



Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Programação de Robôs Antropomórficos na Indústria

Relatório de Estágio

Nelson Manuel Lopes dos Reis

Mestrado em Engenharia Eletrotécnica

Tomar/ Outubro/ 2021



Instituto Politécnico de Tomar

www.ipt.pt



Instituto Politécnico de Tomar

Escola Superior de Tecnologia de Tomar

Nelson Manuel Lopes dos Reis

Programação de Robôs Antropomórficos na Indústria

Relatório de Estágio

Orientado por:

Eng. Francisco Nunes – Instituto Politécnico de Tomar
Eng. António Simões – EXSEPI, Estudos e Projetos Industriais, Lda

Relatório de Estágio apresentado ao Instituto Politécnico de Tomar para cumprimento
dos requisitos necessários à obtenção do grau
de Mestre em Engenharia Eletrotécnica

RESUMO

Os robôs antropomórficos são máquinas polivalentes capazes de realizar tarefas para as quais foram previamente programados. Têm a capacidade de interpretar a alteração das condições do ambiente de trabalho onde estão inseridos e de interagir de forma a tomar decisões com o objetivo de executar as tarefas programadas. Estas máquinas são geralmente associadas ao movimento de um braço humano, capazes de efetuar tarefas análogas às que um ser humano consegue realizar, tem uma boa repetibilidade, uma precisão inferior a um milímetro e uma carga útil até centenas de quilogramas.

Este relatório foi redigido para documentar o trabalho desenvolvido no âmbito de um estágio curricular, para obtenção do grau de Mestre em Engenharia Eletrotécnica, realizado na empresa EXSEPI.

Este estágio teve início no dia 4 de janeiro, terminando no dia 20 de setembro de 2021 e foi coordenado pelo Engenheiro Eletrotécnico António Simões. O estágio focou-se na programação de robôs antropomórficos em ambiente industrial. Foram desenvolvidos quatro trabalhos, todos eles muito semelhantes e que se focam na manipulação e paletização de objetos. O primeiro projeto será explicado de forma detalhada, para que o leitor tenha uma noção de todas as tarefas realizadas até à obtenção do resultado final. Os restantes trabalhos, apresentando a mesma natureza, serão abordados com a intenção de explicar em que consiste o trabalho, os fatos mais relevantes e o seu resultado final.

O primeiro trabalho consistiu na programação de um robô para realizar a etiquetagem e paletização de caixas numa paleta.

No segundo trabalho foi realizada a programação de um robô para colocação de etiquetas numa paleta com garrafas, com o objetivo de identificar as paletes.

O terceiro trabalho teve como objetivo programar um robô para paletizar pasta de argila numa paleta.

Por fim, o quarto trabalho focou-se na alteração da programação de trajetórias de um robô já existente numa ilha robotizada.

Palavras-chave: Robôs antropomórficos, Programação, *RobotStudio*, Ilhas robotizadas, Paletização, Ferramenta, Bases de trabalho

ABSTRACT

Anthropomorphic robots are multipurpose machines capable of performing tasks for which they were previously programmed. They have the ability to interpret the changing conditions of the work environment where they are inserted and to interact in order to make decisions with the objective of carrying out scheduled tasks. These machines are generally associated with the movement of a human arm, capable of performing tasks similar to those that a human being can perform and have good repeatability, an accuracy of less than a millimeter and a payload of up to hundreds of kilograms.

This report was written to document the work developed as part of a curricular internship, to obtain a Master's degree in Electrotechnical Engineering, carried out at the company EXSEPI.

This internship began on January 4, ending on September 20, 2021, and was coordinated by Electrical Engineer António Simões. This internship focused on programming anthropomorphic robots in an industrial environment. Four works were developed, all of them very similar and focused on the manipulation and palletization of objects. The first project will be explained in detail, so that the reader has an idea of all the tasks carried out until the result is obtained. The remaining works, of the same nature, will be approached with the intention of explaining what the work consists of the most relevant facts and its result.

The first work consisted of programming a robot to carry out the labeling and palletizing of boxes on a pallet.

In the second work, a robot was programmed to place labels on a pallet with bottles, in order to identify the pallets.

The third work aimed to program a robot to palletize clay paste on a pallet.

Finally, the fourth work focused on changing the trajectory programming of an existing robot on a robotic island.

Keywords: Anthropomorphic robots, Programming, RobotStudio, Robotic islands, Palletization, Tool, Work bases

AGRADECIMENTOS

Quero de agradecer ao Professor Engenheiro Francisco Nunes pela dedicação e por todo o apoio dado. Sem ele não teria sido possível a concretização deste relatório.

Agradeço também à minha esposa, Maria Carvalho, por me ter acompanhado ao longo do meu percurso académico e profissional, pela paciência e pelo apoio dado ao longo destes últimos anos.

Quero agradecer ainda ao Sr. António Ferreira pela oportunidade de ter podido realizar o estágio na EXSEPI, ao Engenheiro António Simões e ao técnico Pedro Lousado por todo o apoio dado ao longo do estágio e, por fim, a todos os colaboradores da EXSEPI, que me receberam de braços abertos e sempre prontos a ajudar, o que foi fundamental para a minha integração na empresa.

