i> | 141 |
| B.2 | Fluxograma de funcionamento da rotina <i>OPEN_GRIPPER</i> | 142 |
| B.3 | Fluxograma de funcionamento da rotina <i>HOME</i> | 143 |
| B.4 | Fluxograma de funcionamento da rotina <i>MAINTENANCE</i> | 144 |

| | | |
|------|---|-----|
| B.5 | Fluxograma de funcionamento da rotina <i>PICK_BOX</i> | 145 |
| B.6 | Fluxograma de funcionamento da rotina <i>PLACE_BOX</i> | 146 |
| B.7 | Fluxograma de funcionamento da rotina <i>PICK_PALLET</i> | 147 |
| B.8 | Fluxograma de funcionamento da rotina <i>PLACE_PALLET</i> | 148 |
| B.9 | Fluxograma de funcionamento da rotina <i>PICK_INTERLAYER</i> | 149 |
| B.10 | Fluxograma de funcionamento da rotina <i>PLACE_INTERLAYER</i> | 150 |
| | | |
| C.1 | Fluxograma de funcionamento dos transportadores de paletização | 152 |
| C.2 | Fluxograma de funcionamento dos transportadores de transferência | 153 |
| C.3 | Fluxograma de funcionamento dos transportadores de armazenamento de paletes | 154 |
| C.4 | Fluxograma de funcionamento dos transportadores de entrada de paletes | 155 |
| C.5 | Fluxograma de funcionamento das mesas formadoras de mosaicos | 156 |
| C.6 | Fluxograma de funcionamento da sequência de formação do mosaico de caixas grandes | 157 |
| C.7 | Fluxograma de funcionamento da sequência de formação do mosaico de caixas pequenas | 158 |
| C.8 | Fluxograma de funcionamento da leitura dos sensores IO-Link | 159 |

Lista de Tabelas

| | | |
|-----|--|-----|
| 4.1 | Componentes base de um sistema de paletização convencional | 39 |
| 4.2 | Componentes base de um sistema de paletização robotizado | 41 |
| 4.3 | Programas dedicados à programação de robôs para aplicações de pa- letização | 46 |
| 6.1 | Sinais I/O utilizados nos manipuladores FANUC | 82 |
| 6.2 | Rotinas programadas no manipulador robótico | 92 |
| 6.3 | Parâmetros das equações do cálculo da distância de segurança | 108 |
| 6.4 | Parâmetros das equações do cálculo da distância de instalação dos sensores de <i>muting</i> | 110 |
| A.1 | Sinais UI | 133 |
| A.2 | Sinais UO | 134 |
| A.3 | Sinais DI | 135 |
| A.4 | Sinais DI convertidos para GI | 136 |
| A.5 | Sinais DO | 137 |
| A.6 | Sinais GI e Registos | 139 |
| A.7 | Registos de Posição | 140 |

Acrónimos

| Abreviatura | Descrição |
|-------------|---|
| ABB | Asea Brown Boveri |
| AI | <i>Analog Input</i> |
| AO | <i>Analog Output</i> |
| AC | <i>Alternating Current</i> |
| AGV | <i>Automated Guided Vehicle</i> |
| CG | Caixa Grande |
| CNC | <i>Computer Numeric Control</i> |
| CP | Caixa Pequena |
| CPU | <i>Central Processing Unit</i> |
| DC | <i>Direct Current</i> |
| DI | <i>Digital Input</i> |
| DO | <i>Digital Output</i> |
| DOF | <i>Degrees Of Freedom</i> |
| EtherCAT | <i>Ethernet for Control Automation Technology</i> |
| EPAL | European Pallet Association |
| FANUC | Fuji Automatic Numerical Control |
| FBD | <i>Function Block Diagram</i> |
| HMI | <i>Human-Machine Interface</i> |
| IEC | International Electrotechnical Commission |
| IL | <i>Instruction List</i> |
| ISEP | Instituto Superior de Engenharia do Porto |
| KUKA | Keller und Knappich Augsburg |
| LAD | <i>Ladder Diagram</i> |
| PLC | <i>Programmable Logic Controller</i> |
| RIA | Robotic Industries Association |
| SFC | <i>Sequential Function Chart</i> |
| ST | <i>Structured Text</i> |
| USB | <i>Universal Serial Bus</i> |

Capítulo 1

Introdução

1.1 Contextualização

A forma como as empresas, nos dias de hoje, planeiam as suas estratégias de produção sofreu grandes alterações devido às exigências do mercado. Os mercados exigem inovação a nível de produtos e serviços, processos de produção mais otimizados, redução de prazos e custos de fabrico, aumento da produtividade e melhoria de qualidade. Estas necessidades surgem e estão em constante mudança, pois a oferta é superior à procura, a evolução tecnológica origina a rápida obsolescência dos produtos e a grande variedade de produtos disponíveis permite ao cliente fazer a sua escolha em função dos seus requisitos.

Para tal é fundamental o desenvolvimento de sistemas com capacidades de autonomia e inteligência que permitam alcançar a maioria das necessidades enunciadas anteriormente. Com o aparecimento de circuitos integrados, sistemas de controlo computacional e robôs industriais, surgiram os primeiros sistemas que podem ser programados com o objetivo de exercerem o trabalho que, até então, era realizado pelo ser humano. A automação dos processos de fabrico é uma solução viável para que sejam atingidos alguns dos parâmetros acima referidos, visto que aumenta a flexibilidade das operações, tornando-as mais rápidas e eficazes. Automação define-se como a execução de uma sequência de operações com pouca, ou nenhuma, intervenção humana, utilizando dispositivos e equipamentos que executam um programa de instruções e realizam o controlo e monitorização do processo de fabrico [1].

A nível industrial, os equipamentos mais utilizados para realizarem trabalho que até então era realizado pelo ser humano são os braços robóticos e as máquinas que funcionam por controlo numérico computadorizado (*Computer Numeric Control* – CNC).

1.2 Objetivos

As necessidades das empresas de automatizarem os seus processos de produção, com o objetivo de reduzirem tempos de execução e aumentarem a flexibilidade e eficácia do processo, motivaram a realização do presente projeto que estabelece os seguintes objetivos:

- analisar e estudar o processo de paletização a nível convencional e robotizado;
- desenvolver um sistema robótico capaz de efetuar a paletização de caixas com diferentes dimensões, aumentando os índices de produtividade e a eficácia do processo;
- desenvolver a programação do manipulador robótico, do controlador lógico programável e da consola de interface homem-máquina.

1.3 Calendarização

Sendo o presente trabalho a programação e implementação de uma célula robotizada de paletização, foram definidas um conjunto de tarefas, que por sua vez originaram a calendarização apresentada na Figura 1.1.

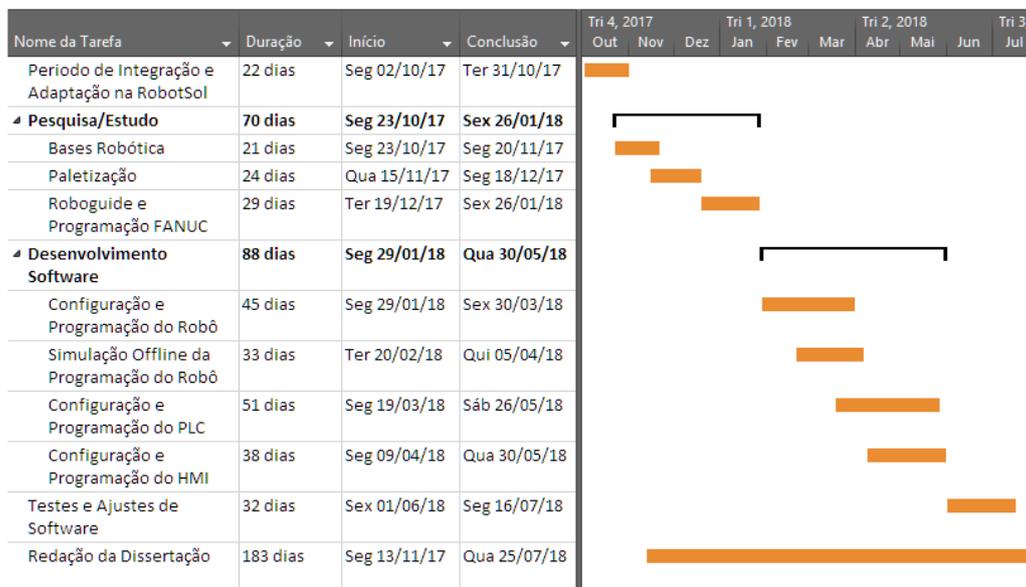


Figura 1.1: Calendarização do projeto.

1.4 Organização do Relatório

Nesta secção apresenta-se a estrutura da dissertação, a qual é constituída por oito capítulos enunciados de seguida. Após o primeiro capítulo introdutório, os três capítulos seguintes são referentes ao estado da arte da dissertação. Após o estado da arte, segue o trabalho desenvolvido composto também por três capítulos, finalizando-se o documento com as conclusões.

No Capítulo 1 é feita uma breve introdução à presente dissertação abordando aspetos como a contextualização, os objetivos, a calendarização e a organização do documento.

Já no Capítulo 2 é apresentado o tema da robótica industrial, abordando tópicos como a evolução histórica, os constituintes de um sistema robótico, as características de um manipulador industrial e os vários métodos de programação.

O Capítulo 3 aborda o enquadramento industrial dos manipuladores robóticos e a forma como estes podem ser integrados num sistema. Para tal serão abordados os controladores lógicos programáveis, como unidade de controlo dos manipuladores robóticos, algumas redes de comunicação existentes a nível industrial, terminando com os protocolos e regras de segurança que devem ser cumpridas.

Segue-se o Capítulo 4, onde é apresentado o processo de paletização, os sistemas de paletização convencionais, a ligação entre a robótica e a paletização, fechando com alguns casos de estudo nos quais foram aplicadas células robotizadas em processos industriais de paletização.

No Capítulo 5 é apresentada a solução da célula de paletização implementada. Após uma breve definição do problema e do funcionamento e lógica de trabalho da célula, é apresentada a sua arquitetura, especificando os vários equipamentos que fazem parte da sua constituição.

O Capítulo 6 apresenta todo o trabalho desenvolvido ao longo desta dissertação, explanando os passos realizados de forma a obter o sistema final. Serão apresentadas as diretrizes seguidas na programação do manipulador robótico, da unidade de controlo e da consola de interface.

Quanto ao Capítulo 7, este apresenta todos os testes que foram realizados durante a implementação do projeto de forma a validar o trabalho desenvolvido e os devidos resultados.

Por fim, o Capítulo 8 reúne as conclusões e possíveis perspetivas de futuros desenvolvimentos no trabalho desenvolvido.

Capítulo 2

Robótica Industrial

"What is an industrial robot? A simple definition is: a programmable, mechanical device used in place of a person to perform dangerous or repetitive tasks with a high degree of accuracy." [1]

Nos dias de hoje, a utilização de robôs no sector industrial é uma realidade bastante comum e, mesmo apesar de estes se encontrarem instalados em grande número e de forma global em várias empresas, não existe uma definição universalmente reconhecida para este equipamento. Segundo a Robotic Industries Association (RIA), um robô industrial é um dispositivo mecânico e programável utilizado para substituir um ser humano em trabalhos perigosos e repetitivos com um elevado grau de precisão.

Assim, um sistema robótico de forma a ser produzido em grande escala e ser o mais versátil possível para que consiga substituir de forma eficiente o ser humano deve atender a uma vasta gama de potenciais aplicações (montagem, paletização, pintura, soldadura, maquinação e manipulação de materiais). Contudo, isso não é uma tarefa simples e apesar dos manipuladores robóticos serem sistemas com alguma versatilidade, existem manipuladores com características mais indicadas para realizar, por exemplo, tarefas de pintura e outros com características para realizar tarefas de paletização. Um robô que seja projetado para executar um trabalho de pintura e um que tenha sido projetado para executar uma tarefa de paletização possuem características diferentes em relação à capacidade de carga, número de eixos, volume de trabalho, entre outras.

Por forma a que o robô execute o trabalho desejado e de forma segura, este vem acompanhado de equipamentos e sistemas que auxiliem na execução da tarefa (unidade de controlo, ferramenta de trabalho ou garra, atuador, sensores, equipamentos de segurança e de transporte de materiais) [2]. Uma das configurações que pode ser adotada numa célula a nível industrial é um, ou mais, robôs

com características análogas ao braço humano, normalmente denominados por manipuladores. Os veículos auto-guiados, do inglês, *Automated Guided Vehicle* (AGV) são também utilizados em algumas aplicações para fazerem o transporte de materiais entre vários pontos do espaço de trabalho fabril como, por exemplo, o transporte de um produto acabado entre o final de uma linha de produção e um armazém.

A utilização de robôs ao nível industrial tem vindo a crescer consideravelmente, pois estes podem substituir os humanos em ambientes de trabalho mais perigosos, as tarefas são executadas com mais consistência e repetibilidade e podem ser reprogramados para executar uma tarefa diferente do que aquela para a qual foram programado inicialmente. Assim, neste capítulo serão abordados vários aspetos importantes que dizem respeito aos robôs industriais, em concreto os manipuladores, como a constituição de um sistema robótico, conceitos a ter em consideração na escolha de um manipulador para um projeto e a forma como estes podem ser programados.

2.1 Evolução Histórica

Acredita-se que o primeiro robô antropomórfico a ser construído foi entre 1495 e 1497 por Leonardo Da Vinci, devido a várias páginas reveladas pertencentes ao seu livro *Codex Atlanticus* que se presume que continham estudos de um robô. Hoje em dia chamado de “Robot Knight”, o robô de Da Vinci foi projetado para se sentar, acenar com os braços, mover a cabeça e abrir e fechar o maxilar. Estudos sugerem que o robô tinha dois sistemas mecânicos independentes. O primeiro, as pernas com três graus de liberdade, tornozelos, joelhos e ancas. O segundo, os braços com quatro graus de liberdade, ombros articulados, cotovelos, pulsos e mãos [3, 4]. Com inspiração nos desenhos de Leonardo Da Vinci, em 2002, Mark Rosheim construiu um modelo físico deste robô, como é possível observar na Figura 2.1

Apesar de terem existido outros desenvolvimentos na área da automação e robótica depois de Leonardo Da Vinci, foi apenas em 1921 que o termo “robô” foi introduzido através de uma peça, do dramaturgo checo Karel Capek, intitulada *Rossum’s Universal Robots*. O termo checo “*robot*”, utilizado na peça de Capek, significa trabalho forçado [6].

No ano de 1941, o escritor e cientista Isaac Asimov, através da sua história de ficção científica *Runaround*, introduz a palavra “robótica” para denominar a ciência que estuda e trabalha com robôs. Foi também Asimov que enunciou as leis da robótica que ainda hoje são aplicadas. Numa primeira instância foram introduzidas três leis fundamentais, sendo mais tarde introduzida a lei zero [7].



Figura 2.1: *Robot Knight* de Leonardo Da Vinci, criado por Mark Rosheim [5].

- 1ª lei: Um robô não pode fazer mal a um ser humano e nem, por inação, permitir que algum mal lhe aconteça.
- 2ª lei: Um robô deve obedecer às ordem dos seres humanos, exceto quando estas contrariarem a primeira lei.
- 3ª lei: Um robô deve proteger a sua integridade física, desde que com isto não contrarie as duas primeiras leis.
- Lei zero: Um robô não pode fazer mal à humanidade e nem, por inação, permitir que ela sofra algum mal.

Em 1954, o britânico Cyril W. Kenward desenvolveu um manipulador que se movia num sistema de eixos de três dimensões (x-y-z), tendo sido a patente emitida em 1957 [6]. Quando se fala da história da robótica é importante referir também o americano George C. Devol. Primeiro inventou um dispositivo, por volta de 1946 com patente emitida em 1952, que permitia gravar magneticamente sinais elétricos para que estes sinais pudessem ser depois reproduzidos para controlar o funcionamento das máquinas. Em segundo é creditado pela invenção de um equipamento na década de 1950, com patente emitida em 1961, denominado “*Programmed Article Transfer*” destinado a manipulação de peças [6]. Apesar de cronologicamente o robô de Kenward ter sido o primeiro, o de Devol provou ser

mais importante no desenvolvimento e comercialização de robôs devido a uma união com Joseph Engelberger.

Os dois formaram uma empresa e trabalharam em conjunto por forma a melhorar e comercializar o dispositivo desenvolvido por Devol. Lançaram o primeiro produto com o nome Unimate, representado na Figura 2.2, utilizado para descarregar uma máquina de fundição na General Motors [2].



Figura 2.2: Unimate a realizar uma operação de manipulação [2].

Naturalmente, com o avanço tecnológico surgiram novas configurações robóticas. Os robôs evoluíram para sistemas com melhores capacidades e desempenho, mecanismos de movimentos e de controlo mais avançados, tornando-se em máquinas mais versáteis e capazes de executarem tarefas com maior eficiência. Hoje em dia existem variadíssimas marcas, com uma vasta gama de modelos de robôs industriais, com capacidades distintas para o tipo de aplicação onde se pretende implementar o robô. É importante definir de forma correta a tarefa que o robô vai executar por forma a escolher um modelo que cumpra com os requisitos.

A evolução da robótica teve marcos importantes, desde a progressão tecnológica à capacidade de integração e colaboração (dos sistemas robóticos) com os humanos. Se há umas décadas atrás, o conceito de um robô trabalhar ao lado de um operador era impensável, por representar um risco de segurança, hoje em dia essa ideia é menos atrevida com o aparecimento dos robôs colaborativos. Como o próprio nome indica, são robôs projetados para trabalharem em colaboração com o ser humano e para tal possuem algumas limitações ao nível de tamanho, velocidade de trabalho e capacidade de carga, quando comparados com os robôs industriais mais comuns. Na Figura 2.3 está representado um robô colaborativo da empresa Asea Brown Boveri (ABB) com dois braços mecânicos.



Figura 2.3: Robô colaborativo YuMi – IRB 14000 da ABB [8]

2.2 Sistema Robótico

Como citado no início deste capítulo, um robô industrial só é capaz de executar determinada tarefa, de forma correta e eficaz, quando interligado com outros equipamentos, formando um sistema com algum nível de autonomia e apto a realizar a sua função com pouca, ou nenhuma, intervenção humana. A arquitetura geral de um sistema robótico encontra-se representada na Figura 2.4 e serão abordados cada um dos seus componentes nas subsecções seguintes.

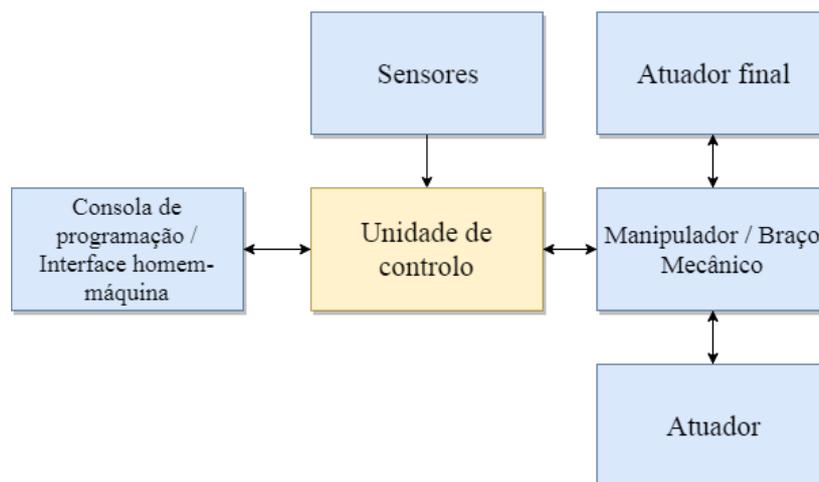


Figura 2.4: Arquitetura geral de um sistema robótico

2.2.1 Manipulador

O manipulador é a estrutura mecânica que permite a execução do trabalho proposto a um sistema robótico. A nível industrial o tipo de manipulador mais adotado é constituído, por analogia ao braço humano, por três componentes: a base, o braço e o punho.

A base é a estrutura que sustenta o robô e que permite, habitualmente, um movimento rotacional de todo o manipulador em torno do seu eixo vertical. O braço é a parte do robô que permite posicionar a extremidade do robô no espaço operacional, enquanto que a função do punho é fazer a orientação do atuador final, possibilitando que este atinja os pontos necessários para a correta execução da tarefa [9]. O manipulador é constituído por partes rígidas, os elos, ligados entre si pelas juntas. As juntas do manipulador fornecem a capacidade de movimento relativo entre dois elos, o elo de entrada e o elo de saída. Este movimento entre as juntas e os elos permite mover, posicionar e orientar o robô para a realização da operação. O número total de movimentos independentes que um manipulador consegue efetuar denomina-se por graus de liberdade, do inglês, *degrees-of-freedom* (dof). Um manipulador, de forma a alcançar qualquer ponto com a orientação desejada dentro do seu volume de trabalho necessita de pelo menos seis dof, três para o posicionamento e três para a orientação [7].

As juntas podem ser divididas em diferentes tipos, como ilustrado na Figura 2.5, e dependendo do tipo de movimento (linear ou rotacional) as suas denominações são [2, 6]:

- Linear ou Prismática, tipo L ou P, que permitem o movimento linear de um dos elos em relação ao outro, desde que os eixos dos elos sejam paralelos. Este tipo de junta permite um tipo de movimento análogo a um ponteiro telescópico caracterizado por apenas um grau de liberdade;
- Ortogonal, tipo O, que têm um funcionamento idêntico às juntas lineares, mas os elos de entrada e saída encontram-se perpendiculares entre si durante o movimento;
- Rotacional, tipo R, caracterizada por um movimento rotacional entre o elo de entrada e de saída, sendo o eixo de rotação perpendicular aos eixos dos elos;
- Torção, tipo T, análoga à junta rotacional, ou seja, permite um movimento rotacional entre os elos, mas com um eixo de rotação paralelo aos eixos dos elos;
- Revolvente, tipo V, que permite também um movimento rotacional entre os elos de entrada e saída, mas ao contrário das juntas do tipo R e do tipo

T, o eixo do elo de entrada é paralelo ao eixo de rotação e o eixo do elo de saída é perpendicular.

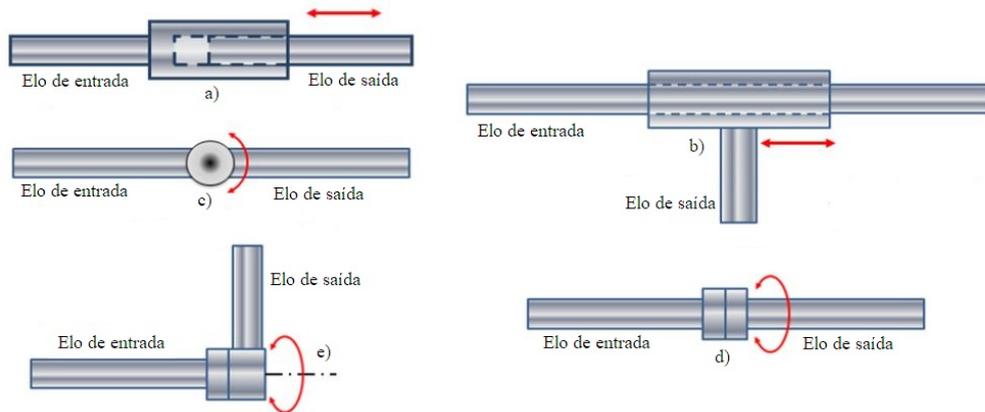


Figura 2.5: Tipos de juntas de um manipulador robótico: a) linear, b) ortogonal, c) rotacional, d) torção, e) revolvente [10]

A forma como os elos e as juntas se encontram interligados definem a configuração do manipulador. De uma forma geral, é possível dividir as configurações em dois grupos: configurações em série e configurações em paralelo. Nas configurações em série os elos e as juntas encontram-se ligados em sequência, ou seja, um elo de entrada de uma junta é também o elo de saída da junta que o antecede. Já nas configurações em paralelo, os elos, como o próprio nome indica, encontram-se ligados em paralelo, ou seja, a uma junta podem estar ligados vários elos.

Nos manipuladores série é possível identificar várias configurações, sendo que esta é baseada nas três primeiras juntas a contar da sua base. É possível obter diversas configurações distintas visto que cada junta pode proporcionar ao robô um movimento linear ou rotacional e podem estar interligadas com os elos de diferentes formas. Assim sendo, as configurações mais comuns de manipuladores industriais são [7, 11, 12]:

- Cartesiana

A configuração cartesiana é caracterizada pela existência de três eixos ortogonais de movimento linear através de juntas prismáticas. Geralmente robôs com esta configuração possuem uma estrutura em pórtico que se distingue através da sua estrutura que suporta os eixos, como é possível observar na Figura 2.6. Este tipo de manipulador é caracterizado por um volume de trabalho grande, capacidade de carga elevada e uma estrutura simples que proporciona rigidez ao manipulador.

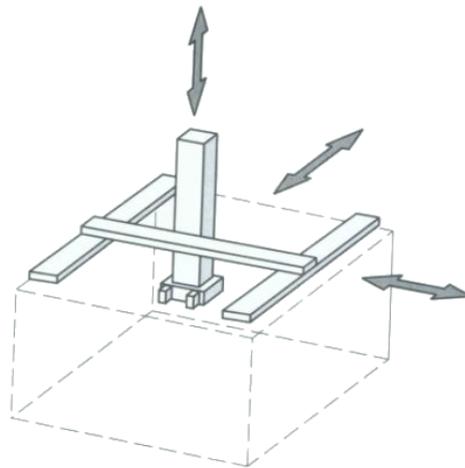


Figura 2.6: Manipulador com configuração cartesiana e respetivo volume de trabalho [13].

- Cilíndrica

A configuração cilíndrica distingue-se pela combinação de duas juntas lineares com uma rotacional, como se pode observar na Figura 2.7. Permite definir a posição do atuador final em relação à base do manipulador em coordenadas cilíndricas. Manipuladores com esta configuração apresentam uma estrutura simples, o que lhes permite elevadas velocidades de trabalho.

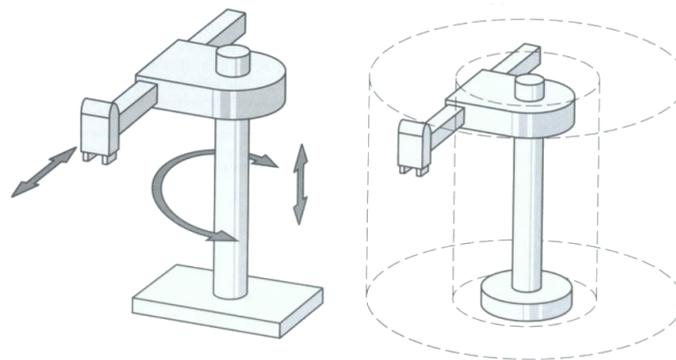


Figura 2.7: Manipulador com configuração cilíndrica (esquerda) e respetivo volume de trabalho (direita) [13].

- Esférica

Esta configuração é identificada pela combinação de dois eixos de movimento rotacional com um movimento linear, como é possível observar na

Figura 2.8. Permite definir a posição do atuador final em relação à base em coordenadas esféricas e é uma configuração que permite ao manipulador uma capacidade de carga e parâmetros como velocidade, precisão e repetibilidade elevados.

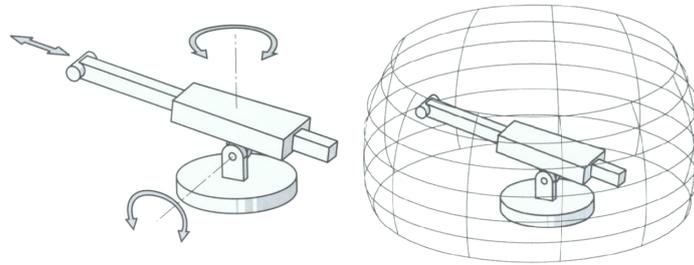


Figura 2.8: Manipulador com configuração esférica (esquerda) e respetivo volume de trabalho (direita) [13].

- *Selective Compliance Assembly Robot Arm - SCARA*

Os manipuladores de configuração SCARA consistem em duas juntas rotacionais paralelas, que proporcionam um movimento rotacional no plano horizontal, e por uma junta prismática que permite um movimento linear vertical, ver Figura 2.9. São manipuladores geralmente utilizados em tarefas de montagem e caracterizados pela elevada precisão e repetibilidade, elevada velocidade e livres de vibrações para ciclos curtos e controlo avançado para a precisão da trajetória.

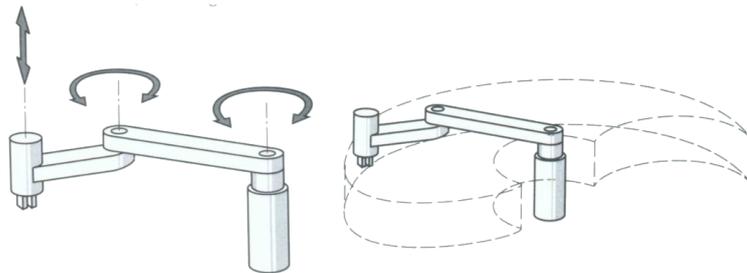


Figura 2.9: Manipulador com configuração SCARA (esquerda) e respetivo volume de trabalho (direita) [13].

- Articulada ou antropomórfica

Os manipuladores com configuração articulada são caracterizados pela existência de apenas juntas rotacionais. É um tipo de configuração com grandes

semelhanças ao braço humano sendo, por isso, também designada por configuração antropomórfica, como é possível observar na Figura 2.10. Apresenta várias vantagens como boa manobrabilidade, velocidade elevada e um grande volume de trabalho para uma área de montagem pequena.

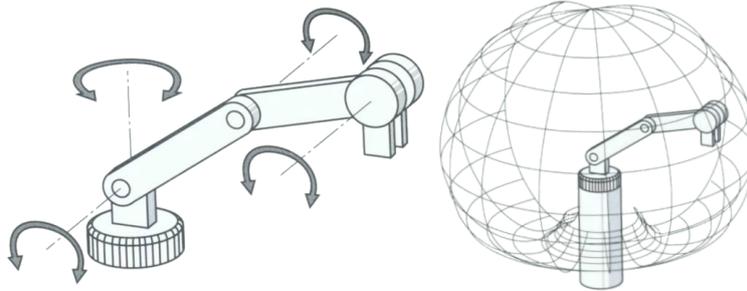


Figura 2.10: Manipulador com configuração articulada (esquerda) e respetivo volume de trabalho (direita) [13].

- Paralela

As configurações anteriormente descritas são configurações associadas a manipuladores série. Nos manipuladores paralelos, como referido, a uma junta podem estar ligados diversos elos, como se pode observar na Figura 2.11. Devido às características desta configuração é possível fixar o robô de diversas formas e obter velocidades de ciclo extremamente elevadas.

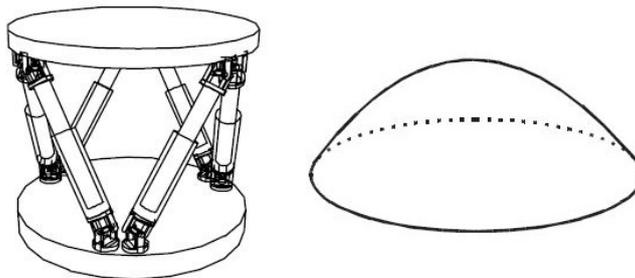


Figura 2.11: Manipulador com configuração paralela (esquerda) e respetivo volume de trabalho (direita) [11].

2.2.2 Controlador e Consola de Programação

Se, por analogia ao corpo humano, se pode dizer que o manipulador é o “corpo” do robô, então é possível referir o controlador como o seu “cérebro”. O controlador

do robô consiste em *hardware* eletrônico e *software* capaz de fazer o controlo do movimento das juntas durante a realização da operação de forma a garantir que os elos e as juntas se encontrem devidamente posicionados e que façam a trajetória desejada [1].

O controlador aciona os atuadores ligados a cada junta e coordena o seu movimento por forma a colocar o atuador final, que se encontra instalado no punho do manipulador, na posição correta. Cabe também ao controlador fazer a gestão dos sinais de entrada (*input*) e saída (*output*), normalmente denominados por sinais de I/O, de modo a fazer o controlo de dispositivos externos (*inputs*), como por exemplo, ler o estado de sensores que indiquem a abertura e o fecho de uma garra mecânica [11].

A consola de programação, ou *teach pendant*, é o dispositivo associado ao controlador que proporciona ao utilizador uma interface mais intuitiva com o sistema robótico. Permite fazer o controlo manual dos movimentos das juntas, realizar a programação *online* do manipulador e configurar diversos parâmetros do robô como, por exemplo, o seu endereço de rede. Na Figura 2.12 é possível observar três controladores da ABB e a respetiva consola de programação.



Figura 2.12: Controlador IRC5 da ABB e respetivo *teach pendant* [14].

2.2.3 Atuador

O atuador é o equipamento que gera e impõe movimento às partes mecânicas do manipulador recorrendo a forças e binários, através da conversão de energia. Por meio das forças aplicadas pelo atuador é possível contrariar a força gravitacional

e inércia, permitindo o movimento das juntas com o objetivo de alterar a posição da extremidade do robô conforme o desejado.

2.2.4 Atuador Final

Um manipulador industrial só é capaz de exercer a sua função quando equipado com um atuador final, que é instalado no punho do robô e, muitas das vezes, concebido especificamente para uma dada aplicação. Normalmente faz-se a divisão do atuador final em duas categorias: garras e ferramentas [1].

As garras são utilizadas para fazer a manipulação/movimentação de objetos em aplicações, como por exemplo, de paletização e carga e descarga de máquinas. Devido às várias características diferentes que os objetos a ser manipulados podem ter (forma, material, tamanho, peso) as garras podem ser de diversos tipos, sendo alguns deles [1, 6, 12]:

- mecânicas, que consistem em dois ou mais dedos que realizam o movimento de abertura e fecho, exercendo uma força suficiente para segurar os objetos a manipular. Por forma a aumentar a eficácia da garra, os dedos devem possuir uma forma geométrica o mais aproximada possível da forma do objeto, aumentando assim a área de contacto evitando-se e diminuindo deslocamentos relativos entre o objeto e a garra, e com o intuito de aumentar o atrito entre os dedos e o objeto, o interior dos dedos da garra pode ser revestido de uma substância de borracha [12]. Na Figura 2.13 é possível observar uma garra mecânica da marca Schunk.
- magnéticas, para manuseamento de materiais ferromagnéticos, como chapas ou placas metálicas. Neste tipo de garra a forma geométrica do objeto não é tão relevante pois apenas existe uma superfície de contacto com o objeto [12]. Na Figura 2.13 é possível observar uma garra magnética da marca Schunk;
- sucção, possuem na extremidade uma, ou mais, ventosas que usam o vácuo para agarrar o objeto. Tal como nas magnéticas, apenas necessitam de uma superfície de contacto e são um tipo de garra aplicável a uma variedade mais extensa de materiais apresentando limitações em objetos que possuam perfurações ou que sejam bastante maleáveis, como por exemplo, o tecido [12];
- agulhas, indicadas para manusear materiais que possam ser perfurados sem qualquer problema [12];
- adesivas, insufláveis ou de diafragma, com dedos articulados, com tecnologia de *jammimg*, universais, sendo que este conjunto de garras é utilizado em

menor escala e para um conjunto de aplicações de carácter mais específico [12];



Figura 2.13: Garra mecânica Schunk SWG (esquerda) e garra magnética Schunk GSW-M (direita) [15, 16].

As ferramentas são aplicadas em tarefas onde o robô deve realizar uma operação no objeto de trabalho. Podem ser ferramentas rotativas para tarefas de maquinação, rebarbagem, polimento e furação, tochas para soldadura por arco elétrico ou pinças para soldadura por pontos, pistolas de pintura, entre outras [6, 12].

Em alguns casos existe a necessidade de um sistema mais flexível que desempenhe tarefas diferentes. Para tal, em vez de se colocar um segundo sistema robótico, pode ser mais vantajoso utilizar um sistema automático de troca do atuador final. Assim, dependendo da tarefa que tem de executar, o manipulador é capaz de aceder a um armazém onde se encontram vários atuadores e fazer a troca, de forma rápida, do atuador final.

Dependendo das especificações da aplicação pode ser necessário projetar e conceber o atuador final devido à elevada variedade de setores na indústria, que trabalham com materiais diferentes, onde é possível automatizar processos.

2.2.5 Sensores

De forma a que seja feito o controlo do sistema robótico, devido à possibilidade de ocorrência de erros ou outros fatores que coloquem em risco o correto funcionamento do sistema, são usados sensores. São dispositivos que auxiliam no correto funcionamento do sistema fornecendo informação ao controlador. Dependendo da origem da informação, os sensores podem ser classificados como

proprioceptivos e exteroceptivos [2]. Os sensores proprioceptivos são aqueles que fornecem informação sobre o próprio manipulador, como a posição atual de cada junta, temperatura, entre outros. Já os sensores exteroceptivos geram informação acerca do ambiente externo do manipulador, como por exemplo, distância a um certo objeto.

2.3 Características de um Manipulador

Analisados os vários constituintes de um sistema robótico e as possíveis configurações que o manipulador pode adotar, é possível concluir que existem várias características que apresentam comportamentos diferentes, como por exemplo, o volume de trabalho. Desta forma, é importante definir bem quais os requisitos da aplicação para que a seleção do modelo do robô seja adequada. Nesta secção serão abordadas as características mais importantes a ter em consideração na escolha do manipulador robótico.

2.3.1 Velocidade

A velocidade de um manipulador é a variação da sua posição ao longo do tempo e é medida na extremidade do punho do manipulador. Esta determina o tempo de ciclo de uma dada tarefa [12]. A velocidade é uma característica importante e a ter em consideração visto que deve ser determinada consoante a precisão com que se pretende posicionar o manipulador e o peso do objeto que está a ser manipulado.

2.3.2 Graus de Liberdade

Os graus de liberdade de um manipulador (dof) são representados pelo número total de movimentos independentes que este consegue executar [11]. Como forma de exemplo, se o manipulador é capaz de se mover linearmente em três direções e rodar em torno de três eixos tem seis graus de liberdade.

2.3.3 Volume de Trabalho e Alcance

Entende-se por volume de trabalho do robô o conjunto de todos os pontos que este é capaz de alcançar. O volume de trabalho difere dependendo da configuração do manipulador, tal como foi possível observar na Subsecção 2.2.1, onde foram abordadas as várias configurações de um manipulador robótico [11]. Relativamente ao alcance, este é definido pela distância máxima que o atuador final consegue alcançar face à base do manipulador. Na Figura 2.14 é possível observar o volume de trabalho e alcance de um robô da empresa Keller und Knappich Augsburg (KUKA).

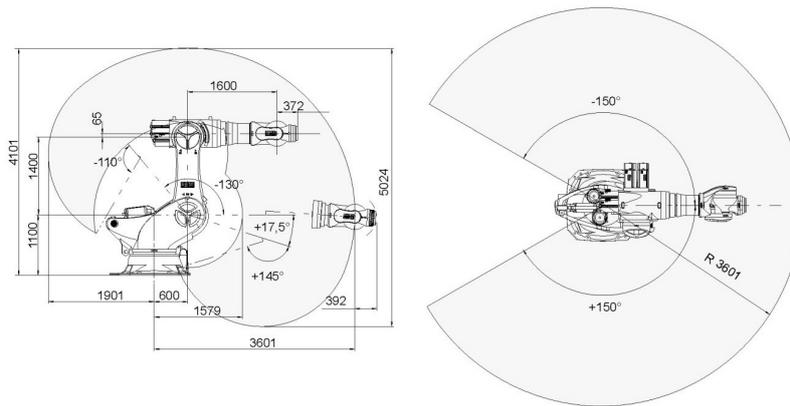


Figura 2.14: Volume de trabalho e alcance do robô KUKA KR 1000 L750 Titan [17].

2.3.4 Capacidade de Carga

A capacidade de carga é a massa total que o robô é capaz de manipular. Normalmente é possível distinguir dois tipos de capacidade de carga: a carga nominal e máxima. A capacidade de carga nominal é a carga máxima que o manipulador consegue manipular à velocidade máxima sem perda de precisão. A capacidade de carga máxima é a carga máxima com que o manipulador consegue trabalhar a velocidade baixa sem perda de precisão [11, 12].

De notar que a capacidade de carga do robô diminui com o aumento da distância do centro de massa do conjunto garra mais carga que é transportada à falange do manipulador. Na Figura 2.15 é possível observar as especificações de capacidade de carga de um robô KUKA e o respetivo diagrama de carga.

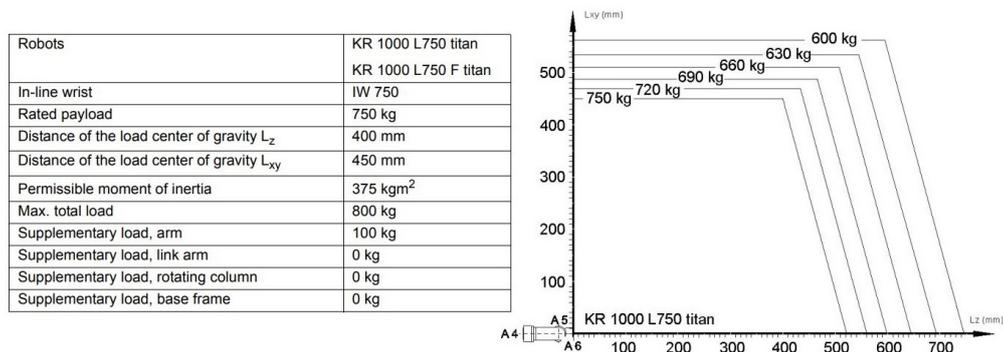


Figura 2.15: Capacidade de carga do robô KUKA KR 1000 L750 Titan [17].

2.3.5 Resolução, Precisão e Repetibilidade

Resolução representa o menor movimento incremental que as juntas do manipulador conseguem executar [2].

Precisão é a capacidade que o robô possui de posicionar o seu atuador final num ponto programado. Este parâmetro é obtido medindo a distância entre o ponto que o robô atinge e o ponto programado [2].

Repetibilidade traduz-se na capacidade com que o robô se consegue reposicionar repetidamente na mesma posição [2].

Para que a aplicação seja o mais eficaz possível é importante que o manipulador possua uma boa precisão e repetibilidade. Na Figura 2.16 é possível observar o comportamento de um manipulador robótico dependendo das suas capacidades de precisão e repetibilidade.

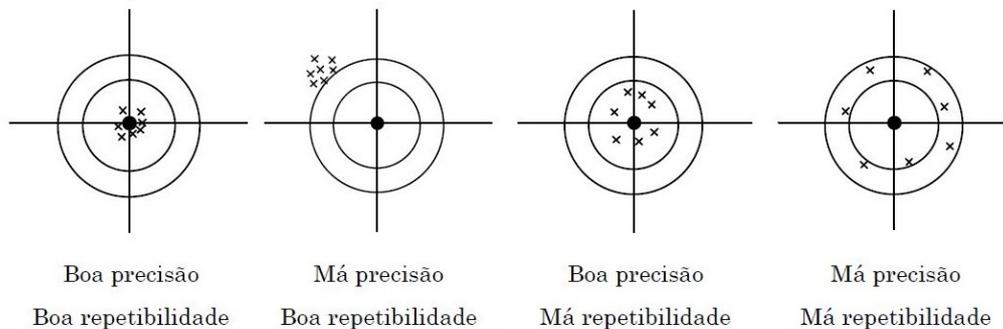


Figura 2.16: Comportamento de um manipulador com diferentes capacidades de precisão e repetibilidade [18].

2.4 Métodos de Programação

Sendo o robô industrial uma máquina projetada para exercer trabalho de forma independente, é necessário proceder à sua programação de forma a que este execute a tarefa que lhe será atribuída. A programação de um manipulador é o processo no qual se definem os pontos e trajetórias que este deverá executar durante o seu ciclo de trabalho. É uma função que consome bastante tempo e por isso é necessário o uso de ferramentas que permitam facilitar este processo. A programação de um robô industrial pode ser realizada através de dois métodos: programação *online* ou programação *offline* [11].

O método de programação *online* implica o uso do próprio robô na definição dos pontos e trajetórias que este tem de executar. Este método de programação pode ser executado através do uso da consola de programação (*teach pendant*)

ou aplicando um processo, designado por *lead-through programming*, que consiste no ensino da trajetória usando uma alavanca que pode ser fixada no manipulador [11].

De forma geral, o método de programação *online* através do uso do *teach pendant* é aplicado quando se pretende definir alguns pontos específicos no espaço de trabalho do manipulador, sendo um método comum para aplicações de *pick and place*. O *lead-through programming* é utilizado quando se pretende definir uma sequência de pontos ou uma trajetória complexa, comuns em aplicações de pintura.

No método de programação *offline* não é necessária a utilização do manipulador, sendo utilizada principalmente em aplicações complexas que exigem períodos de desenvolvimento dos programas longos [11]. Este tipo de programação pode ser realizado através do desenvolvimento de código recorrendo a linguagens de programação ou através de sistemas de simulação gráfica. As linguagens de programação utilizadas nos robôs são semelhantes às linguagens de programação mais conhecidas, como o C, ou seja, permitem definir sub-rotinas que são chamadas no programa principal, fazer a definição de constantes e variáveis e recorrer a instruções e ciclos, como *if-then-else*. Possuem também instruções para a definição do tipo de movimento (linear ou nas juntas), definição de velocidade, instruções de *offset*, que se podem aplicar a um ponto, entre outras. Quando a programação do robô é realizada recorrendo apenas à linguagem de programação é bastante difícil o programador conhecer exatamente as coordenadas dos pontos que têm de ser definidos no código. De forma a contornar este problema, o programador, numa primeira instância, desenvolve a estrutura base do código e depois, com o auxílio do manipulador, faz a edição dos pontos num processo conhecido como *touchup*.

Na programação *offline* com recurso a sistemas de simulação gráfica o utilizador é capaz de modelar toda a célula de trabalho e definir os pontos e movimentos que o manipulador deve executar. De seguida o código é carregado para o controlador e caso seja necessário fazer correções estas podem ser feitas na modelação gráfica da célula ou através do código gerado pelo *software*. O *software* de simulação pode ser usado também como uma plataforma de testes e verificações, caso o utilizador tenha escrito o programa recorrendo à linguagem de programação do robô.

Em suma, cabe ao utilizador decidir qual o método de programação que será implementado dependendo da sua preferência. Comparando os dois métodos, é possível referir que a programação *online* tem a vantagem de ser mais simples de aprender e implementar mas, em contrapartida, pode implicar a paragem da célula de trabalho, e conseqüentemente, a perda de produtividade visto que é necessário o uso do robô no processo de programação. A programação *offline* de

um manipulador é um processo mais complicado de se realizar, visto que é mais difícil determinar de forma precisa as coordenadas dos pontos da trajetória em relação ao sistema de coordenadas utilizado. Por outro lado não é necessária a utilização do robô e pode ser utilizada em paralelo com um sistema de simulação gráfica de forma a analisar e validar o programa final.

2.5 Conclusão

Assim, neste capítulo foram abordados os principais conceitos sobre a robótica industrial, desde a evolução histórica, os principais constituintes que formam um sistema robótico, as características de um manipulador, como os graus de liberdade e volume de trabalho e, por fim, os métodos de programação que podem ser utilizados. Todos estes conceitos auxiliam na interpretação do funcionamento de um sistema robótico e devem ser tidos em consideração nas fases de projeto e implementação de uma célula de trabalho com este tipo de sistemas.

Capítulo 3

Enquadramento Industrial

Terminado o estudo das noções gerais sobre sistemas robóticos que podem ser implementados a nível industrial, é importante referir como é que estes podem ser enquadrados num projeto de forma a que se obtenha uma célula de trabalho autónoma e segura.

Assim, neste capítulo serão abordados os controladores lógicos programáveis, do inglês, *Programmable Logic Controllers* (PLC), protocolos de comunicação que podem ser utilizados na gestão de uma célula de trabalho e que interligam os vários componentes como o PLC, manipulador e sensores, e por fim, mas não menos importante, as normas e regras de segurança que devem ser estabelecidas e cumpridas para que se evitem acidentes.

3.1 Programmable Logic Controllers

Habitualmente, num projeto de uma célula de trabalho composta, não só pelo sistema robótico, como também, por tapetes de transporte de materiais, máquinas que executam operações de maquinação, dispositivos de segurança, entre outros, a quantidade de sensores que auxiliam no controlo da célula é bastante elevada. Para tal, é introduzido na célula um dispositivo que fica encarregue de fazer o controlo de todos estes sensores, o PLC, que após fazer a leitura e análise dos seus estados dá ordens ao sistema através da atuação de saídas. Como exemplo, o PLC ao detetar o sinal de um sensor que esteja colocado no fim de um tapete transportador de caixas, dá ordem ao manipulador robótico para fazer o transporte da caixa para outro local da célula.

Desta forma, nesta secção serão abordados aspetos gerais sobre os PLC, desde a sua definição e constituição geral, até aos métodos e linguagens de programação utilizadas.

3.1.1 Definição Geral

Um PLC é um equipamento de controlo baseado em microprocessadores que utiliza memória programável para armazenar instruções e implementar funções como lógica, sequenciamento, temporização, contadores e operações aritméticas, com o objetivo de fazer o controlo de máquinas e processos [19].

O princípio de operação dos PLC é a aquisição de dados através de entradas, como sensores, e através das instruções do programa são atuadas saídas do sistema, como é possível verificar na Figura 3.1.

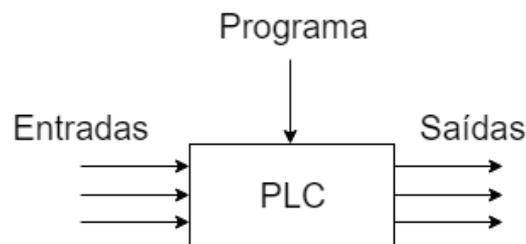


Figura 3.1: Princípio de funcionamento de um PLC.

Os PLC são equipamentos flexíveis visto que permitem realizar o controlo de uma vasta gama de sistemas com naturezas diferentes, bastando para tal alterar as instruções que definem a lógica de controlo do sistema. A nível de construção são equipamentos robustos, concebidos para suportar condições ambientais adversas como vibrações, temperatura, humidade e ruído, usualmente presentes em ambientes a nível industrial.

A nível de *hardware* este equipamento é constituído pela unidade central de processamento, do inglês, *Central Processing Unit* (CPU), fonte de alimentação, dispositivo de programação, unidade de memória e interfaces de entradas, saídas e comunicações [19]. Apesar da sua estrutura se ter mantido praticamente inalterada desde a sua introdução, características como a memória de programa e capacidades de processamento e desempenho foram sofrendo melhorias consoante a evolução tecnológica ao longo do tempo. Nos dias que correm o dispositivo mais comum para a programação de um PLC é o computador sendo que cada fabricante, normalmente, possui um *software* proprietário com as devidas funcionalidades de forma a facilitar este processo.

3.1.2 Tipos de PLC

Com o constante avanço tecnológico, o mercado sente a necessidade de estudar e desenvolver novas soluções de unidades de controlo lógico. Hoje em dia, o tempo de execução de um PLC e a sua flexibilidade são aspetos que condicionam a escolha de um determinado modelo em relação a outro.

Desta forma, atualmente existem modelos mais indicados para determinadas aplicações devido à flexibilidade de montagem, ao número de atuadores necessários a interligar ao sistema, memória e tempo de processamento que a unidade de controlo precisa para executar todo o seu programa. Portanto, a nível de mercado é possível diferenciar dois modelos de PLC: modulares e compactos [19].

Os modulares são, de forma geral, utilizados em soluções de maior dimensão e com um grau de complexidade mais elevado ou em projetos que podem sofrer alterações futuras. Os PLC modulares consistem em módulos independentes para a fonte de alimentação, processador, cartas I/O, cartas de rede, expansões de memória, entre outras. Os vários componentes são ligados entre si, de acordo com as necessidades do utilizador, o que permite obter um sistema com maior flexibilidade.

Relativamente aos modelos compactos, estes são usualmente utilizados em sistemas mais pequenos, em que toda a unidade de controlo é constituída pela fonte de alimentação, entradas e saídas digitais, memória e o processador. Na Figura 3.2 é possível observar um PLC modular e compacto da *Omron*.



Figura 3.2: PLC Omron modular da série CJ2M (esquerda) e PLC Omron compacto da série CP1E (direita) [20, 21].

3.1.3 Linguagens de Programação

De forma a normalizar o processo de programação dos controladores lógicos programáveis foi desenvolvida pela International Electrotechnical Commission (IEC) uma norma internacional, IEC 61131, que define as várias linguagens de programação. Estas podem ser divididas em dois grupos: linguagens gráficas e textuais.

Dentro do grupo das linguagens gráficas identificam-se os diagramas *ladder*, diagramas de blocos e os diagramas de funções sequenciais, respetivamente do inglês, *Ladder Diagram* (LAD), *Function Block Diagram* (FBD), *Sequential Function Charts* (SFC). Já no grupo das linguagens textuais podem ser identificadas as listas de instruções e o texto estruturado, do inglês, *Instruction List* (IL) e *Structured Text* (ST) [19, 22, 23].

Na Figura 3.3 é possível observar como se distribuem as diferentes linguagens de programação nos dois grupos referidos.

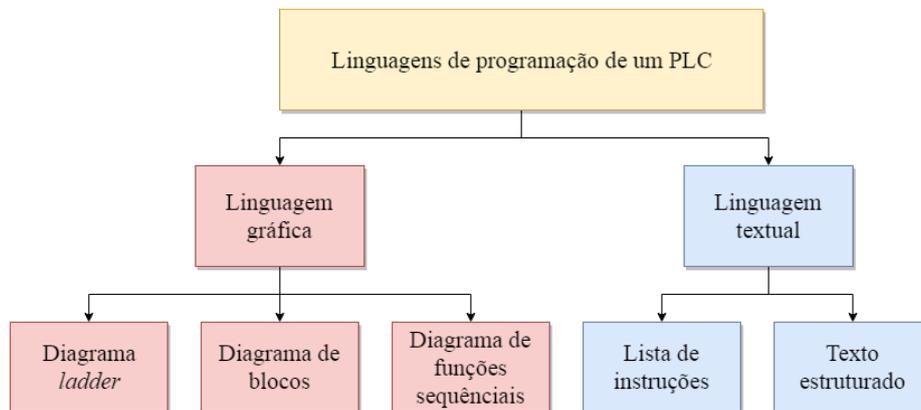


Figura 3.3: Linguagens de programação de PLC de acordo com a norma IEC 61131.

- *Ladder Diagram*

Método de programação de PLC muito comum, sendo que o princípio de escrita do programa se baseia na construção de circuitos com interruptores, normalmente designados na programação em *ladder* por contactos. São diagramas compostos por linhas verticais que representam as linhas de alimentação de um circuito tradicional, que se encontram interligadas por linhas horizontais, denominadas por *rungs*, onde estão representados os vários componentes do circuito [19, 23].

- *Function Block Diagrams*

Um FBD é um tipo de programa, na forma de um bloco que atua uma ou várias saídas, dependendo do estado das entradas e da lógica que o próprio implementa. Os blocos podem ter funções padrão, como operações lógicas, contadores ou temporizadores, ou funções definidas pelo utilizador, como um bloco para obter um valor médio [19].

É um tipo de linguagem que pode ser reutilizado, bastando alterar as variáveis de entrada e saída na nova chamada ao bloco. Como exemplo, para uma aplicação onde o utilizador deseje controlar da mesma forma vários tapetes, cada um com um sensor de entrada e um sensor de saída, pode recorrer ao mesmo FBD e nas várias instâncias de chamada do bloco pode apenas alterar os endereços das variáveis de entrada dos sensores e o endereço da variável de saída relativo a cada tapete.

- *Sequencial Function Charts*

A linguagem de programação SFC permite a representação gráfica de operações de um sistema de forma a demonstrar a sua sequência, composta pelas diversas etapas, transições e ações [19, 23].

O funcionamento do sistema é representado por um conjunto de símbolos. As etapas ou estados são representados por um retângulo, devidamente enumerado, e o estado inicial representado por um retângulo com linhas verticais duplas. A ligar cada estado existe uma linha horizontal, que representa as transições, ou seja, as condições que devem ser verificadas para o sistema transitar de um estado para o outro. Desta forma os estados devem estar sempre separados por uma transição. As saídas do sistema, são representadas por retângulos ligados na horizontal do estado correspondente.

- *Instruction List*

As listas de instruções são uma forma de programação de PLC onde o programa é escrito como uma série de instruções, sendo cada instrução escrita numa linha nova e compostas por um operador seguidas de um, ou vários, operandos [19, 23].

É um tipo de linguagem de baixo nível e mais adequada para programas pequenos e simples. Encontra-se em desuso, sendo que alguns dos fabricantes de PLC já não suportam este tipo de linguagem e optam apenas por uma linguagem de programação textual de mais alto nível, como é o caso do texto estruturado.

- *Structured Text*

Ao contrário das listas de instrução, o texto estruturado apresenta-se como uma linguagem de programação de mais alto nível, algo parecida com a linguagem Pascal. Os programas são escritos como uma série de declarações separadas por ponto e vírgula [19].

O texto estruturado é um tipo de linguagem de programação adequada para aplicações com um grau de complexidade e controlo mais elevados e apresenta vantagens em relação ao *ladder* no que toca à leitura e interpretação do programa. No estado de monitorização e *debug* apresenta mais limitações quando comparada com os diagramas *ladder* pois é uma linguagem do tipo textual e não gráfica.

Apresentadas as várias linguagens de programação de um PLC é possível concluir que todas apresentam as suas vantagens e desvantagens quando comparadas entre si. Em suma, cabe ao utilizador decidir aquela que pretende utilizar de acordo com a aplicação que pretende implementar e as suas preferências. A grande vantagem é que todas as linguagens fazem parte e respeitam a norma IEC 61311, pelo que podem ser utilizadas em conjunto no mesmo projeto. Ou seja, o utilizador para uma secção do programa que necessite de um controlo mais simples pode utilizar os diagramas em *ladder* e para uma secção de controlo do sistema mais complexa pode utilizar o texto estruturado.

3.2 Redes de Comunicação

O aparecimento e introdução de redes de comunicação foi um acontecimento que permitiu um desenvolvimento considerável no setor de automação. Nos dias de hoje, são bastantes os setores que usam redes de comunicação, não só na área de automação industrial, como: controlo de processos, automação residencial, distribuição de energia, sistemas aeronáuticos, entre outros. Desta forma, existem redes apropriadas para os requisitos de cada aplicação nas mais diversas áreas.

A nível industrial, mais propriamente no setor de automação, é normal existirem diferentes níveis de processamento de informação, cada um com diferentes necessidades. De forma geral, é possível identificar o nível da administração e o nível de produção. No nível da administração é preciso processar grandes volumes de informação, como por exemplo, relatórios de controlo, sendo que o tempo de resposta não é crítico podendo variar de alguns minutos, ou até horas. Já ao nível da produção, visto que se faz uso de máquinas e equipamentos que necessitam de trocar informação em tempo real, a velocidade de envio e receção é mais crítica, não podendo ocorrer falhas de comunicação. Desta forma, neste nível são utilizadas redes de campo, redes próprias para transferirem um volume pequeno de informação mas com uma velocidade bastante elevada (na ordem dos

micro-segundos). No nível de produção, os dispositivos que se interligam com a rede de campo são: máquinas de controlo numérico, manipuladores robóticos, controladores lógicos programáveis, sensores e atuadores.

3.2.1 Ethernet for Control Automation Technology - EtherCAT

A rede Ethernet for Control Automation Technology (EtherCAT) é uma rede de campo de alto desempenho baseada na rede Ethernet. A principal razão para o seu desenvolvimento foi a adoção do Ethernet para aplicações de automação, onde são necessários tempos de ciclo curtos e poucos problemas de comunicação.

É um tipo de rede baseado numa abordagem mestre/escravo com uma topologia em anel. Apenas permite um dispositivo mestre, característica que é adequada para conectar uma unidade de controlo, como um PLC, a periféricos como sensores ou atuadores. É o mestre que tem o controlo completo da rede e que inicia a comunicação sendo responsável por toda a troca de dados entre os escravos. Visto ser uma rede com topologia em anel, a trama é inicialmente enviada pelo mestre para o escravo que se encontra imediatamente no nó seguinte. O escravo processa a trama e de seguida envia-a para o escravo que o sucede. Este processo é repetido até a trama chegar ao último nó da rede e, neste ponto, a trama é enviada novamente para o dispositivo mestre [24, 25]. Na Figura 3.4 é possível observar a arquitetura de um sistema que utilize a rede de comunicação EtherCAT.

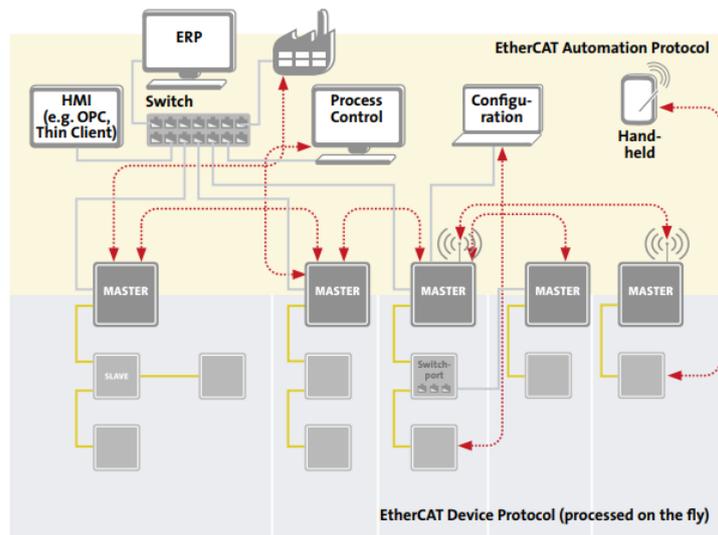


Figura 3.4: Arquitetura de um sistema com comunicação EtherCAT [25].

3.2.2 IO-Link

O aumento do uso de micro-controladores em sensores e atuadores possibilitou a adição de processos de diagnóstico e configuração desses mesmos dispositivos. Assim, surgiu a necessidade de criar uma rede capaz de fazer a troca dessa informação dos dispositivos com uma unidade de controlo como, por exemplo, um PLC, através de uma comunicação digital mantendo a compatibilidade com os tradicionais sinais digitais de entrada e saída.

Desta forma surgiu o IO-Link, que se baseia numa comunicação série ponto a ponto e bidirecional para transmissão de sinais e alimentação dos dispositivos. Assim, o IO-Link define uma interface para conectar sensores e atuadores a uma unidade mestre, unidade esta que pode servir como um ponto de ligação (*gateway*) entre nós remotos de entradas e saídas. Qualquer equipamento que seja compatível com o IO-Link pode ser conectado a qualquer porta do dispositivo *master*. No caso de sensores que necessitem de fazer conversão analógica-digital, esta é feita no próprio dispositivo que comunica o resultado da conversão num formato padrão digital. Desta forma, com a utilização do IO-Link não existe a necessidade de utilizar diferentes módulos de entradas e saídas digitais, do inglês, *Digital Input (DI)* e *Digital Output (DO)* bem como módulos de entradas e saídas analógicas, do inglês, *Analog Input (AI)* e *Analog Output (AO)* [24, 26]. Na Figura 3.5 é possível observar a arquitetura de um sistema que utilize o IO-Link como rede de comunicação entre os vários equipamentos

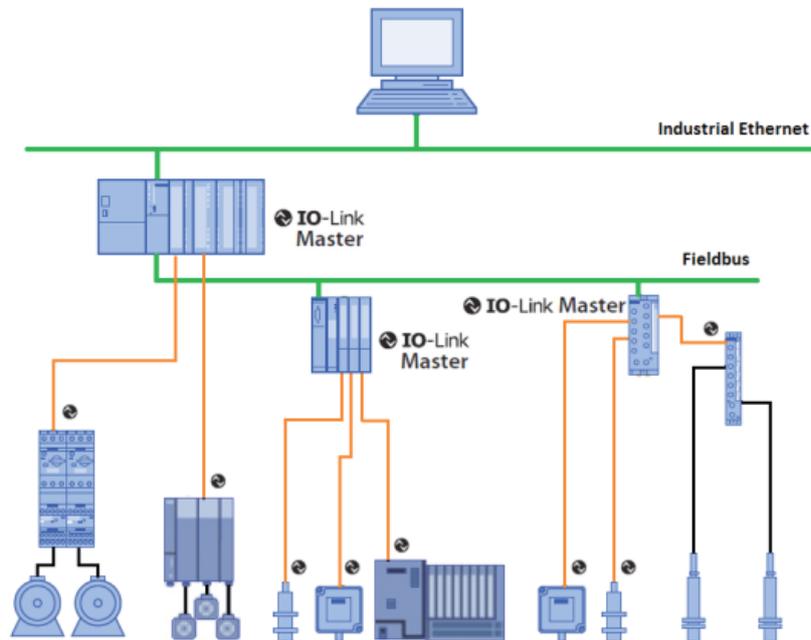


Figura 3.5: Arquitetura de um sistema com comunicação IO-Link [26].

No que toca às capacidades de diagnósticos, o IO-Link permite três níveis de diagnóstico: resposta imediata por necessidades de automação durante a fase de produção, resposta a médio prazo através da intervenção do operador e para comissionamento e manutenção através de informações de diagnóstico de longo prazo.

3.3 Segurança

Aquando do processo de projeto e implementação de uma célula de trabalho que recorra a sistemas de automação, como os manipuladores robóticos industriais, um dos aspetos a que se deve prestar especial atenção é a segurança. A incorreta implementação dos sistemas de segurança coloca em risco, não só o correto funcionamento da célula como, no pior dos casos, a vida de um operador que possa entrar em contacto com os equipamentos.

O desrespeito das normas e regras de segurança e a inadequada implementação destes sistemas podem originar perigos para os operadores como: impacto provocado pelo movimento do manipulador robótico ou atuador final a alta velocidade ou até mesmo o entalamento do operador em relação a outro equipamento que faça parte da célula. Algumas das causas que estão na origem destes acontecimentos são:

- entrada não autorizada de pessoas na área de trabalho;
- erros de programação e manutenção;
- falhas elétricas, hidráulicas ou pneumáticas;
- falhas mecânicas;

Assim, devem ser aplicadas medidas de proteção com o objetivo de diminuir a ocorrência destes acontecimentos e tornar o local de trabalho o mais seguro possível. Desta forma, é possível identificar as medidas de proteção ativas e passivas.

As medidas de proteção passivas podem ser definidas como o conjunto de regras de conduta a serem implementadas de forma a evitar a ocorrência de situações que possam originar um acidente, baseando-se no conhecimento adquirido pela experiência e regras de bom senso [12]. As medidas de proteção passivas englobam regras/ações como:

- formação dos operadores que poderão trabalhar com o equipamento;
- limitação do espaço de trabalho dos robôs;

- definição das zonas de segurança de zonas de trabalho;
- utilização de marcações e sinais que indicam as zonas de perigo, tendo sempre em consideração na definição destas zonas, o volume de trabalho do robô, tempos de atuação do sistema de paragem de emergência e o tempo que o robô demora até se imobilizar completamente;

Sempre que as medidas passivas não sejam capazes de evitar os acidentes, devem ser aplicadas as medidas ativas destinadas a limitar os efeitos ou consequências dos acidentes [12]. Dentro deste tipo de medidas é possível identificar:

- circuitos de paragem de emergência, onde se podem diferenciar dois tipos de paragem. Paragem de emergência havendo a necessidade de fazer o rearme do sistema para iniciar de novo a operação do robô e paragem de segurança na qual não é necessário o rearme do sistema;
- sinais de segurança do robô que executam a paragem do mesmo, como por exemplo, sinal de paragem (*hold*);
- sistemas para impedir o acesso à área de trabalho do robô como, vedações, barreiras luminosas, cortinas de luz, *scanners*, trincos magnéticos, entre outros, que devem ser ligados ao circuito de paragem de emergência do controlador do robô;
- sistemas de proteção contra colisões, ou seja, dotar o robô com um dispositivo de proteção contra colisões que deve ser ligado ao circuito de paragem de emergência.

Em suma, estas medidas não evitam por completo a possibilidade de ocorrência de um acidente, mas quando cumpridas com rigor ajudam a diminuir a probabilidade de ocorrência. Cabe, sobretudo, ao operador agir com o maior cuidado possível visto que os manipuladores são máquinas que podem causar perigo e que operam frequentemente com outras máquinas. Assim, o operador quando executa uma função perto de uma célula que contém um manipulador, nunca deve em nenhuma circunstância entrar no volume de trabalho do mesmo quando este tem a sua alimentação ligada e assumir que o robô não se irá movimentar.

3.4 Conclusão

Assim sendo, neste capítulo foram apresentadas tecnologias que podem ser adotadas e aplicadas numa célula de trabalho composta por um sistema robótico, como os controladores lógicos programáveis e as devidas linguagens de programação, bem como alguns protocolos de comunicação que podem ser estabelecidos

entre os vários equipamentos. No entanto, a interligação destas tecnologias com o sistema robótico só se apresenta como uma solução viável desde que todas as normas e regras de segurança sejam cumpridas.

Capítulo 4

Aplicações de Paletização

Paletização é um termo utilizado na indústria, que consiste em transportar e colocar determinados produtos sobre uma palete. Tem vindo a ser um processo automatizado, visto que pode envolver cargas elevadas para o ser humano, tornar-se repetitivo, fator que facilita a sua automatização e, devido ao aumento dos índices de produção, o ser humano apresenta dificuldades em realizar o processo com níveis de precisão e rapidez elevados. A decisão de recorrer ao uso de soluções de automação para melhorar o processo de paletização recai sobre vários aspetos que devem ser tidos em consideração, tais como: velocidade da linha, *layout* da fábrica, tamanho, peso e forma dos produtos, capacidade de formação de padrões/mosaicos durante o processo de produção, tempo de utilização do equipamento, custo do sistema, entre outros.

Assim, neste capítulo serão abordados os tipos de sistemas que podem ser implementados com o objetivo de melhorar o processo de paletização. No caso dos sistemas robóticos, serão abordadas as características mais importantes que um manipulador deve possuir para este tipo de aplicações, bem como as tecnologias que podem ser utilizadas no atuador final e os *softwares* de programação dedicados ao processo de paletização. Em relação aos sistemas de paletização convencionais, serão apresentadas soluções que podem ser implementadas e as respetivas diferenças. Por fim, referenciar alguns casos de estudo, onde se recorreu à implementação de uma célula com, pelo menos, um manipulador robótico para se efetuar a automatização do processo de paletização.

4.1 O Processo de Paletização

Como referido na introdução ao capítulo, o processo de paletizar consiste na colocação de produtos sobre uma palete, logo é necessário dispor dos equipamentos

necessários, formando um sistema capaz de executar o processo de paletização de forma automática. Analisando as opções existentes no mercado, é possível dividir os sistemas dedicados à paletização em dois grupos: os sistemas convencionais e as células robotizadas. Apesar de serem sistemas diferentes, a tarefa que executam é a mesma, logo os componentes e módulos que constituem este tipo de sistemas são similares como será abordado nas Seções 4.2 e 4.3.

Um dos aspectos a ter em consideração na escolha do tipo de sistema paletizador é a sua capacidade de formação de mosaicos. O mosaico que é formado pelo sistema é, talvez, o parâmetro que mais diferencia um sistema do outro, ou seja, na prática todos os sistemas de paletização, de uma forma ou de outra, recebem o produto a ser paletizado, a paleta e a camada de intercalar, no entanto, o mosaico formado pode ser diferente de sistema para sistema.

O número de mosaicos diferentes que podem ser construídos é teoricamente ilimitado, no entanto, existem padrões mais comuns na indústria devido, em grande parte dos casos, às medidas padrão das paletes que transportam o produto e à necessidade de aproveitar ao máximo a área da paleta com o objetivo de reduzir custos de armazenamento e transporte.

Partindo do princípio que os produtos colocados na paleta são da mesma forma e dimensão, de todos os mosaicos que podem ser formados, aqueles que se usam com mais frequência nestes casos são o entrelaçado e o sobreposto, ou em colunas, representados na Figura 4.1 [27].

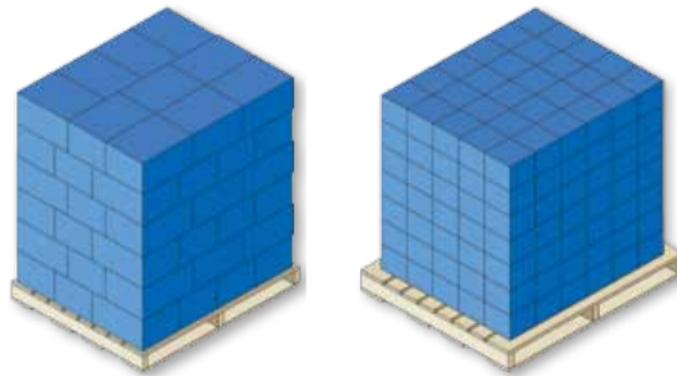


Figura 4.1: Exemplo de mosaico entrelaçado (esquerda) e em coluna (direita) [28].

Existe ainda a possibilidade de, na mesma paleta, serem colocados produtos diferentes. O setor da distribuição é um dos que realiza com mais frequência processos de paletização onde se colocam produtos mistos na mesma paleta, processo conhecido como *mixed palletizing*, que possibilita aos fornecedores construir uma paleta com produtos que mais se adequem a determinado cliente com o objetivo de limitar a quantidade de inventário nos armazéns [2]. De todos os mosaicos que podem ser construídos nestas circunstâncias, os tipos de paletes com cargas mistas mais comuns são: *display pallet*, *mixed layer pallet*, e por fim, *true mixed pallet*, representadas na Figura 4.2 [28].

No caso das *display pallets*, os produtos colocados têm as mesmas dimensões e quantidades, ou seja, uma paleta deste tipo pode ser composta por uma coluna do produto "A", uma coluna do produto "B" e assim sucessivamente. No que toca às *mixed layer pallets*, estas são semelhantes à referida anteriormente, diferenciando apenas a quantidade de produto colocado, ou seja, pode conter duas colunas do produto "A" e apenas uma coluna do produto "B". Por fim, as *true mixed pallets* são construídas quando os vários produtos são de dimensões totalmente diferentes e colocados na paleta em quantidades diferentes também. Este tipo de paleta permite ao fornecedor construir cargas de acordo com o cliente.

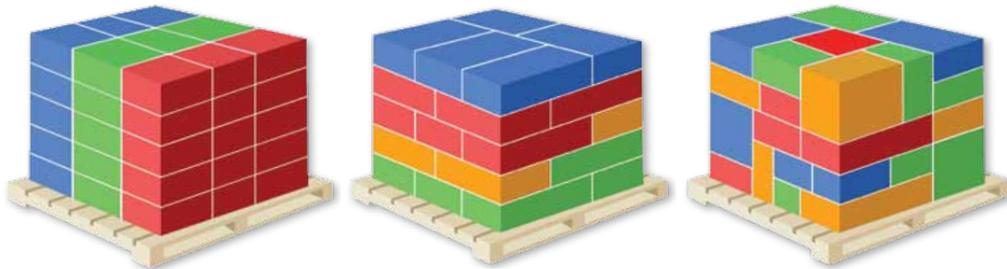


Figura 4.2: Exemplo de *display pallet* (esquerda), *mixed layer pallet* (centro) e *true mixed pallet* (direita) [28].

Dado o elevado número de combinações que podem ser feitas na construção de uma paleta, é importante que o sistema escolhido para realizar a operação de paletização seja versátil para que seja capaz de trabalhar com o maior número de mosaicos possível. Deste modo, nas secções que se seguem serão apresentados os vários tipos de sistemas paletizadores, convencionais e robóticos, mais comuns a nível industrial.

4.2 Sistemas de Paletização Convencionais

Como introduzido na Secção 4.1, um dos tipos de sistemas automáticos de paletização são os convencionais. Dentro deste tipo de sistemas é possível diferenciar os sistemas baixo nível (*floor level*) e os sistemas de alto nível (*high level*) [28], representados na Figura 4.3.



Figura 4.3: Exemplo de paletizador convencional *floor level* (esquerda) e *high level* (direita) [29, 30].

Comparando os dois tipos de sistemas, é possível observar que possuem estruturas diferentes mas, apesar desse fator, funcionam de forma semelhante. O produto chega ao sistema através de tapetes transportadores, que se encontram a alturas diferentes dependendo do tipo de sistema. À medida que estes passam pelo tapete, um dispositivo de rotação faz rodar o produto para a orientação correta até este se deslocar para a área de formação de fila. A fila formada é empurrada para a área de formação de camadas e, uma vez que esta esteja completa, é colocada sobre a palete. No caso dos sistemas *floor level*, a camada é baixada ou elevada sendo que a palete permanece estática durante todo o processo. Já no caso dos sistemas *high level*, é a palete que executa movimentos verticais de modo a que a camada seja colocada.

Este funcionamento apenas é possível graças aos vários módulos que fazem parte da constituição de um paletizador convencional. Apesar de estes sistemas apresentarem estruturas diferentes, os módulos que os constituem são semelhantes. Mesmo que ambos executem a paletização de determinado produto e apresentem módulos semelhantes, a sequência de trabalho destes pode ser ligeiramente diferente, dado a forma como os dois tipos de sistemas executam a tarefa de paletização. Na Figura 4.4 e na Tabela 4.1 é possível observar os vários módulos de trabalho que fazem parte da constituição de um sistema de paletização convencional.

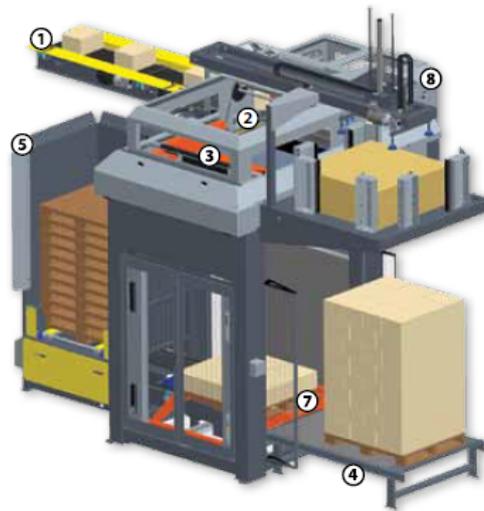


Figura 4.4: Componentes base de um sistema de paletização convencional [28].

Tabela 4.1: Componentes base de um sistema de paletização convencional

| Identificador | Componente |
|---------------|----------------------------------|
| 1 | Alimentador |
| 2 | Formador de filas |
| 3 | Área de formação de camada |
| 4 | Transportador de descarga |
| 5 | Dispensador de palete |
| 7 | Estação de palete |
| 8 | Dispensador de camada intercalar |

Em suma, e comparando os dois tipos de sistemas, é possível identificar características inerentes a cada tipo de paletizador, como por exemplo: os paletizadores *floor level* têm, de forma geral, um custo mais reduzido que os paletizadores *high level*, a manutenção nos sistemas *floor level* é mais fácil de realizar devido à facilidade de acesso aos vários componentes, um sistema *high level* oferece um rendimento maior em relação ao tamanho de área que ocupa, usando de forma mais eficiente o espaço da instalação e para aplicações de alta velocidade é mais adequada a utilização de um paletizador de alto nível, visto que oferece índices de produção mais elevados quando comparado com os de baixo nível. Desta forma, ambos os sistemas apresentam vantagens e desvantagens e cabe ao utilizador decidir aquele que melhor se adequa às suas necessidades.

4.3 Robótica e Paletização

Uma das alternativas aos sistemas convencionais de paletização, e que tem vindo a ganhar espaço na indústria, são os sistemas robóticos. Os manipuladores robóticos, ao contrário dos sistemas de paletização convencionais, utilizam um atuador final que permite ao robô pegar no produto vindo de um tapete transportador, por exemplo, e colocá-lo sobre a paleta, iniciando o processo de paletização. Dos vários tipos de configurações existentes de manipuladores robóticos, como aprofundado na Subsecção 2.2.1, as mais utilizadas em aplicações de paletização são a cartesiana, SCARA, em portico, e por fim, a articulada [28].

Apesar dos sistemas convencionais serem compostos por vários módulos, não é possível alterar a configuração geral da máquina dada a elevada interdependência dos vários periféricos. Tal não acontece nos sistemas robóticos, pois, desde que o manipulador tenha alcance suficiente para interagir com os devidos periféricos, a posição relativa dos vários componentes é indiferente para o robô, bastando para tal realizar a programação dos pontos. Esta característica dá aos paletizadores robotizados uma grande flexibilidade pois é possível configurar a célula de acordo com o espaço de instalação disponível. Ainda assim, e apesar desta maior flexibilidade de configuração oferecida pelos sistemas robóticos, os constituintes base de uma célula de paletização robotizada são semelhantes aos dos paletizadores convencionais, como se pode observar na Figura 4.5 e na Tabela 4.2.

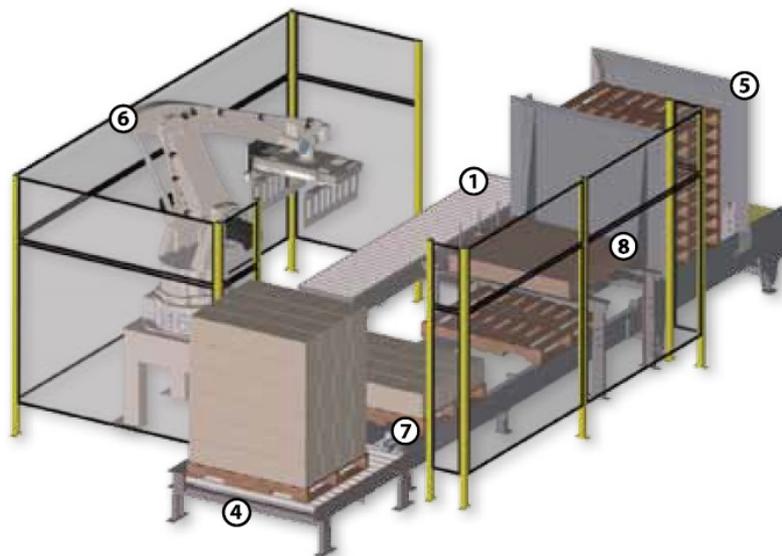


Figura 4.5: Componentes base dos sistemas de paletização robotizados [28].

Tabela 4.2: Componentes base de um sistema de paletização robotizado

| Identificador | Componente |
|---------------|----------------------------------|
| 1 | Alimentador |
| 4 | Transportador de descarga |
| 5 | Dispensador de palete |
| 6 | Manipulador Robótico |
| 7 | Estação de palete |
| 8 | Dispensador de camada intercalar |

Fora os periféricos base mencionados acima, os equipamentos de segurança, como trincos magnéticos, barreiras de segurança, entre outros, devem ser considerados na constituição do sistema para garantir a segurança dos operadores que circulam perto da área de trabalho da célula. Em alguns casos, onde poderá ser necessário um índice de produção mais elevado, é comum a utilização simultânea de dois manipuladores, um que pode ficar ao encargo de formar a camada e o segundo pela colocação da mesma em cima da palete.

Para além da constituição base dos sistemas de paletização robotizados, é importante referir quais as características que um manipulador deste tipo deve possuir a nível de *hardware*, bem como as tecnologias que podem ser utilizadas na ferramenta de manipulação. Já ao nível de *software* é relevante enunciar programas dedicados para este tipo de aplicações.

4.3.1 Manipuladores Robóticos de Paletização

Praticamente todos os fabricantes de manipuladores industriais dividem a sua vasta gama de produtos pelas diversas aplicações que estes podem executar, desta forma, disponibilizam ao consumidor vários tipos de manipuladores que divergem na sua configuração e especificações, de acordo com a aplicação que estes deverão executar. Assim, um manipulador pensado para executar, por exemplo, uma tarefa de pintura irá possuir características, como capacidade de carga e volume de trabalho, totalmente diferentes quando comparado com um manipulador mais apropriado para tarefas de paletização. Apesar de existirem robôs com configurações diferentes que são capazes de executar tarefas de paletização, no âmbito desta dissertação serão abordados com mais detalhe os manipuladores com configuração articulada.

Os manipuladores de paletização, de forma geral, apresentam uma configuração mais simples, possuindo normalmente 4 eixos, ao contrário dos robôs de pintura que possuem 6 eixos. Esta característica deve-se ao facto dos movimen-

tos e trajetórias inerentes a uma aplicação de paletização serem mais simples, como se verifica na Figura 4.6. Deste modo, o robô apenas necessita de, praticamente, rodar o eixo mais próximo da base (eixo 1), subir e descer a carga (eixo 2 e 3) e orientar o atuador final (eixo 4). Visto que a maior parte dos manipuladores de paletização possuem apenas 4 eixos faz com que apresentem algumas limitações de movimentos, quando comparados com os de 6 eixos, no entanto, possuem um volume de trabalho grande em relação à área de trabalho que ocupam.

Em relação à capacidade de carga são modelos que apresentam uma grande variedade de valores, existindo no mercado manipuladores para efeitos de paletização com capacidades de carga que podem variar entre os 50 kg, como por exemplo o modelo da FANUC M-710iC/50H, até aos 1300 kg como o modelo KUKA KR 1000 1300 Titan PA. São também capazes de trabalhar a velocidades elevadas, o que permite um nível alto de produção, têm uma repetibilidade na ordem das décimas de milímetro e um alcance horizontal que pode variar entre os 2 metros e 3,5 metros.