ÍNDICE

RESUMO	III
ABSTRACT	V
AGRADECIMENTOS.....	VII
ÍNDICE	IX
ÍNDICE DE ILUSTRAÇÕES.....	XI
ÍNDICE DE TABELAS	XIII
LISTA DE ABREVIATURAS E SIGLAS	XV
1. Introdução	1
1.1 Contexto e motivação.....	1
1.2 Objetivo.....	1
1.3 Enquadramento e Projetos propostos	2
1.4 Organização do relatório	3
2. Apresentação da Empresa EXSEPI.....	5
2.1 História da empresa.....	6
2.2 Missão	6
2.3 Visão	6
2.4 Áreas de negócio	7
3. Estado da Arte	9
3.1 História.....	9
3.2 Estrutura e tipologia dos principais robôs industriais.....	13
3.3 Cinemática.....	20
3.4 Tipos de linguagem de programação.....	29
4. Desenvolvimento do Estágio.....	33
4.1 Programação de um paletizador de caixas	33
4.1.1 Objetivo.....	33
4.1.2 Ferramenta.....	36
4.1.3 Layout	39
4.1.4 Bases de trabalho.....	45
4.1.5 Posições alvo e trajetórias	48
4.1.5.1 Posição home.....	49
4.1.5.2 Posição de recolha de caixa.....	49
4.1.5.3 Posição de entrega de caixa.....	51
4.1.5.4 Posição de recolha de palete de cartão	51
4.1.5.5 Posição de recolha de cartões.....	52

4.1.6	Desenvolvimento da aplicação para paletização de caixas	53
4.1.6.1	Sinais com PLC, sensores e atuadores	54
4.1.7	Programação da aplicação de paletização de caixas.....	64
4.1.8	Ensaio e comissionamento.....	82
4.2	Programação de robô para colocação de etiquetas na paleta.....	92
4.2.1	Objetivo.....	92
4.2.2	Layout	93
4.2.3	Ferramenta.....	95
4.2.4	Bases de trabalho.....	96
4.2.5	Posições e trajetórias	97
4.2.6	Desenvolvimento da aplicação.....	98
4.2.7	Ensaio e comissionamento.....	102
4.3	Paletização de pasta de argila.....	105
4.4	Alteração de ilha robotizada.....	108
5.	Conclusão	113
	Referências	117
	Anexos.....	119
	Anexo A	119
	Anexo B	120
	Anexo C	122

ÍNDICE DE ILUSTRAÇÕES

Figura 1 - Instalações da Exsepi.	5
Figura 2 - Primeiro Transístor, Nokia Bell Labs [2].	10
Figura 3 - À esquerda William Bradford Schokley (1910-1989) [26], seguindo-se John Bardeen (1908-1991) [24] e à direita Walter Houser Brattain(1902- 1987) [25].	10
Figura 4 - Karel Čapek, (1890 - 1938) [3].	11
Figura 5 - A revolta dos robôs, numa versão da peça pelo Theatre Guild em 1928 [5].	12
Figura 6 - Robô Unimate [6].	12
Figura 7 -Tipos de juntas: de rotação (à esquerda) e prismática (à direita).	13
Figura 8 - Representação do sistema de coordenadas tridimensional.	14
Figura 9 - Principais configurações de robôs [21].	15
Figura 10 - Exemplo de configuração de robô cartesiano e seu respetivo espaço de trabalho.	16
Figura 11 - Exemplo de robô cilíndrico e do seu espaço de trabalho.	17
Figura 12 - Exemplo de robô esférico e do seu espaço de trabalho.	18
Figura 13 - Exemplo de robô SCARA e do seu espaço de trabalho.	18
Figura 14 - Exemplo de robô antropomórfico e do seu espaço de trabalho.	19
Figura 15 - Sistema de referência	22
Figura 16 - Transformação direta de coordenadas.	22
Figura 17 - Representação de um sistema de coordenadas de um robô	23
Figura 18 - Notação de Denavit-Hartenberg(DH) [7]	24
Figura 19 - Configuração do elemento terminal de um robô	25
Figura 20 – Exemplo de rotações elementares, Roll, Pitch e Yaw [27].	26
Figura 21 - Exemplo ilustrativo de um robô com quatro configurações diferentes para atingir o mesmo ponto no espaço de trabalho.	27
Figura 22 - Problema da cinemática inversa: determinar uma configuração θ correspondente a uma posição desejada x	28
Figura 23 - 1 Robô, 2 controlador, 3 teach pendant(consola) e 4 ligações.	29
Figura 24 - Exemplo do simulador RobotStudio da ABB	30
Figura 25 - Fluxograma geral de funcionamento.	35
Figura 26 - Ferramenta para paletizador de caixas	36
Figura 27 - Modelo da ferramenta em 3D	37
Figura 28 - Coordenadas de referência do zero do modelo 3D da ferramenta	38
Figura 29 - Identificação dos TCP's: à esquerda a garra e à direita as ventosas	39
Figura 30 - Layout inicial da célula robotizada.	40
Figura 31 - Volume de trabalho do robô	42
Figura 32 - Robô na área da mesa de alimentação, já corrigida.	43
Figura 33 - Correção da posição do robô em relação à mesa de paletização.	44
Figura 34 - Layout da célula robotizada, com todos os equipamentos.	44
Figura 35 - Posicionamento do robô em relação à sua coordenada mundo.	45
Figura 36 - Localização da base de trabalho para paletização.	46
Figura 37 - Localização da base de trabalho da mesa de alimentação.	47
Figura 38 - Localização da base de trabalho do armazém de paletes	47
Figura 39 - Localização da base de trabalho do armazém de cartões	48
Figura 40 - Robô na posição home.	49
Figura 41 - Exemplo de entrega da caixa na ilha e posição de recolha da caixa.	50
Figura 42 - Representação da posição de entrega da caixa na paleta	51
Figura 43 - Representação da posição de recolha de paletes.	52
Figura 44 - Representação da posição de recolha de cartões.	53
Figura 45 - Modulo DSQC 1030 de entradas e saídas digitais.	55
Figura 46 - Esquema de entradas digitais.	56
Figura 47 - Representação do sensor de presença de paletes no armazém.	57
Figura 48 - Sensores de abertura e fecho da pinça.	57
Figura 49 - Sistema de vácuo(venturi) e vacuostato.	58
Figura 50 - Pressostato e electroválvula para abertura e fecho da pinça.	59
Figura 51 - Electroválvula para acionar sistema de vácuo venturi.	59