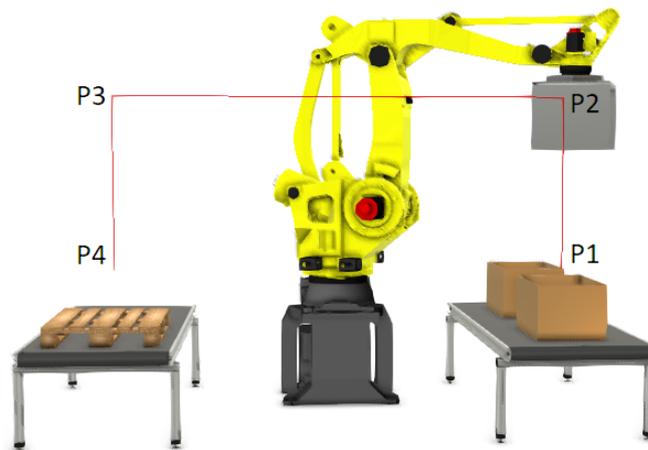


Figura 4.6: Trajetória efetuada por um manipulador no processo de paletização.

4.3.2 Atuadores Finais para Aplicações de Paletização

O atuador final, ou garra, é o componente que mais diferencia a flexibilidade de um sistema robótico. A tecnologia utilizada para realizar a manipulação do produto é um fator determinante, e que deve ser tida em grande consideração, para que o sistema seja o mais versátil possível. Alterando a ferramenta, ou utilizando uma tecnologia apropriada para lidar com vários tipos de produtos,

um manipulador pode ser capaz de realizar o processo de paletização a caixas de cartão, sacos ou qualquer outro tipo de embalagem.

É comum projetar um atuador final que utilize várias tecnologias de manipulação, permitindo ao robô manipular diversos tipos de objetos, sendo que em alguns casos este é responsável pela manipulação da paleta, do produto e da folha separadora de camadas. Devido à elevada variedade de produtos que podem ser utilizados no processo de paletização, existem vários tipos de atuadores finais, que dependendo das suas características, são mais indicados para manipular determinados produtos. Em alguns casos mais particulares, é necessário projetar e desenhar a garra especificamente para a aplicação; no entanto, a norma é criar tipos de garras padrão. Dento deste grupo destacam-se as garras com as seguintes tecnologias: grampo (*clamp*), garfo (*fork*), garra (*finger* ou *claw*) e de vácuo [28].

As garras do tipo *clamp* são constituídas por um, ou vários, painéis laterais que fecham e prendem o objeto, como representado na Figura 4.7. Podem possuir um sistema de controlo de curso nos painéis de forma a ser possível adaptar a pega a produtos com diferentes dimensões. É uma ferramenta de uso geral, capaz de manipular vários produtos quando estes se encontram na mesma orientação; no entanto, o atrito criado pelos painéis pode não ser suficiente para segurar o produto e originar a sua queda.

As ferramentas do tipo *fork*, representadas na Figura 4.7, têm a particularidade de possuir um sistema em forma de dedos que fornecem um suporte na parte inferior do produto, eliminando o risco de queda do mesmo. Para além deste sistema, possuem também uma prensa capaz de manter a forma de objetos menos rígidos durante o transporte. Devido à forma como é feita a pega do produto, este tipo de garra exige a utilização de um equipamento de transporte de produto que seja compatível com a forma da garra, ou seja, nestes casos ao invés de se utilizar um transportador de tela será mais indicado um transportador de rolos.

Já os atuadores finais do tipo *claw* ou *finger*, representados na Figura 4.7, possuem uma estrutura semelhante às do tipo *fork*. A grande diferença é a presença de dois sistemas em forma de dedos, ao invés de apenas um. É um tipo de ferramenta mais adequado para o transporte de sacos e tal como as do tipo *fork*, necessitam de um transportador adequado que permita realizar a pega do produto

Por último, os atuadores do tipo vácuo, representados na Figura 4.7, são constituídos por um conjunto de ventosas. No processo de pega é atuado o vácuo para fazer a pega do produto pela sua face superior e aquando o momento de largar o produto na paleta o vácuo é desligado. Apresentam a desvantagem de não serem eficazes em superfícies irregulares ou com furos. No entanto, possuem um sistema de controlo mais simples quando comparada com os outros tipos

de ferramentas e, em alguns casos, quando o vácuo é suficientemente forte, são utilizadas para fazer a manipulação do produto, como por exemplo caixas, e da própria paleta.

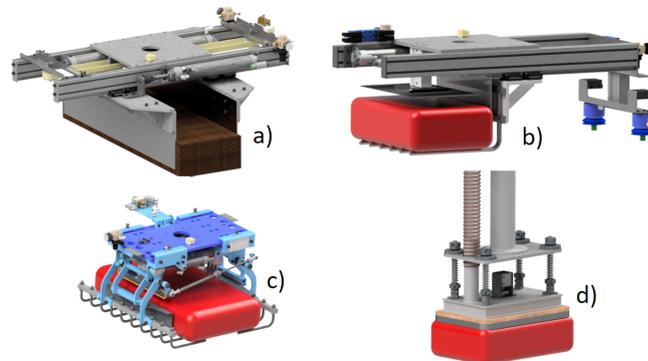


Figura 4.7: Tipos de atuadores finais mais comuns em aplicações de paletização: a) *clamp*, b) *fork*, c) *finger* ou *claw*, d) vácuo [31].

Com o objetivo de tornar o sistema robótico mais flexível é comum usar uma combinação de vários tipos de tecnologias no mesmo atuador final. Como se pode observar na Figura 4.8, o atuador final à esquerda utiliza dois sistemas do tipo *claw*, um para manipular o produto e outro para o transporte da paleta. Já a ferramenta da direita utiliza um sistema do tipo *clamp* para pegar no produto e um sistema vácuo para colocar a folha intercalar de camadas. Assim, apenas com uma ferramenta o robô é capaz de manipular o produto, a paleta e, nos casos onde se aplicam, a folha intercalar de camadas.



Figura 4.8: Atuadores finais com dois tipos de sistemas de manipulação [31].

Apesar dos sistemas apresentados serem os mais comuns e padronizados na indústria, em alguns casos, onde a variedade de produtos que entram na linha de paletização é elevada, poderá ser necessário adotar um sistema menos comum e que necessite de ser desenhado e projetado para a aplicação em específico. Na Figura 4.9 encontra-se representado um atuador final da Brenton Engineering com

um funcionamento menos comum na indústria. O sistema tipo cortina permite ao atuador suportar toda a camada do produto pela parte inferior. De forma a pegar no produto, o manipulador aproxima-se da camada, e em simultâneo, empurra a camada para a ponta do transportador e baixa o sistema em cortina. Caso a área de trabalho da célula seja de dimensões mais reduzidas e não permita fazer este movimento lateral do produto para a ponta do transportador, é possível utilizar um transportador com um sistema elevatório que possibilite realizar a pega do produto.



Figura 4.9: Atuador final da Brenton LLC com sistema em cortina [32].

De todos os tipos de atuadores finais disponíveis, cabe sempre ao utilizar analisar e definir aquele que mais se adequa ao seu tipo de aplicação, tendo sempre em conta o tipo de produto que cada atuador final consegue manipular e o tipo de controlo necessário. Na maior parte dos casos o acionamento é feito de forma pneumática mas, em algumas aplicações, pode ser necessário um controlo mais preciso do curso da ferramenta, podendo recorrer-se à implementação de eixos elétricos.

4.3.3 Softwares Dedicados à Paletização

Tal como foi abordado na Secção 2.4, na programação *offline* de um manipulador é comum utilizar *softwares* de simulação no decorrer do processo de programação. Os fabricantes, geralmente, disponibilizam ao consumidor uma versão base do *software* que possui funções mais generalizadas para todo o tipo de aplicações. Contudo, visto que algumas aplicações como pintura, paletização ou soldadura, são mais comuns na indústria, os fabricantes optam por oferecer pacotes de ex-

pansão ao programa base, conhecidos como *add-ons*, ou até mesmo um programa independente dedicado a este tipo de aplicações, com funções mais específicas, que agilizam o processo de programação do manipulador. Na Tabela 4.3 encontram-se referenciados os programas/pacotes de expansão dedicados à programação de aplicações de paletização de vários fabricantes. Neste sentido, no decorrer desta secção serão abordadas as funcionalidades e vantagens/desvantagens destes sistemas de programação *offline*.

Tabela 4.3: Programas dedicados à programação de robôs para aplicações de paletização

| Fabricante | <i>Software</i> |
|-------------------|------------------------|
| ABB | Palletizing PowerPac |
| FANUC | PalletPRO PalletTool |
| Kawasaki Robotics | K-SPARC |
| KUKA | FlexPal |
| Yaskawa Motoman | PalletSolver |

Apesar de existirem vários *softwares* dedicados à programação de manipuladores para aplicações de paletização, como indica a Tabela 4.3, as funcionalidades e o método de funcionamento destes são semelhantes. Este aspeto deve-se ao facto dos parâmetros e requisitos necessários e inerentes ao processo de paletização serem sempre os mesmos e independentes do *software*, ou seja, quer esteja o utilizador a trabalhar com o *software* da ABB ou com o *software* da KUKA, em ambos terá de fornecer, por exemplo, como variáveis de entrada o *layout* do mosaico que o manipulador terá de formar ou as camadas onde será introduzida a folha intercalar.

Analisando o caso da FANUC, visto ser do mesmo fabricante que o robô utilizado no âmbito desta dissertação, o fabricante disponibiliza no mercado o seu *software* base e mais generalizado para várias aplicações, o Roboguide. Para aplicações de paletização existem expansões que podem ser adquiridas, o PalletPro e o PalletTool [33].

O PalletPro é o *software* de simulação que pode ser utilizado para criar e testar uma aplicação de paletização *offline*. Permite ao utilizador criar um modelo em três dimensões da célula, indicar a localização das estações de alimentação de produto, tipo de produto a ser manipulado, mosaico a ser construído, localização da camada intercalar, entre outros. Após a conclusão da modelação da célula no PalletPro, toda a informação gerada pode ser descarregada para o controlador do robô, que deve possuir o PalletTool. Este é responsável por analisar toda a infor-

mação recebida e gerar o código do manipulador. Possui ainda funcionalidades que facilitam a produção, como comando de fim de produção de lote, que permite ao PLC enviar dados de mudança de produto para o controlador do robô, e funcionalidades de recuperação na ocorrência de erros, como por exemplo, a queda de um produto [33]. Na Figura 4.10 é possível observar a janela de configuração dos parâmetros que devem ser introduzidos no PalletPro, como as dimensões do produto a paletizar, dimensões da paleta, entre outros.

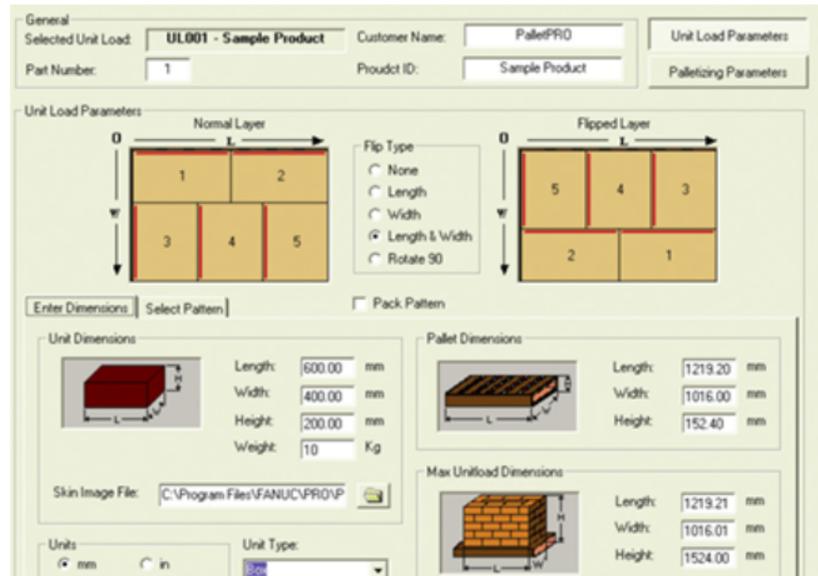


Figura 4.10: Janela de configuração de parâmetros no *software* PalletPro [34].

Assim, este tipo de sistemas de programação e simulação dedicados a um tipo de aplicação possui vários benefícios. Não só os *softwares* da FANUC, como os dos restantes fabricantes, permitem validar processos de paletização num cenário virtual, sem que seja necessária a utilização do robô ou de outros periféricos, como o atuador final ou os tapetes transportadores de produto. Permitem testar e validar vários cenários através da verificação de alcances, possibilidade de colisões e índices de produção. Dadas as funcionalidades que estes sistemas apresentam para um tipo de aplicação em concreto, o tempo, custo de integração e programação da célula diminuem.

4.4 Casos de Estudo

Por todo o mundo existem inúmeros casos de empresas que decidem inovar os seus métodos de produção, que utilizam tecnologias de automação tradicionais, por células de trabalho compostas por sistemas robóticos. Com isto, as empresas pretendem melhorar os seus processos de paletização, tornando-os mais rápidos, o que se traduz em maiores quantidades de produtos produzidos, e com índices de qualidade mais elevados.

Desta forma, nesta secção serão analisados alguns casos de empresas que implementaram um sistema robótico nos seus processos de paletização e abordados aspetos como o problema que incentivou a empresa a recorrer a uma solução desta natureza e os requisitos que o sistema teve de cumprir até aos resultados obtidos e benefícios alcançados.

4.4.1 Forbes Chocolate

A empresa Forbes Chocolate é uma fornecedora de cacau e aromas utilizados na indústria alimentar, em produtos como o leite, gelado, iogurte, entre outros. Fundada em 1901, a empresa tem sede em Cleveland, Ohio, e faz parte da família Geringer [35].

A opção de instalar um manipulador robótico deveu-se ao facto das instalações possuírem espaço limitado disponível para a automatização do processo de paletização, tal como foi dito por Keith Geringer, vice presidente da Forbes. *"Our operation is definitely space-challenged. The space we had available ruled out just about everything but robotic palletizers. A robot was the better fit."* [36]. Os objetivos da Forbes eram, sobretudo, melhorar o rendimento de produção, eliminando o engarrafamento do sistema de paletização manual e aliviar os funcionários de uma tarefa tediosa e pesada.

O sistema consiste num robô FANUC M-410iB de quatro eixos e capacidade para 160 kg, um tapete de entrada de produto, um tapete de armazenamento de paletes, uma garra capaz de manusear sacos e o painel de *interface* da FANUC, iPendant. O iPendant permite ao operador introduzir os parâmetros de paletização, como por exemplo, a altura da carga, e visualizar informações de diagnóstico. Para além do iPendant, existe também um painel com um botão utilizado para informar o sistema que os componentes estão prontos e este pode trabalhar em modo automático. A garra é mecânica e capaz de três movimentos de aperto. Dois movimentos são utilizados para apertar o saco durante a tarefa de *pick* e o terceiro movimento para evitar qualquer mobilidade do saco durante a transferência [35]. Na Figura 4.11 é possível observar o sistema implementado a executar a paletização dos sacos e a garra adotada para realizar a operação.

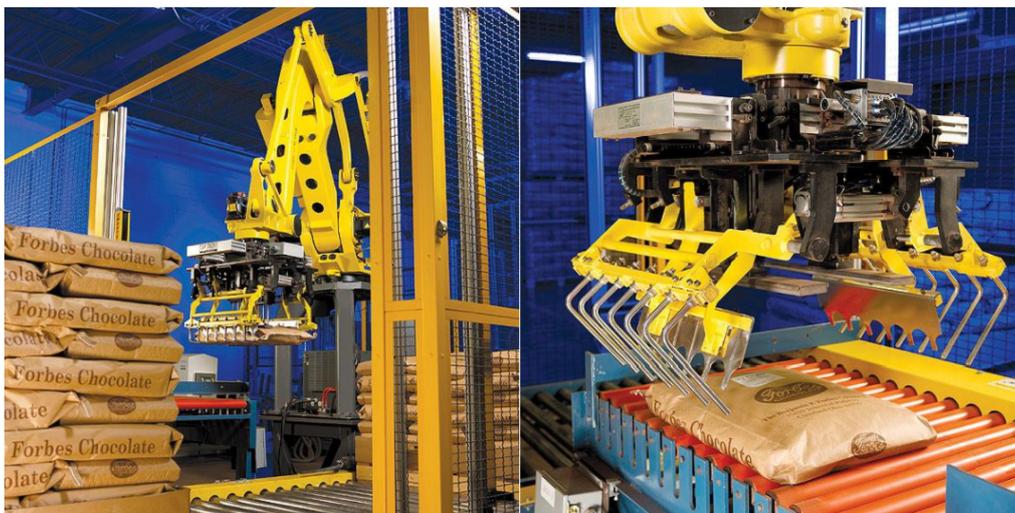


Figura 4.11: Sistema implementado na empresa Forbes Chocolate (esquerda) e garra mecânica utilizada no manipulador (direita) [36].

Como requisitos, o sistema teria de ser capaz de manipular quatro tamanhos e estilos diferentes de sacos, podendo o peso variar entre os 7 kg e os 27 kg. As características dos sacos variam dependendo da densidade das misturas usadas e do volume de enchimento dos mesmos. O sistema teria também de realizar a paletização do produto em dois tamanhos diferentes de paletes e para diferentes alturas de carga, parâmetros que teriam de ser pré-especificados com base nos requisitos dos cliente. Devido a estes requisitos, a ferramenta de manipulação teria de ser concebida de forma a lidar corretamente com os vários tipos de sacos e o programa do robô permitir flexibilidade para ajustar facilmente as posições de paletização de acordo com o tipo de saco [35, 36].

O funcionamento começa com o operador a posicionar a paleta vazia na extremidade do tapete. Estando a paleta preparada, o operador carrega no botão que se encontra no painel para indicar ao robô que esta se encontra pronta. O controlador do robô, R-J3iB, ao receber o sinal do botão, dá ordem ao tapete para transportar a paleta para a zona de paletização. Os sacos, após passarem por um processo de etiquetagem, chegam à entrada do tapete de alimentação da célula onde são pegados pelo manipulador, iniciando assim o processo de paletização. Quando o conjunto estiver completo, o controlador informa o tapete que transporta a carga para outra zona, onde será posteriormente removida [35].

Este processo é possível graças à utilização de dois *softwares* da FANUC. O primeiro é o *software* Pallet Pro que consiste num pacote de simulação, onde é simulado o funcionamento do sistema e os padrões de paletes que serão utilizados. O segundo *software* é o Pallet Tool, que corre no controlador do robô, respon-

sável por calibrar e controlar as funções do iPendant, bem como gerar o código baseando-se nas informações geradas pelo Pallet Pro [36].

Com o sistema implementado e a funcionar da forma esperada, a Forbes Chocolate reconhece que os objetivos principais de aumentar a produção e aliviar os trabalhadores foram atingidos. O aumento de produção deve-se, principalmente, ao facto do manipulador conseguir construir pilhas mais altas, em alguns casos 2 m, quando comparadas com as que eram feitas no processo de paletização manual. Este aspeto permitiu também à Forbes aumentar a capacidade do armazém e reduzir os custos de envio como afirma Geringer: *“Now we can stack higher, so we use fewer pallets. Even more important, we save on shipping costs, since a lot of our shipments go at a pallet rate, regardless of whether the pallet holds 500 or 2 pounds.”* [35, 36].

4.4.2 Y&S Candies

A Y&S Candies empresa de doces sediada em Lancaster, PA, adquirida pela Hershey Company em 1977, em Abril de 2009 decidiu atualizar as suas quatro linhas de paletização de caixas, que já tinham 20 anos.

Resolveram então substituir os robôs antigos que tinham instalados nas suas instalações por novos modelos com características dedicadas a aplicações de paletização. Embora o projeto seja apenas a substituição direta dos robôs mais antigos por novos modelos, e mesmo com apenas duas das quatro células totalmente substituídas, Alex Diaz, engenheiro da Hershey, explica como a atualização teve grandes impactos no processo de paletização: *“We’ve seen an increase in speed of as much as 20 percent, jumping from 8 cases/min. to 10 cases/min.”* [37]. Alex refere ainda que os manipuladores instalados, da marca Yaskawa Motoman, modelo EPL 160 de quatro eixos, executam os seus movimentos de forma mais rápida, com uma aceleração superior, o que permite pegar e soltar o produto de forma mais rápida, diminuindo assim o tempo de ciclo. Menciona ainda que os incidentes devido à queda de caixas diminuíram graças à nova garra de manipulação instalada que permite uma pega mais segura para os produtos mais pesados.

As células da Y&S Candies, como representado na Figura 4.12, recebem as caixas através de tapetes transportadores, que se encontram interligados a um tapete principal que recebe os produtos após o processo de empacotamento. As caixas que o manipulador terá de paletizar terão dimensões bastante diferentes, podendo o seu peso chegar até aos 28 kg. Dos 149 produtos diferentes, 125 são empilhados com um padrão em coluna e 24 têm a sua camada superior rodada de 180 graus. Todos recebem uma folha de intercalar na base de cada palete e 17 uma camada intermédia para evitar que o produto deslize [37].



Figura 4.12: Célula de trabalho instalada na Y&S Candies [37].

Existem 52 padrões de paletização diferentes baseados nas exigências dos clientes e, portanto, o sistema tem de estar preparado para estas situações. Para além do novo modelo da Motoman, foi ainda integrado no sistema um novo *software* de simulação de paletização que permite criar um novo padrão em 10 minutos. Além disto, durante o processo de produção, o operador pode alterar o padrão de paletização para outro através da interface homem-máquina, do inglês *Human-Machine Interface - HMI*, em menos de 1 minuto. Por outro lado, o *software* tem a capacidade de calcular a trajetória que cada robô tem de executar, com base num algoritmo pré-definido aplicado a cada padrão de paleta a ser produzida. O programa encontra-se alojado num computador central da empresa que permite o controlo de todos os robôs da Y&S e indica quais os trabalhos a serem executados por cada robô na presença de um padrão diferente de paleta [37].

Através do *software* é possível alterar rapidamente o tamanho da caixa e da paleta, conforme especificado pelo cliente, para um padrão de paleta pré-existente, permitindo gerar de forma rápida e automática o novo programa. Sempre que o manipulador inicia a construção de um novo padrão descarrega o respetivo programa, que contém as dimensões das caixas, instruções para a rotação das camadas e localização das folhas de intercalar, do computador central. O sistema é capaz de suportar 30 camadas e aproximadamente 100 caixas/camada [37].

Em cada manipulador encontra-se instalada uma garra, com princípio de funcionamento a vácuo, que permite pegar nos diversos tipos de caixas com facili-

dade. As áreas de vácuo são ativadas com base na forma e localização das caixas, conforme programado no *software*. Segundo Alex Diaz, o facto da garra ser composta por uma placa de espuma faz com que esta consiga selar as caixas de forma mais eficaz do que a garra de ventosas anteriormente utilizada. Conclui ainda que, graças à nova garra, os robôs agora são capazes de manipular as caixas mais pesadas de uma forma mais eficaz sem terem de diminuir a sua velocidade: *“That allows the robots to more reliably handle the bigger and heavier cases without having to slow down, as they previously were required to do.”* [37].

4.4.3 Voit

O setor alimentar não é o único que recorre à implementação de células robóticas para automatizar os seus processos de paletização e a empresa Voit é um desses exemplos. Sediada na Alemanha e fundada em 1919, a empresa Voit insere-se no ramo da carpintaria e especializa-se no fabrico de mobiliário e acessórios de interior de alta qualidade. Com 50 funcionários, a empresa é fornecedora de mobiliário para escritório e salas a uma vasta gama de fabricantes alemães na indústria automóvel [38].

De forma a agilizar o seu processo de produção, a empresa adquiriu uma máquina para fazer a colocação do acabamento lateral das placas e pretendia colocar apenas um funcionário responsável por configurar, operar e alimentar a máquina com as placas. À saída da máquina o objetivo seria colocar um manipulador robótico responsável pelo processo de paletização das mesmas. Assim, a Voit adicionou um manipulador robótico ao seu processo de produção, o modelo KR 180 R3200 PA da KUKA, com uma capacidade de carga de 180 kg e um alcance de 3,2 m. Sendo assim, o processo de produção inicia-se com a introdução das placas, por parte do operador, na máquina responsável por fresar as placas e colar o acabamento lateral. Após este processo as placas são transportadas com o auxílio de um tapete de rolos e empilhadas pelo robô, que utiliza uma garra com mecanismo de vácuo [38]. Na Figura 4.13 é possível observar o manipulador a realizar a paletização das placas de madeira.

O programa utilizado controla o sistema com precisão para que seja possível manipular de forma correta os diferentes tamanhos das placas e para que o robô consiga paletizar de acordo com os padrões de camadas pré-definidos. Uma vez que a pilha se encontre terminada é movida para o exterior da célula, através de um sistema de tapetes. Segundo Roland Fisher, proprietário da Fischertech empresa responsável pela implementação do sistema na Voit, o programa instalado é capaz de ler parâmetros introduzidos pelo cliente como comprimento, largura ou até espessura. No caso da Voit estes parâmetros irão corresponder às medidas das placas a serem paletizadas: *“All of our customers receive a custom-written program that merely contains certain parameters, such as length, width and thick-*

ness ... All the operator has to be able to do is modify these parameters; no other prior knowledge is required.” [38].



Figura 4.13: Manipulador KUKA a paletizar as placas de madeira [38].

Para além do manipulador e respetiva garra, foi também instalado um sistema de segurança que deteta se a pilha se está a mover corretamente para fora da célula e se algum operador tenta entrar na célula. Desta forma, em caso de perigo o sistema de segurança é capaz de automaticamente parar o manipulador.

Com a implementação do sistema a empresa foi capaz de melhorar o seu índice de produtividade, tal como afirma Stefan Voit diretor da empresa: *“Today, with the robot, we are able to stack 2,400 individual parts per shift, compared with 40 percent less in the past”*. Stefan afirma ainda que a implementação do sistema foi menos complicada e mais rápida do que o esperado e que existe a possibilidade de instalar um segundo robô para introduzir as placas na máquina ao invés do operador [38].

4.5 Conclusão

Em suma, após a apresentação dos sistemas robóticos e das tecnologias que podem ser utilizadas para interligar os vários equipamentos, neste capítulo foram abordados aspetos importantes sobre o processo de paletização. Com o objetivo de automatizar este processo podem ser utilizados os paletizadores convencionais ou os manipuladores robóticos. No caso dos paletizadores convencionais é possível fazer a distinção entre dois tipos, os *floor level* e os *high level*. No que toca aos manipuladores robóticos foram abordados os vários tipos de atuadores finais e os *softwares* dedicados à programação de manipuladores paletizadores.

Capítulo 5

Célula de Paletização Robotizada

O objetivo do presente trabalho é a implementação de uma célula robotizada para efetuar o processo de paletização de caixas. A necessidade de implementação de uma célula deste género surge pois, nos dias de hoje, o mercado é bastante competitivo e o constante melhoramento dos índices de produção é uma das maiores preocupações dos fabricantes. Desta forma, foi necessário recolher toda a informação necessária de modo a projetar uma célula com um manipulador robótico e restantes equipamentos, que cumpra os requisitos necessários e que seja o mais flexível possível.

A célula é constituída por diversos equipamentos, sendo os principais: manipulador robótico FANUC, atuador final, transportadores de produto, mesas de formação de mosaico, armazéns de palete e de folha intercalar, sistema de transporte automático (*charriot*), consola de interface com o operador e todos os dispositivos de segurança.

Assim, este capítulo tem como propósito identificar quais as necessidades do sistema, abordando aspetos como a definição do problema e análise de requisitos, configuração (*layout*) da célula, lógica de trabalho e a arquitetura do sistema onde serão apresentados os vários equipamentos de trabalho e respetivo funcionamento.

5.1 Definição do Problema

A RobotSol, empresa dedicada ao desenvolvimento de soluções de engenharia industrial, recebeu uma proposta de trabalho baseada na resolução de um problema para uma empresa que atua no sector da alimentação. O cliente em questão possui duas linhas de produção de pão congelado onde a paletização das caixas com produto que saem das linhas de produção é feita de forma manual. Desta forma, o desafio apresentado é a criação e implementação de uma célula capaz de exe-

cutar o processo de paletização e o transporte da paleta completa para a entrada de uma envolvedora já em funcionamento nas instalações do cliente.

A primeira etapa passa pela definição do problema, onde será feito o levantamento de todos os requisitos que devem ser cumpridos para que a célula seja capaz de realizar todos os processos propostos e definidos. Para além da paletização de caixas, o sistema terá de ser capaz de construir o mosaico, transportar paletes vazias para os respetivos armazéns e mover a paleta completa para um ponto de saída.

Sendo o foco do trabalho o processo de paletização, a primeira variável que deve ser analisada é o tipo de produto que o robô irá manipular. Assim sendo, foram apresentados dois modelos de caixas com tamanhos diferentes. Os modelos, diferenciados como Caixa Grande (CG) e Caixa Pequena (CP), apresentam ambas duas variantes no que toca ao tamanho, obtendo-se um total de quatro caixas com dimensões diferentes. Assim, dentro do modelo CG tem-se a variante com dimensão de $600 \times 400 \times 265$ mm e outra com $600 \times 400 \times 226$ mm. No que toca ao modelo CP, existe a variante de $400 \times 300 \times 265$ mm e a de $400 \times 300 \times 210$ mm. Apesar do comprimento e largura serem os mesmos entre as variantes dos dois modelos, a altura varia o que impõe a colocação de um número de camadas de caixas por paleta diferente. A paleta utilizada segue as normas da *European Pallet Association* (EPAL) de dimensões $1200 \times 800 \times 144$ mm. Desta forma, analisando os tamanhos das diversas caixas e o espaço disponível na paleta, será necessário construir dois mosaicos diferentes, como representado na Figura 5.1.

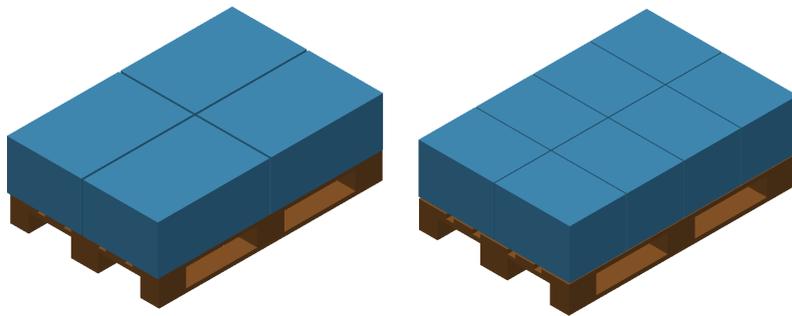


Figura 5.1: Mosaicos definidos para a célula de paletização

Relativamente à composição da célula, foram identificados vários requisitos que ditaram a sua configuração final: a existência de duas linhas de produção, o espaço disponível, a necessidade de transportar paletes vazias para dentro da célula, transportar a paleta finalizada para a entrada de uma envolvedora, que se encontra afastada do local reservado para a instalação da célula, e o facto de existir uma área entre a célula e a envolvedora onde pode ocorrer a passagem de operadores.

O facto de existirem duas linhas de produção distintas obriga a que a célula tenha duas entradas de produto de forma a receber as caixas provenientes das duas linhas, que saem a uma cadência de aproximadamente dez caixas por minuto. No que toca ao espaço disponível para a instalação da célula, verificou-se que se dispunha de uma área com dimensões de aproximadamente 7×6 m e altura de 4 m. A existência de uma envolvente já instalada obrigou a que fosse implementado um sistema que realizasse o transporte da paleta completa (*charriot*) para o transportador de entrada da envolvente e que os módulos da célula fossem instalados à mesma altura deste mesmo transportador. Relativamente à área onde pode ocorrer a passagem de operadores, teve de ser realizada uma análise mais aprofundada pois esta interceta o percurso que o *charriot* tem de fazer, o que cria riscos que devem ser tidos em conta da definição da segurança da célula. Assim, a configuração adotada para a célula de paletização encontra-se representada na Figura 5.2.

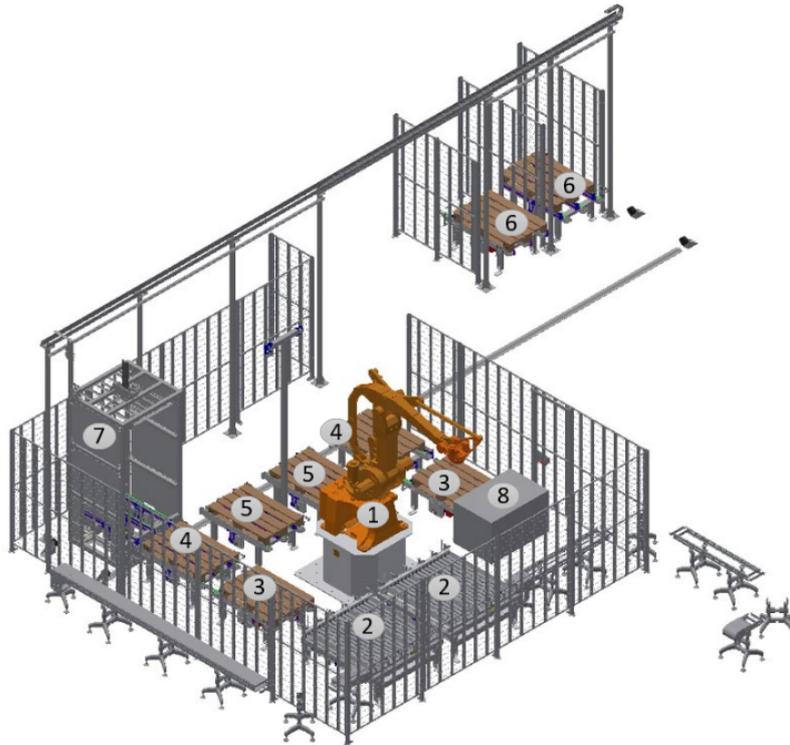


Figura 5.2: Configuração da célula implementada: 1) robô, 2) mesas formadoras de mosaico, 3) zona de paletização, 4) zona de transferência, 5) armazém de paletes, 6) zona de alimentação de paletes, 7) *charriot*, 8) armazéns de cartão intercalar.

Analisando a Figura 5.2, é possível identificar os vários módulos que constituem a célula e a sua configuração. Dada a entrada de produto ser realizada por dois locais distintos e em simultâneo, foi necessário equipar o sistema de forma a que permita ao robô manipular as caixas provenientes das duas linhas sem causar engarrafamento. Assim, o núcleo da célula é composto pelo manipulador e respetivo atuador final, duas mesas de rolos que realizam a formação dos mosaicos, seis transportadores de correntes, um armazém de folha intercalar e dois transportadores de tela que alimentam a célula com caixas. Fora do núcleo da célula encontram-se dois transportadores de correntes, onde são colocadas as paletes para alimentar a célula, e a envolvente. O *charriot* é o sistema encarregue da interligação dos equipamentos colocados no interior do núcleo com os equipamentos instalados no exterior.

5.2 Funcionamento da Célula

Enunciados os vários componentes do sistema, a segunda etapa é expor a sequência de trabalho definida. Para este sistema foram definidos três modos de funcionamento distintos: automático, semi-automático e manual.

No modo automático todos os módulos estão ativos e a célula é capaz de operar sem a intervenção humana. O modo semi-automático é usado no caso de avaria do *charriot*, permitindo que o núcleo da célula continue a executar trabalho sem que esteja dependente do sistema de transporte. O modo manual pode ser utilizado para realizar uma produção mais específica que não justifique colocar em funcionamento todos os componentes da célula. Neste modo o manipulador não exerce trabalho e todos os comandos devem ser dados pelo operador através do HMI.

Apesar do sistema possuir o modo semi-automático e manual, estes serão utilizados de forma menos recorrente pois foram pensados para se usarem em situações mais excecionais ou de avaria. O modo automático é aquele que aproveita todas as capacidades de célula, sendo o seu funcionamento explicado de seguida.

Antes de se iniciar o ciclo de trabalho, o operador deve realizar alguns passos de preparação. O primeiro é colocar uma pilha de paletes vazias em cada um dos transportadores de correntes localizados no exterior do núcleo da célula e alimentar o armazém de intercalar. De seguida, o operador deve, através do HMI, indicar as referências das caixas, por cada linha de produção, que irão alimentar a célula, de modo a que o robô saiba que mosaico manipular e que as mesas de formação saibam que mosaico formar. Através da consola devem também ser indicadas as camadas da pilha onde se pretende colocar a folha intercalar (*slip sheet*). Por fim, o operador deve certificar-se do estado das emergências, devendo fazer o rearme no caso de alguma estar ativa. Após estes passos iniciais, todas as condições de arranque do sistema estão verificadas e o ciclo de trabalho inicia-se

após ser dada a ordem, através do botão de *start*, que se encontra no painel de interface homem-máquina.

A unidade de controlo (PLC), após receber a ordem de início de ciclo, realiza as inicializações dos vários módulos e verifica se o sistema se encontra com as condições de arranque adequadas. Em caso afirmativo, o sistema entra no modo de funcionamento automático e inicia o seu ciclo de trabalho. Neste estado, até que o manipulador inicie o processo de paletização várias tarefas acontecem em simultâneo. Assim sendo, as caixas começam a circular pelos transportadores até chegarem às mesas formadoras, que formam o mosaico pretendido e indicado pelo operador. Em paralelo, o *charriot* inicia o transporte das pilhas de paletes vazias, colocadas nos tapetes exteriores da célula, para os respetivos armazéns dentro da célula.

Assim que seja dado o sinal ao manipulador que, pelo menos uma, das pilhas de paletes se encontra no local e devidamente indexada, este manipula uma das paletes da pilha para a zona de paletização. Após a paleta ser colocada nesta zona e a mesa indicar que o mosaico se encontra terminado, o robô pega no mosaico e coloca-o sobre a paleta. Terminada a colocação da camada, a unidade de controlo verifica se é necessário colocar a *slip sheet* e, em caso afirmativo, dá ordem ao manipulador para tal. Este processo é repetido para as duas linhas até se obter uma paleta completa. Logo após a paleta estar completa, é iniciada a transferência da mesma para os transportadores de espera. Desta forma, o robô é capaz de iniciar novo processo de paletização sem que haja a necessidade de retirar de imediato a paleta terminada do interior da célula. Encontrando-se a paleta terminada nos transportadores de espera, o *charriot* encarrega-se do transporte da mesma para a entrada da envolvente. Todo este processo é repetido até que haja uma ordem de fim de produção ou atuação de alguma emergência.

A diferença do modo automático para o semi-automático encontra-se apenas na lógica de alimentação da pilha de paletes vazias e na forma de descarga da paleta completa, visto que o *charriot* se encontra desativado. Por esta razão, o operador deve colocar a pilha de paletes vazias diretamente nos transportadores de armazenamento do interior da célula e fazer a descarga da paleta diretamente dos transportadores de espera. De forma a proteger o operador, sempre que este entre na área reservada à célula todos os movimentos são parados e deve ser feito o rearme do sistema para que seja retomado o ciclo de trabalho.

5.3 Arquitetura do Sistema

Dada a lógica de funcionamento e a sequência de trabalho da célula explanadas na Secção 5.2, é possível compreender que todos os módulos existentes executam funções em paralelo e são todos dependentes uns dos outros. No caso do manipulador, este só executa uma rotina de trabalho se, por exemplo, a mesa formadora

de mosaicos indicar que terminou a sua tarefa e que existe uma paleta indexada na zona de paletização.

Esta interdependência dos módulos não acontece só com o manipulador mas com todos os componentes da célula, ou seja, os equipamentos só executam uma tarefa quando recebem *feedback* dos restantes módulos de que as condições para iniciar a realização de uma tarefa foram cumpridas. Por este motivo, todos os componentes da célula são controlados por uma unidade de controlo (PLC) e a arquitetura do sistema encontra-se representada na Figura 5.3.

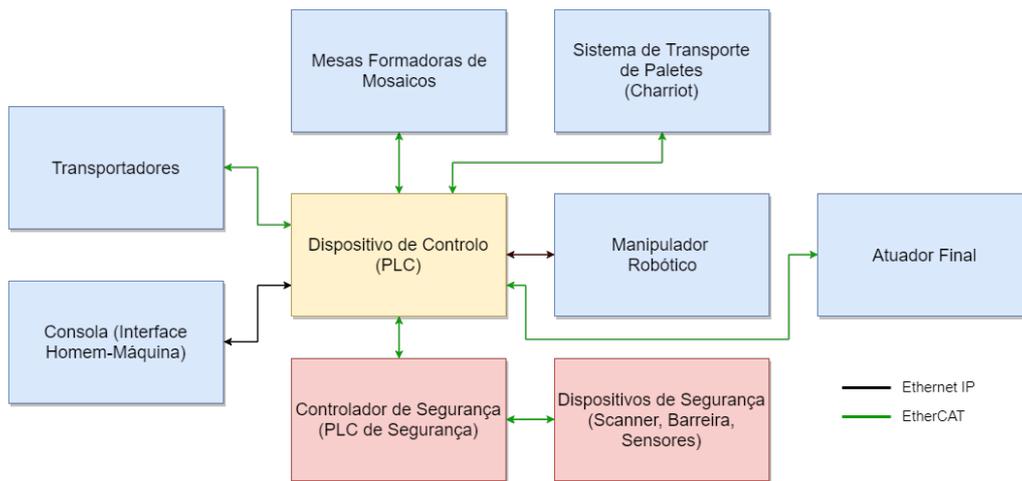


Figura 5.3: Arquitetura da célula implementada

5.3.1 Unidade de Controlo

Face aos requisitos deste trabalho, a unidade de controlo escolhida foi um PLC Omron, modelo NX1P2-9024DT1, representado na Figura 5.4. Encontra-se na gama média de controladores disponibilizados pela Omron e compreende os requisitos exigidos por esta aplicação.

De características gerais, possui uma capacidade de programa de 1,5 MB, 24 pontos de entradas e saídas (do inglês, *Inputs/Outputs* - I/O), possibilidade de utilização dos protocolos Ethernet e EtherCAT, capacidade de controlo de motores e servo-motores e uma vasta gama de cartas que permitem expandir as capacidades do controlador. Dentro do largo leque de cartas de expansão podem ser salientadas as de entradas e saídas digitais, módulos opcionais de comunicação série, cartas que permitem a utilização de sensores com interface IO-Link, possibilidade de adição de cabeceiras EtherCAT que possibilitam a expansão da rede do controlador, cartas para controlo de servo-motores, cartas de entradas e saídas de segurança, entre outras. A programação deste controlador pode ser feita através da linguagem *ladder* ou ST, através do *software* proprietário da Omron,

Sysmac Studio, que conta com uma biblioteca alargada de funções que auxiliam na sua programação [39, 40].