Figura 52 - Esquema das saídas digitais.	60
Figura 53 - Laser para medição de distância.	61
Figura 54 - Configuração de uma rede ethernet/ip para comunicar com o PLC.	62
Figura 55 - Declaração do número de bytes de entrada e saída por via da comunicação ethernet/ip.	62
Figura 56 - Exemplo da configuração de um sinal.	63
Figura 57 - Fluxograma do programa 'MAIN', primeira parte.	66
Figura 58 - Fluxograma do programa 'MAIN', segunda parte	67
Figura 59 - Condição que verifica se o robô está na posição 'HOME'.	68
Figura 60 - Condição de espera de nova tarefa.	68
Figura 61 - Verificação da necessidade de colocar o robô na posição de segurança, de manutenção ou KM0 do dispensador de etiquetas.	69
Figura 62 - Opção de o robô ir para a posição de manutenção, ou efetuar o KM0 da etiquetadora.	69
Figura 63 - Representação da mensagem que confirma que o robô está na posição de manutenção.	70
Figura 64 - Subprograma 'KM0_Etiq'.	70
Figura 65 - Mensagem de alerta e confirmação de início de trajetória para a posição 'KM0_Etiquetadora' do dispensador de etiquetas.	71
Figura 66 - Mensagem a informar que o robô se encontra na posição 'KM0_Etiquetadora' e indicação do procedimento para colocar o dispensador de etiquetas na posição correta.	72
Figura 67 - Opção de o robô trocar de receita.	72
Figura 68 - Página da lista de receitas no HMI.	74
Figura 69 - Exemplo da estrutura de uma receita.	75
Figura 70 - Opção do robô para reinicializar a paletização.	76
Figura 71 - Subprograma para reinicializar a paletização.	77
Figura 72 - Verificações antes do robô começar a paletizar.	78
Figura 73 - Fluxograma geral do subprograma 'Prog_1'	79
Figura 74 - Subprograma 'Prog_1'.	81
Figura 75 - Páginas do HMI com a informação enviada ao PLC e recebida do robô.	83
Figura 76 - Verificação de sinais do módulo local de entradas e saídas.	83
Figura 77 - RobotStudio: verificação da informação trocada entre o PLC e o robô.	84
Figura 78 - Exemplo de método de calibração [11].	84
Figura 79 - Apontador posicionado no centro de origem da base de paletização.	85
Figura 80 - Apontador numa posição aproximada do ponto de origem da base de trabalho.	86
Figura 81 - À esquerda (a), o robô pega numa paleta de cartão e à direita (b) deixa a mesma paleta na mesa-tapete.	87
Figura 82 - À esquerda (a), o robô a pegar numa caixa. À direita (b), a entrega da caixa na paleta.	88
Figura 83 - À esquerda (a), o robô a pegar num cartão no armazém e à direita (b) a entregá-lo na paleta.	88
Figura 84 - Paleta na envolvente de filme.	89
Figura 85 - Ajuste do dispensador de etiquetas: robô na posição KM0 da etiquetadora.	90
Figura 86 - Colocação de etiqueta na caixa.	90
Figura 87 - Exemplo de quatro paletizações diferentes.	91
Figura 88 - Layout da implantação do robô na linha de etiquetagem.	93
Figura 89 - Mesa das impressoras, tabuleiros e pistola de aplicação da cola.	94
Figura 90 - Ferramenta para pegar, fotografar e colocar a etiqueta na paleta.	95
Figura 91 - Representação do sistema de coordenadas da base de trabalho de etiquetagem.	96
Figura 92 - Representação do sistema de coordenadas da mesa dos tabuleiros.	97
Figura 93 - Posições alvo para etiquetagem da paleta.	97
Figura 94 - Posições alvo para os tabuleiros e pistola de aplicar cola.	98
Figura 95 - Fluxograma do programa MAIN	100
Figura 96 - Fluxograma do subprograma para aplicar etiqueta na paleta 'Prog_ETQ'.	101
Figura 97 - Ferramenta aplicada no terminal do robô.	102
Figura 98 - Tabuleiros para etiquetas de formato A4 e A5, gravador RFID e tabuleiro de rejeição.	103
Figura 99 - Exemplo de como fica aplicada a etiqueta na paleta, neste caso na face lateral.	104
Figura 100 - Layout da ilha robotizada para paletização de pasta de argila.	106
Figura 101 - Ferramenta para pegar na pasta e paletes.	107
Figura 102 - Exemplo do robô a efetuar a paletização de pasta em formato de cilindro.	107
Figura 103 - Consola do controlador do robô.	108
Figura 104 - Layout anterior: robô a passar na frente da estrutura de alumínio.	109
Figura 105 - Transportador inserido na ilha e robô na nova trajetória.	110
Figura 106 - Posição de deixar peça no suporte do transportador.	110

ÍNDICE DE TABELAS

Tabela 1 - Alguns exemplos de fabricantes e suas linguagens de programação.....	31
Tabela 2 - Dados correspondentes a cada paletização.	41
Tabela 3 - Exemplo de dados que são trocados entre o PLC e o controlador do robô.....	63

LISTA DE ABREVIATURAS E SIGLAS

HMI – *Human Machine Interface*

IP – *Internet Protocol*

PCB – *Printed Circuit Board*

PLC – *Programmable Logic Controller*

RF TAG – Etiqueta de Radiofrequência

RFID – *Radio Frequency Identification*

RUR – *Rossumovi Univerzální Roboti*

SCARA – *Selective Compliance Assembly Robot Arm*

TCP – *Tool Center Point*

1. Introdução

Este capítulo tem o intuito de efetuar uma introdução de como é constituído este relatório e para isso está dividido em quatro secções. O primeiro tem como propósito explicar o contexto do estágio e a motivação que levou à sua realização na empresa EXSEPI. No segundo é apresentado o objetivo do estágio. No terceiro é feito um enquadramento do tema do relatório assim como uma breve descrição dos projetos propostos. Por último, é feita uma descrição da organização do relatório.

1.1 Contexto e motivação

Com o objetivo de realizar o estágio curricular para obtenção do grau de Mestre em Engenharia Eletrotécnica, foi escolhida a empresa EXSEPI pelo fato de se tratar de uma empresa que exerce a sua atividade principal na área de estudos e desenvolvimento de projetos industriais, orientados para a automação e robótica industrial. O estágio teve como principais objetivos obter conhecimento em vários *softwares* de desenvolvimento de programação de robôs, desenvolver a capacidade de trabalhar com equipas multidisciplinares, analisar e desenvolver projetos de robótica e incrementar a experiência de trabalho em ambiente industrial.

Pode-se afirmar que estagiar na EXSEPI foi um grande privilégio, pelo fato de estar envolvida com colaboradores munidos de conhecimento e experiência em várias áreas distintas, apresentando um contexto ideal para obter e aprofundar conhecimentos e promover o desenvolvimento pessoal.

1.2 Objetivo

A empresa EXSEPI tem, na sua estrutura, uma equipa de três colaboradores que efetuam trabalhos de automação, sendo que uma destas equipas também realiza trabalhos de robótica. Tendo só uma pessoa para efetuar dois trabalhos distintos, foi proposta a criação de uma equipa de robótica, constituída por duas pessoas. Nesta equipa, um colaborador ficará dedicado exclusivamente à área da robótica enquanto o outro, que também já auxiliava nos trabalhos de robótica, terá mais disponibilidade para também se envolver

em trabalhos de automação. Assim sendo, todos os trabalhos de robótica passaram a ser efetuados por uma só pessoa, dedicada a esse efeito que neste caso é o estagiário. Os trabalhos consistem em programar, parametrizar e até criar soluções de robótica para problemas que serão propostos.

1.3 Enquadramento e Projetos propostos

Durante o estágio foram realizados vários trabalhos, denominados projetos, que consistiram principalmente na programação de robôs, em particular robôs antropomórficos.

O robô antropomórfico, devido à sua constituição ser semelhante à de um braço humano, é capaz de realizar tarefas idênticas, com repetibilidade e precisão, e até operar em condições severas que humanamente são impossíveis.

Nos dias de hoje, as empresas estão cada vez mais sensibilizadas para os problemas relacionados com o estado de saúde dos seus colaboradores, inquietas com a falta da mão de obra e empenhadas na otimização dos custos associados à produção. Todos estes fatores vão ao encontro da introdução de ilhas robotizadas nas suas empresas.

A realização de cada um dos projetos, à exceção do último, foi dividida em várias fases. Numa primeira fase foi efetuada uma análise do *layout* da ilha robotizada, com recurso ao *software* de simulação e desenvolvimento correspondente ao robô de cada projeto, que teve como objetivo identificar e corrigir possíveis problemas que, numa primeira análise, poderiam não ter sido identificados pela equipa de desenvolvimento de projetos. Na segunda fase foram criados sistemas de coordenadas locais, bases de trabalho, sistemas de coordenadas da ferramenta do robô e posições alvo principais. Na terceira fase desenvolveu-se a configuração do controlador do robô, incluindo comunicações com outros equipamentos e sinais de entrada e saída digitais. Na quarta fase efetuou-se toda a programação necessária, no *software* de desenvolvimento, para que o robô realize todas as tarefas para as quais foi destinado. E por último, na fase de teste e comissionamento, foram testados os programas em ambiente real, permitindo efetuar correções e ajustes para garantir o bom funcionamento em todas as tarefas a realizar pelo robô. Resumidamente, o comissionamento, é todo o processo de instalação, validação do funcionamento e entrega da ilha ao cliente.

No último trabalho foram simplesmente efetuadas alterações de trajetórias para a criação de um ponto de entrega no controlador de um robô já em pleno funcionamento.

1.4 Organização do relatório

O primeiro capítulo do presente relatório é dedicado à sua introdução.

No segundo capítulo é feita uma breve apresentação da empresa EXSEPI: como surgiu e as suas missão, visão e áreas de negócio em que atua.

No terceiro capítulo está elaborado o estado da arte, em que é abordada muito resumidamente a história do aparecimento dos robôs na indústria e da palavra robô. Também, e devido à sua complexidade, é feito um resumo da estrutura e da tipologia dos robôs industriais, assim como da sua cinemática e dos tipos de linguagens utilizadas para a sua programação.

No quarto capítulo serão abordados todos os trabalhos desenvolvidos ao longo do estágio. Para isso foi dividido em quatro secções, destinados a descrever os trabalhos individualmente.

No primeiro trabalho foi proposta a programação um robô para a realização da paletização de caixas. Para além de paletizar caixas, o robô tem de realizar outras tarefas tais como colocar uma paleta de cartão antes de iniciar a paletização, colocar cartões ente camadas ou planos e afixar etiquetas para identificação do produto nas caixas. Para a realização deste trabalho foram definidos os seguintes objetivos: introduzir o modelo 3D da ferramenta no simulador do respetivo robô e criar o seu sistema de coordenadas, denominado por *TCP (Tool Center Point)*; introduzir no simulador do respetivo robô o modelo 3D de toda a ilha e detetar possíveis constrangimentos para o robô no *layout*; criar sistemas de coordenadas, denominadas bases de trabalho, no robô que facilitem a sua programação; criar posições alvo e trajetórias; desenvolver toda a aplicação, que consiste na parametrização e programação do robô para realizar as tarefas pretendidas; por fim, executar a fase de testes e comissionamento da ilha no cliente.