Figura 5.4: PLC industrial Omron NX1P2-9024DT1 [40]

À unidade de controlo foram adicionadas várias cartas de expansão. Para a leitura de entradas digitais, como por exemplo sensores, foram utilizadas cartas NX-ID5442 que possuem 16 pontos de entradas digitais. Já para a atuação de saídas, como por exemplo o comando de avanço de um inversor, foram utilizadas as cartas NX-OD5256 que possuem também 16 pontos de saídas digitais. Para os sensores com interface IO-Link é necessário uma carta específica, sendo a escolhida o modelo NX-ILM400 que permite a ligação de quatro sensores distintos. Dada a limitação da unidade de controlo NX1P2 de apenas permitir 8 cartas de I/O no seu circuito principal, foram adicionadas cabeceiras EtherCAT NX-ECC203 de modo a aumentar a flexibilidade e expansibilidade do sistema. Estas cabeceiras possuem a capacidade de ligar 63 unidades de I/O e o seu uso é necessário quando se pretende implementar um controlador de segurança, que será apresentado na Subsecção 5.3.7 [41]. As cabeceiras NX-ECC203, para além da conexão de entrada EtherCAT, possuem também uma conexão de saída permitindo ligar outras cabeceiras presentes na rede. Contudo, se um dos nós da rede EtherCAT falhar ou entrar em erros, todos os nós seguintes deixam de receber informação e ficam inoperáveis. Por esta razão, foi utilizado um *switch* EtherCAT, modelo GX-JC06, que possui 6 portas EtherCAT. Desta forma, no caso de um dos nós falhar os outros não são afetados e continuam no modo de operação normal.

5.3.2 Manipulador e Atuador Final

A escolha do manipulador robótico é uma tarefa que deve ser feita após analisar os vários requisitos que este terá de cumprir. Para o sistema em questão, cada caixa que entra na célula tem um peso aproximado de 10 kg. Sendo que um

dos mosaicos leva oito caixas, obtém-se um peso máximo por mosaico de 80 kg. Sendo que o peso do atuador final é de aproximadamente 200 kg, a soma total de carga que o robô deve ser capaz de manipular ronda os 280 kg. Assim sendo, o manipulador escolhido foi o M410iC/315 do fabricante FANUC.

O manipulador de 4 eixos, representado na Figura 5.5, classificado como um paletizador, possui como características gerais uma capacidade de carga de 315 kg, um alcance horizontal de 3143 mm e uma repetibilidade de aproximadamente 0,5 mm [42]. Para este modelo, o fabricante disponibiliza duas variantes, com ou sem pedestal incorporado. A opção de escolha foi a versão com pedestal de forma a aumentar o alcance vertical do manipulador. Em relação ao controlador, para robôs de maior dimensão como é o utilizado nesta aplicação, o painel frontal que contém a botoneira de emergência e o seletor de modo não vem anexado ao controlador, permitindo instalar este painel fora do alcance do robô sendo que o controlador fica alojado dentro do pedestal. Quanto à comunicação entre o robô e o PLC, esta será feita através do protocolo EthernetIP e a sua implementação e configuração será abordada no capítulo seguinte.



Figura 5.5: Manipulador robótico industrial FANUC M410iC/315 (esquerda) e respetivo controlador R-30iB e consola de programação iPendant (direita) [42, 43].

Apesar das características do manipulador irem de encontro às necessidades do sistema, estas só serão aproveitadas com o devido atuador final. Para esta célula, o robô terá de ser capaz de manipular os dois mosaicos diferentes, mencionados na Subsecção 5.1, a palete e a folha intercalar. Assim sendo, o atuador final desenvolvido para esta aplicação encontra-se representado na Figura 5.6.

Como se pode observar, é um atuador final mecânico composto por três sistemas de manipulação. O primeiro é composto por oito painéis que irão exercer pressão nas caixas e permitir que o manipulador pegue nelas. No caso do mosaico composto por oito caixas, cada painel irá exercer contacto apenas em uma caixa.

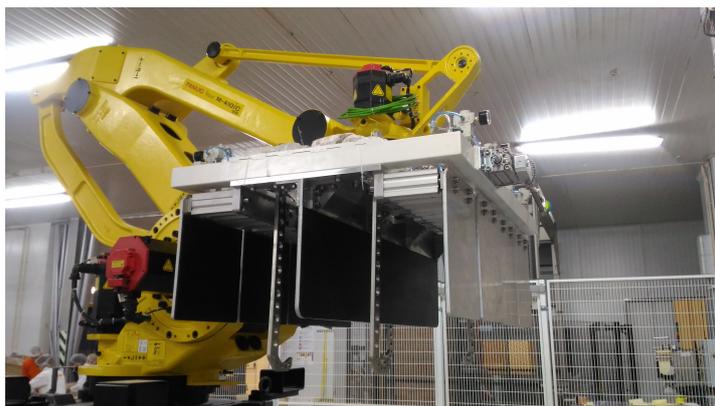


Figura 5.6: Atuador final implementado

Já no mosaico com as quatro caixas maiores, serão dois painéis a exercer contacto por caixa. O segundo sistema, é um conjunto de garras em forma de "garfo" que permitem a manipulação da paleta. Por fim, o terceiro sistema é composto por duas ventosas para a manipulação do cartão intercalar de camadas.

O controlo dos vários sistemas de manipulação do atuador final pode ser realizado de forma independente, ou seja, a cada componente móvel está associado um atuador pneumático que será responsável por exercer movimento. Assim sendo, o atuador final é composto por doze atuadores pneumáticos: oito para os painéis laterais, dois para o sistema de garfos e mais dois para o sistema de ventosas. O controlo da atuação dos cilindros é feito por eletroválvulas.

Para os painéis laterais foram utilizadas oito eletroválvulas, SY5400-5U1 da SMC, uma para cada painel, representadas na Figura 5.7. Estas eletroválvulas são bi-estáveis, com três posições, ou seja, o cilindro é capaz de manter uma posição intermédia. Esta característica é importante pois as caixas possuem dimensões diferentes e o curso que o cilindro terá de realizar será também diferente para cada tipo de caixa. Quanto ao sistema de garfos para a manipulação de paletes e para o sistema de ventosas, foram utilizadas duas eletroválvulas, SY5200-5U1, uma para cada sistema. São também bi-estáveis mas possuem apenas duas posições, ou seja, o cilindro só é capaz de se manter nas posições de totalmente avançado ou recuado. Por fim, tem-se uma eletroválvula SY5A00-5U1 para a ativação e desativação do vácuo que irá agarrar na folha intercalar [44, 45].

Para a perceção da posição e movimento de cada cilindro foram colocados sensores D-M9PSAPC da SMC, ilustrados na Figura 5.7, que verificam se este se encontra na posição recuada ou avançada, obtendo-se um total de vinte e quatro sensores. Como já enunciado, o curso de fecho que o sistema de manipulação de caixas deve fazer varia consoante o tipo de caixa. Por esta razão, os sensores de

posição dos cilindros não são suficientes para que se garanta que o atuador final agarrou corretamente as caixas. Desta forma, foram introduzidos oito pressóstatos digitais ISE30A-01-F-G da SMC, ilustrados na Figura 5.7, que atuam uma saída digital quando a câmara de ar do cilindro se encontra sobre determinada pressão, que pode ser programada no pressóstato, indicando que o painel entrou em contacto com a caixa.

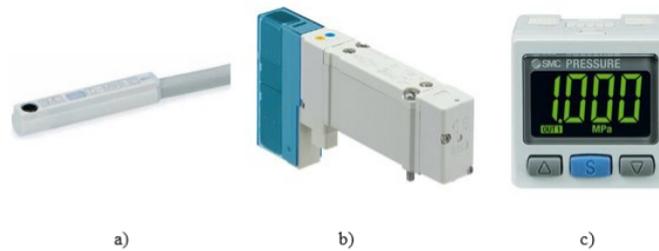


Figura 5.7: Componentes pneumáticos utilizados no atuador final: a) sensor de posição de cilindro, b) eletroválvula, c) pressóstato digital [44, 45, 46].

Dado o elevado número de sensores, eletroválvulas e atuadores pneumáticos necessários para realizar o controlo do atuador final, este será feito pelo PLC e não pelo controlador do robô. Desta forma, e com o objetivo de tornar o sistema mais compacto, optou-se pela utilização de um sistema modular do fabricante SMC. O sistema, representado na Figura 5.8, é composto por uma unidade de controlo EX600-SEC1 com capacidade de comunicação EtherCAT, por quatro cartas de entradas digitais EX600-DXPC, sendo que cada carta possui oito entradas, e pelas onze eletroválvulas referidas anteriormente. Com esta solução obtém-se um sistema de controlo para o atuador final mais compacto visto que todos os sensores e pressóstatos são reunidos no módulo da SMC, sendo apenas necessário passar um cabo de comunicação EtherCAT para fazer a comunicação com o PLC.

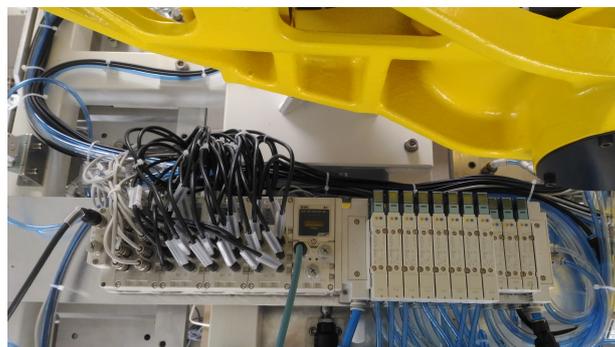


Figura 5.8: Sistema de controlo do atuador final

5.3.3 Interface Homem-Máquina

Com o objetivo do operador conseguir monitorizar e controlar o sistema, foi implementado um painel de interface homem-máquina. Este painel, representado na Figura 5.9, é composto por uma consola programável da Omron, modelo NB10W-TW01B, um seletor com três estados para se seleccionar o modo de funcionamento, uma botoneira de emergência e os botões *Start*, *Stop* e *Reset*.



Figura 5.9: Painel de controlo

O botão de *Start*, como o próprio nome indica, servirá para o operador dar ordem de início ao sistema. O botão de *Stop* terá duas funcionalidades: a primeira será colocar o sistema em estado de pausa, a segunda será realizar uma paragem forçada do sistema sem que seja necessário ativar uma emergência. Para o primeiro caso o operador apenas necessita de fazer um clique simples no botão, já para realizar a paragem forçada deve manter o botão premido durante cinco segundos. Por fim, o botão de *Reset* servirá para rearmar o sistema em caso de emergência.

A consola selecionada é de 10 polegadas, tem uma resolução de ecrã de 800×480 , permite fazer o armazenamento de dados para um dispositivo *Universal Serial Bus* (USB) e pode ser conectada ao PLC através do protocolo Ethernet. A nível de programação, esta é feita no *software* NB-Designer, que permite a implementação de funcionalidades como criação de receitas de produção, representação de gráficos, níveis de segurança de utilizadores que restringem o acesso a ecrãs escolhidos pelo programador, listagem de alarmes, entre outros [47, 48].

Será através da consola que o operador poderá controlar e monitorizar a célula. O operador será capaz de, por exemplo, verificar os índices de produção, observar os estados de todos os componentes, realizar a seleção de produção, averiguar a

lista dos alarmes ocorridos e fazer o controlo manual de todos os componentes para efeitos de testes ou manutenção.

5.3.4 Transportadores

Como referido ao longo deste capítulo, a célula é constituída por oito tapetes de correntes, ilustrados na Figura 5.10. A utilização de tapetes de correntes, ao invés de transportadores de tela ou rolos, incidiu sobre o facto de estes permitirem uma melhor tração à superfície da paleta. Todos os tapetes possuem estrutura e tamanho igual. A única diferença é que nos transportadores da zona de paletização e de armazenamento de paletes, estes possuem batentes mecânicos para evitar a queda da paleta na sua indexação.

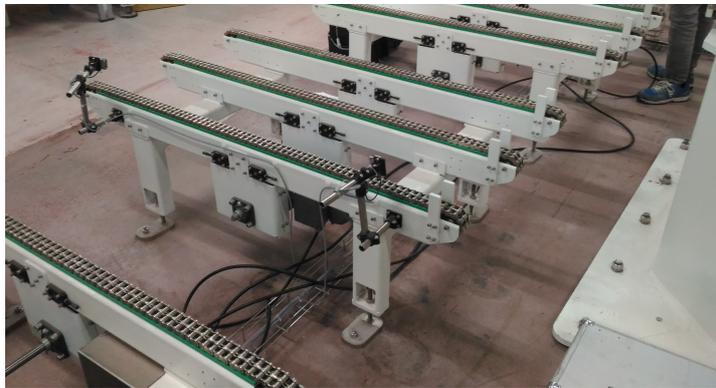


Figura 5.10: Tapetes de correntes implementados

O tapete é movido através de um motor de corrente alternada, do inglês *Alternating Current* (AC), que por sua vez é controlado por um inversor, também conhecido como variador de frequência, da Omron, modelo 3G3MX2-AB004-V1. A grande vantagem de utilizar um inversor é que este é capaz de variar a sua frequência de saída de acordo com a velocidade pretendida de rotação do motor [49].

Quanto à sensorização dos tapetes para fazer o controlo de funcionamento dos mesmos, em cada tapete de correntes foram instalados dois sensores fotoelétricos de reflexão no objeto. Um sensor para detetar a entrada/indexação da paleta e outro para detetar que esta se encontra na saída do tapete. O sensor escolhido foi o ML100-8-H-350-RT/103/115a da Pepperl Fuchs, representado na Figura 5.11. É um sensor fotoelétrico de reflexão no objeto com um intervalo de deteção entre os 5 e os 350 mm [50]. Nos tapetes de armazenamento da pilha de paletes, para além destes dois sensores, foi instalado um sensor laser de medição de distância VDM28-8-L1-IO/73c/110/122 da Pepperl Fuchs, representado na Figura 5.11,

cuja função é indicar ao robô a que altura se encontra a pilha de paletes. Este sensor possui um intervalo de medição compreendido entre os 0,2 e 8 m, uma repetibilidade inferior a 5 mm, uma saída digital, uma saída analógica e ainda interface IO-Link [51].



Figura 5.11: Sensor de reflexão no objeto (esquerda) e sensor de medição de distância (direita) [50, 51].

Devido ao elevado número de motores e cabos de potência nas imediações destes sensores, decidiu-se utilizar sensores com interface IO-Link. A vantagem de sensores IO-Link é que a conversão analógica-digital é realizada no próprio sensor e não numa carta analógica do PLC. Assim o PLC recebe logo um valor digital e desta forma reduz-se o risco da leitura ser errada devido a ruídos eletromagnéticos que possam ser induzidos pela comutação dos inversores e pelos motores.

5.3.5 Mesas Formadoras de Mosaicos

Dado a existência de duas linhas de produção que alimentam a célula, foram implementadas duas mesas de rolos para formação do mosaico, representadas na Figura 5.12. O objetivo da implementação destes componentes é que o robô não fique ocupado a criar o mosaico, permitindo uma maior fluidez no ciclo de trabalho da célula, diminuindo a probabilidade de engarrafamento da mesma.

Para que seja possível formar os mosaicos desejados, cada mesa é composta por dois empurradores, três *stoppers* e por vários sensores que detetam o posicionamento das caixas. Os empurradores têm como principal função empurrar a caixa que entra na mesa para a posição correta. Os três *stoppers* têm cada um a sua funcionalidade. O primeiro faz a gestão da entrada de caixas para a mesa, ou seja, só se forem verificadas as devidas condições é que este *stopper* recua e deixa entrar uma caixa para a zona de formação de mosaico. O segundo tem como objetivo fazer a separação das filas do mosaico que contêm oito caixas, para que o atuador final consiga manipular o mosaico. Por fim, o terceiro *stopper* serve



Figura 5.12: Mesas de formação de mosaico implementadas

como batente, não permitindo que as caixas caiam da mesa. Com a mesma filosofia que foi utilizada para o atuador final, também para as mesas foi utilizado um sistema modular da SMC que reúne todos os sensores e eletroválvulas, fazendo a comunicação com o PLC através do protocolo EtherCAT.

Em relação aos sensores utilizados para auxiliar o controlo das mesas, cada uma possui três sensores fotoelétricos par emissor-recetor, modelo M100/MV100-RT/35/76a/98/103 da Pepper Fuchs, que possui um alcance de deteção de 20 m [52]. Destes três sensores, dois foram instalados de forma cruzada no topo da mesa com o objetivo de indicar ao PLC se alguma caixa abriu no momento de pega pelo manipulador robótico e o terceiro instalado à entrada da mesa formadora para que seja possível identificar se existe alguma caixa pronta a entrar na zona de formação de mosaicos. Instalados no interior das mesas e colocados de forma a que seja possível identificar se as caixas se encontram na posição correta durante o processo de formação de mosaico, foram utilizados dez sensores fotoelétricos ML100-8-H-350-RT/103/115a, como utilizados nos transportadores. Com o objetivo de identificar qual a posição dos atuadores pneumáticos, sendo que para os *stoppers* existem sensores de avançado e recuado, e para os empurradores existem sensores de avançado, recuado, curso de 150 mm e curso de 450 mm para que seja possível controlar o posicionamento das caixas de forma mais eficaz, foram instalados sensores D-M9PSAPC, como utilizados no atuador final.

Quanto à motorização da mesa, devido a limitações de espaço, ao invés de se utilizar um motor como nos tapetes para exercer movimento nos rolos, a solução foi a utilização de rolos motorizados, do fabricante Interroll. O rolo motorizado RollerDrive EC310 é acionado pelo controlador DriveControl 20, representados na Figura 5.13, que permite definir qual o sentido e velocidade de rotação do rolo motorizado através de entradas e saídas digitais que podem ser controladas pelo PLC [53].



Figura 5.13: Rolo motorizado RollerDrive EC310 (esquerda) e controlador DriveControl 20 (direita) [54, 55]

5.3.6 Sistema de Transporte de Paletes - charriot

Visto que um dos requisitos e necessidade do sistema é o transporte de paletes de um local para o outro, foi preciso desenvolver um sistema de transporte (*charriot*), que se encontra representado na Figura 5.14. O principal foco deste equipamento é realizar o transporte da pilha de paletes vazias para o interior da célula, mais concretamente para os transportadores de armazenamento de paletes, e realizar a descarga das paletes completas que se encontram nos transportadores de transferência para a entrada da envolvente.

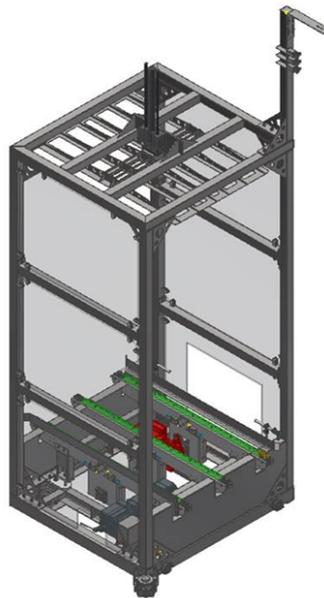


Figura 5.14: Desenho do sistema de transporte de paletes implementado

De forma geral, o *charriot* baseia-se num transportador com estrutura semelhante aos restantes que foram implementados na célula, mas com a particularidade de se conseguir deslocar no espaço. Assim, este equipamento é composto por dois motores: um que motoriza o transportador de forma a realizar a transferência das paletes e um segundo encarregue de deslocar toda a estrutura por um carril. O controlo de posicionamento é realizado através de um sensor fotoelétrico de medição de distância que se encontra instalado na estrutura. Através da leitura deste sensor, é realizada a comparação entre a posição atual e a posição pretendida. De seguida, por *software*, é analisado qual o erro da comparação e consoante o resultado obtido, a frequência do inversor é alterada com o objetivo de acelerar e desacelerar o *charriot*. Para que seja possível realizar a alteração da frequência do inversor, este deve vir equipado com uma carta EtherCAT opcional. A opção de implementação de uma solução de controlo de posicionamento desta natureza, ao invés da utilização de um servo-motor, recaiu sobretudo em fatores económicos e na possibilidade de existir deslizamento do *charriot* no piso, já que este será lavado com água com frequência.

Visto que o *charriot* irá realizar o transporte de uma carga que pode atingir uma altura máxima de 2 m, foram instalados painéis laterais e um sistema de prensa que impedem que a carga caia numa situação de paragem de emergência.

Quanto à sensorização do equipamento, este possui os mesmos dois sensores de reflexão no objeto que foram utilizados nos restantes transportadores de correntes, com o objetivo de verificar se a transferência da paleta foi realizada com sucesso. Em relação aos sensores de verificação da posição do atuador pneumático, foram utilizados dois com a mesma referência dos já mencionados para o atuador final e para as mesas formadoras. O sensor de medição de distância escolhido para o controlo de posição foi o modelo VDM28-15-L1-IO/73c/110/122, com capacidade de medição para 15 m e com interface IO-Link. Foram também inseridos vários dispositivos de segurança cuja decisão de incorporação no *charriot* deve-se ao facto de, como já referido na Secção 5.1, existir uma área onde pode ocorrer a passagem de operadores que interceta o seu percurso. Por esta razão, foram instalados dois *scanners* e três interruptores de fim de curso de segurança. O método de funcionamento destes equipamentos, e de que forma estes afetam o comportamento do *charriot* e garantem a segurança da célula, será abordada na Subsecção 5.3.7, onde serão referenciados todos os equipamentos de segurança e as suas funcionalidades.

5.3.7 Dispositivos de Segurança

Um dos aspetos mais importantes na implementação de um sistema desta natureza é garantir que todas as condições de segurança são cumpridas, de forma a não colocar em risco a integridade física dos operadores. Assim, nesta subsecção

serão abordados todos os equipamentos de segurança e devidas funcionalidades implementados no sistema.

Com o objetivo de se obter um controle de segurança mais flexível e que permita alterações futuras em caso de necessidade, optou-se por introduzir no sistema um PLC de segurança, modelo NX-SL3300 da Omron, ilustrado na Figura 5.15, que permite ligar um máximo de 256 pontos de I/O. O controlador NX-SL3300 é utilizado especificamente para efeitos de segurança e a sua programação é realizada através de *Function Blocks* de segurança incluídos na biblioteca da Omron. Associado ao controlador de segurança encontram-se as cartas de entradas de segurança, NX-SID800, e as cartas de saídas de segurança, NX-SOD400, ambas ilustradas na Figura 5.15.



Figura 5.15: Controlador de segurança e respetivas cartas de entradas e saídas de segurança [56].

Estas cartas possuem, respetivamente, oito pontos de entradas e quatro pontos de saídas de segurança. De realçar que nos dispositivos de segurança é necessária a utilização de dois canais ao invés de um, pelo que os oito pontos de entrada e quatro pontos de saídas se traduzem, respetivamente, em quatro dispositivos diferentes de segurança e dois sinais de saída diferentes [56]. Com esta configuração é possível reunir todos os dispositivos de segurança no PLC de segurança, como por exemplo botoneiras de emergência, e através da programação definir quais os cenários em que deve ser atuado o estado de emergência do sistema.

Em conjunto com o PLC de segurança e respetivas cartas, foram implementados relés de segurança G9SE-401 da Omron, representados na Figura 5.16. Estes relés recebem os sinais da carta de saídas de segurança e, em caso de emergência, atuam sobre o circuito de emergência dos vários equipamentos forçando a sua paragem. Visto que os manipuladores robóticos em geral possuem dois circuitos de emergência distintos, um para paragens de emergência através de botoneiras e outro para paragens por abertura de portas, foram utilizados dois relés de segurança.



Figura 5.16: Relé de segurança [57].

O dispositivo de segurança mais simples e comum em todos os sistemas de automação é a botoneira de emergência. Sendo um dispositivo seguro, é representado por um contacto normalmente fechado e, em caso da sua atuação, o circuito é aberto fazendo com que o sistema entre em estado de emergência. Neste sistema foram implementadas várias botoneiras de emergência de dois canais, ou seja, com dois contactos normalmente fechados, como as representadas na Figura 5.17, sendo que o próprio controlador e consola do manipulador, bem como a fechadura eletrónica de portas, que será apresentada de seguida, já possuem incorporadas em si uma botoneira de emergência.



Figura 5.17: Botoneira de emergência.

Na medida em que a célula possui uma porta de acesso, esta deve ser equipada com um dispositivo de segurança que faça a gestão de quando é que o operador pode ou não entrar no espaço de trabalho da célula. Por esta razão, foi instalada uma fechadura eletrónica de portas, ou PSENGate, como ilustrado na Figura 5.18, modelo SG2C-5LPPLLE da Pilz. Este elemento é composto por um trinco de segurança, que é trancado e destrancado sobre ordem do PLC, quatro botões que podem ser configurados para executar uma tarefa programada, uma botoneira de emergência, e um sistema de abertura mecânica para destrancar a PSENGate em caso de algum colaborador ficar retido dentro da célula. Analisando a fechadura

de cima para baixo, uma das configurações padrão que pode ser implementada para os quatro botões programáveis é: botão de *start*, botão de *stop*, botão para pedido de acesso/destrancamento da fechadura, botão de trancamento da fechadura [58].



Figura 5.18: Fechadura eletrônica de portas.

Tal como enunciado na Subsecção 5.3.6, o charriot foi equipado com dois *scanners* e três sensores mecânicos de segurança. Os *scanners* de segurança são basicamente sensores óticos que determinam a localização de obstáculos que entram na zona de monitorização pré-definida e programada. Os *scanners* instalados são o modelo OS32C-BP-DM-4M da Omron, representado na Figura 5.19.



Figura 5.19: *Scanner* de segurança Omron [59].

Este *scanner* permite a monitorização de três campos em simultâneo: uma zona de segurança e duas zonas de aviso, como mostra a Figura 5.20. A zona de segurança é utilizada para detetar um obstáculo presente na área definida. Na presença de um obstáculo nesta zona, o *scanner* envia um sinal seguro de paragem de emergência que imobilizará o *charriot*. As duas zonas de aviso podem

ser configuradas com uma distância de monitorização maior tendo como objetivo a diminuição da velocidade de movimentação do *charriot* ou a ativação de um alarme sonoro. É possível realizar a configuração de 70 conjuntos de zonas, sendo cada conjunto composto pela zona de segurança e pelas duas zonas de aviso [59].

O sensor é capaz de detetar intrusões dentro da zona de segurança com um raio máximo de 4 m, para obstáculos com dimensão mínima de 70 mm, e duas zonas de aviso com um raio máximo de 15 m, cobrindo um ângulo de varrimento máximo de 270°. Este modelo possui também comunicação EthernetIP, que permite realizar a monitorização de vários parâmetros, como por exemplo o conjunto de zonas ativado ou o motivo de paragem do sistema [59, 60].

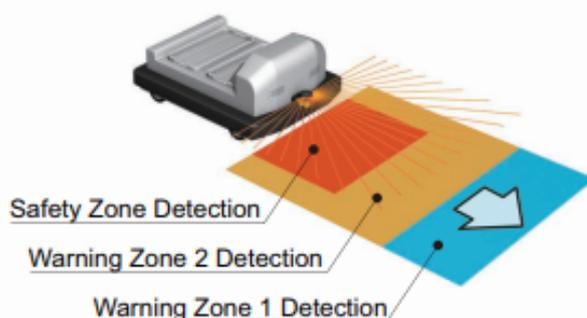


Figura 5.20: Zonas de deteção do *scanner* [59].

Em relação aos modos de operação, é possível distinguir o modo *Automatic Start*, *Start Interlock* e *Start/Restart Interlock*. No modo *Automatic Start*, o *scanner* entra automaticamente no estado de funcionamento, depois de ligado, se nenhuma falha for detetada durante as inicializações e auto-testes e se nenhuma intrusão for detetada dentro da zona de segurança. Se entrar um obstáculo na zona de segurança o *scanner* entra em estado de emergência, desligando o sinal de segurança. Quando a zona de segurança se encontrar livre o *scanner* liga o sinal de segurança e entra em modo de funcionamento normal de forma automática. Já no modo *Start Interlock*, o dispositivo entra em modo de bloqueio após realizar as verificações iniciais. Para sair deste estado deve ser dado o sinal de *start* e o *scanner* entra em modo de funcionamento. No caso de obstrução da zona de segurança o seu comportamento é igual ao modo *Automatic Start*. Quanto ao modo *Start/Restart Interlock*, o *scanner* entra no modo bloqueio após as verificações iniciais, tal como no modo *Start Interlock*. Após dar o sinal de *start* este entra no modo de funcionamento e no caso de existir obstrução da zona de segurança o equipamento entra no estado de bloqueio [59].

Em relação aos três interruptores de fim de curso de segurança, estes têm o propósito de indicar a posição do *charriot* com o objetivo de realizar a comutação das zonas de monitorização do *scanner*. A forma de atuação dos interruptores é realizada através de calhas instaladas ao longo do percurso do *charriot*, sendo que cada calha faz a atuação de apenas um interruptor. O modelo escolhido D4N-1C22 da Omron, ilustrado na Figura 5.21, possui duas saídas normalmente fechadas e uma saída normalmente aberta [61].



Figura 5.21: Interruptor de fim de curso de segurança [61].

Visto que o *charriot* circula para dentro e para fora do núcleo da célula, existe um espaço livre de vedação de proteção permitindo a sua circulação livre. Este espaço livre de vedação representa um perigo, pois possibilita a entrada de um operador para dentro do núcleo da célula enquanto o manipulador robótico se encontra em movimento. Com o objetivo de eliminar este perigo foi instalada uma barreira de segurança neste espaço.

A barreira de segurança F3SG-4RA1870-30 da Omron, trata-se de um emissor e recetor de feixes luminosos, como se pode observar na Figura 5.22, que entra em estado de emergência no caso de os feixes serem interrompidos. Este modelo tem uma altura de proteção de 1870 mm e uma capacidade de deteção de 30 mm. A seleção de uma barreira com estas características é o resultado do cálculo da distância de segurança que deve ser cumprida para o sistema parar em segurança, que será demonstrado na Subsecção 6.6.1 [62].

Assim, utilizando uma barreira de segurança, sempre que algum operador ou objeto interromperem os seus feixes, esta entra em estado de emergência e o sistema é parado. No entanto, este cenário não pode acontecer sempre pois a barreira só deve entrar em estado de emergência quando for um operador a interromper os seus feixes e não o *charriot*. Para tal, a barreira escolhida possui



Figura 5.22: Barreira de segurança

a funcionalidade de *muting*. A função de *muting* permite desabilitar temporariamente as funções de segurança da barreira, mantendo a saída de segurança ligada mesmo se os seus feixes estiverem bloqueados. Esta funcionalidade possibilita a passagem de sistemas de transporte móveis, mantendo os níveis de segurança e de produtividade. A função de *muting* é ativada através de sensores, normalmente fotoelétricos, que devem ser instalados e configurados de uma forma específica, que será abordada na Subsecção 6.6.1 [62]. Os sensores usados para ativar a funcionalidade de *muting* da barreira foram o par emissor-recetor M100/MV100-RT/35/76a/98/103, utilizados também nas mesas formadoras de mosaicos.

5.4 Conclusão

Neste capítulo foram apresentados vários aspetos importantes que determinaram o tipo de solução a implementar. Após a definição do problema e da análise de requisitos, foi possível determinar qual a melhor configuração para a célula e a sua lógica de funcionamento. Foram expostos os vários componentes que fazem parte da arquitetura do sistema e o papel que desempenham no funcionamento da célula.

Capítulo 6

Implementação Prática

Neste capítulo serão descritos todos os procedimentos realizados no processo de programação e implementação da célula.

Será feita uma abordagem, em primeiro lugar, à simulação *offline* realizada de modo a validar o *layout* definido da célula e analisar as trajetórias do manipulador e possíveis colisões. De seguida, será explicada a lógica de programação do robô, bem como da consola HMI e do PLC, que é responsável pelo controlo geral da célula. Por último, são expostos os protocolos e regras de segurança aplicadas na célula e a programação realizada no PLC de segurança.

6.1 Simulação Offline

O primeiro passo da implementação do sistema foi realizar a simulação *offline* da célula. Desta forma, utilizando o *software* de simulação Roboguide da FANUC, realizou-se a modelação da célula com o objetivo de validar o *layout* inicial, analisando pontos como o alcance do manipulador aos restantes equipamentos da célula, trajetórias e possíveis colisões.

Numa primeira abordagem ao sistema, foi possível concluir que o manipulador não possuía alcance vertical suficiente para formar uma paleta com 2 m de altura sobre o tapete de paletização que possui, aproximadamente, 500 mm de altura. Por esta razão, foi necessário colocar o manipulador em cima de um pedestal. A altura do pedestal foi analisada tendo em conta dois fatores: o primeiro é o pé-direito do edifício de apenas 4 m e o segundo a altura necessária de instalação do manipulador para que este consiga formar a paleta com 2 metros. Assim, após a simulação deste cenário, foi possível concluir que o pedestal teria de possuir uma altura de aproximadamente 850 mm possibilitando a formação da paleta com 2 m

de altura não ocorrendo o risco do manipulador bater no teto do edifício. Ambos os casos podem ser visualizados na Figura 6.1.

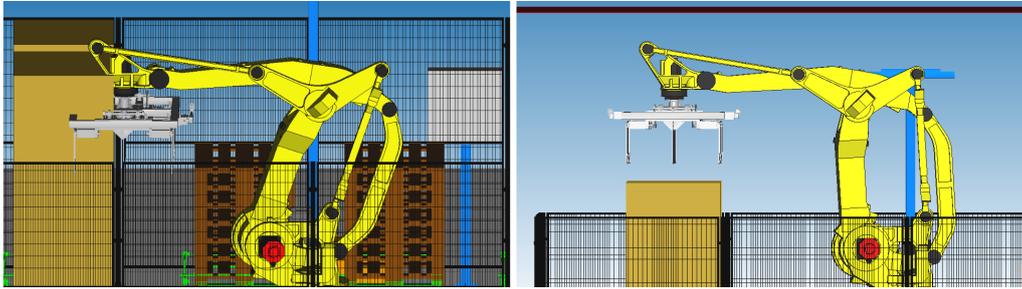


Figura 6.1: Alcance vertical do manipulador sem pedestal (esquerda) e com pedestal (direita).

Outro cenário analisado através da simulação *offline* foi a forma como o manipulador deve realizar o transporte da palete para o transportador de paletização no caso de existir uma paleta já completa no transportador de transferência. Como é possível observar na Figura 6.2, quando se encontra uma paleta completa na zona de transferência o robô não possui espaço suficiente para passar com a paleta vazia por cima da paleta completa. Por esta razão, o robô terá de realizar uma trajetória de forma a passar com a paleta no espaço disponível entre a paleta completa e o próprio manipulador.

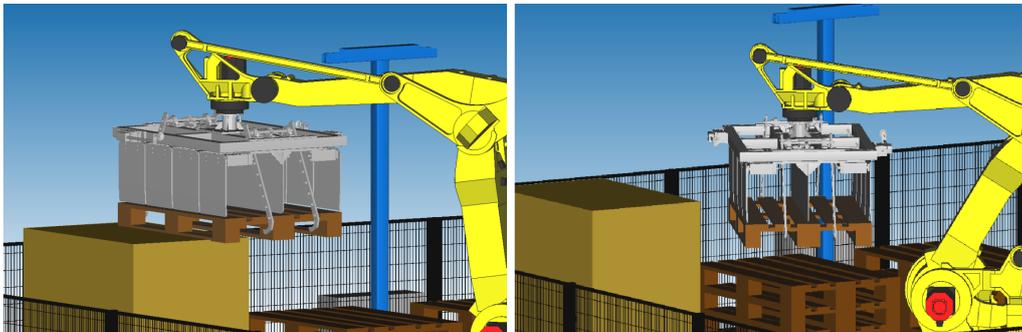


Figura 6.2: Análise da trajetória para a manipulação da paleta

Por fim, foi possível concluir que o robô possui alcance suficiente em relação a todos os módulos da célula e que não existia mais nenhum cenário onde a sua trajetória tivesse de ser avaliada devido ao espaço disponível para a movimentação, como se pode observar na Figura 6.3.

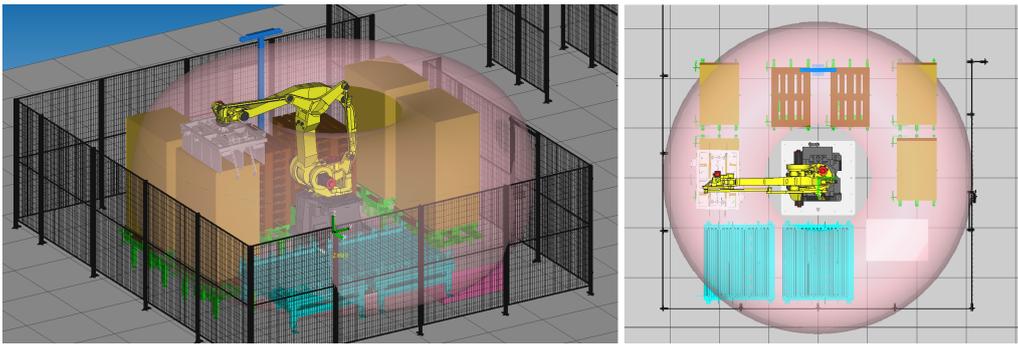


Figura 6.3: Validação do volume de trabalho do manipulador robótico

6.2 Configuração do Manipulador Robótico

O passo seguinte na implementação do sistema foi a configuração do manipulador robótico. Assim, esta secção tem como propósito abordar as configurações efetuadas no manipulador FANUC antes de se iniciar o processo de programação.

6.2.1 Sistemas de Coordenadas

De modo a que o robô se consiga orientar no espaço, existem diversos sistemas de coordenadas que podem ser definidos. Nos manipuladores FANUC, os mais comuns são: o sistema de coordenadas mundo (*World*); os sistemas de coordenadas definidos pelo utilizador (*User Frame - Uframe*); o sistema de coordenadas da ferramenta do robô (*Tool Frame*). Estes sistemas de coordenadas representam uma referência tri-dimensional sobre a qual são memorizadas todas as posições do programa do robô [63].

Na prática, é comum ser definido um sistema de coordenadas *Uframe*, por exemplo, numa mesa de trabalho onde o robô irá realizar a manipulação do objeto, e no caso da necessidade de reajustar a posição dessa mesa de trabalho basta redefinir a *User Frame* e não todos os pontos do programa do robô.

Durante a implementação deste sistema foram definidos 7 sistemas de coordenadas *User Frame*, sendo um para cada elemento da célula onde o robô irá realizar a manipulação de objetos (mesas formadoras de mosaicos, transportadores de paletização, transportadores de armazenamento de paletes e armazém de cartão intercalar de camadas). Em relação ao sistema de coordenadas *Tool Frame* foi definido apenas um para o atuador final.

O método de configuração utilizado para o sistema de coordenadas *Uframe* foi o método dos três pontos. Neste método é necessário memorizar três pontos distintos: a origem, a orientação do eixo x e a orientação do eixo y . Na Figura 6.4,

pode ser observado o procedimento de marcação da *Uframe* de uma das mesas formadoras com o auxílio de duas ponteiros, uma presa no atuador final e outra presa na mesa. Já na configuração do sistema de coordenadas *Tool Frame* foi utilizado o método de introdução direta dos valores. Neste método devem ser introduzidas de forma manual todas as coordenadas que definem a localização da ferramenta a utilizar. Assim, todas as coordenadas foram deixadas com o valor pré-definido zero e apenas a coordenada z foi colocada com o valor de 560mm, com o objetivo de definir a *Tool Frame* na base do atuador final e centrada com a flange do robô.

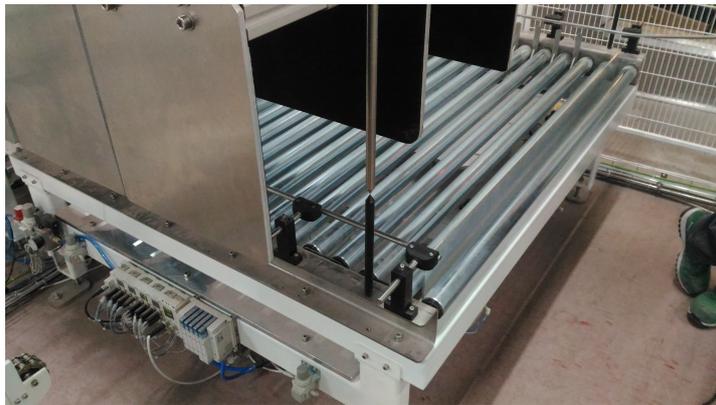


Figura 6.4: Sistemas de coordenadas de um manipulador robótico

6.2.2 Carga de Trabalho

Para o uso de forma eficaz do manipulador é recomendado configurar adequadamente a informação sobre a carga de trabalho que o robô deverá manipular. Esta configuração permite um aumento de rendimento dos movimentos e tempos de ciclo mais curtos, visto que o controlador do robô adequa a trajetória de movimentos ao tipo de carga que este transporta [63].

Para configurar corretamente a carga de trabalho é necessário indicar qual o centro de massa do atuador final e os momentos de inércia em cada eixo, dados estes fornecidos pelo Departamento Mecânico. Para este sistema foram definidas três cargas de trabalho distintas, como se pode observar na Figura 6.5. Uma carga de trabalho para quando o atuador final se encontra vazio ou a manipular a folha intercalar, sendo que o peso da folha é praticamente desprezável, uma carga de trabalho para a situação onde o atuador final manipula a palete e outra para a manipulação do mosaico.

| No. | PAYLOAD [kg] | Comment |
|-----|--------------|-------------------|
| 1 | 200.00 | [Gripper_Vazio] |
| 2 | 230.00 | [Gripper_Pallet] |
| 3 | 250.00 | [Gripper_Box] |
| 4 | 315.00 | [] |
| 5 | 315.00 | [] |
| 6 | 315.00 | [] |
| 7 | 315.00 | [] |
| 8 | 315.00 | [] |
| 9 | 315.00 | [] |
| 10 | 315.00 | [] |

Active PAYLOAD number = 1

Figura 6.5: Cargas de trabalho definidas no manipulador robótico

6.2.3 Sinais de Entradas e Saídas

Nos manipuladores FANUC existem vários tipos de sinais de entrada e saída que podem ser configurados e utilizados com o objetivo de realizar o controle do robô. Os vários tipos de sinais, bem como as suas funcionalidades, encontram-se representados na Tabela 6.1 [63].