No segundo trabalho foi proposta a realização da programação de um robô com o propósito de etiquetar paletes com garrafas de vidro. Neste trabalho o robô também realiza outras tarefas cruciais para atingir o seu principal objetivo. Estas tarefas consistem em inspecionar a etiqueta impressa, pegar na etiqueta e aplicar cola. Para a realização deste trabalho, e sendo do mesmo tipo de estrutura do trabalho anterior, foram definidos

os mesmos objetivos do projeto anterior. Por serem repetidos, serão descritos de uma forma mais simplificada.

No terceiro trabalho foi proposta a programação de um robô para realizar a paletização de pasta de argila em paletes. O robô, para além de colocar a pasta de argila numa paleta, também coloca a paleta de madeira antes de começar a paletizar. Não sendo diferente dos projetos anteriormente apresentados, os objetivos serão os mesmos, sendo estes também descritos de uma forma resumida.

No quarto e último trabalho, foi proposto reprogramar um robô já existente numa ilha robotizada. Este robô recolhe peças de um tapete para realizar o acabamento final e coloca-as em duas máquinas: numa delas, após a peça estar finalizada, coloca-a num carrinho que, quando estiver completo com peças, é retirado por um operador para ser esvaziado. A outra máquina, simplesmente recebe a peça para efetuar o acabamento e envia-a para a linha seguinte. O objetivo deste trabalho é que o robô consiga realizar a entrega das peças já prontas num transportador, que as leva para outra máquina fora da ilha robotizada. Para este trabalho foram definidas as seguintes tarefas: análise e identificação de trajetórias efetuadas pelo robô; alteração de trajetórias; por fim, adicionar mais um ponto de entrega de peças maquinadas.

No último capítulo é feita uma retrospectiva do estágio, através de um resumo de todo o trabalho realizado, no qual se explica os trabalhos e as suas dificuldades e o contributo do estágio para o desenvolvimento profissional e pessoal do estagiário.

2. Apresentação da Empresa EXSEPI

A EXSEPI é uma empresa sediada em Anadia desde 1991 e tem como principal área de negócio a prestação de serviços de engenharia, com trabalhos realizados em várias empresas do sector industrial e terciário no mercado nacional e internacional.



Figura 1 - Instalações da Exsepi.

A EXSEPI atua desde a fase de conceção inicial e projeto, até à instalação e funcionamento final de soluções integradas de todas as especialidades de automação industrial, processo e controlo industrial, comunicações e sistemas de informação, ambiente e eficiência energética.

Devido ao elevado grau de rigor que impõe em cada projeto, aos vastos conhecimentos técnicos e à capacidade de inovação dos seus quadros, a EXSEPI colocou desde sempre um desempenho notável em todas as fases de projeto, respondendo aos problemas do presente com soluções de futuro.

A EXSEPI tem uma filial, EXSEPI-IBERIA, que foi constituída em 2015 com o objetivo de prestar serviços de forma mais estreita no mercado espanhol. O escritório está localizado no Parque Empresarial de La Moraleja – Madrid [1].

2.1 História da empresa

Na região de Aveiro existe um grande número de empresas ligadas à cerâmica. O resultado deste prodígio deve-se ao facto de a região ser rica numa matéria-prima que é a argila. Existindo este afluente de empresas de cerâmica na região, um grupo de pessoas ligadas às mais diversas áreas, tais como eletrónica, programação e gestão, reuniram-se e visualizaram e identificaram a necessidade de criar uma empresa prestadora de serviços ligada às áreas de manutenção, programação e instalações elétricas industriais. Deste modo foi criada a EXSEPI, começando a prestar serviços na área de projetos de automação ligados à cerâmica e, com o passar dos anos, foi criando conhecimento e evoluindo em outras áreas industriais. Nos dias de hoje a EXSEPI tem aproximadamente 40 colaboradores que se complementam nas áreas de desenho técnico, desenho de esquemas elétricos, programação, robótica, segurança, comercial, marketing, eletrónica e operacional.

2.2 Missão

A missão da EXSEPI é a execução de estudos, desenvolvimento e execução de projetos e instalação de soluções industriais integradas. Compromete-se a promover e desenvolver a atividade com “Qualidade e Inovação”, de forma coerente com a Política e Objetivos Organizacionais, garantindo a sustentabilidade do negócio e do Ambiente e as melhores condições de Segurança e Saúde no Trabalho [1].

2.3 Visão

A EXSEPI tem como visão ser uma organização prestadora de serviços líder de mercado, gerando uma imagem de solidez pela apresentação de uma estrutura flexível, competitiva e capaz, em permanente procura das melhores soluções para o cliente [1].

2.4 Áreas de negócio

A EXSEPI é uma empresa dotada de colaboradores especialistas em várias áreas distintas e tem a capacidade de atuar em várias áreas da tecnologia vocacionadas para a indústria, tais como:

- Sistemas de Automação e Controlo Industrial;
- Robótica Industrial e Visão Artificial;
- Projetos e Instalações Elétricas;
- Sistemas de Pesagem Eletrónica;
- *Software* de Supervisão e Controlo;
- Ambiente:
- Construção de Equipamentos por Medida;
- Sistemas de Gestão de Energia,
- Desenho e Projeto;
- Manutenção Industrial.

3. Estado da Arte

Nos dias de hoje vive-se num mundo dada vez mais desenvolvido e devido aos rápidos avanços tecnológicos em robótica e automação, o setor da indústria tem testemunhado uma maior adoção da engenharia e tecnologia robótica nos seus processos de produção. Robôs industriais estão a ser utilizados para realizar tarefas com alta precisão e repetibilidade, resultando em produtos de qualidade. A capacidade dos robôs industriais de trabalhar continuamente melhora a *performance* das indústrias e aumenta a sua produção. Além disso, os *robots* podem trabalhar em ambientes perigosos e prejudiciais para o ser humano, melhorando assim as condições de trabalho e a segurança na instalação de produção. Portanto, as diversas vantagens que os robôs trazem são um bom incentivo para que os fabricantes queiram integrar diferentes tipos de robôs industriais nas suas linhas de produção, aumentando a eficiência e o lucro da produção.

3.1 História

Ao longo dos anos, a produção industrial tem evoluído muito rapidamente, tendo de acompanhar as necessidades de consumo, que têm sido cada vez maiores. Com isto, tem sido todos os dias um desafio constante para as empresas conseguirem produzir da maneira mais eficaz, mais económica e mais rápida.

Muitas das tarefas de produção são repetitivas, por vezes pesadas e prejudiciais para a saúde do ser humano. É neste paradoxo que se quer produzir mais e melhor, com custo e desperdício de recursos mais reduzidos, para não falar da mão de obra, cada vez mais escassa em alguns sectores. As empresas veem cada vez mais os robôs como um meio alternativo para determinadas tarefas de produção, sejam elas de manipulação ou de transporte e armazenamento. Desta forma, podem alocar os seus operadores a outras tarefas que sejam mais difíceis de ser efetuadas por robôs, tais como administrativas, manutenção, monitorização e controlo de máquinas ou células robotizadas, entre outras. Os robôs estão presentes em quase tudo no nosso quotidiano, desde um simples aspirador de casa até ao robô de exploração no espaço, passando pelos robôs de inteligência artificial (humanoides), assistência a operações cirúrgicas, robôs assistentes sociais, robôs de condução autónoma, robôs industriais, entre outros.

1947 foi o ano em que a tecnologia iniciou um grande progresso, com a invenção do primeiro transistor. Com o desenvolvimento deste tipo de tecnologia, foi possível, nos anos seguintes, criar os primeiros circuitos integrados, sendo talvez a invenção que mais revolucionou a tecnologia nos anos posteriores.



Figura 2 - Primeiro Transistor, Nokia Bell Labs [2].

O transistor veio substituir as válvulas de vácuo, o que revolucionou toda a indústria. A descoberta e as pesquisas relacionadas com semicondutores foram o contributo para que em 1956 fosse atribuído o prémio nobel da Física a *William Bradford Shockley*, *John Bardeen* e *Walter Houser Brattain* [2].

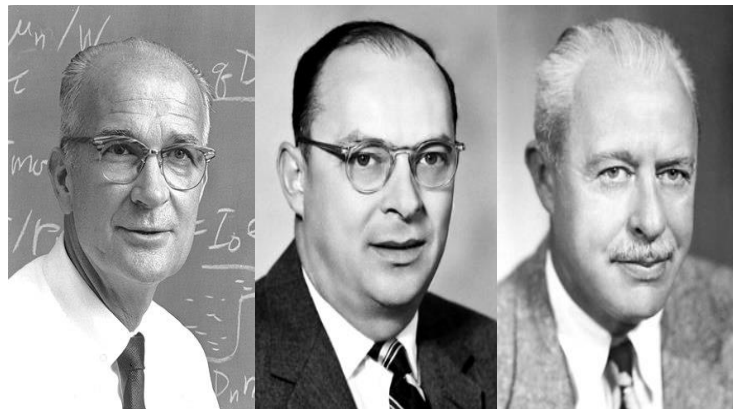


Figura 3 - À esquerda William Bradford Schokley (1910-1989) [26], seguindo-se John Bardeen (1908-1991) [24] e à direita Walter Houser Brattain(1902- 1987) [25].

A definição de robô, vem originalmente do idioma checo ‘*robota*’, que significa “trabalho forçado”.

O termo robô, foi criado por *Karel Čapek* (1890-1938), escritor checo que escreveu um romance famoso em 1921 chamado, “*R.U.R.*” (“Robôs Universais de Rossum”) [3].



Figura 4 - Karel Čapek, (1890 - 1938) [3].

Em resumo, na peça de *Čapek*, Robôs Universais de *Rossum*, os robôs são produzidos em massa por outros robôs em linhas de montagem. A idealista *Helena Glory*, presidente da *The Humanity League*, acredita que os robôs têm (ou têm em vias de desenvolvimento) alma e acha que eles deveriam ser libertados.

Os robôs podem claramente pensar por si próprios, embora estejam contentes por servir. Eles lembram-se de tudo, mas não pensam em nada original ou único. O excêntrico cientista Old Rossum estava decidido a assumir o papel do criador, ao reproduzir artificialmente um homem em detalhes intrincados, enquanto o pragmático economista industrial Young Rossum produzia versões simplificadas da humanidade, para serem vendidas como trabalhadores baratos, ou seja, robôs.

De vez em quando, um dos robôs deixa o seu trabalho e começa a ranger os dentes. Embora muitos discordem, incluindo o *Dr. Hellman*, psicólogo-chefe, *Helena Glory* sente que é uma evidência e um sinal da alma em constante evolução nos robôs. Depois de se casar com *Harry Domin*, gerente geral da RUR, *Helena* pressiona os cientistas a modificar alguns dos robôs, para que as suas “almas” possam desenvolver-se mais rapidamente. Um dos robôs modificados de *Helena* emite o prenúncio de um plano: “Robôs do mundo, vocês receberam ordens de exterminar a raça humana... O trabalho não deve parar!”.