Para este sistema, visto que o robô é comandado remotamente pelo PLC, foram utilizados os sinais UOP representados nas Tabelas A.1 e A.2, apresentadas no Anexo A. A configuração destes sinais pode ser realizada através da consola de programação, nas janelas representadas na Figura 6.6. O valor da *Rack* foi configurado com o valor 89 visto que o protocolo de comunicação utilizado para a troca destes sinais com o PLC foi o EthernetIP.

| # | RANGE | RACK | SLOT | START | STAT. |
|---|--------------|------|------|-------|-------|
| 1 | UI [1- 18] | 89 | 1 | 1 | ACTIV |

Device Name : EthernetIP

Figura 6.6: Configuração dos sinais UOP através da consola de programação

Tabela 6.1: Sinais I/O utilizados nos manipuladores FANUC

| Tipo de Sinal | Nomenclatura | Descrição |
|----------------------|---------------------|---|
| Digitais | DI[N] - DO[N] | Utilizados para a gestão de periféricos de linha, como transportadores, alimentadores, entre outros. |
| Analógicos | AI[N] - AO[N] | Utilizados nos casos onde é necessário a gestão de sinais de sensores analógicos. |
| Grupo | GI[N] - GO[N] | Utilizados nos casos onde é preciso agrupar entradas ou saídas digitais. |
| Robô | RI[N] - RO[N] | Sinais digitais utilizados para a gestão da ferramenta de trabalho montada na flange do robô. |
| UOP | UI[N] - UO[N] | Sinais digitais utilizados como interface de comunicação com o PLC. Permitem comandar o robô de forma remota. |
| SOP | SI[N] - SO[N] | Sinais digitais que indicam o estado do sistema. |

Para além dos sinais UOP, foram utilizados os sinais digitais (DI-DO) e os sinais de grupo (GI). Os sinais digitais utilizados encontram-se representados nas Tabelas A.3, A.4 e A.5, presentes no Anexo A, e a sua configuração é semelhante aos sinais UOP. Analisando as Tabelas é possível verificar que o mapeamento dos sinais digitais não se encontra por ordem numérica. Isto deve-se ao facto de, na RobotSol, o mapeamento dos sinais digitais nos manipuladores FANUC ter que seguir determinado padrão com o objetivo de normalizar o processo de implementação, facilitando a leitura e compreensão do código entre as equipas. Desde modo, existem sinais que não foram utilizados no controlo da célula mas que mesmo assim se encontram mapeados no manipulador.

Os sinais de grupo são representados por grupos de sinais digitais possibilitando a conversão de vários sinais booleanos num valor inteiro. Desta forma é possível trocar diretamente com o robô, por exemplo, a altura do produto que irá manipular. Para este tipo de sinal é necessário inserir a *Rack* que, tal como nos sinais UOP e DI/DO, foi preenchida com o valor 89, o endereço de começo do grupo e o número de constituintes, sendo neste caso 16 bits por grupo formando um inteiro. A configuração destes sinais encontra-se representada na Figura 6.7.

The figure shows two screenshots of a FANUC programming console. The left screenshot displays the 'I/O Group In' configuration, and the right screenshot displays the 'I/O Group Out' configuration. Both screens show a table of I/O points with columns for Rack, Slot, Start Point, and Number of Points.

| GI # | RACK | SLOT | START PT | NUM PTS |
|------|------|------|----------|---------|
| 21 | 89 | 1 | 289 | 16 |
| 22 | 89 | 1 | 305 | 16 |
| 23 | 89 | 1 | 321 | 16 |
| 24 | 89 | 1 | 337 | 16 |
| 25 | 0 | 0 | 0 | 0 |
| 26 | 0 | 0 | 0 | 0 |
| 27 | 0 | 0 | 0 | 0 |
| 28 | 0 | 0 | 0 | 0 |
| 29 | 0 | 0 | 0 | 0 |
| 30 | 89 | 1 | 353 | 16 |

| GO # | RACK | SLOT | START PT | NUM PTS |
|------|------|------|----------|---------|
| 9 | 0 | 0 | 0 | 0 |
| 10 | 89 | 1 | 225 | 16 |
| 11 | 89 | 1 | 241 | 16 |
| 12 | 89 | 1 | 257 | 16 |
| 13 | 89 | 1 | 273 | 16 |
| 14 | 89 | 1 | 289 | 16 |
| 15 | 0 | 0 | 0 | 0 |
| 16 | 0 | 0 | 0 | 0 |
| 17 | 0 | 0 | 0 | 0 |
| 18 | 0 | 0 | 0 | 0 |

Figura 6.7: Configuração dos sinais de grupo através da consola de programação

6.2.4 Registos e Registos de Posições

No método de programação de um robô FANUC existem dois tipos de variáveis disponíveis: os registos e os registos de posição. Um registo é, basicamente, uma variável de trabalho que pode armazenar o resultado de um cálculo aritmético auxiliar, um valor constante ou um valor de um sinal de I/O. Já os registos de posição armazenam coordenadas de um ponto do manipulador robótico e podem ser utilizados em operações aritméticas ou como variável auxiliar de controlo, pois é possível aceder a uma coordenada específica de um ponto gravado. Ambos tratam-se de variáveis globais, ou seja, todos os programas têm acesso a todos os registos e registos de posição [63].

Para esta aplicação os registos são usados com dois propósitos: armazenamento dos valores inteiros contidos nos sinais de grupo GI e como variáveis auxiliares de operações aritméticas. Na Tabela A.6, é possível observar quais os registos utilizados e quais os sinais de grupo correspondentes. Quanto aos registos de posição, listados na Tabela A.7, do Anexo A, estes foram utilizados para guardar as coordenadas dos pontos mais importantes no funcionamento do manipulador, como a posição de *home*, a posição de manutenção, a posição de preparação para iniciar o ciclo de trabalho e para operações auxiliares de controlo do programa.

O armazenamento dos valores contidos nos sinais de grupo GI, trocados com o PLC através de comunicação EthernetIP, devem ser transferidos de forma contínua para os registos. Desta forma, é necessário criar um programa que corra de forma contínua e que seja responsável pela transferência dos valores recebidos pelo controlador do robô para os registos. Para alcançar este objetivo é necessário utilizar a funcionalidade de *Background Logic* que será explicada na Subsecção 6.2.5.

6.2.5 Background Logic

A funcionalidade *Background Logic* permite a criação de um programa que corre de forma contínua e em ciclos repetitivos em segundo plano, ou seja, ao mesmo tempo que os restantes programas do manipulador. Mesmo que seja acionada uma paragem de emergência, o programa em *Background Logic* não é interrompido. Assim sendo, foi criado o programa `GI_GO_DATA` responsável pela passagem dos valores obtidos nos sinais GI para os registos. Após a criação do programa é necessário fazer a sua atribuição à função de *Background Logic*, como ilustrado na Figura 6.8 [63].

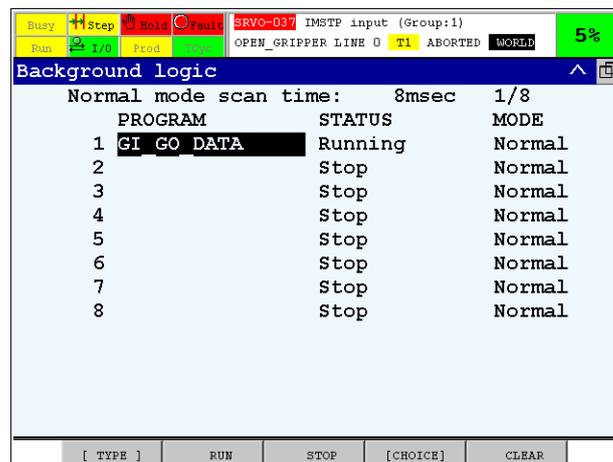


Figura 6.8: Configuração do programa a correr em *Background Logic*

6.2.6 Posições de Referência

Uma posição de referência é uma posição do manipulador onde se especificam tolerâncias, em graus, para cada junta. Esta função permite delimitar uma zona no espaço e sempre que o robô entre nessa zona, ou seja, os valores de posição das suas juntas se encontrem dentro dos limites estabelecidos, pode ser atuada uma saída digital [63].

Para esta aplicação foram configuradas várias posições de referência: uma que indica que o robô se encontra na posição de *home*, outra que indica que o robô se encontra numa posição segura, ou seja, que pode mover o eixo 1 sem limitações garantindo que não colide com nenhum objeto presente na célula, uma que indica que o robô se encontra dentro da área definida para a Linha 1, outra que indica que o manipulador se encontra dentro da área definida para a Linha 2 e, por último, uma que indica que o robô está na posição de manutenção. Na Figura 6.9 encontra-se a janela de configuração das posições de referência através da consola de programação.

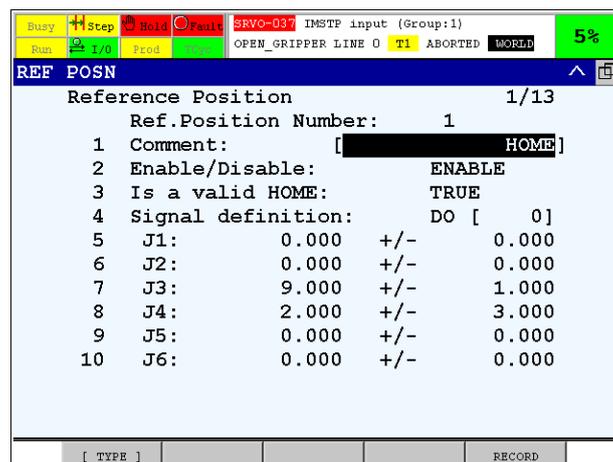


Figura 6.9: Configuração das posições de referência

6.2.7 Configuração de Rede

Como já referido várias vezes ao longo do documento, o método de comunicação entre o controlador do robô e a unidade de controlo é o protocolo EthernetIP. Para que a configuração de rede seja realizada de forma correta, é necessário fazer a sua configuração do lado do robô e do lado do PLC. Do lado do controlador do robô é necessário introduzir o endereço IP dos vários dispositivos e definir qual o tamanho das tramas de entrada e saída, como se pode observar na Figura 6.10. Atendendo ao número de sinais trocados entre o controlador e o PLC, o tamanho da trama de comunicação de entrada e de saída foi configurado com um tamanho de 35 *words*, ou 70 *bytes*. Já a configuração do lado do PLC será detalhada na Subsecção 6.4.2.

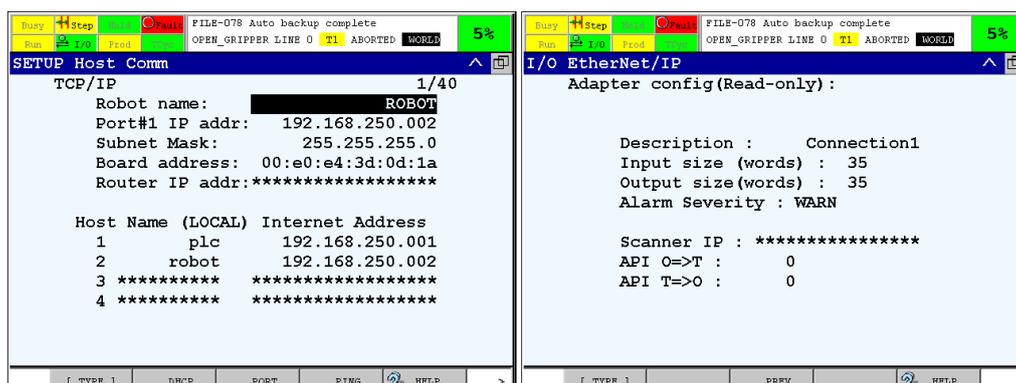


Figura 6.10: Configuração da comunicação EthernetIP do lado do controlador robótico

6.2.8 Limitação dos eixos

Com o objetivo de diminuir a probabilidade e o número de situações onde o manipulador poderá colidir com algum objeto da célula, ou até mesmo, com o teto do edifício, procedeu-se à limitação dos seus eixos. Para tal, movimentou-se o robô em modo manual para os pontos mais críticos e que apresentassem maior probabilidade de colisões, verificou-se qual o valor das juntas nesses mesmos pontos e procedeu-se à sua limitação como se pode observar na Figura 6.11.

| AXIS | GROUP | LOWER | UPPER | 1/84 |
|------|-------|---------|--------|------|
| 1 | 1 | -160.00 | 160.00 | dg |
| 2 | 1 | -44.00 | 100.00 | dg |
| 3 | 1 | -126.00 | 10.00 | dg |
| 4 | 1 | -360.00 | 360.00 | dg |
| 5 | 0 | 0.00 | 0.00 | mm |
| 6 | 0 | 0.00 | 0.00 | mm |
| 7 | 0 | 0.00 | 0.00 | mm |
| 8 | 0 | 0.00 | 0.00 | mm |
| 9 | 0 | 0.00 | 0.00 | mm |
| 10 | 0 | 0.00 | 0.00 | mm |

Figura 6.11: Configuração da limitação dos eixos do manipulador.

6.2.9 Space Functions

Mesmo utilizando a opção de limitação dos eixos do manipulador, como abordado na Subsecção 6.2.8, foram verificados alguns cenários que não conseguiram ser abrangidos por essa limitação.

Assim sendo, recorreu-se à funcionalidade de *space function* que permite a criação de um volume com o objetivo de verificar se o manipulador se encontra dentro ou fora desse espaço. No caso do manipulador, seja em modo automático ou em modo manual, entrar ou sair nos espaços definidos é ativada uma saída digital. Para tal foram configuradas duas *space functions*, representadas na Figura 6.12, uma que engloba toda a área da célula e outra que engloba um sistema de ventilação nas imediações do robô. Assim sendo, sempre que o manipulador sai da zona da célula ou entra na zona do sistema de ventilação, é enviado um sinal ao PLC, que por consequência desativa todos os sinais que permitem a movimentação do robô. Desta forma, são assegurados os perigos de colisão do manipulador com o sistema de ventilação, com as redes da célula e com os transportadores e mesas de formação de mosaicos.

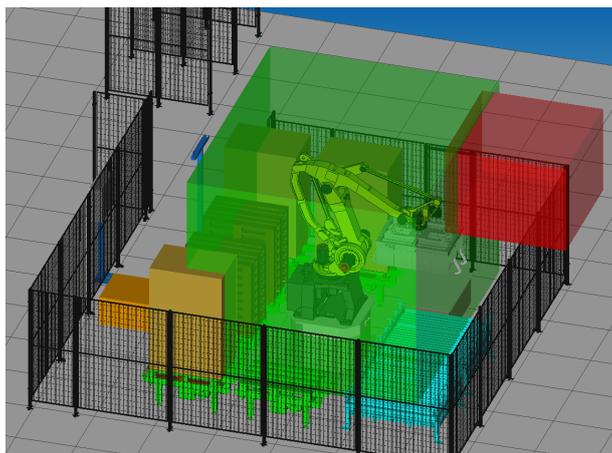


Figura 6.12: *Space Functions* configurados no manipulador.

6.3 Programação do Manipulador

Após realizadas as várias configurações no controlador do robô segue-se a programação das várias rotinas de trabalho. Com o objetivo de obter o correto funcionamento do sistema foram criadas várias rotinas de trabalho para o manipulador. Assim, nesta secção serão mencionadas quais as rotinas criadas e o seu papel no funcionamento do sistema.

A programação de um robô FANUC pode ser realizada de várias formas. Uma delas é através da linguagem de programação KAREL, no entanto o controlador necessita de possuir um pacote opcional de interpretação da linguagem. Outro método é através da consola de programação, método conhecido como TP Programming, que gera ficheiros com a extensão TP interpretados pelo controlador do robô. No entanto, para programas mais complexos e extensos este método não se revela adequado, pois a introdução de comandos não se revela como um procedimento prático. Por esta razão, a FANUC permite ao utilizador escrever o programa num ficheiro de texto que deve ser gravado com a extensão LS. Este programa é depois carregado e compilado no Roboguide, gerando de seguida o ficheiro TP que pode ser interpretado pelo controlador. De notar que o ficheiro escrito pelo utilizador com a extensão LS deve cumprir a sintaxe de programação TP.

Com a finalidade de se perceber de uma forma mais intuitiva a lógica de trabalho do manipulador, na Figura 6.13 encontra-se representado um diagrama de árvore das principais rotinas criadas organizadas hierarquicamente

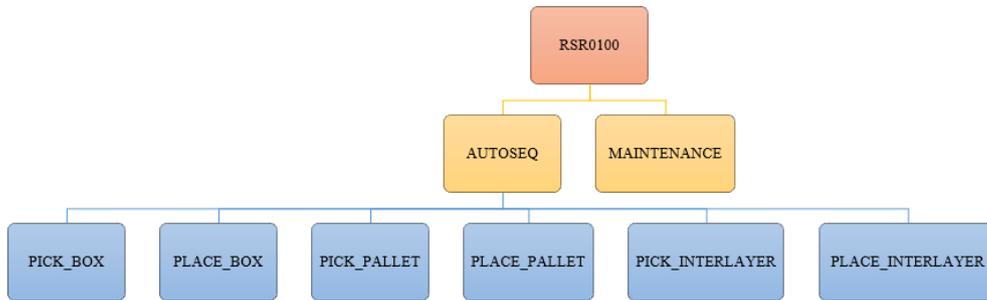


Figura 6.13: Diagrama de árvore dos programas do manipulador robótico

Pela análise da Figura 6.13 é possível dividir a programação do robô em três níveis. O nível mais alto é representado pela rotina RSR0100 que é o programa principal (*main*) chamado no arranque do sistema. Este programa começa com a colocação do robô em posição de *home*, chamando a rotina HOME, e pelas inicializações dos sinais digitais que são trocados com o PLC, através da rotina INIT.

A rotina HOME faz a recuperação automática do manipulador para a posição de *home*. O primeiro passo é a verificação da posição atual do manipulador e no caso de este se encontrar numa posição segura, como abordado da Subsecção 6.2.6, movimenta-se automaticamente para a posição gravada como *home*. No caso de o manipulador não se encontrar numa posição segura, o programa lê a posição atual do robô e é realizado um processo de *offsets* até que o manipulador se encontre numa posição segura e se consiga mover para *home* sem perigo.

Após estes passos, dependendo das escolhas do operador, o programa RSR0100 pode chamar o programa AUTOSEQ ou MAINTENANCE localizados no nível intermédio. A seleção destes dois programas é feita através do valor presente no registo R[1] - nProg. Na Figura 6.14 é possível observar o fluxograma de funcionamento do programa RSR0100.

O programa MAINTENANCE é a rotina que coloca o robô numa posição de manutenção e é chamado sempre que o operador faça um pedido de manutenção através da consola de interface.

Já o programa AUTOSEQ representa o ciclo de produção em modo automático onde o robô fica à espera que o PLC lhe indique qual a tarefa que tem de realizar, dependendo do estado geral da célula. As tarefas do robô são representadas pelas rotinas do nível inferior e a sua seleção é realizada através do valor presente no registo R[2] - nJob. Assim sendo, no caso do PLC verificar, por exemplo, que existem condições de colocar uma paleta no tapete de paletização dá ordem ao robô para executar a tarefa responsável por esse processo.

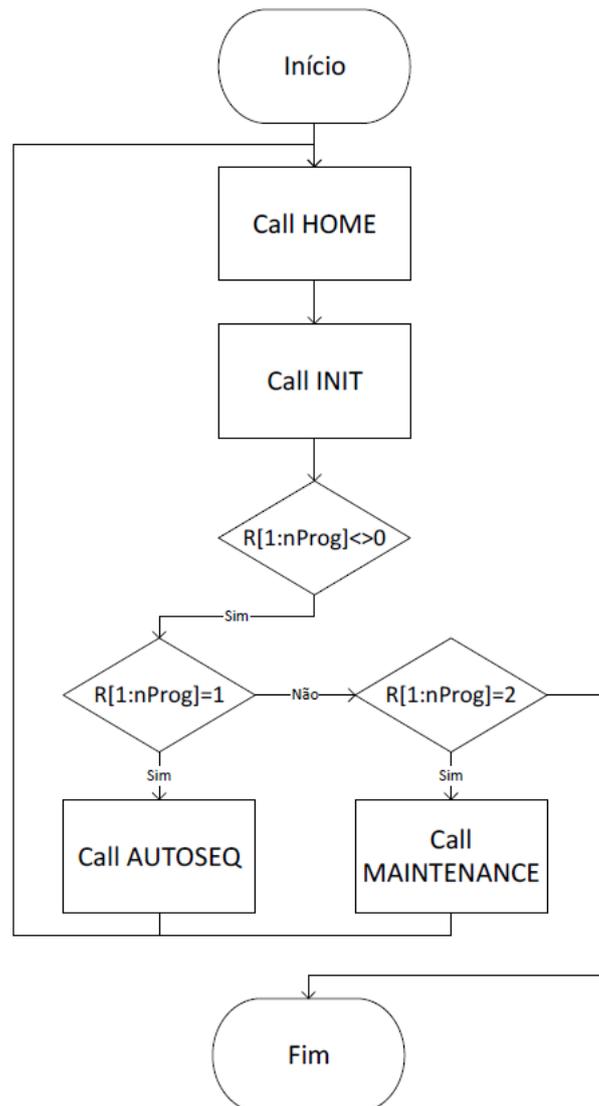


Figura 6.14: Fluxograma do programa RSR0100

Assim que seja iniciada a execução do AUTOSEQ, o manipulador é movido para uma posição de preparação para iniciar o ciclo de trabalho. De seguida é verificado se foi efetuado algum pedido de acesso ou de manutenção. No caso de ter sido feito um pedido de acesso, o programa é pausado até que o operador dê ordem para continuar através do botão verde presente no painel de interface. No caso de ter sido feito um pedido de manutenção, o programa AUTOSEQ é terminado, fazendo com que a execução do programa salte para a rotina RSR0100 que trata de chamar o programa que move o robô para a posição de manutenção.

Não tendo sido feito nem o pedido de acesso nem o pedido de manutenção, o robô fica a espera que o PLC lhe indique qual a tarefa que tem de executar e, dependendo da ordem do PLC, o robô executa uma das rotinas presentes no nível hierárquico inferior. No total existem 6 tarefas distintas, sendo que cada uma é composta pelo ciclo de *pick* e pelo ciclo de *place*. Sempre que é terminada a execução de uma tarefa o robô volta a esperar por nova ordem de tarefa, visto que o programa AUTOSEQ corre de forma cíclica até que seja dada a ordem de paragem. A lógica da rotina AUTOSEQ encontra-se esquematizada no fluxograma da Figura 6.15.

Dada a existência de duas linhas de produção, numa primeira instância foi considerado desenvolver rotinas de *pick* e *place* gerais, ou seja, que fossem re-entrantes recebendo parâmetros como argumentos. Para tal iria ser criada uma *User Frame* de trabalho dinâmica que se iria mover no espaço dependendo do argumento passado nas funções de *pick* e *place*. Devido à forma como o produto chega nas mesas e à orientação das mesmas, esta abordagem não foi aplicada pois teriam de ser marcados vários pontos distintos para cada mesa, eliminando as vantagens de utilizar apenas um ponto para cada mesa através do uso da *User Frame* dinâmica. Assim sendo, o método aplicado foi a criação de várias rotinas distintas para cada linha de produção. Apesar de esta abordagem gerar rotinas com lógica de funcionamento semelhantes, sendo apenas os pontos marcados e os registos para os cálculos de *offset* diferentes, torna-se mais fácil ler e compreender o programa visto que todos os aspetos referentes a uma dada linha de produção se encontram apenas num programa, não se misturando com parâmetros da linha oposta. Desta forma, os programas de *pick* e de *place* dos vários objetos são caracterizadas pelos movimentos que o manipulador terá de fazer para realizar com sucesso a sua tarefa.

Nestas rotinas, antes de se realizar cada movimento, são calculados os pontos de *pick* e de *place*, através de *offsets*, dependendo dos valores fornecidos pelo PLC que se encontram guardados nos registos, como por exemplo, a altura das caixas que deverão ser manipuladas ou então a altura da pilha de paletes vazias. Os fluxogramas de funcionamento destas rotinas encontram-se representados no Anexo B.

As últimas rotinas que faltam mencionar são as de abertura e fecho do atuador final, respetivamente OPEN_GRIPPER e CLOSE_GRIPPER. Ambas as rotinas fazem o respetivo pedido de abertura e fecho do atuador final ao PLC e esperam que este indique que o atuador final se encontra aberto ou fechado, consoante o pedido. Dado que existem três sistemas de manipulação diferentes no atuador final, estas funções recebem como parâmetro qual o sistema de manipulação a abrir ou fechar. Os fluxogramas de funcionamento das rotinas de abertura e fecho do atuador final encontram-se representados no Anexo B.

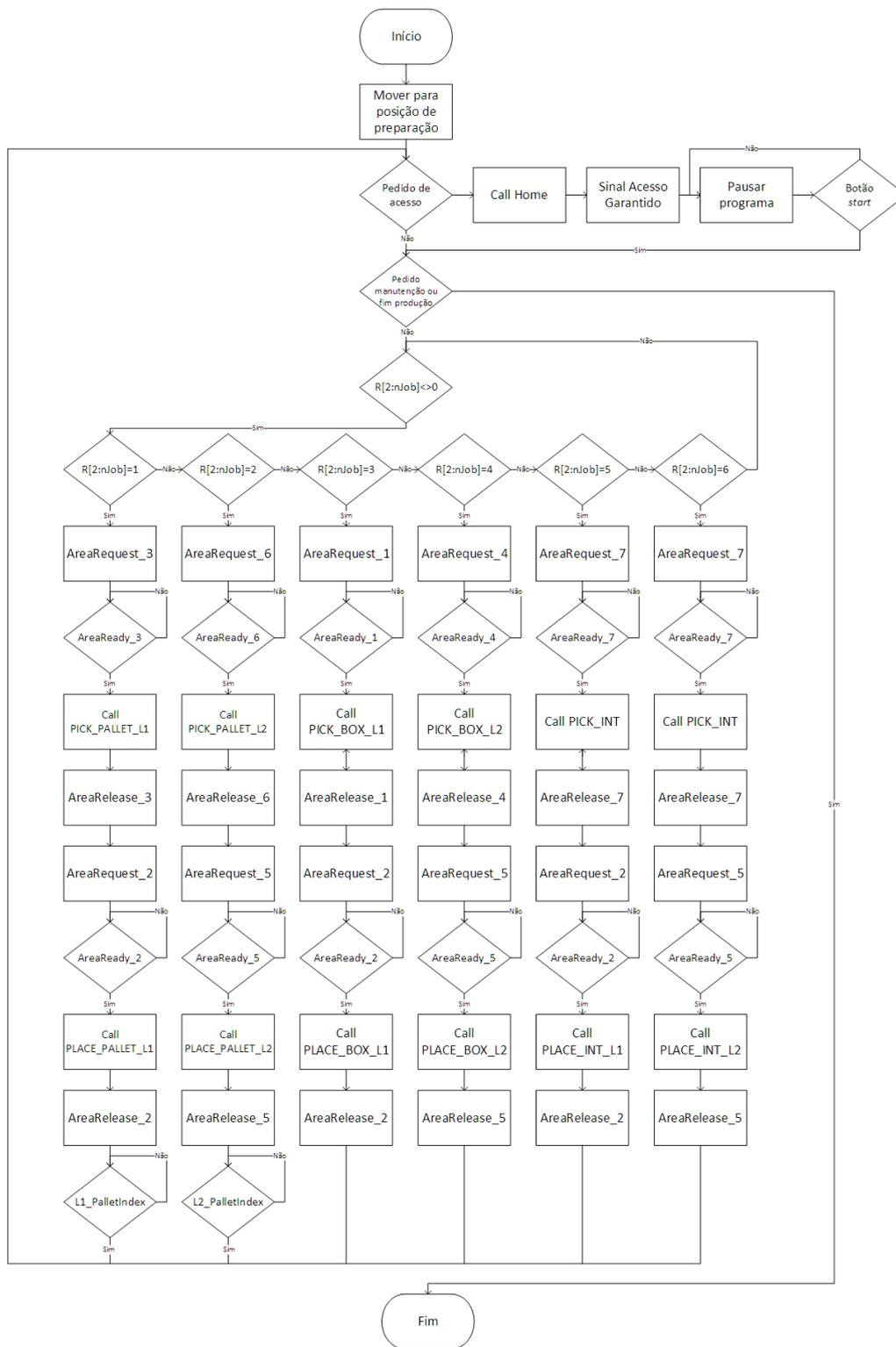


Figura 6.15: Fluxograma do programa AUTOSEQ

Na Tabela 6.2 encontra-se representada uma listagem de todos os programas criados para a programação do manipulador e a sua descrição.

Tabela 6.2: Rotinas programadas no manipulador robótico

| Rotina | Descrição |
|-----------------|---|
| RSR0100 | Programa principal (<i>main</i>) que é chamado no arranque do sistema |
| AUTOSEQ | Programa de produção em modo automático |
| MAINTENANCE | Programa responsável por colocar o robô na posição de manutenção |
| HOME | Programa que coloca o manipulador na posição de <i>home</i> |
| INIT | Programa de inicializações dos sinais digitais trocados entre o robô e o PLC |
| GLGO_DATA | Programa que corre em <i>Background Logic</i> responsável pela atualização dos valores dos registos |
| OPEN_GRIPPER | Programa que faz a requisição de abertura do atuador final ao PLC |
| CLOSE_GRIPPER | Programa que faz a requisição de fecho do atuador final ao PLC |
| PICK_BOX_L1 | Programa que executa a tarefa de pega do mosaico de caixas na Linha 1 |
| PICK_BOX_L2 | Programa que executa a tarefa de pega do mosaico de caixas na Linha 2 |
| PLACE_BOX_L1 | Programa que executa a tarefa de colocação do mosaico de caixas na Linha 1 |
| PLACE_BOX_L2 | Programa que executa a tarefa de colocação do mosaico de caixas na Linha 2 |
| PICK_PALLET_L1 | Programa que executa a tarefa de pega da palete na Linha 1 |
| PICK_PALLET_L2 | Programa que executa a tarefa de pega da palete na Linha 2 |
| PLACE_PALLET_L1 | Programa que executa a tarefa de colocação da palete na Linha 1 |
| PLACE_PALLET_L2 | Programa que executa a tarefa de colocação da palete na Linha 2 |

Tabela 6.2 continuação da página anterior

| Rotina | Descrição |
|---------------------|---|
| PICK_INTERLAYER | Programa que executa a tarefa de pega da folha intercalar |
| PLACE_INTERLAYERL1 | Programa que executa a tarefa de colocação da folha intercalar na Linha 1 |
| PLACE_INTERLAYER_L2 | Programa que executa a tarefa de colocação da folha intercalar na Linha 2 |

6.4 Configuração e Programação do PLC

O PLC é o equipamento do sistema que realiza todo o controlo da célula, definindo a sua lógica e sequência de trabalho. Através da leitura de variáveis de entrada, como por exemplo sensores ou botoneiras, a unidade de controlo envia os comandos de saída aos vários módulos do sistema, como o robô, as mesas formadoras de mosaicos, o *charriot*, entre outros. Assim, dada a importância do PLC no controlo geral da célula, esta secção visa explanar quais as configurações realizadas, bem como a lógica de trabalho implementada através da programação para que o controlo da célula seja realizado de forma efetiva.

6.4.1 Configuração da Rede EtherCAT

Já que um dos protocolos de comunicação implementados no sistema foi o EtherCAT, é necessário realizar a configuração da rede do lado do PLC. Analisando os vários equipamentos com interface EtherCAT no sistema, é possível concluir que apenas os módulos pneumáticos da SMC não fazem parte do leque de produtos da Omron. Desta forma, é necessário realizar um conjunto de passos de configuração do módulo pneumático [64].

O primeiro passo é a introdução do ficheiro de configuração EDS, disponibilizado pela SMC, na biblioteca do Sysmac Studio para que o PLC comunique efetivamente com o equipamento. Após a inclusão do ficheiro EDS, o módulo EX600 aparece na biblioteca Omron.

O passo seguinte foi a configuração manual do módulo EX600. Este módulo possui um conjunto de 8 *dip switches* que devem ser ajustados de acordo com as configurações desejadas pelo utilizador [64]. Os *switches* 1 e 2 configuram o número de saída digitais necessárias, sendo que no caso dos módulos para as mesas formadoras estes foram ajustados para um tamanho de 16 saídas, enquanto que para o módulo do atuador final foram configurados para 24. Quanto aos *switches* 3 e 4, estes habilitam a funcionalidade de trocar informação de diagnóstico entre

o PLC e o módulo, funcionalidade esta aplicada em todos os módulos da célula. Em relação ao *switch* 5, este foi ajustado de forma a que, quando exista uma falha de comunicação, o módulo pneumático mantenha o estado atual de todas as saídas. O *switch* 6 não foi ajustado, permanecendo com o valor padrão do modo de operação normal.

Realizada a configuração dos módulos pneumáticos, o passo seguinte foi introduzir no Sysmac Studio a rede EtherCAT presente no sistema. Este procedimento é realizado na aba *Configurations and Setup - EtherCAT*. Este processo pode ser realizado de forma automática através da opção *Compare and Merge with Actual Network Configuration*, ou pela introdução de cada módulo por parte do utilizador. A configuração de rede EtherCAT do sistema e os respetivos nós de endereço configurados nos dispositivos encontram-se representados na Figura 6.16.

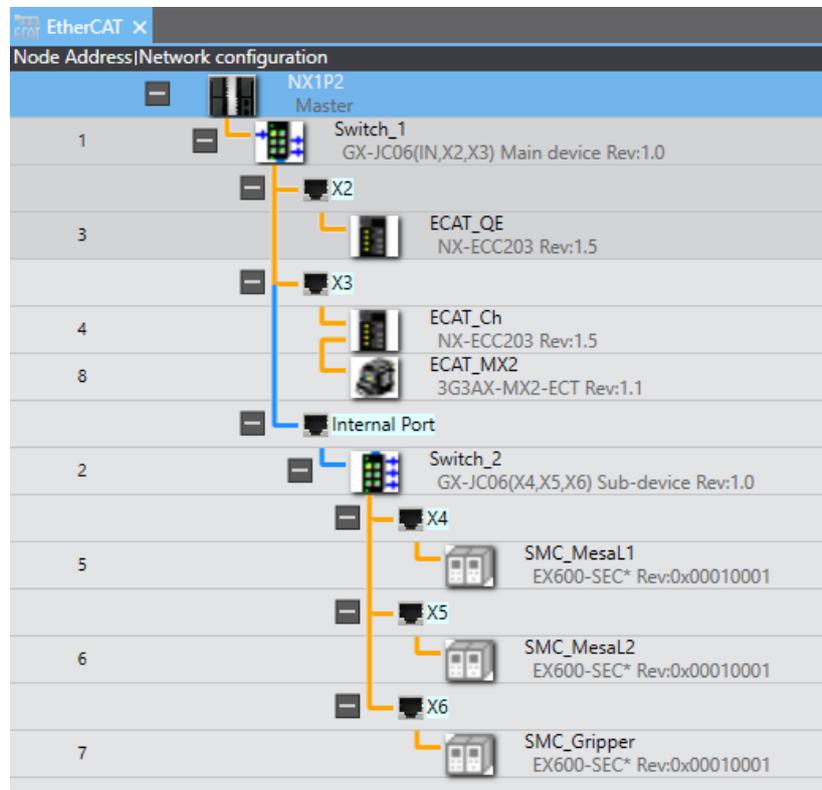


Figura 6.16: Configuração da rede EtherCAT na unidade de controlo.

A configuração EtherCAT só fica concluída após indicar quais as cartas que se encontram ligadas a cada nó da rede (módulos NX-ECC203). Assim, através da opção *Edit Slave Terminal Configuration* é possível indicar quais as cartas ligadas a cada nó e a sua posição na rede. Na Figura 6.17 é possível observar a configuração de cada cabeceira EtherCAT, bem como a configuração da *rack*

principal do PLC.

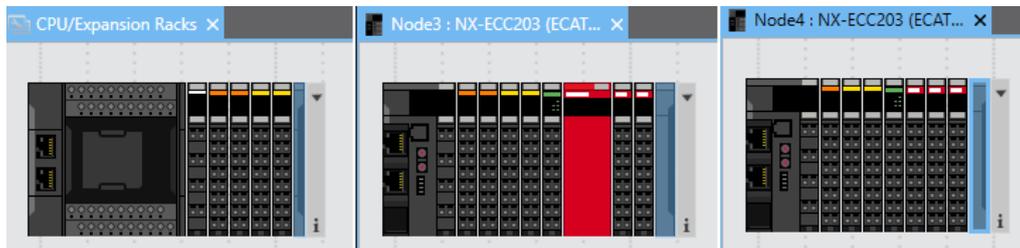


Figura 6.17: Configuração dos nós EtherCAT e da *rack* principal da unidade de controlo.

Pela análise da Figura 6.17 é possível identificar quais as cartas de expansão ligadas a cada nó. Na *rack* principal do PLC tem-se, por ordem, uma carta de alimentação de cartas de expansão, duas cartas de entradas digitais e duas cartas de saídas digitais. Na cabeceira EtherCAT do nó 3 encontram-se ligadas duas cartas de entradas digitais, duas cartas de saídas digitais, uma carta de interface IO-Link, o PLC de segurança, uma carta de entradas de segurança e uma carta de saídas de segurança. Já na cabeceira representada pelo nó 4 encontra-se uma carta de entradas digitais, duas cartas de saídas digitais, uma carta de interface IO-Link, duas cartas de entradas de segurança e uma carta de saídas de segurança.

Concluída a configuração da rede EtherCAT, o passo seguinte foi realizar o mapeamento das variáveis de controlo associadas a cada carta ou módulo da rede. Assim, após serem criadas as variáveis desejadas, na aba *Configurations and Setup - I/O Map* deve ser feito o mapeamento das variáveis referentes aos sinais de entrada e saída das cartas de expansão do PLC e dos módulos EX600.

6.4.2 Configuração da Rede EthernetIP

Terminando a configuração do protocolo de comunicação EtherCAT, o passo seguinte foi a configuração do protocolo de comunicação EthernetIP. No sistema, os únicos dispositivos que comunicam com o PLC através deste protocolo são o robô e os dois *scanners* de segurança. A configuração de rede EthernetIP foi realizada através do *software Network Configurator* da Omron.

Abrindo o *Network Configurator*, o primeiro passo é adicionar os vários dispositivos na rede. Para adicionar tanto o controlador do robô como os *scanners* de segurança é necessário instalar os ficheiros de configuração EDS, ambos fornecidos pela FANUC e pela Omron, respetivamente. Após adicionar os vários dispositivos devem ser definidos os endereços de IP para cada equipamento. De seguida devem ser definidas as tramas de comunicação e o seu tamanho para cada equipamento. Arrastando o ícone, por exemplo, do controlador do robô sobre o

ícone do PLC é apresentada a janela ilustrada na Figura 6.18. Nesta janela devem ser criadas e definidas as várias tramas de comunicação.

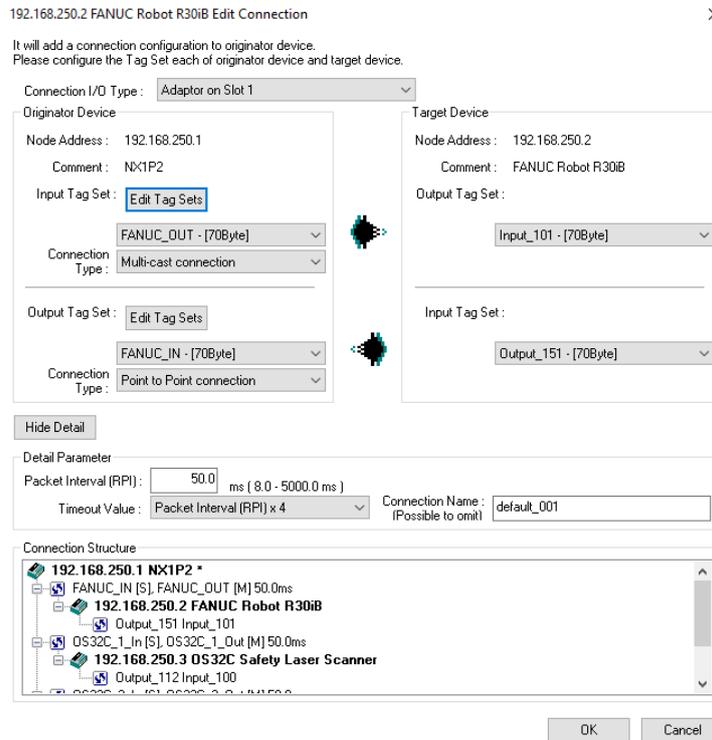


Figura 6.18: Janela de criação das tramas de comunicação EthernetIP.

Para o caso do robô foram definidas tramas de 70 *bytes*, como visto na Subsecção 6.2.7. Para o caso dos *scanners*, estes possuem estruturas de comunicação EthernetIP já predefinidas que podem ser consultadas no manual de utilização [60]. Assim sendo, foram definidas tramas de entrada de 32 *bytes* e tramas de saída de 2 *bytes*. Terminada a configuração e criação das várias tramas de comunicação a configuração da rede deve ser descarregada para o PLC. Na Figura 6.19 encontra-se representada a configuração final de rede EthernetIP implementada.

6.4.3 Programação do PLC e Controlo da Célula

Terminadas as configurações do PLC, o passo seguinte é a programação das várias rotinas de controlo que definem a lógica de trabalho de todo o sistema. A programação da unidade de controlo foi efetuada através da criação de vários programas que realizam o controlo e lógica de trabalho das várias etapas do funcionamento do sistema. Para tal, existem programas destinados à comunicação entre o PLC e o robô fazendo a gestão do arranque e a troca de sinais, programas para o controlo do funcionamento dos constituintes do sistema, como por exem-

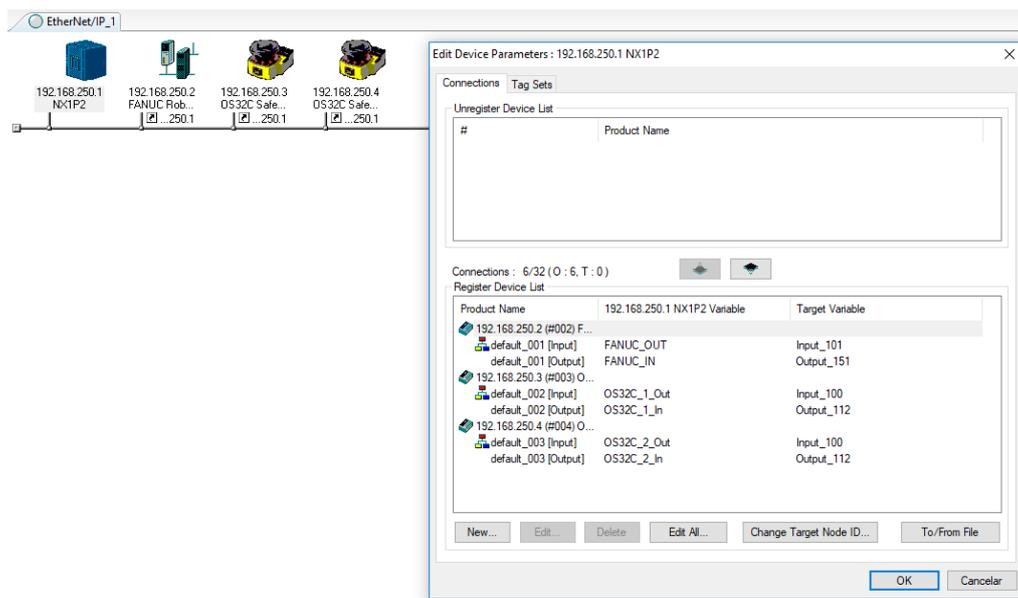


Figura 6.19: Configuração da rede EthernetIP do sistema.

plo os tapetes, as mesas e o *charriot*, programas para a comunicação e troca de variáveis entre o PLC e a interface homem-máquina e programas mais simples que fazem a gestão de equipamentos, como os *scanners*, a *psengate* ou, até, os sensores com interface IO-Link. Na Figura 6.20 encontra-se ilustrado o diagrama de árvore dos programas criados na unidade de controlo.