Domin possui as fórmulas para criar os robôs e planeia usá-las como ferramenta de troca a seu favor. *Helena*, ignorando a verdadeira ameaça em mãos, queima as fórmulas. Os robôs reúnem e matam todos os humanos, deixando apenas o *Clerk* (chamado *Alquist*). O

líder do robô, Damon, tenta vezes sem conta fazer com que o *Clerk* descubra como ajudá-los a povoar a terra, mas sem sucesso, pois *Clerk* não sabia como produzir outros robôs. Eventualmente, dois robôs, *Helena* (um belo robô modificado com o nome de *Helena Glory*), e outro robô chamado *Primus*, apaixonam-se. Com a bênção de Alquist, os amantes casam-se e chamaram-se Adão e Eva (Jerz, "RUR") [4] [5].



Figura 5 - A revolta dos robôs, numa versão da peça pelo Theatre Guild em 1928 [5].

O primeiro robô industrial surgiu em 1954, criado por *George Devol*, ao qual chamou *Unimate*. Este robô foi implementado na produção da *General Motors* em 1961: tinha como função pegar em peças quentes e colá-las nos chassis, ou seja, fundição por injeção. O robô executava as tarefas que estavam gravadas numa fita magnética.

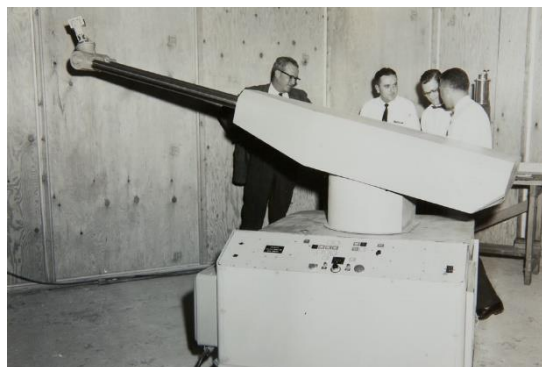


Figura 6 - Robô Unimate [6].

Os braços industriais modernos (ou manipuladores robóticos), aumentaram a sua capacidade e desempenho a partir da década de 1980, com o uso de microprocessadores e linguagens de programação mais avançadas. Estes avanços foram obtidos em parte graças aos grandes investimentos das empresas ligadas ao ramo automóvel [6].

3.2 Estrutura e tipologia dos principais robôs industriais

Os robôs manipuladores foram inventados com base na semelhança a um braço humano. É com esse conceito que, utilizando vários tipos de energia, tais como hidráulica, pneumática e elétrica, foi possível a construção e a evolução dos robôs manipuladores. Os robôs manipuladores são compostos por elementos rígidos aos quais se dá o nome de elos, que estão conectados entre si por juntas (ou eixos), numa cadeia cinemática aberta ou fechada.

Ao primeiro elemento fixo da cadeia dá-se o nome de base e o último é o elemento terminal (ou punho). No punho é montada a ferramenta ou garra que permite efetuar a tarefa para a qual o robô foi concebido. Os exemplos mais comuns de garras podem ser uma pinça ou um sistema de ventosas para manipular objetos. Já a ferramenta pode ser equipamento para efetuar soldadura, polidor, leitor de código de barras para inspeção, pistola para colocação de cola, entre muitas outras aplicações.

A função do braço é o posicionamento e a orientação do punho, no qual está montada a ferramenta anteriormente referida.

As juntas podem ser de rotação (R) ou prismáticas/de translação (P). No sistema de coordenadas cartesianas, o eixo de rotação, na junta rotacional, ou o deslocamento, no caso da prismática, representam-se no eixo z. Os elos são representados por " l_n ", " θ " representa o ângulo de rotação e " d " o deslocamento, como ilustrado na Figura 7.

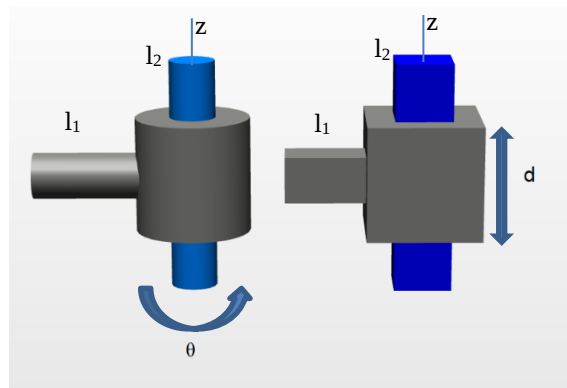


Figura 7 -Tipos de juntas: de rotação (à esquerda) e prismática (à direita).

Para conseguirem alcançar no espaço o ponto desejado, os robôs têm de efetuar uma combinação de movimentos com o conjunto de elos ligados por juntas com que são

constituídos. Para isso, é necessário definir um número de variáveis independentes, capazes de descrever cada movimento. Num sistema mecânico, dá-se o nome de graus de liberdade ou mobilidade, aos movimentos independentes que ele pode efetuar.

Num plano “xyz”, por cada movimento segundo um eixo obtém-se um grau de liberdade: neste caso serão 3 graus de liberdade. Se aplicarmos a mesma metodologia num espaço tridimensional, além dos movimentos de translação é adicionado um movimento de rotação por cada eixo, contabilizando um total de 6 graus de liberdade (Figura 8).

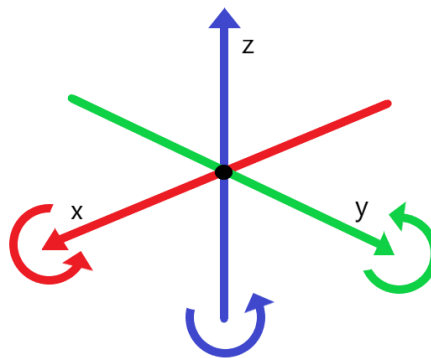


Figura 8 - Representação do sistema de coordenadas tridimensional.