O programa Main inicia-se com a secção Systems_Enable que, dependendo do modo de funcionamento escolhido por parte do operador através do painel de interface (automático ou semi-automático), habilita a utilização dos vários equipamentos do sistema. Assim sendo, o robô, os tapetes e as mesas ficam habilitados quer o operador escolha o modo de funcionamento automático ou semi-automático. Já o *charriot* apenas funciona se o operador escolher o modo automático. Este programa verifica o modo selecionado e atua variáveis auxiliares que, dependendo dos seus estados, permitem ou não a execução dos *Function Blocks* de controlo de cada equipamento.

A secção, Start_Control, cujo fluxograma se encontra representado na Figura 6.21, é responsável por realizar o controlo de arranque do sistema através da verificação das condições necessárias para o sistema arrancar. Assim, após o operador premir o botão de *start*, dando ordem de arranque ao sistema, o programa realiza a verificação do estado das emergências, do estado dos tapetes, das mesas e do *charriot* dependendo do modo de funcionamento escolhido, terminando com a verificação do robô. Caso todas as verificações sejam realizadas com sucesso, o sistema entra no modo de produção em automático. Nesta fase o operador pode

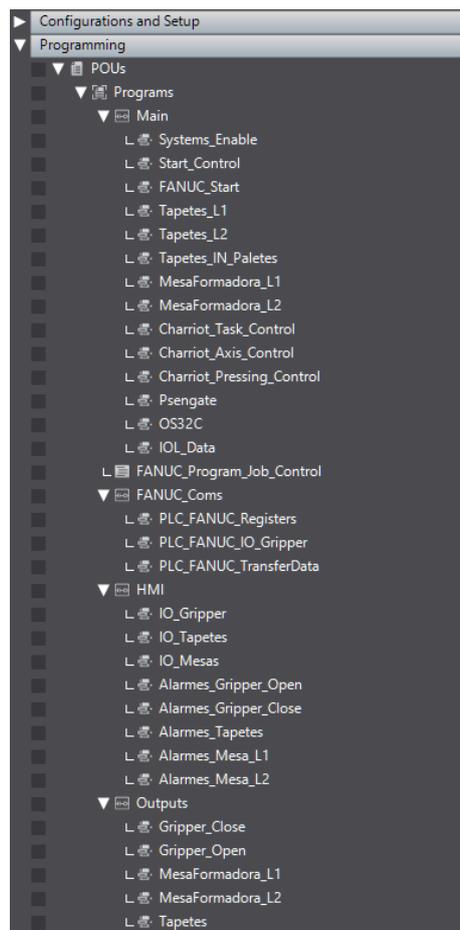


Figura 6.20: Diagrama de árvore dos programas da unidade de controlo.

realizar um pedido de paragem controlada ou pedido de fim de produção. No caso do pedido de paragem controlada, o programa salta para uma etapa onde exerce uma paragem controlada sobre o robô e os restantes equipamentos, sendo que a retoma do ciclo de produção em automático é efetuada pelo botão de *start*. No caso de um pedido de fim de produção, o sistema entra num estado onde espera que o robô termine a sua tarefa e que a célula se encontre vazia.

Após asseguradas estas condições, o programa salta para a etapa de arranque inicial. Durante a execução do programa, se for detetado que o operador premiu o botão de *stop* durante 5 segundos, o programa efetua uma paragem forçada do sistema obrigando a que o operador realize as etapas de arranque inicial.

A secção FANUC_Start é responsável pelo arranque do manipulador através do envio dos sinais IMSTP, SFSPS, HOLD e ENBL, necessários para possibilitar a movimentação do manipulador.

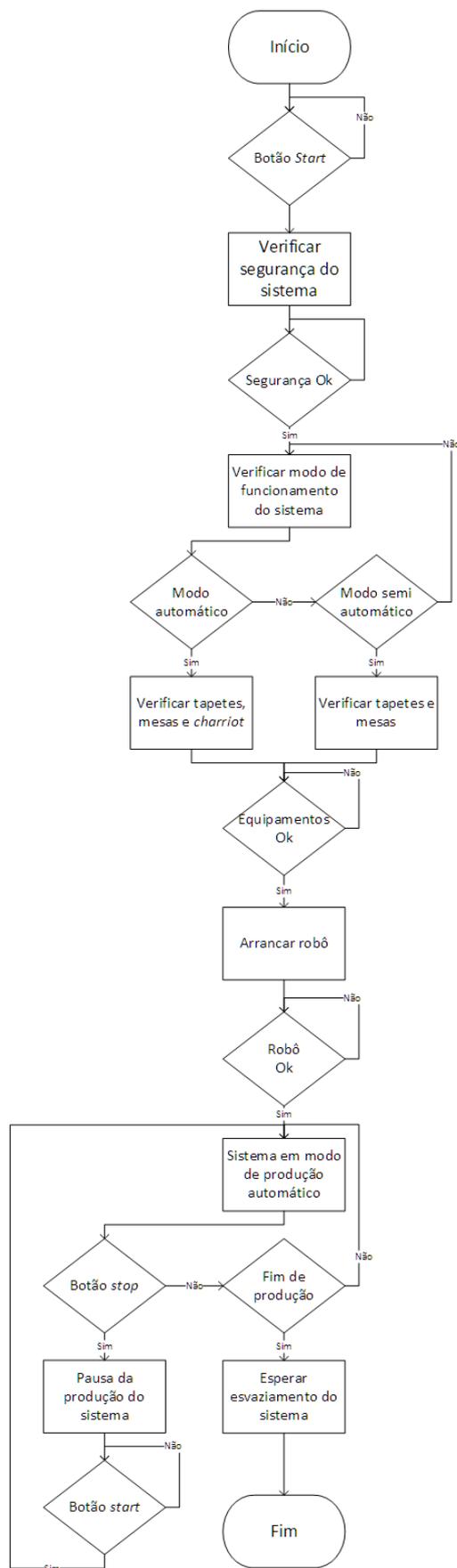


Figura 6.21: Fluxograma da lógica de arranque do sistema.

De seguida, após serem realizadas as verificações de que o robô se encontra em modo remoto, que não tem falhas e que não tem nenhum programa a correr, é enviado o sinal RSR de arranque do programa principal. No caso do manipulador entrar no estado de falha e o operador carregar no botão *start*, o programa procede ao envio do sinal FAULT_RESET. O sinal CSTOPI, responsável por abortar o programa em execução do robô, é enviado caso o operador realize uma paragem forçada do sistema. Na situação de paragem controlada, no momento em que o operador carrega no botão de *start* e o robô possui o seu programa em pausa, é enviado o sinal START com o objetivo de continuar a execução do programa a partir do ponto onde parou.

Visto que ambas as linhas de produção são simétricas e possuem os mesmos componentes, foram criados *Function Blocks* de controlo para cada equipamento. Assim, foram criados os seguintes *Function Blocks*: Tapetes_1 que controla o funcionamento dos transportadores de paletização, Tapetes_2 responsável pelo controlo dos transportadores de transferência, Tapetes_3 que controla o funcionamento dos tapetes de armazenamento da pilha de paletes vazias, Tapetes_IN para a lógica de funcionamento dos transportadores de entrada de paletes e, por fim, Mesa_Formadora responsável pela lógica de trabalho das mesas de formação de mosaicos. Com a criação dos *Function Blocks* é possível chamar várias instâncias que controlam equipamentos diferentes mas com a mesma lógica de trabalho. Assim, para controlar, por exemplo, as mesas formadoras de mosaicos, é chamado duas vezes o mesmo *Function Block* mas as variáveis de entrada e de saída são respetivas a cada mesa. Desta forma, as secções Tapetes_L1, Tapetes_L2, Tapetes_IN_Paletes, MesasFormadoras_L1 e MesasFormadoras_L2 são responsáveis por chamar os *Function Blocks* de controlo dos respetivos equipamentos, sendo que os respetivos fluxogramas de funcionamento estão representados no Anexo C.

As três secções seguintes, Charriot_Task_Control, Charriot_Axis_Control e Charriot_Pressing_Control, são responsáveis pela lógica de trabalho do sistema de transporte automático de paletes. A secção Charriot_Task_Control é responsável por indicar ao *charriot* qual a tarefa que este tem de executar, dependendo do estado atual da célula, ou seja, se a unidade de controlo verificar que é necessário colocar uma pilha de paletes vazias nos transportadores de armazenamento de paletes ou transportar a paleta completa para o exterior da célula, dá ordem de tarefa ao sistema de transporte. A secção Charriot_Axis_Control é responsável pelo controlo de posicionamento do *charriot*. Por fim, a secção Charriot_Pressing_Control é responsável por controlar o sistema de prensa instalado no sistema de transporte. Todas as secções partilham uma dependência entre si, como por exemplo, o programa Charriot_Task_Control faz o pedido de movimentação do sistema de transporte para uma posição do seu percurso ao programa Charriot_Axis_Control, e espera que este lhe envie um sinal a indicar que o *char-*

riot realmente chegou à posição pretendida. Os fluxogramas de funcionamento das várias secções encontram-se representados no Anexo C.

Na rotina *Psengate*, cujo fluxograma se encontra representado no Anexo C, é realizada a gestão de acesso ao interior da célula. Na ocorrência de um pedido de acesso por parte do operador, o programa verifica se o robô se encontra em posição segura e parado e, em caso afirmativo, concede o acesso ao interior da célula destrancando a *psengate*.

O programa OS32C realiza a comutação das zonas de monitorização dos *scanners* através da leitura dos interruptores de fim de curso. A comutação das zonas é realizada através da transferência de um sinal digital para os *scanners*.

Já a secção *IOL_Data* realiza a leitura dos sensores com interface IO-Link. Para a leitura destes sensores é utilizado o *Function Block IOL_ReadObj* da biblioteca da Omron. Para tal é necessário preencher as entradas do FB com tipos de dados específicos ao IO-Link. Nos sensores com interface IO-Link, todos os dados que podem ser lidos têm relacionado um identificador composto por um *index* e *sub-index*. Este identificador deve ser passado através da estrutura interna do *software _sIOLOBJ_ACCESS*. Outra informação que deve ser passada ao FB é a localização do sensor na rede e que tipo de carta de expansão faz a sua leitura, através da estrutura interna *_sDEVICE_PORT*. Após a leitura do sensor é realizada uma operação de verificação do valor obtido antes de este ser enviado para o manipulador. Assim sendo, é verificado se o valor lido no sensor se encontra dentro dos limites mínimo (altura de 1 paleta) e máximo (altura de 10 paletes) que a pilha de paletes vazias deve compreender. No caso de o valor retornado pelo sensor se encontrar dentro do intervalo estabelecido o valor é enviado para o manipulador, caso contrário o programa do robô é abortado e é apresentada uma mensagem de erro na interface HMI.

O programa *FANUC_Program_Job_Control* realiza a gestão de quais programas (*AUTOSEQ* ou *MAINTENANCE*) e tarefas (rotinas de *pick* e *place*) são carregadas para o robô. O programa inicia com a verificação de arranque do robô; se o seu arranque foi realizado corretamente é iniciada a lógica de passagem de tarefas. Após a confirmação inicial de que o robô arrancou corretamente, o programa apura se foi realizado um pedido de manutenção por parte do operador na consola. Em caso afirmativo é carregado o programa *MAINTENANCE* para o robô e em caso negativo é carregado o programa *AUTOSEQ*. No caso de ser carregada a rotina *AUTOSEQ*, através da leitura das saídas dos vários FB das mesas e dos tapetes, o programa decide qual tarefa deve ser enviada para o robô. Assim sendo, como exemplo, se o transportador de armazenamento da pilha de paletes da L1 indicar que está ocupado e o transportador de paletização da L1 indicar que está livre, o programa transfere para o robô a tarefa responsável por realizar o *pick* e *place* de uma paleta na linha 1. No Anexo C encontra-se ilustrado

o fluxograma de funcionamento deste programa.

A secção seguinte, FANUC_Coms, é composta por três programas distintos. O programa PLC_FANUC_Registers lê os parâmetros da receita carregada pelo operador através da interface homem-máquina e transfere os valores para variáveis de trabalho do PLC. O operador, através da consola, indica ao sistema qual o tipo de mosaico que pretende produzir, e a cada mosaico estão relacionados valores, como a altura da caixa, o número máximo de camadas que o robô deve fazer para esse mosaico e a referência do produto. A rotina PLC_IO_Gripper, através da leitura do estado do atuador final, atua sinais que serão enviados ao controlador do robô com o objetivo de indicar se o atuador final se encontra aberto ou fechado. Os sinais de ambos os programas são carregados para variáveis de trabalho e, posteriormente, enviados para o controlador do robô através do programa PLC_FANUC_TransferData. Este programa move o valor dos sinais e das variáveis de trabalho para as tramas de comunicação que serão transferidas para o controlador do manipulador por EthernetIP.

O programa HMI é constituído por várias secções. Cada secção é responsável por transferir para variáveis de comunicação com a consola o estado dos vários módulos do sistema, incluindo informação sobre os sensores ou condições de alarme que possam ocorrer. Se, por exemplo, a unidade de controlo receber o sinal de pedido de abertura do atuador final do robô e verificar que este não se encontra aberto ao fim de um tempo pré-definido, é despoletado um alarme para a consola com o objetivo de indicar ao operador que fenómeno se encontra a bloquear o sistema.

Finalizando, a secção *Outputs* atua as saídas físicas dos vários módulos, dependendo dos sinais de saída dos FB quando o sistema se encontra em automático, ou dos comandos manuais dados pelo operador através da consola quando o sistema se encontra em manual.

6.5 Interface Homem-Máquina

Como mencionado no Capítulo 5, o sistema possui uma consola de interface que permite ao operador realizar o controlo e monitorização da célula. Assim, nesta secção pretende-se descrever as funcionalidades de cada janela da consola de interface homem-máquina. Algumas das funcionalidades implementadas na consola permitem ao operador visualizar o estado geral do sistema, índices de produção, carregar receitas de produção, visualizar a lista de alarmes ocorridos, realizar o comando manual de cada componente, entre outros. Os vários elementos colocados em cada janela, como botões, tabelas, ou indicadores de estado, foram mapeados na memória do PLC, permitindo uma comunicação contínua entre a unidade de controlo e a consola.

A janela HOME, representada na Figura 6.22, apresenta dados como o estado atual do sistema, o utilizador com sessão iniciada, os índices de produção para cada linha de produção, permite reiniciar os contadores de produção e realizar um pedido de fim de produção.

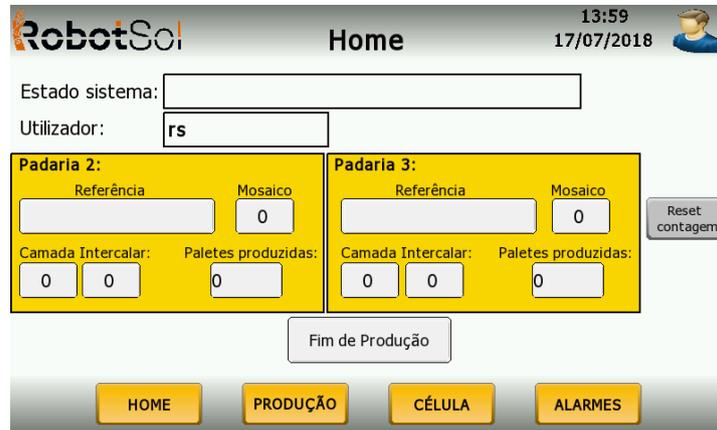


Figura 6.22: Janela HOME da consola.

Na janela Seleção de Produção, ilustrada na Figura 6.23, o operador é capaz de seleccionar qual a referência do mosaico a produzir e indicar se o robô deve colocar, ou não, a camada intercalar. Se a colocação de intercalar for ativada, o operador deve indicar quais as camadas que irão receber o intercalar. Através desta janela é também possível aceder ao menu de configuração de receitas, onde o operador tem a possibilidade de adicionar novas receitas ou editar as já existentes.

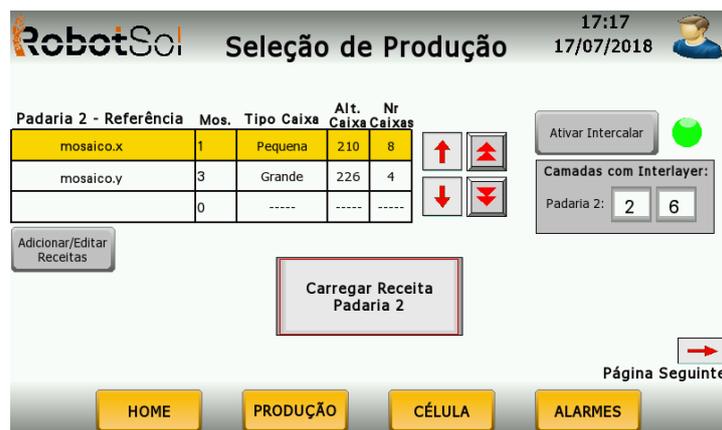


Figura 6.23: Janela Seleção de Produção da consola.

A janela Estados Célula, ilustrada na Figura 6.24, apresenta o *layout* do sistema e possibilita ao operador selecionar um componente da célula com o objetivo de verificar o seu estado. Tocando sobre um dos componentes, o operador será redirecionado para a página de estados do respetivo equipamento.



Figura 6.24: Janela Estados Célula da consola.

Na Figura 6.25 encontra-se representada a janela de estado de uma das mesas formadoras de mosaicos. Estes ecrãs permitem ao operador verificar o estado dos sensores de cada componente, bem como a ordem que o PLC está a enviar através dos sinais das eletroválvulas. Assim, no caso de alguma anomalia durante o funcionamento da mesas, é possível apurar, por exemplo, se um dos *stoppers* não recuou por não ter recebido ordem do PLC, ou por causa de algum travamento mecânico.

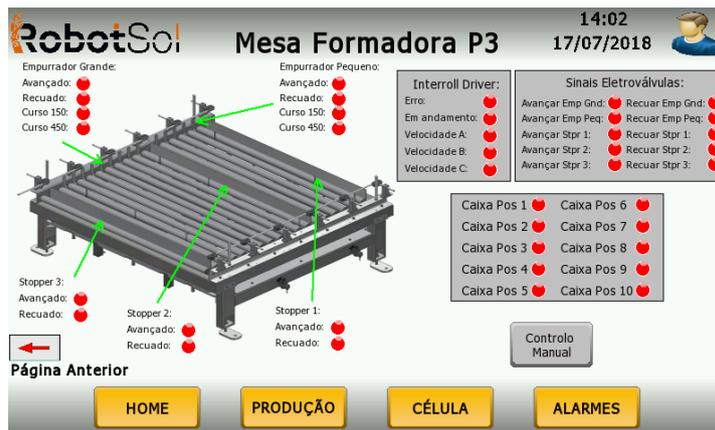


Figura 6.25: Janela de estado da mesa formadora de mosaicos.

Com o objetivo de facilitar o processo de manutenção, foram implementados ecrãs que permitem o comando manual de cada equipamento. Estes ecrãs

possibilitam ao operador movimentar de forma individual os vários componentes de cada equipamento. Na Figura 6.26 encontram-se representadas as janelas de comando manual das mesas formadoras de mosaico.



Figura 6.26: Janela de comando manual da mesa formadora de mosaicos.

Na janela Alarmes, ilustrada na Figura 6.27, encontra-se representado um historial de todos os alarmes ocorridos até à data. O historial apresenta a data e hora de ocorrência do alarme, a hora de visualização por parte do utilizador, a hora de cancelamento do alarme e a sua descrição. Os alarmes podem indicar a ocorrência de uma situação de emergência, ou até um comportamento não conforme num dos vários componentes da célula, como por exemplo, um dos sistemas de manipulação do atuador final não fechar corretamente.



Figura 6.27: Janela de historial de alarmes.

Por fim, outra funcionalidade implementada na consola é a possibilidade de adição/remoção de novos utilizadores. Esta opção permite ao cliente criar uti-

- Método 1: instalação da barreira de forma vertical, onde a zona de detecção é ortogonal à direção de aproximação.

Neste método a distância de segurança é calculada segundo a Equação 6.1, sendo o parâmetro C dado pela Equação 6.2. Os restantes parâmetros de cada equação e as respetivas descrições encontram-se representados na Tabela 6.3.

$$S = K \times T + C \quad (6.1)$$

$$C = 8 \times (d - 14) \quad (6.2)$$

Assim, substituindo o valor de T e de C obtém-se a Equação 6.3.

$$S = K \times (T_s + T_m) + 8 \times (d - 14) \quad (6.3)$$

- Método 2: instalação da barreira de forma vertical, onde a zona de detecção é ortogonal à direção de aproximação e existe a possibilidade de a zona perigosa ser alcançada passando por cima da zona de detecção.

Este método entra em consideração com a possibilidade de o operador alcançar a zona de perigo por cima da barreira. Neste cenário, o cálculo da distância de segurança deve ser feito através da Equação 6.4. O valor resultante deve ser comparado com o resultado calculado através do Método 1 e deve ser considerada a distância de segurança maior.

$$S = K \times (T_s + T_m) + Cro \quad (6.4)$$

Tabela 6.3: Parâmetros das equações do cálculo da distância de segurança

| Parâmetro | Descrição |
|------------|--|
| S [mm] | Distância de segurança |
| K [mm/s] | Velocidade de aproximação |
| T [s] | Tempo total de paragem do sistema |
| C [mm] | Distância adicional dependente da resolução da barreira |
| d [mm] | Capacidade de deteção da barreira |
| Ts [s] | Tempo de resposta da barreira |
| Tm [s] | Tempo de resposta da máquina/robô |
| Cro [mm] | Distância adicional determinada com base na altura da zona perigosa e na altura da borda superior da zona de deteção. O seu valor é tabelado e pode ser consultado no manual de operação da barreira |

Com o objetivo de diminuir ao máximo a distância de segurança necessária por motivos de pouco espaço de instalação, definiu-se a utilização de uma barreira que possua a borda superior da zona de deteção a uma altura de aproximadamente 2200 mm. Desta forma o operador não é capaz de alcançar o perigo por cima da barreira e não tem de ser considerada a distância adicional Cro prevista no Método 2. Visto que a barreira pode ser instalada a uma altura máxima de 300 mm do chão, o modelo escolhido possui aproximadamente 1900 mm de altura. Assim, aplicando o Método 1 foram realizados os cálculos da distância de segurança para barreiras com capacidade de deteção de 14 mm e 30 mm. Seguindo o exemplo do manual de operação da barreira e considerando uma velocidade de aproximação de 2000 mm/s, um tempo de resposta da barreira de 0,008 s e um tempo de resposta do robô de 0,644 s obtiveram-se os resultados das Equações 6.5 e 6.6.

$$S = 2000 \times (0,008 + 0,644) + 8 \times (30 - 14) = 1432 \text{ mm} \quad (6.5)$$

$$S = 2000 \times (0,008 + 0,644) + 8 \times (14 - 14) = 1304 \text{ mm} \quad (6.6)$$

Visto em ambos os casos os resultados excederem os 500 mm, segundo o manual é possível considerar uma velocidade de aproximação de 1600 mm por

segundo, obtendo-se os resultados das Equações 6.7 e 6.9

$$S = 1600 \times (0,008 + 0,644) + 8 \times (30 - 14) = 1171 \text{ mm} \quad (6.7)$$

$$S = 1600 \times (0,008 + 0,644) + 8 \times (14 - 14) = 1043 \text{ mm} \quad (6.8)$$

Analisando os resultados obtidos e o espaço disponível de instalação, foi possível implementar uma barreira com uma altura de aproximadamente 1900 mm e uma capacidade de detecção de 30 mm, instalada a uma distância de segurança de aproximadamente 1171 mm, como obtido na Equação 6.7.

No que toca à funcionalidade de *muting* da barreira, esta pode ser implementada através de vários métodos, dependendo da aplicação. No caso particular deste sistema, o método adotado encontra-se representado na Figura 6.29. Este método utiliza quatro sensores fotoelétricos e permite que o *charriot* intercete a barreira, independentemente da direção com que se esteja a mover.

A funcionalidade de *muting* só é ativada se a sequência de interceção dos sensores seguir um padrão. Desta forma, quando os sensores A1 e B1 são ativados por esta ordem, o estado de *muting* é habilitado e a barreira desativa as suas funções de segurança. Dada a configuração de instalação, quando os sensores A1 e B1 ficam desativados os sensores A2 e B2 comutam de estado, mantendo-se a funcionalidade de *muting* ativada. No momento em que o sensor B2 não deteta o *charriot*, o estado de *muting* é libertado e a barreira ativa as suas funcionalidades de segurança novamente.

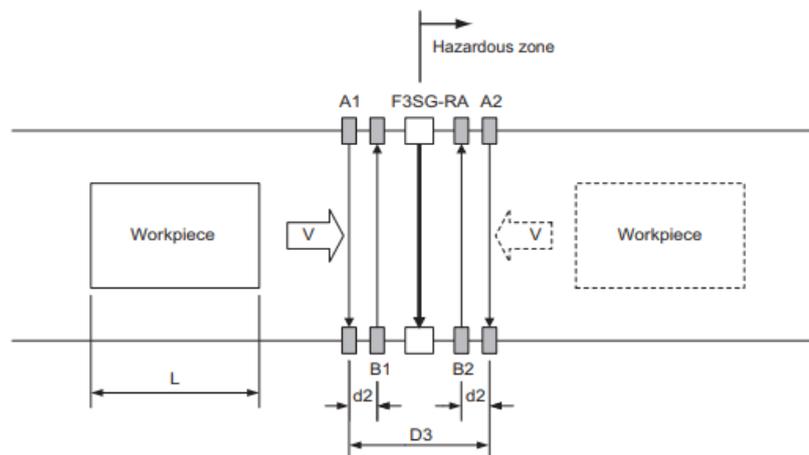


Figura 6.29: Método de *muting* implementado na barreira de segurança.

Para que este método funcione, as distâncias de instalação entre os sensores devem respeitar as Equações 6.9 e 6.10, sendo que os respectivos parâmetros encontram-se descritos na Tabela 6.4.

$$D3 < L \quad (6.9)$$

$$V \times T1min < d2 < V \times T1max \quad (6.10)$$

Substituindo os valores, obtém-se os resultados das Equações 6.11 e 6.12

$$D3 < 1,5m \quad (6.11)$$

$$0,4 \times 0,1 < d2 < 0,4 \times T1max \Leftrightarrow 0,04m < d2 < 1,6m \quad (6.12)$$

Tabela 6.4: Parâmetros das equações do cálculo da distância de instalação dos sensores de *muting*

| Parâmetro | Descrição |
|-------------|---|
| $D3$ [m/s] | Distância mínima necessária para os sensores manterem a função de <i>muting</i> ativada |
| $d2$ [m/s] | Distância máxima necessária para os sensores manterem a função de <i>muting</i> ativada |
| L [m] | Comprimento do objeto |
| V [m/s] | Velocidade de aproximação do objeto |
| $T1min$ [s] | Tempo limite mínimo da funcionalidade de <i>muting</i> (configurável) |
| $T1max$ [s] | Tempo limite máximo da funcionalidade de <i>muting</i> (configurável) |

6.6.2 Scanners de Segurança

Os *scanners* de segurança instalados no *charriot* devem também ser configurados. Tal como foi abordado na Subsecção 5.3.7, estes possuem vários modos de operação e zonas de monitorização configuráveis. A configuração dos *scanners*

é realizada através do *software OS32C Configuration Tool* da Omron. Através deste *software*, é permitido escolher o modo de operação do equipamento, o número de zonas a monitorizar, o tempo de comutação entre zonas, a resolução dos objetos a detetar, entre outros parâmetros.

Os *scanners* implementados foram configurados com o modo de operação *Automatic Start*, permitindo o rearme automático do estado de segurança sempre que o sensor deixa de detetar algum objeto na zona de monitorização, uma monitorização para múltiplas zonas, neste caso duas zonas para cada *scanner* e a comutação do estado da saída auxiliar quando a lente do sensor se encontrar contaminada de forma a indicar ao operador que esta deve ser limpa para assegurar o correto funcionamento do dispositivo.

Na Figura 6.30 encontram-se ilustradas as zonas de monitorização configuradas nos *scanners*. A zona de monitorização 1 é a utilizada quando o *charriot* se encontra na área onde pode ocorrer a passagem de operadores. Esta zona possui um perfil horizontal grande com o objetivo de detetar algum operador nas imediações do corredor. A zona de monitorização 2 possui um perfil horizontal mais estreito pois é utilizada quando o *charriot* se encontra dentro da célula rodeado de vedações. Nesta situação a zona de monitorização dos *scanners* deve ser mais estreita, evitando que o dispositivo entre em estado de segurança por detetar uma vedação. A configurações das zonas de monitorização teve em conta as dimensões da célula e do corredor.

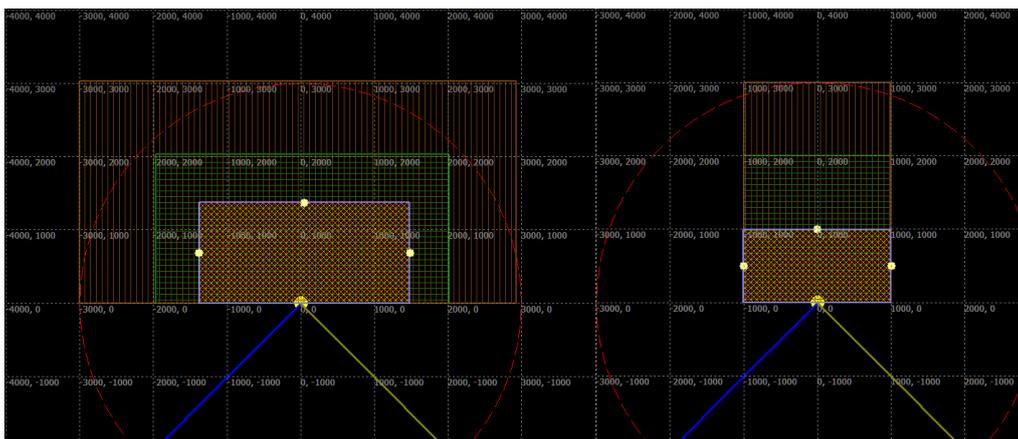


Figura 6.30: Zonas de segurança configuradas nos *scanners*: zona de monitorização no corredor de passagem de operadores (esquerda) e zona de monitorização dentro da célula (direita).

6.6.3 Programação do PLC de Segurança

O PLC de segurança é o dispositivo que irá realizar a gestão de todos os equipamentos de segurança da célula. A sua programação é integrada no *Sysmac Studio*, o que possibilita a troca de variáveis entre os dois PLC. A única configuração necessária no PLC de segurança é o mapeamento dos sinais nas cartas de entrada e saída. Através do menu *Configuration and Setup - Safety - Safety I/O* é possível atribuir um periférico de segurança aos pontos de entradas e saídas das cartas de segurança, como se pode observar na Figura 6.31.

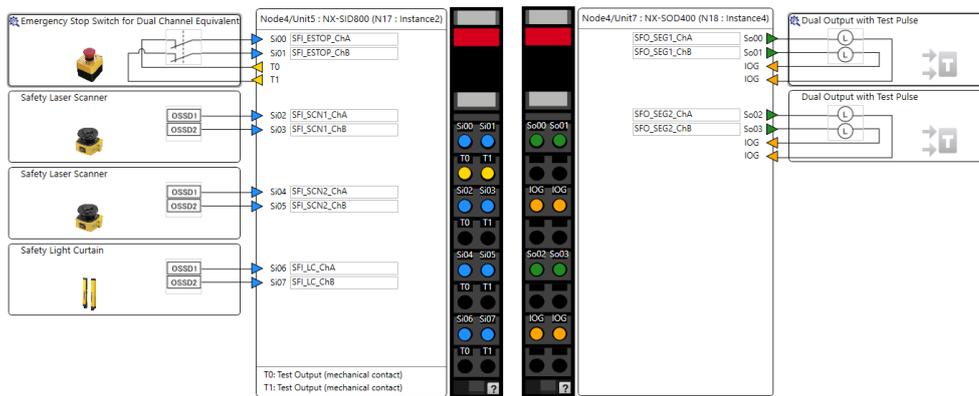


Figura 6.31: Configuração das cartas de entradas (esquerda) e saídas (direita) de segurança.

Assim, após a configuração das cartas e do mapeamento dos sinais das botoneiras de emergência, da barreira luminosa, dos *scanners*, dos interruptores de fim de curso e do botão de rearme, o passo seguinte foi a programação da lógica de segurança do PLC. O método de programação baseia-se na utilização de *Function Blocks* de segurança incluídos na biblioteca da Omron. Desta forma, para programar o cenário de ocorrência de emergência quando pressionada uma botoneira de segurança, ou para programação do cenário de *muting* da barreira de segurança, basta chamar os blocos correspondentes a cada função e completar com os sinais de entrada e de saída, como ilustrado na Figura 6.32.

As saídas dos *Function Blocks* chegam aos relés de segurança e no caso de uma situação de emergência os relés atuam diretamente nos equipamentos necessários, como, por exemplo, os inversores que controlam o movimento dos tapetes ou no circuito de emergência do controlador do robô, obrigando à paragem do mesmo.

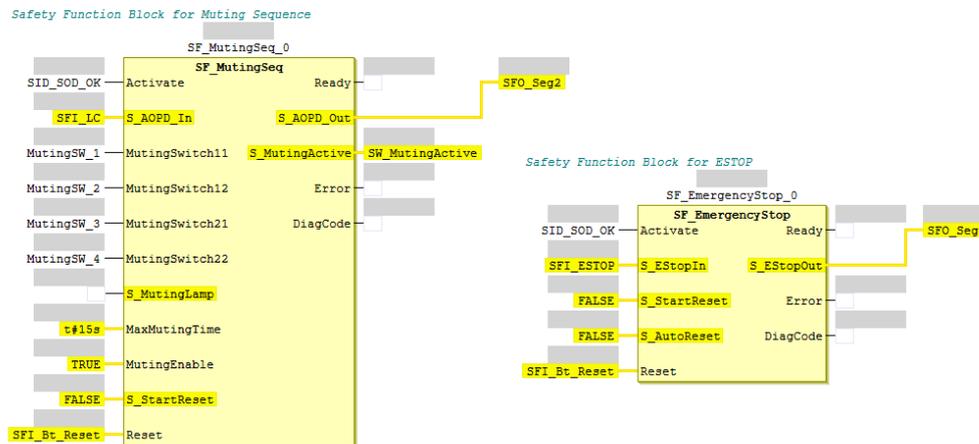


Figura 6.32: *Function Block* de segurança para a sequência de *muting* (esquerda) e *Function Block* de segurança para botoneiras de emergência (direita).

6.7 Conclusão

Em suma, este capítulo teve como objetivo explicar todos os passos seguidos ao longo da programação e implementação do sistema. Numa primeira fase foi abordada a simulação *offline* realizada e os pontos analisados que tiveram interferência direta na constituição da célula e na programação do robô. De seguida foram abordadas as etapas de configuração dos vários parâmetros do manipulador, bem como as rotinas criadas e a lógica de funcionamento. Quando à unidade de controlo, foram também enumeradas as configurações efetuadas e os programas criados para efetuar o controlo do sistema. Em relação à consola de interface homem-máquina foram apresentadas as várias janelas e as devidas funcionalidades implementadas. Finalizando com o PLC de segurança e os respetivos equipamentos, foram explicadas as regras que se devem seguir no processo de escolha e instalação de uma barreira luminosa de segurança, bem como a configuração das zonas de monitorização dos *scanners*. Quanto ao PLC de segurança, foram evidenciadas as configurações necessárias e a lógica de programação através dos *Function Block* de segurança incluídos na biblioteca da Omron. Assim, é possível concluir que a implementação prática é um processo moroso e que deve ser realizado com cuidado, visto que podem ser gerados erros que possam colocar em risco a concretização do projeto.

Capítulo 7

Testes e Resultados

O presente capítulo tem como finalidade expor todos os testes efetuados e os devidos resultados. Após a simulação e programação *offline* do manipulador, foram realizados vários testes às trajetórias programadas no manipulador e à comunicação e interface com o PLC. Em relação aos vários módulos constituintes da célula, foram conduzidos testes de funcionamento individuais com o objetivo de aprovar o conceito mecânico e o *software* desenvolvido. Após validação dos vários equipamentos, a etapa seguinte é a sua interligação de forma a testar o controlo geral de lógica e funcionamento da célula.

7.1 Simulação Offline do Código do Robô

Um dos primeiros passos realizados na etapa de testes foi iniciar a validação das rotinas desenvolvidas para o robô. Através das capacidades de simulação do Roboguide, as rotinas foram executadas passo-a-passo com o objetivo de validar a lógica de controlo do programa implementado, os cálculos de *offset* dos pontos e as trajetórias do manipulador.

Visto que as rotinas necessitam de sinais e variáveis do PLC, na fase de simulação recorreu-se à funcionalidade *I/O Panel* do Roboguide. Esta funcionalidade permite simular os sinais de entrada digitais do robô, como também os sinais de grupo, que neste caso determinam qual o programa e tarefa a executar.

Assim sendo, na Figura 7.1 encontra-se representada a trajetória de *pick* e *place* de uma palete, obtida na simulação. Analisando a Figura 7.1, na fase de arranque e início da execução do programa o robô verifica a sua posição atual e executa a recuperação automática para a posição de *home* (secção mais à direita da trajetória). Após assegurar essa posição, inicia o processo de *pick* e *place*

dos vários produtos, sendo neste caso de uma paleta (secção mais à esquerda da trajetória).

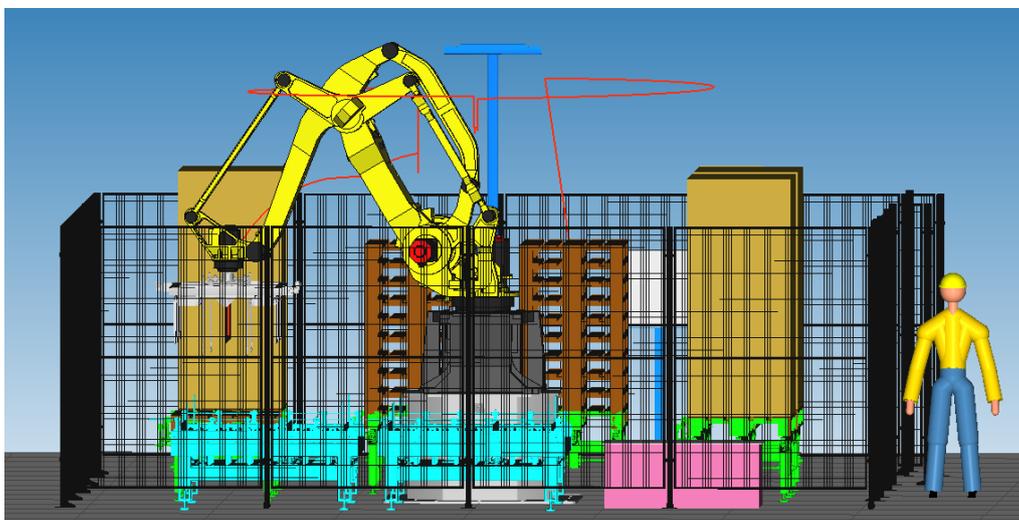


Figura 7.1: Trajetória de *pick* e *place* de paleta.

Para além da aprovação da lógica do programa e das trajetórias, os testes realizados permitiram a aprovação das configurações realizadas nos sistemas de coordenadas (*User Frame* e *Tool Frame*), nas posições de referência, na funcionalidade de *Background Logic* e no arranque do programa RSR através do envio dos sinais UI. No entanto, numa fase seguinte de implementação da célula no cliente, os sistemas de coordenadas *User Frame* e os pontos marcados necessitaram de ser reajustados, visto que as coordenadas foram obtidas em ambiente de simulação e poderão não corresponder às coordenadas reais.

7.2 Comunicação Robô-PLC

A etapa seguinte focou-se na validação da comunicação EthernetIP entre o controlador do robô e a unidade de controlo. Para tal, foi desenvolvido um programa de teste no PLC responsável por atuar os vários sinais de entrada mapeados no robô e que atribuisse valores às variáveis de registo. Na Figura 7.2 é possível verificar que o controlador do robô recebe corretamente os sinais digitais, bem como os valores introduzidos nos registos, através da conversão dos sinais DI para GI e, posteriormente, de GI para Registo.

Assim sendo, foi possível validar as configurações realizadas do lado do robô no que toca ao mapeamento dos sinais digitais (DI e GI) e à comunicação EthernetIP, sobretudo na atribuição dos endereços de rede e no tamanho das tramas de comunicação. Do lado do PLC foi possível validar a configuração de rede e o

tamanho das tramas de comunicação introduzidas no *software Network Configurator*. De forma análoga, também os sinais digitais de saída do robô (DO) foram testados, encontrando-se devidamente mapeados e recebidos pelo PLC.

| # | SIM | STATUS | |
|----------|-----|--------|--------------------|
| DI[173] | U | OFF | [Error_Code] |
| DI[174] | U | OFF | [Error_Code] |
| DI[175] | U | OFF | [Error_Code] |
| DI[176] | U | OFF | [Error_Code] |
| DI[177] | U | ON | [F_IN_Grp_Box_Ope] |
| DI[178] | U | OFF | [F_IN_Grp_Box_Clo] |
| DI[179] | U | OFF | [F_IN_Grp_Fork_Op] |
| DI[180] | U | ON | [F_IN_Grp_Fork_C1] |
| DI[181] | U | ON | [F_IN_Grp_Air_Ope] |
| DI[182] | U | OFF | [F_IN_Grp_Air_Clo] |
| DI[183] | U | OFF | [CMD_Gripper] |

| R[| Value |
|--------------------|--------|
| 5: |]=0 |
| 6: |]=0 |
| 7: |]=0 |
| 8: |]=0 |
| 9: |]=0 |
| 10:L1_LayerCounter |]=8 |
| 11:L1_TotalLayers |]=8 |
| 12:L1_LayerHeight |]=265 |
| 13:IN_1_PalHeight |]=2000 |
| 14:IN_1_ProdRef |]=4 |
| 15: |]=0 |

Figura 7.2: Teste de comunicação de sinais digitais e registos entre o robô e o PLC.

7.3 I/O Check

A validação dos sinais de entrada e saída, processo conhecido como *IO Check*, é uma tarefa que deve ser realizada na fase inicial da implementação prática do sistema com o objetivo de verificar e validar todas as ligações elétricas e o mapeamento dos sinais do lado da unidade de controlo.

Assim sendo, para realizar o *IO Check* foi necessário atuar de forma individual todos os sinais de entrada e de saída do PLC, como por exemplo, sensores, botões ou eletroválvulas. Durante o teste verificou-se que alguns sinais não chegavam corretamente à unidade de controlo ou que algumas saídas não eram atuadas devido a erros de ligações elétricas, que tiveram de ser posteriormente corrigidas. Após as várias correções elétricas, concluiu-se que o PLC era capaz de receber todos os sinais de entrada e atuar as respetivas saídas de forma correta.

Este processo permitiu validar a configuração e implementação da rede EtherCAT entre o PLC, as cabeceiras e os módulos pneumáticos SMC e o mapeamento dos vários sinais nas restantes cartas do PLC. Também nesta fase foram testados os controlos manuais de cada módulo através do HMI, como por exemplo, o avanço e recuo dos tapetes ou dos cilindros do atuador final, permitindo validar o mapeamento das variáveis de comunicação entre o PLC e a consola de interface homem-máquina. Na Figura 7.3 é possível observar a receção dos sinais dos sensores do atuador final na consola.

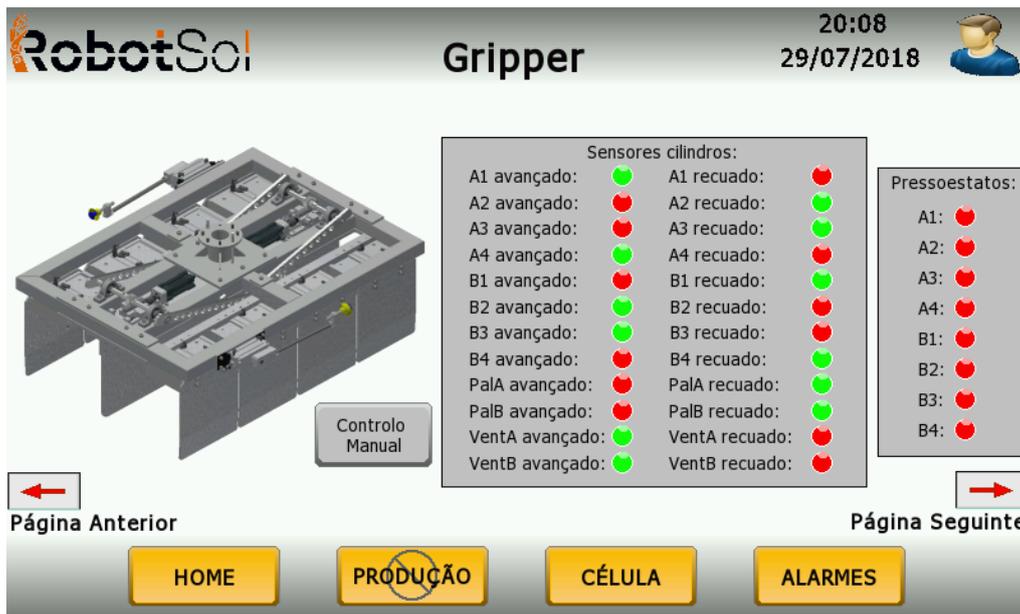


Figura 7.3: Receção dos sinais dos sensores do atuador final na consola HMI.

7.4 Sequência de Formação de Mosaicos e Transferência de Paletes

Posteriormente à validação da configuração de rede e do mapeamento dos sinais, iniciaram-se os testes de lógica de funcionamento individual de cada módulo da célula.

Assim sendo, foi testada a sequência de transferência de palete entre os transportadores e a sequência de formação de mosaico para as várias referências de caixas. Os resultados finais foram os esperados, tendo sido necessárias pequenas correções de *software* no decorrer dos testes e alguns ajustes pneumáticos no que toca à velocidade de avanço e de recuo dos empurradores das mesas. Para além da validação da lógica de trabalho dos vários transportadores e das mesas de formação de mosaico, estes testes permitiram validar a funcionalidade de seleção de receita de produção introduzida na consola de interface e recebida pelo PLC.

7.5 Movimentação do Robô e Manipulação de Objetos

Terminadas as validações iniciais de comunicações, mapeamento dos sinais e da movimentação do manipulador e trajetórias programadas, através do *software* de simulação, iniciaram-se os testes em ambiente real a baixa velocidade. Realizaram-se testes aos valores das coordenadas, obtidas em ambiente de simu-

lação, das *Uframes* e dos pontos marcados. Como esperado, verificou-se que as coordenadas não correspondiam exatamente à realidade, pelo que foi necessário refazer o processo de marcação das *Uframes* e realizar o ajuste (*touchup*) dos pontos de *pick* e de *place*. Após as correções, procedeu-se à execução das várias trajetórias e os resultados obtidos foram os esperados, com o manipulador a alcançar os pontos programados corretamente.

Também nesta fase, foram testadas as ligações do circuito de emergência do robô. A paragem do manipulador, em caso de emergência, é um processo que deve ser testado e assegurado pelo implementador, pois no caso de ocorrência de erros podem ser originadas situações que coloquem em risco a integridade física do operador. Assim sendo, enquanto eram executadas as rotinas de movimento do robô, foram atuadas as botoneiras de emergência e, tal como esperado, o manipulador interrompeu o seu movimento.

Terminados os testes das trajetórias em vazio, ou seja, sem utilizar o produto, o passo seguinte foi o teste de manipulação dos produtos. No caso da palete e da camada intercalar, os testes realizados apresentaram bons resultados sendo que o robô foi capaz de manipular o produto sem problemas. Em relação ao mosaico de caixas, os primeiros testes apresentaram piores resultados, visto que devido ao peso das caixas o atuador final não era capaz de conferir aderência suficiente à face da caixa, permitindo que esta deslizasse. De forma a resolver este problema foi adicionado no atuador final uma tela, como se pode observar na Figura 7.4, com o objetivo de aumentar a aderência do atuador final ao mosaico. Após a adição da tela e alguns ajustes pneumáticos, o robô é capaz de manipular o mosaico de caixas sem qualquer tipo de problema.

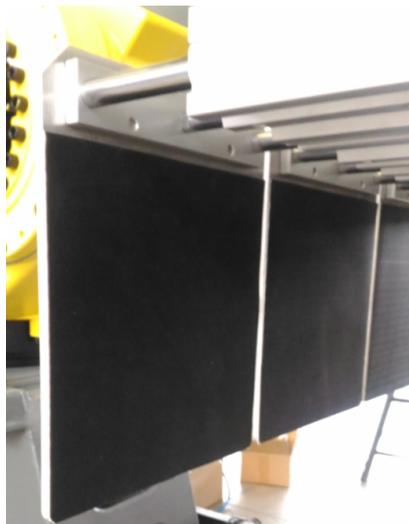


Figura 7.4: Tela adicionada ao sistema de manipulação do atuador final.

7.6 Seguranças

Como referido na Secção 7.5, um dos testes realizados foi a validação das ligações do circuito de segurança do controlador do robô. Contudo, foi também necessário testar os restantes cenários de segurança e a configuração e programação da barreira e dos *scanners* de segurança.

Assim sendo, para além de se ter verificado a paragem do robô aquando o pressionar de uma botoneira de emergência, verificou-se também a paragem dos tapetes e do corte geral pneumático para impedir o movimento do atuador final e das mesas formadoras de mosaicos. Os testes apresentaram resultados positivos, com a paragem dos transportadores e do corte de ar, impedindo a movimentação dos cilindros do atuador final e das mesas.

Em relação à barreira de segurança foram realizados os testes da sequência de *muting* sem *charriot*. Como esperado, visto que não era o *charriot* a executar a sequência de interrupção dos sensores, quando os feixes da barreira eram interrompidos esta entrava no modo de emergência despoletando uma paragem dos vários componentes da célula.

No que toca à fechadura eletrónica, foram efetuados testes de pedido de acesso à célula enquanto o robô executava uma tarefa. Como esperado, só depois de o manipulador terminar a sua tarefa (realizar o *pick* e o *place* do objeto) e depois de regressar à posição *home*, o acesso à célula era permitido. Após o operador sair, trancar a porta e carregar no botão de *start* o sistema continuava com o seu ciclo de trabalho.

Quanto aos *scanners*, os testes realizados consistiram na violação das zonas de monitorização, através da colocação de um objeto no percurso. Assim sendo, foi verificado que quando o objeto se encontrava dentro das zonas de aviso a frequência do inversor diminuía, representando a diminuição de velocidade de movimentação do *charriot*. Assim que o objeto passava para dentro da zona de segurança, era atuada a saída de segurança do *scanner* e o inversor entrava no estado de emergência, colocando o seu valor de frequência a zero, simulando a paragem de movimento do *charriot*.

Desta forma, os testes realizados permitiram validar a configuração e programação do PLC de segurança, validar os resultados calculados para a instalação dos sensores de *muting* da barreira de segurança e, por fim, a configuração das zonas de monitorização dos *scanners*.

7.7 Limitações de movimentos

Após validar o ciclo normal de funcionamento da célula, procederam-se os testes que permitiram concluir se as configurações realizadas ao nível das limitações do

manipulador por *software* foram realizadas de forma correta.

Assim sendo, com o objetivo de validar a limitação por *software* efetuada ao nível das juntas, tentou-se deslocar o manipulador contra um pilar presente no interior da célula e, quando o manipulador se aproximou deste, parou o seu movimento. Em relação à configuração das *space functions*, realizou-se a movimentação do robô no sentido de colidir com as redes de vedação, com o sistema de ventilação e com os próprios constituintes da célula (mesas e transportadores). Em todos os casos, o manipulador parou o seu movimento, sendo apresentada a mensagem "erro de limites" na sua consola.

7.8 Arranque do Sistema e Controlo de Funcionamento

Concluídos os vários testes de funcionamento individual dos módulos da célula, o passo seguinte é ensaiar o arranque do sistema em modo automático. Esta etapa deve ser realizada com bastante precaução, visto que todos os equipamentos da célula irão funcionar em simultâneo. Para tal, deve ser seguido um conjunto de passos, de forma a assegurar todas as condições de arranque evitando erros. Assim sendo, de modo a arrancar o sistema corretamente o operador deve:

1. assegurar que a célula se encontra livre de pessoas e os módulos sem produto;
2. verificar o estado dos dispositivos de segurança do sistema, como por exemplo as botoneiras;
3. colocar o modo de funcionamento do robô em AUTO;
4. seleccionar o modo de funcionamento do sistema (automático ou semi-automático) através do seletor posicionado no painel de interface com o operador;
5. carregar o tipo de produção através do HMI;
6. pressionar o botão *start*.

Efetuados estes passos, o sistema inicia a sequência de arranque fazendo a verificação de todas as condições necessárias para iniciar a produção em modo automático. Numa primeira fase o sistema arrancava corretamente mas bloqueava no momento em que o PLC envia ao robô qual a tarefa que deve executar. Este erro era consequência da lógica implementada no controlo de gestão de tarefas do *software*. O erro foi facilmente corrigido, pois a filosofia utilizada para a implementação de grande parte do código do PLC foi a utilização de máquinas de estados, permitindo identificar qual o estado de bloqueio da máquina. Após esta correção e a otimização do código no que toca à troca de sinais entre o robô e o PLC, o sistema arranca corretamente e todos os módulos executam as devidas tarefas, como esperado.

No que toca ao controlo de funcionamento geral da célula, visto que a sequência de funcionamento já tinha sido validada de forma individual para cada módulo, não foram encontrados grandes erros que bloqueassem o funcionamento do sistema. Surgiram apenas algumas falhas menores no que toca ao tratamento de variáveis auxiliares e atuação de algumas saídas no estado errado da máquina de estados implementada para cada sistema.

7.9 Conclusões

Concluindo, neste capítulo foram abordados os vários testes conduzidos para a validação do sistema. A filosofia de validação do sistema baseou-se em confirmar o funcionamento individual de cada módulo, partindo depois para o teste geral da célula. De forma geral, os testes apresentaram os resultados esperados surgindo, como seria de esperar, alguns erros de implementação e programação. Contudo, os erros foram facilmente detetados e corrigidos dado a programação ter sido realizada no conceito de máquinas de estados, permitindo averiguar de forma rápida qual o estado que bloqueava o sistema ou que exercia uma ação errada.

Capítulo 8

Conclusões

A presente dissertação teve como principal objetivo a programação e implementação de uma célula robotizada para uma aplicação de paletização de caixas. Não sendo apenas a robótica o foco principal do projeto, este trabalho motivou um estudo alongado de temas relacionados com tecnologias e processos utilizados ao nível da automação industrial e o método de interligação com a área da robótica e o processo de paletização.

Assim, neste capítulo será descrito o balanço entre os objetivos iniciais propostos e os resultados obtidos, bem como os principais obstáculos encontrados. A finalizar o capítulo serão mencionadas algumas perspetivas de futuras melhorias ao projeto.

8.1 Solução Adotada

Apresentados e analisados os requisitos do sistema, a solução adotada baseou-se na utilização de um manipulador robótico de 4 eixos em conjunto com um alargado leque de equipamentos (transportadores, unidade de controlo, equipamentos de segurança, mesas de mosaicos, entre outros) formando uma célula capaz de realizar o processo de paletização.

Dada a complexidade do sistema e o número de equipamentos a controlar em simultâneo, justificou-se o uso de um PLC como unidade de controlo do sistema em conjunto com um PLC de segurança, responsável apenas pela gestão dos equipamentos de segurança, assegurando o cumprimento das normas de segurança. Ambos os modelos escolhidos cumprem com as necessidades do sistema oferecendo flexibilidade de expansão em caso de necessidade. A utilização do protocolo de comunicação EtherCAT fornece também ao sistema um elevado nível de flexibilidade e expansibilidade, visto que é possível conectar em cada cabeceira

EtherCAT sessenta e três cartas de expansão, para além das elevadas velocidades de transmissão de dados.

Apesar de o robô possuir apenas quatro eixos, o que leva a maiores restrições a nível de complexidade de trajetórias, cumpre os requisitos de uma aplicação de paletização que, de forma geral, não necessita de trajetórias complexas. No que toca à sua programação e configuração, é um processo de aprendizagem acessível visto que pode ser realizado através do simulador, e posteriormente carregado para o controlador do robô, ou diretamente através do *teach pendant*. Em relação ao nível de programação, esta apresenta algumas limitações quando comparadas com manipuladores de outros fabricantes, como por exemplo o simples facto de na mesma linha de código não poderem ser realizadas operações aritméticas diferentes.

Quanto ao atuador final, numa fase inicial surgiram problemas de manipulação do mosaico, contudo o problema foi resolvido com a adição da tela. O método de controlo do atuador final, através dos sensores de posição dos cilindros e dos pressóstatos, mostrou-se efetivo, permitindo diferenciar com facilidade quando o atuador final se encontrava a manipular um produto ou se encontrava vazio. Uma das alternativas ao atuador final mecânico seria a utilização de um atuador final com tecnologia de vácuo, contudo o consumo de ar comprimido seria bastante elevado, acarretando um custo adicional para o cliente.

As mesas formadoras apresentam-se como um sistema mecanicamente robusto, capaz de formar os mosaicos necessários. Contudo, no caso do cliente precisar de adicionar outro tipo de mosaicos na linha de produção, a mesa poderá apresentar algumas limitações dada a forma como foi mecanicamente concebida.

Em relação aos transportadores, o uso de correntes oferece uma melhor tração às paletes permitindo a sua transferência de forma eficaz.

No que toca ao sistema de transporte de paletes (*charriot*), devido a problemas internos por parte da RobotSol e atrasos nas datas prevista, o sistema não ficou pronto a tempo de ser testado na íntegra. Contudo é possível retirar algumas conclusões sobre a forma como foi concebido. A sua estrutura com painéis laterais e o uso de um sistema de prensa conferem-lhe uma maior segurança no transporte da paleta completa. Relativamente à sensorização do *charriot*, este não possui nenhum sistema capaz de detetar qual a carga que transporta, com o objetivo de assegurar que nenhum operador entra para o seu interior. Foi analisada uma solução desta natureza, que consistia em aplicar no *charriot* uma barreira de segurança projetada especificamente para aplicações de paletização, capaz de diferenciar uma paleta de outro tipo de objetos através da distância entre os tacos da paleta e os feixes luminosos da barreira interrompidos. Outra solução estudada seria realizar a monitorização da carga que o *charriot* transporta. Contudo, ambas as opções não foram implementadas devido a restrições

orçamentais. Em relação ao controlo de posição, a programação foi realizada, contudo a solução não foi testada devido aos atrasos da RobotSol, no que toca à montagem completa do sistema.

Abordando o tema da segurança, a solução apresentada cumpre com as necessidades exigidas pelo sistema. A implementação de uma barreira com funcionalidade de *muting* permite diferenciar se é um operador ou outro objeto que circula para dentro e fora da célula. No caso de ser o *charriot* a sequência de *muting* deve ser validada e o sistema continua em funcionamento; no caso de ser um operador a sequência de *muting* não é validada e o sistema entra em estado de emergência. Quanto aos *scanners* de segurança, estes oferecem uma grande flexibilidade visto que possuem uma área de monitorização grande e a possibilidade de monitorizar três áreas distintas (duas zonas de aviso e uma zona de segurança). Esta funcionalidade permite abrandar a velocidade do *charriot* no caso de ter sido violada a zona de aviso, possibilitando uma paragem do sistema mais suave em caso de emergência.

Em suma, é possível concluir que o objetivo principal do projeto, que consistia na programação de uma célula de paletização, foi cumprido. O sistema de transporte automático era uma funcionalidade adicional que foi pensada numa fase posterior do projeto. O núcleo da célula, composto pelo robô, transportadores e mesas cumpre os objetivos propostos executando as tarefas de forma correta.

8.2 Perspetivas Futuras

Terminada a implementação do sistema, fazendo uma análise aos resultados obtidos é possível identificar algumas alterações que favoreciam a qualidade e flexibilidade do projeto. Assim, algumas possíveis alterações são:

1. substituição das mesas formadoras de mosaicos por um sistema capaz de formar mais variedade de mosaicos, com o objetivo de aumentar a flexibilidade e capacidade de resposta da célula;
2. dotar o *charriot* de equipamentos capazes de diferenciar a carga que este transporta, assegurando-se que nenhum operador entra para o seu interior;
3. no atuador final, ao invés de um controlo pneumático, utilizar eixos elétricos com o objetivo de ser possível parametrizar os cursos de abertura e fecho, possibilitando a manipulação de um leque mais alargado de mosaicos;
4. utilização de um atuador final apenas com tecnologia de vácuo, facilitando o seu controlo e diminuindo a probabilidade de problemas mecânicos e consequentes manutenções;

5. implementação de mais funcionalidades na consola de interface, como por exemplo uma base de dados que guardasse um histórico de produção do sistema.

8.3 Conclusão

A possibilidade de ter realizado um projeto em ambiente industrial revelou-se como uma oportunidade bastante enriquecedora, tendo em conta todo o conhecimento e experiência adquirida e a possibilidade de colocar em prática alguns dos conhecimentos obtidos ao longo do percurso académico. O contacto com o mundo de trabalho na área da robótica e automação industrial permitiu assimilar novos conhecimentos técnicos, bem como os níveis de exigência e qualidade impostos pelos clientes.

Bibliografia

- [1] Mikell P. Groover, *Fundamentals of Modern Manufacturing.* , 2010. [citado na p. 1, 5, 15, 16]
- [2] B. Siciliano and O. Khatib, *Handbook of Robotics.* Springer, 2008. [citado na p. 5, 8, 10, 18, 20, 37]
- [3] M. E. Rosheim, *In the Footsteps of Leonardo.* , 1997. [citado na p. 6]
- [4] M. E. Moran, *Epochs in Endourology: The Da Vinci Robot.* , 2006. [citado na p. 6]
- [5] History Computer, “The Automata of Leonardo Da Vinci.” [Online]. Disponível em: <http://history-computer.com/Dreamers/LeonardoAutomata.html> [Acedido: Outubro 2017]. [citado na p. 7]
- [6] Mikell P. Groover, *Automation, Production Systems and Computer-Integrated Manufacturing.* , 2001. [citado na p. 6, 7, 10, 16, 17]
- [7] Reza N. Jazar, *Theory of Applied Robotics.* Springer, 2010. [citado na p. 6, 10, 11]
- [8] ABB, “Robô colaborativo ABB YuMi-IRB 14000 Datasheet.” [Online]. Disponível em <http://new.abb.com/products/robotics/industrial-robots/yumi> [Acedido: Outubro 2017]. [citado na p. 9]
- [9] J. N. Pires, *Automação Industrial.* , 2012. [citado na p. 10]
- [10] NPTEL, “CNC Programming and Industrial Robotics.” [Online]. Disponível em <http://nptel.ac.in/courses/112103174/module7/lec5/2.html> [Acedido: Novembro 2017]. [citado na p. 11]
- [11] S. Y. Nof, *Handbook of Industrial Robotics.* , 1999. [citado na p. 11, 14, 15, 18, 19, 20, 21]

- [12] M. Silva, *Apontamentos Teóricos de Robótica Industrial.* , 2016. [citado na p. 11, 16, 17, 18, 19, 31, 32]
- [13] Illinois State University, “Industrial Robots: Robots Configurations.” [Online]. Disponível em http://my.ilstu.edu/kldevin/Introduction_to_robotics2/Introduction_to_robotics7.html [Acedido: Novembro 2017]. [citado na p. 12, 13, 14]
- [14] ABB, “IRC5 Industrial Robot Controller Datasheet.” . [citado na p. 15]
- [15] Schunk, “Garra Mecânica Shunck SWG.” [Online]. Disponível em https://schunk.com/be_en/gripping-systems/series/swg/ [Acedido: Novembro 2017]. [citado na p. 17]
- [16] Schunk, “Garra Magnética Shunck GSW-M.” [Online]. Disponível em https://schunk.com/be_en/gripping-systems/series/gsw-m/ [Acedido: Novembro 2017]. [citado na p. 17]
- [17] KUKA, “KR 1000 Titan, KR 1000 L750 Specifications.” . [citado na p. 19]
- [18] Vítor Santos, “Robótica Industrial: Apontamentos teóricos.” . [citado na p. 20]
- [19] W. Bolton, *Programmable Logic Controllers.* Newnes, 2009. [citado na p. 24, 25, 26, 27, 28]
- [20] Omron, “CP-series CP1E CPU Units Datasheet.” . [citado na p. 25]
- [21] Omron, “CJ-series CJ2M CPU Units, Pulse I/O Modules Datasheet.” . [citado na p. 25]
- [22] Frank D. Petruzella, *Programmable Logic Controllers.* McGraw-Hill Education, 2017. [citado na p. 26]
- [23] International Electrotechnical Commission, *Programmable controllers - Part 3: Programming languages.* IEC, 2003. [citado na p. 26, 27]
- [24] Richard Zurawski, *Industrial Communication Technology Handbook.* CRC Press: Taylor & Francis Group, 2015. [citado na p. 29, 30]
- [25] EtherCAT Technology Group, *EtherCAT – the Ethernet Fieldbus.* , 2018. [citado na p. 29]
- [26] IO-Link, *IO-Link System Description: Technology and Application.* IO-Link Community, 2016. [citado na p. 30]
- [27] Yunjie Xu, Yumei Liu, Lina Hao, Hongtai Cheng, *Design of Palletizing Algorithm Based on Palletizing Robot Workstation.* IEEE International Conference on Real-time Computing and Robotics, 2016. [citado na p. 36]

- [28] Robin A. Popple, *The Science of Palletizing*. Columbia Machine, Inc., 2009. [citado na p. 36, 37, 38, 39, 40, 43]
- [29] Brenton Engineering, “Brenton Engineering: LLP Low Level Infeed Palletizer.” [Online]. Disponível em <http://www.brentonengineering.com/products/low-level-infeed-palletizer-llp/> [Acedido: Abril 2018]. [citado na p. 38]
- [30] Brenton Engineering, “Brenton Engineering: HLP-High Speed High Level Infeed Palletizer.” [Online]. Disponível em <http://www.brentonengineering.com/products/high-level-infeed-palletizer-hlp-high-speed/> [Acedido: Abril 2018]. [citado na p. 38]
- [31] Columbia Okura, “Columbia Okura: End Effectors.” [Online]. Disponível em <http://columbiaokura.com/products/end-effectors/> [Acedido: Abril 2018]. [citado na p. 44]
- [32] Brenton Engineering, “Brenton Engineering: Master-Pal Bulk (de)Palletizer EOAT.” [Online]. Disponível em <http://www.brentonengineering.com/products/bulk-de-palletizer-eoat-masterpal/> [Acedido: Abril 2018]. [citado na p. 45]
- [33] FANUC, “R-30iB PalletTool Setup and Operations Manual.” . [citado na p. 46, 47]
- [34] Motion Controls Robotics, “Perfect Your Palletizing with PalletPRO.” [Online]. Disponível em <http://motioncontrolsrobotics.com/perfect-palletizing-palletpro/> [Acedido: Março 2018]. [citado na p. 47]
- [35] RIA, “Robotic Bag Palletizing System Helps Forbes Chocolate Productivity.” [Online]. Disponível em https://www.robotics.org/content-detail.cfm/content_id/287 [Acedido: Fevereiro 2018]. [citado na p. 48, 49, 50]
- [36] Arnie Orloski, “Robot takes 'load' off workers.” [Online]. Disponível em <https://www.packworld.com/article/applications/food/confectionery/robot-takes-load-workers> [Acedido: Fevereiro 2018]. [citado na p. 48, 49, 50]
- [37] RIA, “A Sweet Success Story: Candy Factory Modernizes its Robotic-Palletizing Cells.” [Online]. Disponível em https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Case-Studies/A-Sweet-Success-Story-Candy-Factory-Modernizes-its-Robotic-Palletizing-Cells/content_id/2123 [Acedido: Fevereiro 2018]. [citado na p. 50, 51, 52]

- [38] KUKA, “KUKA KR QUANTEC in operation at woodworking company Voit.” [Online]. Disponível em <https://www.kuka.com/en-de/industries/solutions-database/2016/09/solution-robotics-fischertec> [Acedido: Março 2018]. [citado na p. 52, 53]
- [39] Omron Industrial Automation, “Controlo Avançado para Máquinas Compactas: Controlador de Máquina NX1.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#nx1p> [Acedido: Junho 2018]. [citado na p. 61]
- [40] Omron Industrial Automation, “Machine Automation Controller: NX-series NX1P2 CPU Unit Hardware User’s Manual.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#nx1p> [Acedido: Junho 2018]. [citado na p. 61]
- [41] Omron, “Omron: NX-Series I/O.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#nx-series> [Acedido: Junho 2018]. [citado na p. 61]
- [42] FANUC, “FANUC: Manipulador de Paletização M410iC/315 Datasheet.” [Online]. Disponível em <https://www.fanuc.eu/uk/en/robots/robot-filter-page/m-410-series/m-410ic-315> [Acedido: Junho 2018]. [citado na p. 62]
- [43] FANUC, “FANUC: Industrial Robot Controllers.” [Online]. Disponível em <https://www.fanucamerica.com/home/products-services/robots/Controllers/other-fanuc-robot-controllers> [Acedido: Junho 2018]. [citado na p. 62]
- [44] SMC, “SMC: General Purpose Auto Switch.” [Online]. Disponível em <https://www.smcusa.com/products/Auto-Switch-Solid-State-General-Purpose-30777> [Acedido: Junho 2018]. [citado na p. 63, 64]
- [45] SMC, “SMC: SY5000 Solenoid Valves.” [Online]. Disponível em <https://www.smcusa.com/products/SY5000-5-Port-Solenoid-Valve-All-Types-New-Style-50611> [Acedido: Junho 2018]. [citado na p. 63, 64]
- [46] SMC, “SMC: ISE30A Digital Pressure Switch.” [Online]. Disponível em <https://www.smcusa.com/products/ISE30A-Digital-Pressure-Switch-2-Color-Display-High-Precision-for-Positive-Pressure-39270> [Acedido: Junho 2018]. [citado na p. 64]
- [47] Omron, “Omron: HMI Série NB Catalogo.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#nb> [Acedido: Junho 2018]. [citado na p. 65]

- [48] Omron, “Omron: NB-Designer Operation Manual.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#nb> [Acedido: Junho 2018]. [citado na p. 65]
- [49] Omron, “Omron: MX2 User’s Manual.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#mx2-ip54> [Acedido: Junho 2018]. [citado na p. 66]
- [50] Pepperl Fuchs, “Pepperl Fuchs: Photoelectric Sensor ML100-8-H-350-RT/103/115a Datasheet.” [Online]. Disponível em <https://www.pepperl-fuchs.com> [Acedido: Junho 2018]. [citado na p. 66, 67]
- [51] Pepperl Fuchs, “Pepperl Fuchs: Distance sensor VDM28-8-L1-IO/73c/110/122 Datasheet.” [Online]. Disponível em <https://www.pepperl-fuchs.com> [Acedido: Junho 2018]. [citado na p. 67]
- [52] Pepperl Fuchs, “Pepperl Fuchs: Thru-beam sensor M100/MV100-RT/35/76a/98/103 Datasheet.” [Online]. Disponível em <https://www.pepperl-fuchs.com> [Acedido: Junho 2018]. [citado na p. 68]
- [53] Interroll, “Interroll: Drive Control User’s Manual.” [Online]. Disponível em <https://www.interroll.co.uk/support/product-information/drives-controls/> [Acedido: Junho 2018]. [citado na p. 68]
- [54] Interroll, “Interroll: Drive Control 20.” [Online]. Disponível em <https://www.interroll.co.uk/products/drives-and-controls/controls-drivers/drivecontrol-20/> [Acedido: Junho 2018]. [citado na p. 69]
- [55] Interroll, “Interroll: Roller Drive EC310.” [Online]. Disponível em <https://www.interroll.co.uk/products/drives-and-controls/rollerdrive/rollerdrive-ec310-ip66/> [Acedido: Junho 2018]. [citado na p. 69]
- [56] Omron, “Omron: NX-Series Safety Control Unit User’s Manual.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#nx-series> [Acedido: Julho 2018]. [citado na p. 71]
- [57] Omron, “Omron: Safety Relay Units G9SE Series.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#g9se> [Acedido: Julho 2018]. [citado na p. 72]
- [58] Pilz, “Pilz: PSEN sg2c Operating Manual.” [Online]. Disponível em <https://www.pilz.com/pt-PT/eshop/0010600222704880J7/PSENsgate#Operating%20Manual-en> [Acedido: Julho 2018]. [citado na p. 73]

- [59] Omron, “Omron: Safety Laser Scanner OS32C Series User’s Manual.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#os32c> [Acedido: Julho 2018]. [citado na p. 73, 74]
- [60] Omron, “Omron: OS32C-DM Safety Laser Scanner with Ether-Net/IP.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#os32c> [Acedido: Junho 2018]. [citado na p. 74, 96]
- [61] Omron, “Omron: Safety Limit Switch D4N Datasheet.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#d4n> [Acedido: Julho 2018]. [citado na p. 75]
- [62] Omron, “Omron: Safety Light Curtain F3SG Series User’s Manual.” [Online]. Disponível em <https://industrial.omron.pt/pt/services-support/technical-tools/downloads#f3sg-re> [Acedido: Julho 2018]. [citado na p. 75, 76, 106]
- [63] FANUC, “FANUC: Curso de programação TPE níveis A e B.” Disponibilizado pela FANUC. [citado na p. 79, 80, 81, 83, 84]
- [64] Omron, “Omron: EtherCAT Connection Guide SMC Corporation.” Disponibilizado pela Omron. [citado na p. 93]

Anexo A

Sinais Trocados com o Manipulador FANUC

Tabela A.1: Sinais UI

| Sinal | Nome | Descrição |
|----------|------------------------|--|
| UI[1] | IMSTP | Paragem imediata de <i>software</i> |
| UI[2] | Hold | Paragem externa controlada |
| UI[3] | SFSPD | Habilitação de velocidade reduzida de segurança |
| UI[4] | Cycle Stop | Paragem de ciclo |
| UI[5] | Fault Reset | Rearme de falhas |
| UI[6] | Start | Arranque do programa |
| UI[7] | Home | Mover robô para posição gravada como HOME |
| UI[8] | Enable | Habilitação de movimento |
| UI[9-16] | RSR1/PNS1 RSR8/PNS8 | Sinais para a seleção do programa RSR ou PNS |
| UI[17] | PNS strobe | Sinal de validação do número de programa PNS |
| UI[18] | Prod Start | Sinal de arranque de produção para programas PNS |

Tabela A.2: Sinais UO

| Sinal | Nome | Descrição |
|--------------|------------------------|--|
| UO[1] | CMDENBL | Robô em modo remoto |
| UO[2] | SYSRDY | Sistema pronto |
| UO[3] | Progrun | Programa em execução |
| UO[4] | Paused | Programa em pausa |
| UO[5] | Motion Held | Movimento do robô detido |
| UO[6] | Fault | Falha de sistema |
| UO[7] | AtPerch | Sinal que o robô alcançou uma posição de referência pré-definida |
| UO[8] | TPENBL | Consola de programação habilitada |
| UO[9] | BATALM | Falha nas baterias |
| UO[10] | Busy | Robô executando um programa |
| UO[11-18] | ACK1/SNO1 ACK8/SNO8 | Sinais de reconhecimento do programa RSR/PNS |
| UO[19] | Snack | Sinal de confirmação do programa PNS |
| UO[20] | Reserved | Reserva de sistema |

Tabela A.3: Sinais DI

| Sinal | Nome | Descrição |
|--------------|----------------------------------|--|
| DI[81-87] | IN_AreaReady_1 IN_AreaReady_7 | Sinal que indica que área de trabalho se encontra pronta e manipulador pode entrar |
| DI[150] | IN_ProdEnd | Sinal de fim de produção |
| DI[151] | IN_AcessRequest | Sinal que indica um pedido de acesso realizado pelo operador |
| DI[152] | IN_MaintenanceRequest | Sinal que indica um pedido de manutenção realizado pelo operador |
| DI[153] | IN_L1_PalletIndex | Sinal que indica ao manipular que a palete foi devidamente indexada na Linha 1 |
| DI[154] | IN_L2_PalletIndex | Sinal que indica ao manipular que a palete foi devidamente indexada na Linha 2 |
| DI[177] | IN_Grp_Box_Open | Indica que sistema de manipulação de caixas está aberto |
| DI[178] | IN_Grp_Box_Close | Indica que sistema de manipulação de caixas está fechado |
| DI[179] | IN_Grp_Pallet_Open | Indica que sistema de manipulação de paletes está aberto |
| DI[180] | IN_Grp_Pallet_Close | Indica que sistema de manipulação de paletes está fechado |
| DI[181] | IN_Grp_Air_Open | Indica que sistema de manipulação de ventosas está aberto |
| DI[182] | IN_Grp_Air_Close | Indica que sistema de manipulação de ventosas está fechado |

Tabela A.4: Sinais DI convertidos para GI

| Sinal DI | Sinal GI | Descrição |
|-----------------|---------------------------|--|
| DI[193-208] | GI[1] - nProg | Número de programa |
| DI[209-224] | GI[2] - nJob | Número de tarefa |
| DI[225-240] | GI[11] - L1_TotalLayers | Número de camadas totais por pa- lete (Linha de produção 1) |
| DI[241-256] | GI[12] - L1_LayerHeight | Altura de cada camada (Linha de produção 1) |
| DI[257-272] | GI[13] - L1_PalletHeight | Altura de pilha de paletes (Linha de produção 1) |
| DI[273-288] | GI[14] - L1_ProdRef | Referência do mosaico a manipular (Linha de produção 1) |
| DI[289-304] | GI[21] - L2_TotalLayers | Número de camadas totais por pa- lete (Linha de produção 2) |
| DI[305-320] | GI[22] - L2_LayerHeight | Altura de cada camada (Linha de produção 2) |
| DI[321-336] | GI[23] - L2_PalletHeight | Altura de pilha de paletes (Linha de produção 2) |
| DI[337-352] | GI[24] - L2_ProdRef | Referência do mosaico a manipular (Linha de produção 2) |
| DI[353-368] | GI[30] - InterlayerHeight | Altura da pilha de folhas intercalar |

Tabela A.5: Sinais DO

| Sinal | Nome | Descrição |
|--------------|--------------------------------|--|
| DO[21] | OUT_Auto | Seletor do controlador em modo automático |
| DO[22] | OUT_T1 | Seletor do controlador em modo T1 |
| DO[23] | OUT_T2 | Seletor do controlador em modo T2 |
| DO[24] | OUT_ESTOP | Sinal que indica que se encontra atuada uma botoneira de emergência |
| DO[33] | OUT_Prog_Request | Sinal de pedido de programa ao PLC |
| DO[34] | OUT_Door_Acess_Granted | Sinal de acesso à célula concedido |
| DO[36] | OUT_Job_Request | Sinal de pedido de tarefa ao PLC |
| DO[40] | OUT_RefPos_L1 | Sinal que indica que manipulador se encontra dentro da área reservada para a Linha 1 |
| DO[41] | OUT_RefPos_L2 | Sinal que indica que manipulador se encontra dentro da área reservada para a Linha 2 |
| DO[47] | OUT_SF_CellSpace | Sinal que indica que manipulador se encontra dentro do volume da célula |
| DO[48] | OUT_SF_Ventilation | Sinal que indica que manipulador se encontra dentro do volume do sistema de ventilação |
| DO[65-71] | OUT_AreaReq_1 OUT_AreaReq_7 | Sinais de pedido de acesso a área de trabalho |
| DO[81-87] | OUT_AreaRel_1 OUT_AreaRel_7 | Sinais que indicam que robô saiu da área de trabalho |
| DO[145] | OUT_L1_PalletFinish | Paleta terminada (Linha de produção 1) |
| DO[145] | OUT_L2_PalletFinish | Paleta terminada (Linha de produção 2) |
| DO[147] | OUT_L1_PalletPicked | Paleta pegada (Linha de produção 1) |
| DO[148] | OUT_L2_PalletPicked | Paleta pegada (Linha de produção 2) |

Tabela A.5 continuação da página anterior

| Sinal | Nome | Descrição |
|---------|-------------------------|--|
| DO[149] | OUT_L1_PalletPlaced | Palete colocada/indexada (Linha de produção 1) |
| DO[150] | OUT_L2_PalletPlaced | Palete colocada/indexada (Linha de produção 2) |
| DO[151] | OUT_L1_MosaicPicked | Mosaico pegado (Linha de produção 1) |
| DO[152] | OUT_L2_MosaicPicked | Mosaico pegado (Linha de produção 2) |
| DO[153] | OUT_L1_MosaicPlaced | Mosaico colocado (Linha de produção 1) |
| DO[154] | OUT_L2_MosaicPlaced | Mosaico colocado (Linha de produção 2) |
| DO[155] | OUT_InterlayerPicked | Camada intercalar pegada |
| DO[156] | OUT_L1_InterlayerPlaced | Camada intercalar colocada (Linha de produção 1) |
| DO[157] | OUT_L2_InterlayerPlaced | Camada intercalar colocada (Linha de produção 2) |
| DO[159] | OUT_SafePosition | Robô em posição segura |
| DO[160] | OUT_MaintenancePosition | Robô em posição de manutenção |
| DO[177] | OUT_Grp_Box_Open | Pedido de fecho de sistema de manipulação de caixas |
| DO[178] | OUT_Grp_Box_Close | Pedido de fecho de sistema de manipulação de caixas |
| DO[179] | OUT_Grp_Pallet_Open | Pedido de abertura de sistema de manipulação de paletes |
| DO[180] | OUT_Grp_Pallet_Close | Pedido de fecho de sistema de manipulação de paletes |
| DO[181] | OUT_Grp_Air_Open | Pedido de abertura de sistema de manipulação de ventosas |
| DO[182] | OUT_Grp_Air_Close | Pedido de fecho de sistema de manipulação de ventosas |

Tabela A.6: Sinais GI e Registos

| Sinal - Registo | Nome | Descrição |
|------------------------|------------------|--|
| GI[1] - R[1] | nProg | Número de programa |
| GI[2] - R[2] | nJob | Número de tarefa |
| GI[10] - R[10] | L1_LayerCounter | Contador de camadas (Linha de produção 1) |
| GI[11] - R[11] | L1_TotalLayers | Número de camadas totais por pa- lete (Linha de produção 1) |
| GI[12] - R[12] | L1_LayerHeight | Altura de cada camada (Linha de produção 1) |
| GI[13] - R[13] | L1_PalletHeight | Altura de pilha de paletes (Linha de produção 1) |
| GI[14] - R[14] | L1_ProdRef | Referência do mosaico a manipular (Linha de produção 1) |
| GI[20] - R[20] | L2_LayerCounter | Contador de camadas (Linha de produção 2) |
| GI[21] - R[21] | L2_TotalLayers | Número de camadas totais por pa- lete (Linha de produção 2) |
| GI[22] - R[22] | L2_LayerHeight | Altura de cada camada (Linha de produção 2) |
| GI[23] - R[23] | L2_PalletHeight | Altura de pilha de paletes (Linha de produção 2) |
| GI[24] - R[24] | L2_ProdRef | Referência do mosaico a manipular (Linha de produção 2) |
| GI[30] - R[30] | InterlayerHeight | Altura da pilha de folhas intercalar |

Tabela A.7: Registos de Posição

| Registo | Nome | Descrição |
|----------------|----------------------|--|
| PR[1] | Home_Position | Posição Home |
| PR[2] | Maintenance_Position | Posição de manutenção |
| PR[8] | Home_J_Offset | Registo de posição auxiliar para recuperação automática de posição de Home |
| PR[10] | Point_Offset_Aux | Registo de posição auxiliar para realizar <i>offsets</i> entre pontos de trabalho |
| PR[15] | Change_Lines | Registo de posição auxiliar para realizar a movimentação do manipulador para uma posição segura sempre que exista a mudança de tarefa para a outra linha de produção |

Anexo B

Fluxogramas Rotinas FANUC

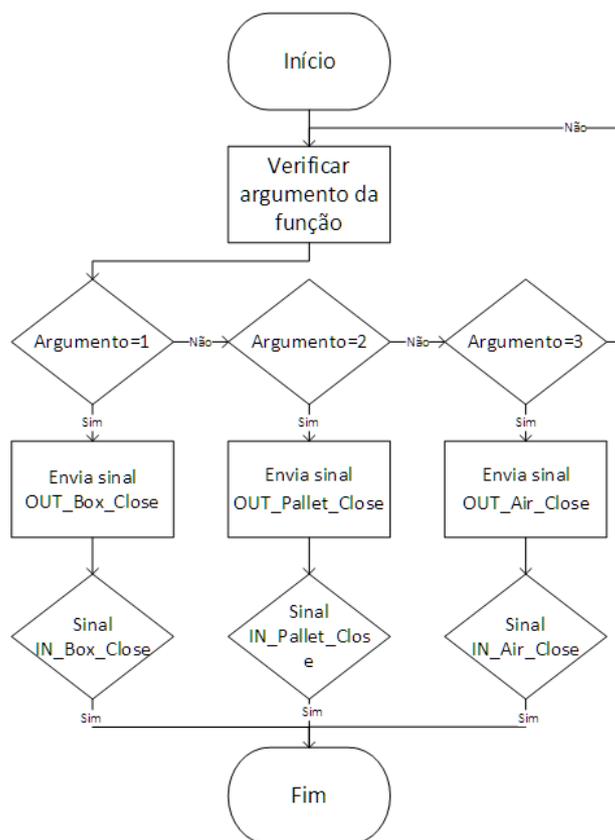


Figura B.1: Fluxograma de funcionamento da rotina *CLOSE_GRIPPER*.

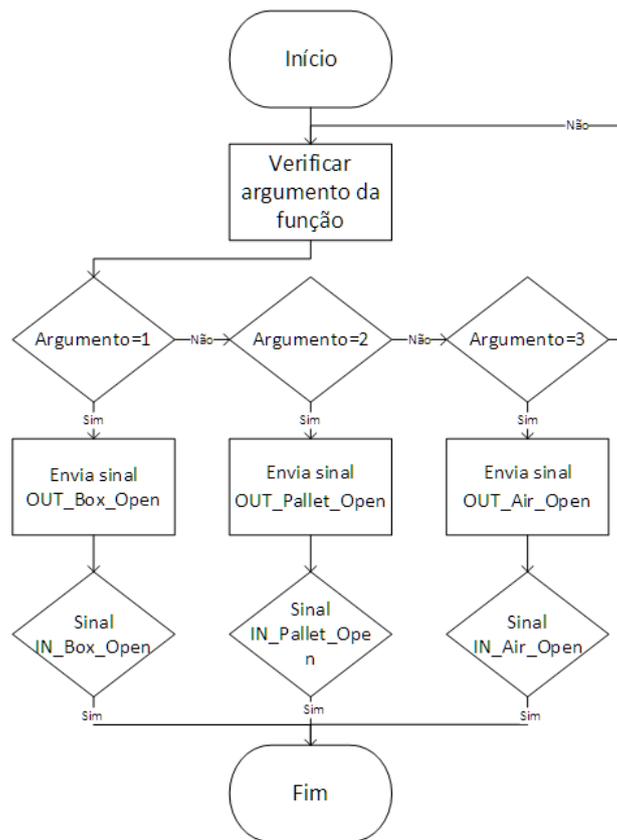


Figura B.2: Fluxograma de funcionamento da rotina *OPEN_GRIPPER*.

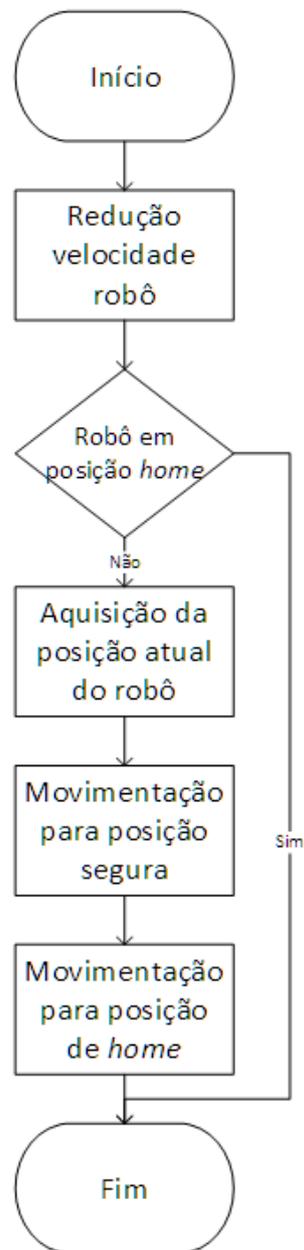


Figura B.3: Fluxograma de funcionamento da rotina *HOME*.

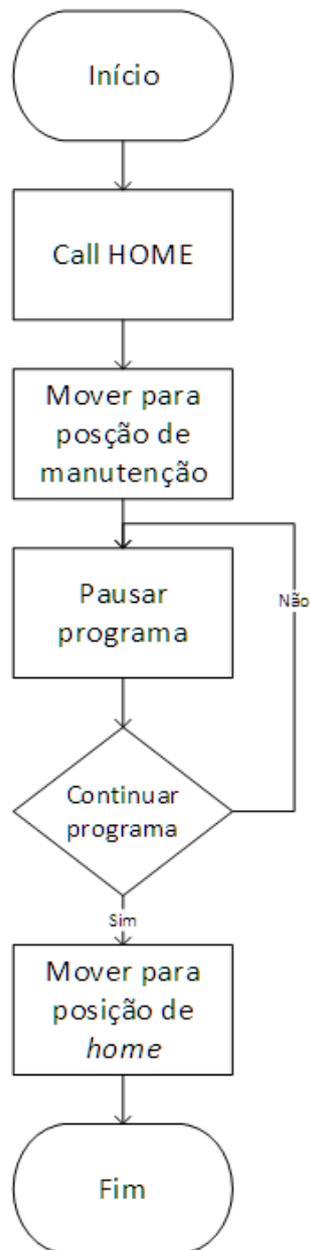


Figura B.4: Fluxograma de funcionamento da rotina *MAINTENANCE*.

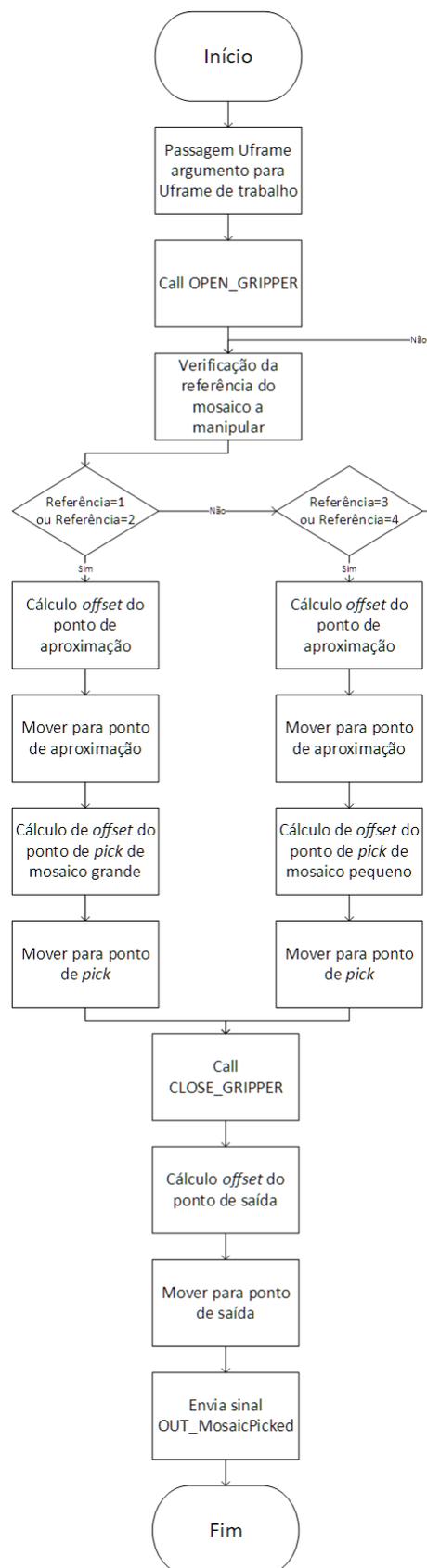
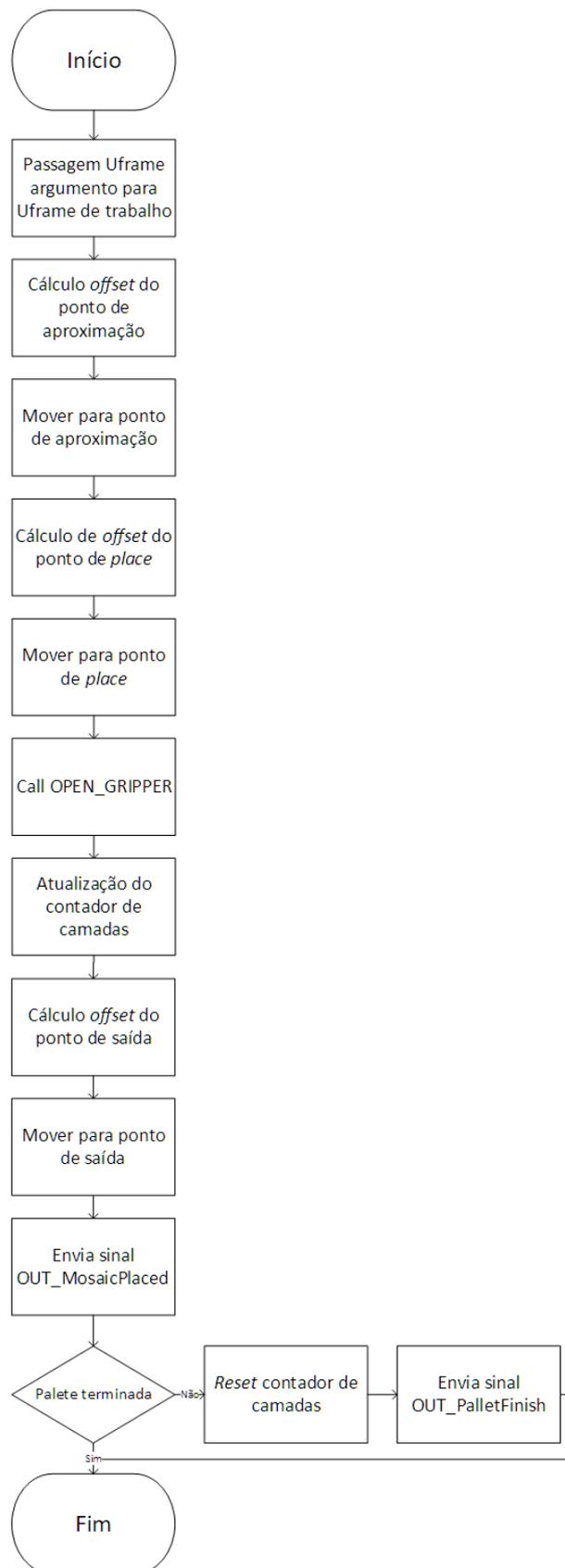


Figura B.5: Fluxograma de funcionamento da rotina *PICK_BOX*.

Figura B.6: Fluxograma de funcionamento da rotina *PLACE_BOX*.

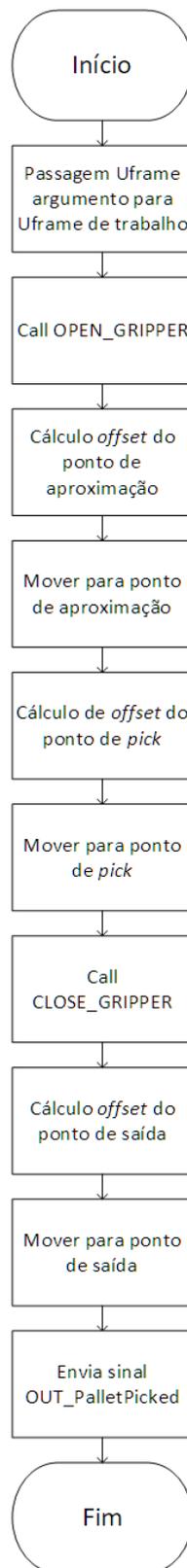
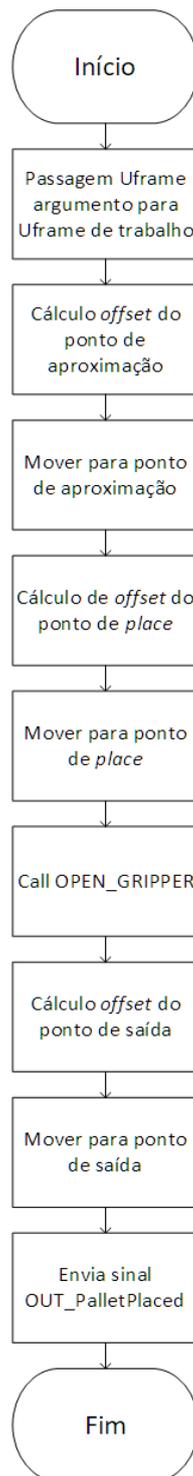


Figura B.7: Fluxograma de funcionamento da rotina *PICK_PALLET*.

Figura B.8: Fluxograma de funcionamento da rotina *PLACE_PALLET*.

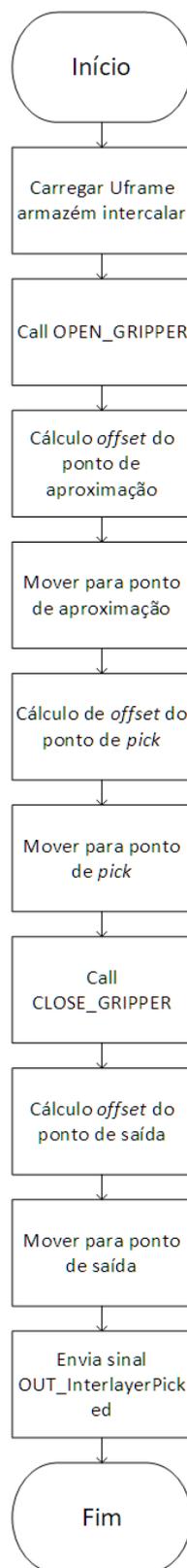
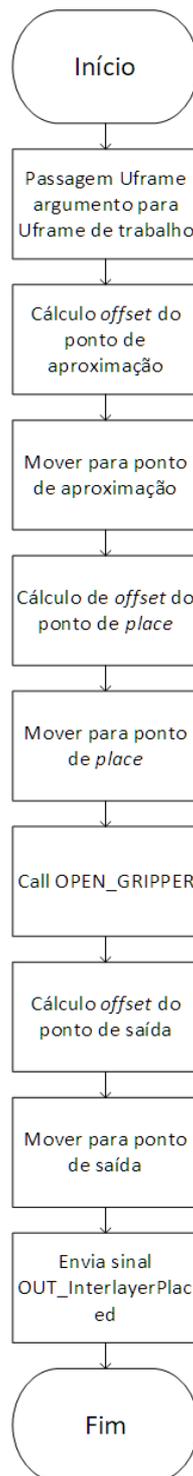


Figura B.9: Fluxograma de funcionamento da rotina *PICK_INTERLAYER*.

Figura B.10: Fluxograma de funcionamento da rotina *PLACE_INTERLAYER*.

Anexo C

Fluxogramas Programas PLC

Dada a dimensão dos fluxogramas dos programas FANUC_Program_Job_Control e Charriot_Task_Control, estes não foram incluídos no relatório mas podem ser consultados nos ficheiros do CD em anexo.

(Este espaço foi deixado em branco intencionalmente)

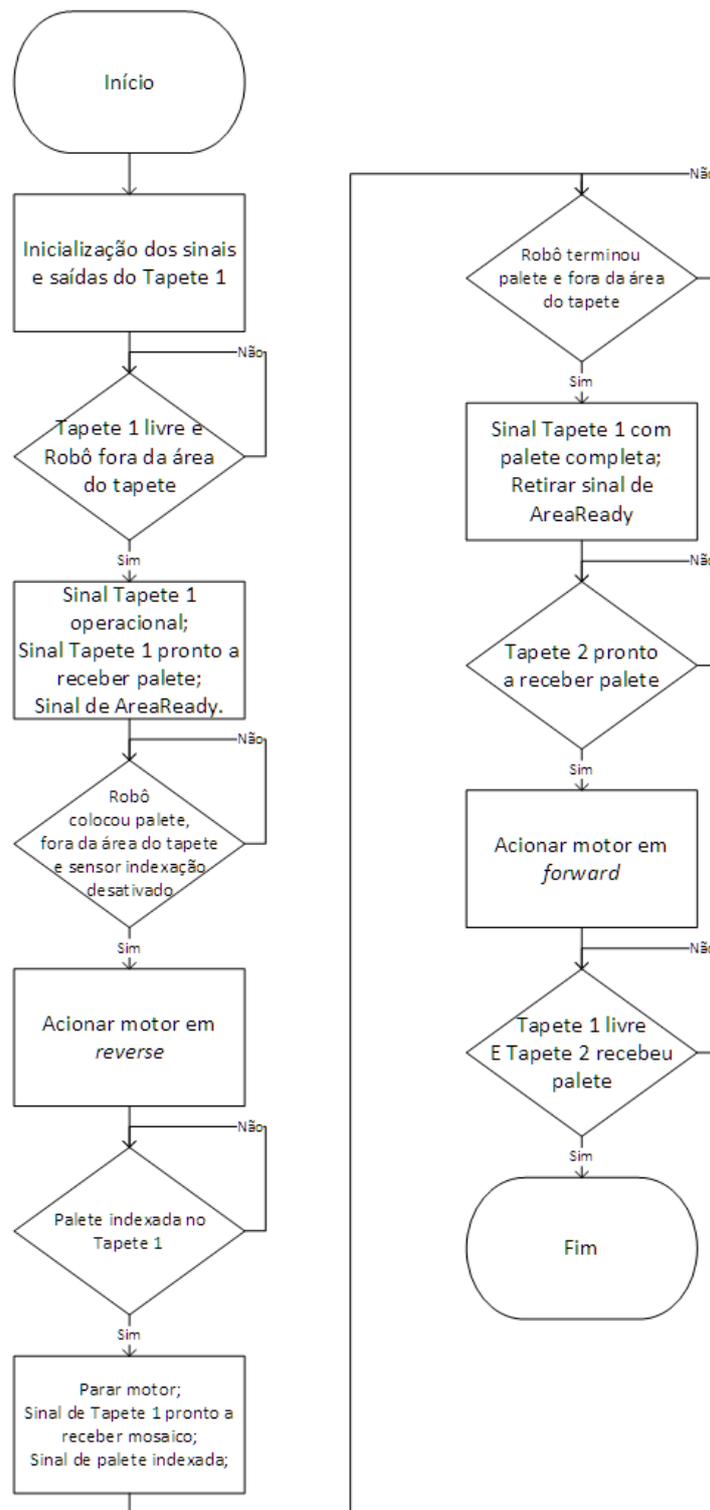


Figura C.1: Fluxograma de funcionamento dos transportadores de paletização.

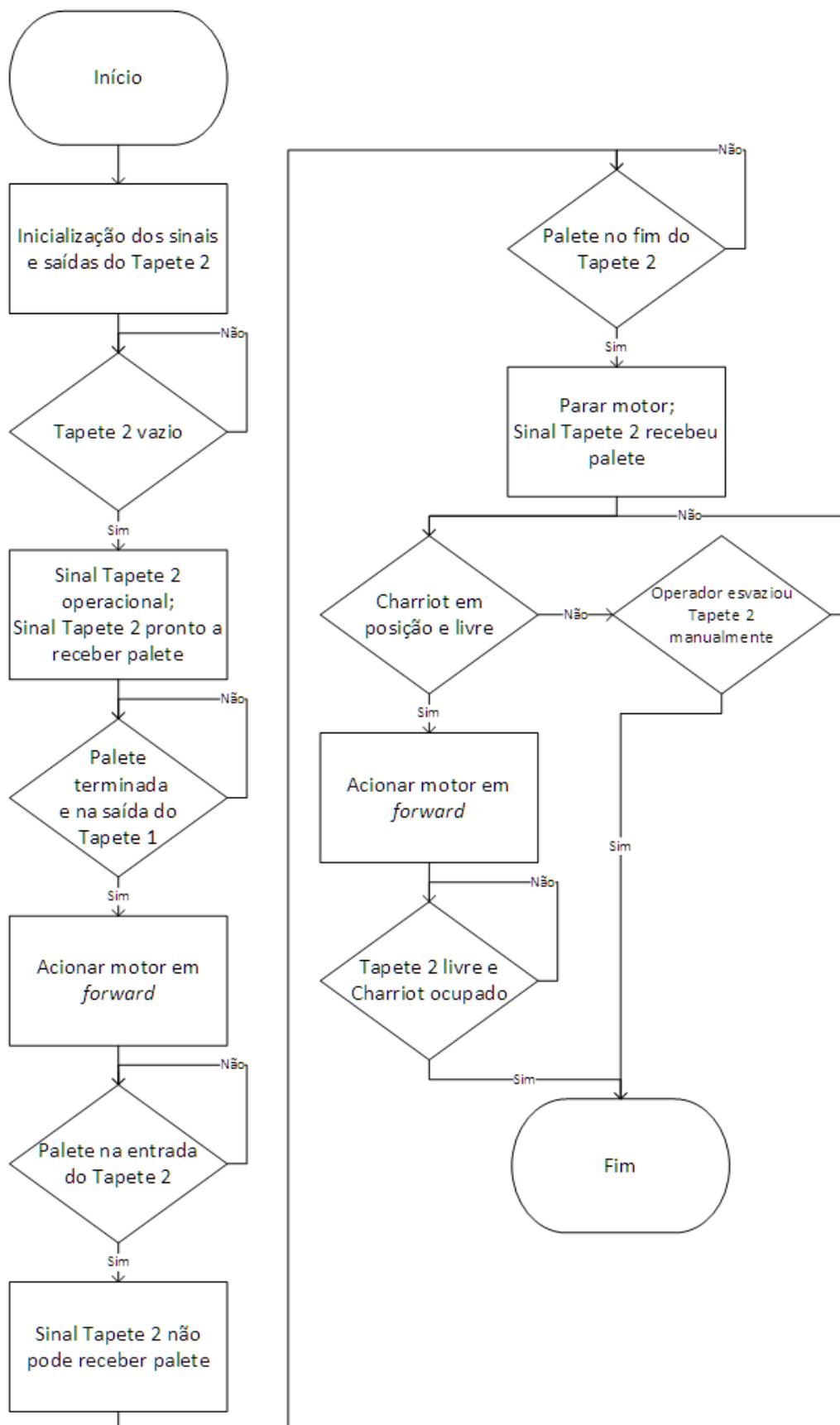


Figura C.2: Fluxograma de funcionamento dos transportadores de transferência.

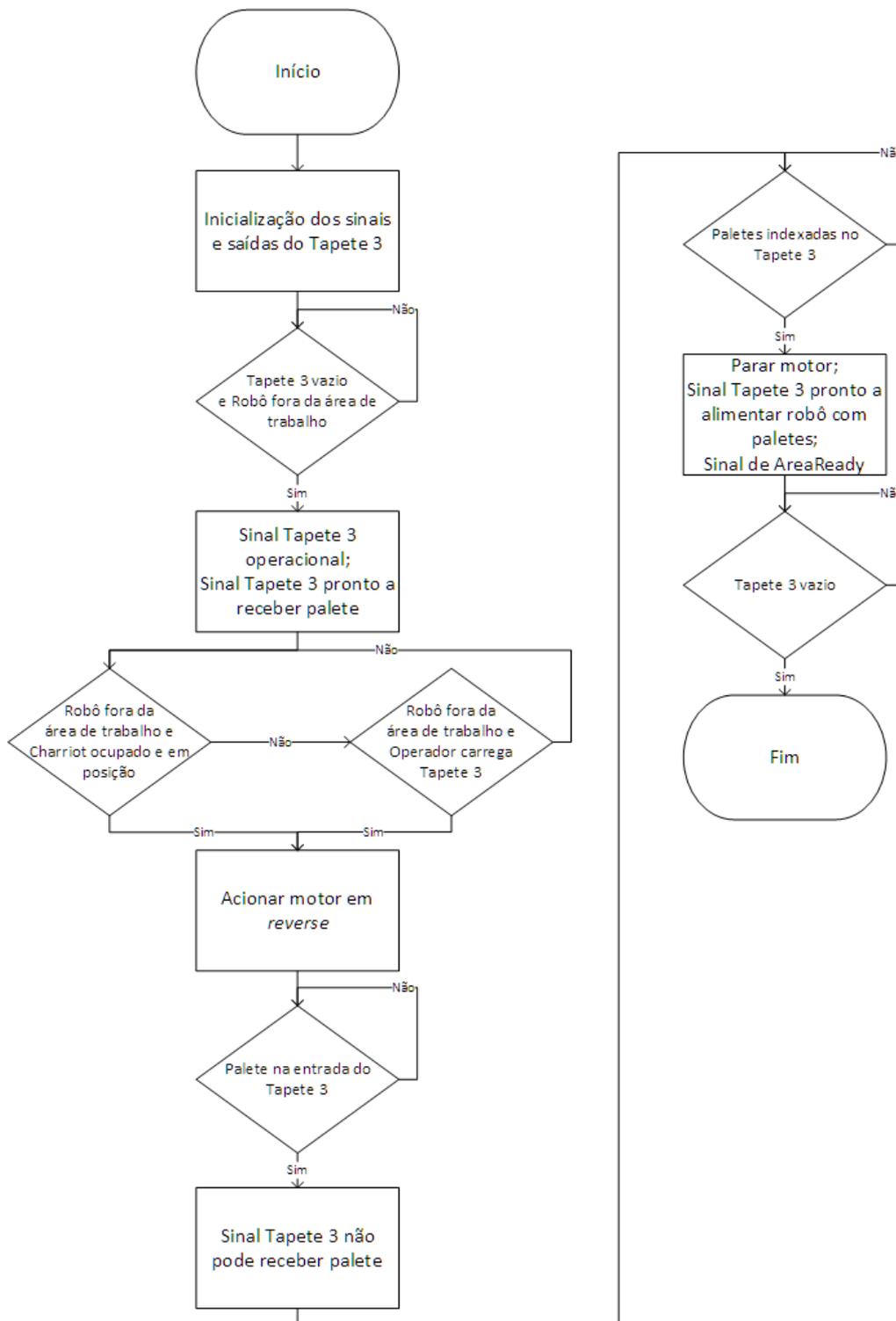


Figura C.3: Fluxograma de funcionamento dos transportadores de armazenamento de paletes.

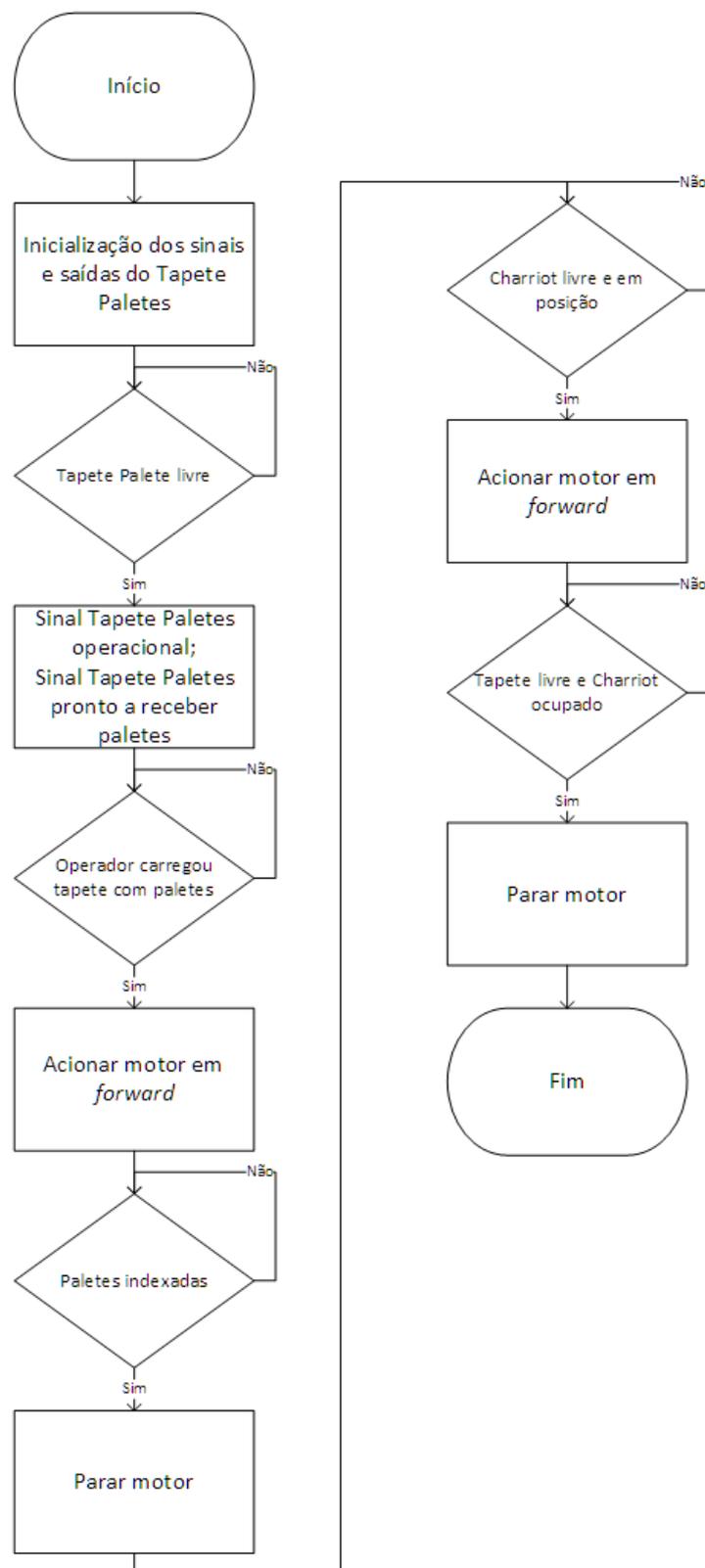


Figura C.4: Fluxograma de funcionamento dos transportadores de entrada de paletes.

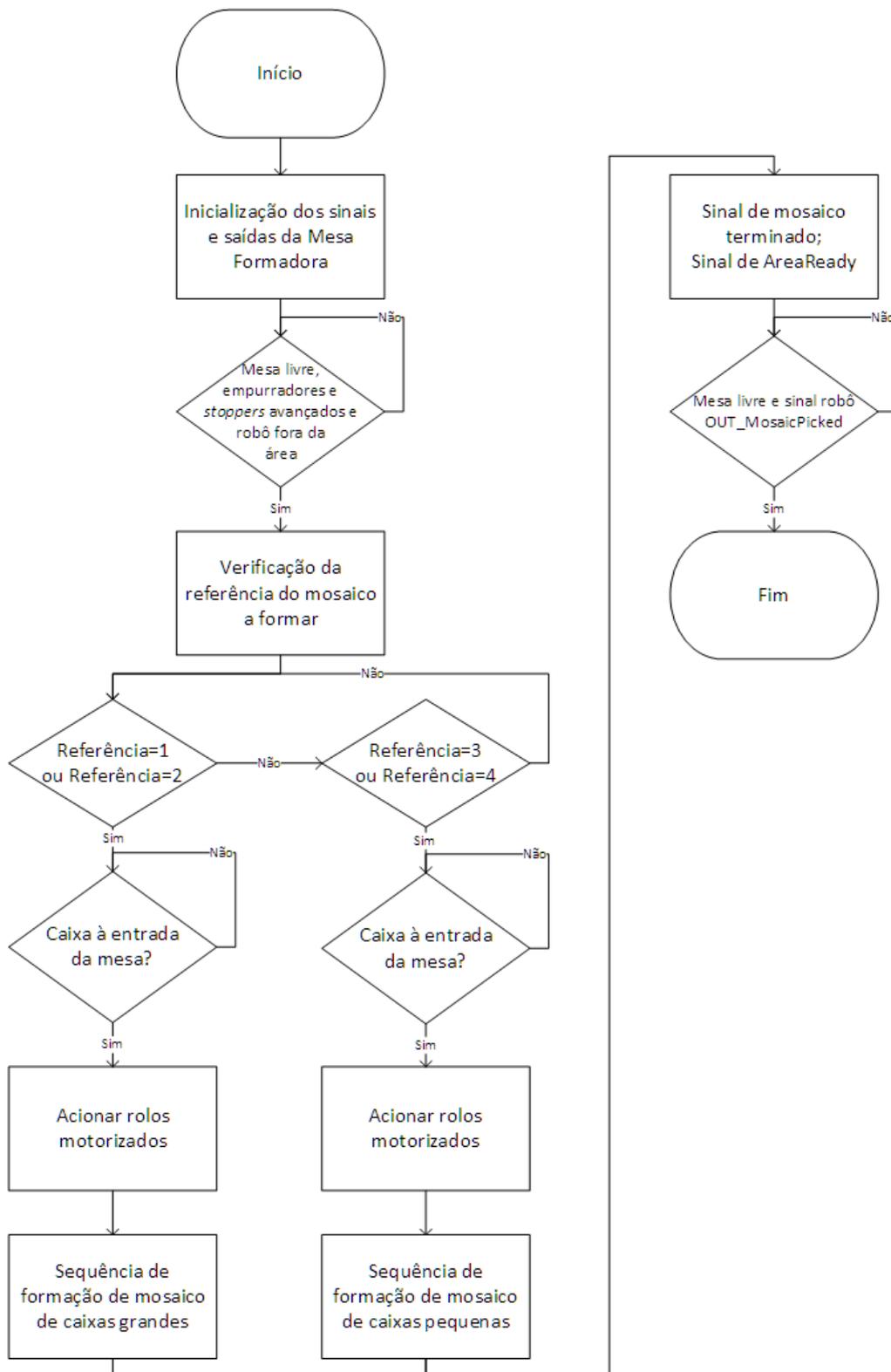


Figura C.5: Fluxograma de funcionamento das mesas formadoras de mosaicos.

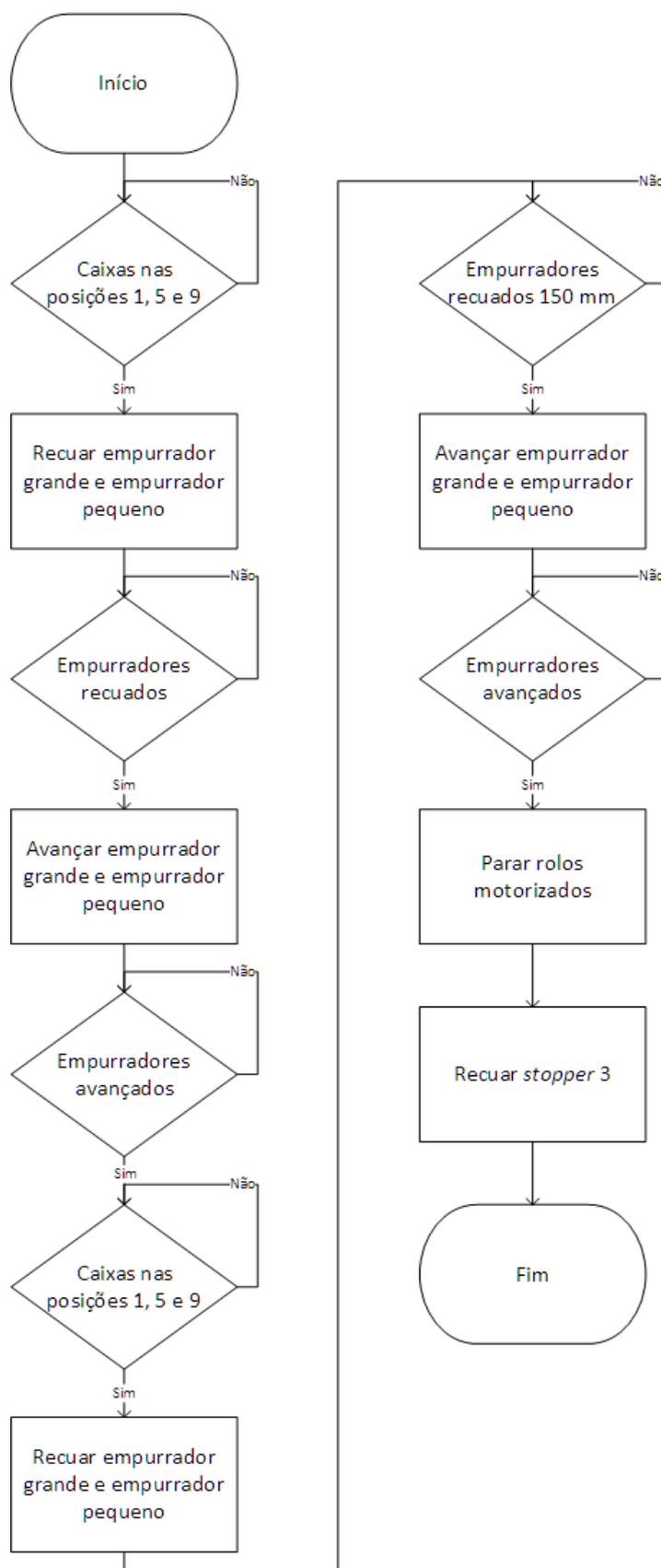


Figura C.6: Fluxograma de funcionamento da sequência de formação do mosaico de caixas grandes.

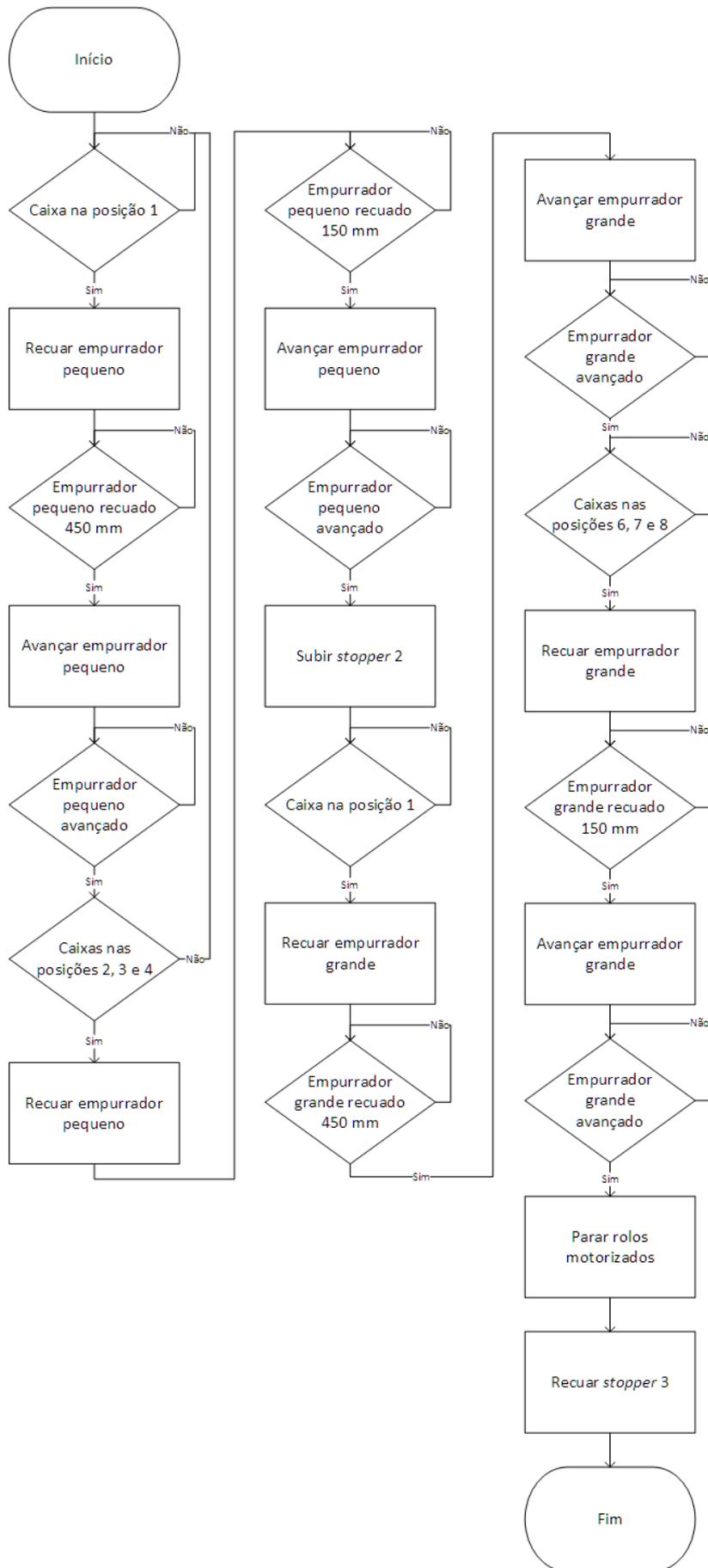


Figura C.7: Fluxograma de funcionamento da sequência de formação do mosaico de caixas pequenas.

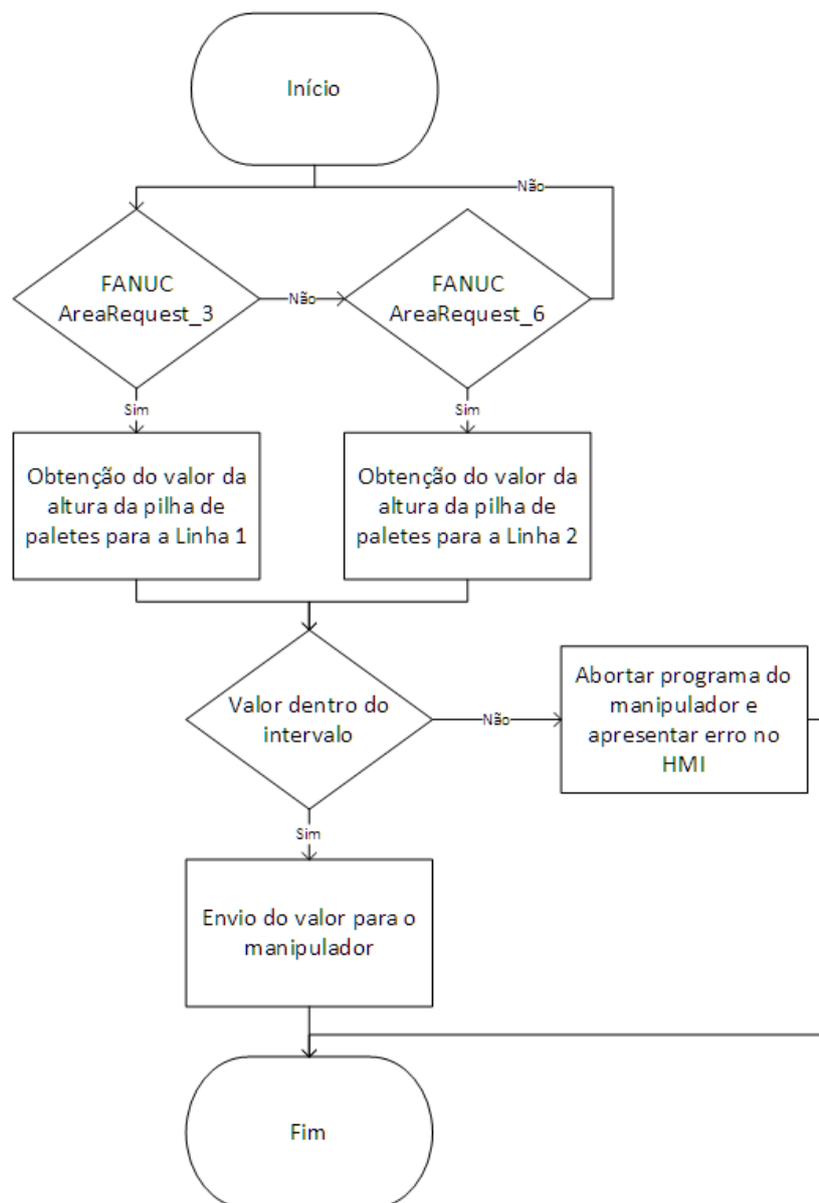


Figura C.8: Fluxograma de funcionamento da leitura dos sensores IO-Link.