Para posicionar o elemento terminal no espaço, o robô necessita de três graus de liberdade e para definir a sua orientação necessita de mais três. Uma vez que só são necessários 6 graus de liberdade para se movimentar no espaço, como anteriormente referido, um robô com mais eixos torna-se redundante.

Com base na configuração mecânica dos três primeiros graus de liberdade, a contar a partir da base, os robôs manipuladores industriais podem ser classificados em 5 principais tipos: cartesiano, cilíndrico, polar (ou esférico), articulado (ou antropomórfico) e SCARA (*Selective Compliance Assembly Robot Arm*) que, em português, significa, Braço Robótico para Montagem de Conformidade Seletiva. Além da configuração mecânica, os robôs industriais podem ser caracterizados com base no controlo de movimento, controlo da fonte de alimentação e características físicas.

No que se refere à fonte de alimentação, as juntas de um robô podem ser acionadas por atuadores elétricos, que são os mais utilizados, hidráulicos ou pneumáticos.

Os atuadores elétricos, têm a vantagem de ser mais limpos, silenciosos e mais fáceis de controlar. Os hidráulicos têm como principal vantagem a possibilidade de gerar grandes forças. A necessidade de equipamentos periféricos tais como uma bomba, que pode gerar ruído, um tanque, uma central de comandos e tubagens, são as suas principais desvantagens. Os atuadores pneumáticos são simples de aplicar e são baratos. A sua baixa precisão e a dificuldade de efetuar o seu controlo apresentam-se como as suas principais desvantagens.

O controlo dos atuadores pode ser efetuado em malha aberta ou fechada. Em malha aberta, os movimentos são limitados por batentes físicos, não existindo controlo sobre os mesmos. Em malha fechada, podem existir dois tipos de controlo: o primeiro é o controlo de ponto a ponto, sem que exista um controlo na trajetória do ponto terminal do robô, sendo normalmente programados vários pontos para o robô seguir uma determinada sequência ao efetuar uma tarefa; no segundo tipo de controlo, a trajetória do punho do robô pode ser controlada, podendo-se programar vários pontos tais que a trajetória entre eles possa ser uma reta ou uma curva, conseguindo-se ainda definir a orientação do punho, a aceleração e a velocidade. Este último tipo de controlo é efetuado por controladores(processadores), com os quais são efetuados cálculos de cinemática.

O volume de trabalho de um robô representa o espaço envolvente que o punho ou a ferramenta, se esta estiver instalada no punho, consegue alcançar. A forma do volume depende da estrutura do manipulador bem como das limitações das juntas.

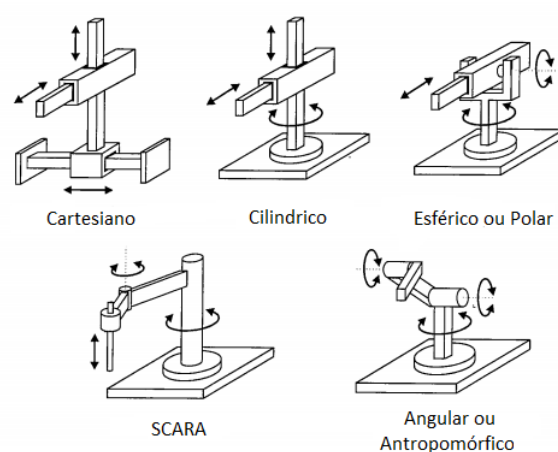


Figura 9 - Principais configurações de robôs [21].

Os robôs cartesianos, apresentam uma configuração básica de três juntas prismáticas (PPP) e podem realizar movimentos lineares em 3 dimensões (Figura 10). São de cinemática simples e de fácil programação, mas limitados na orientação do elemento terminal do robô. Devido à configuração deste tipo de robôs, o espaço de trabalho é muito reduzido relativamente ao espaço que ocupa. Este tipo de robô, pode ser utilizado para aplicações de movimentação de peças, paletização, impressão, maquinação ou soldadura. Em algumas aplicações, pode ser adicionada uma junta rotacional no elemento terminal do robô, para orientação da ferramenta ou garra.

Um exemplo muito comum, nos dias de hoje, de um robô cartesiano, é uma impressora 3D.

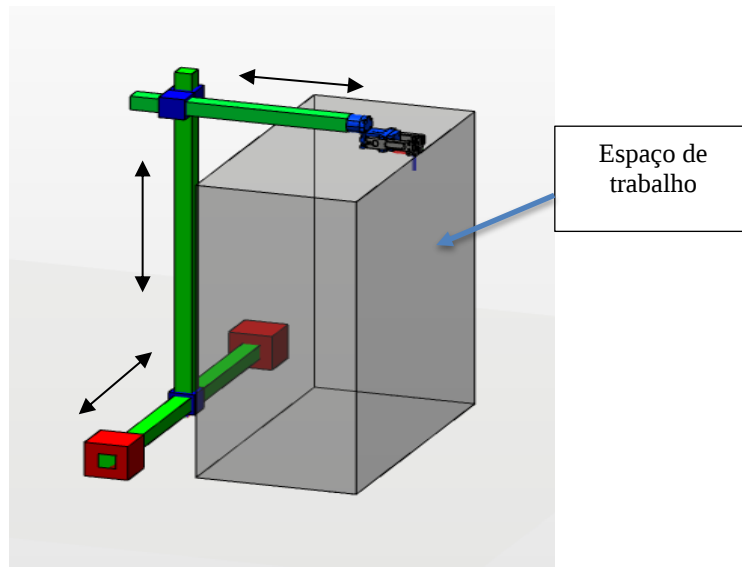


Figura 10 - Exemplo de configuração de robô cartesiano e seu respetivo espaço de trabalho.

Os robôs cilíndricos apresentam uma configuração básica de uma junta de rotação na sua base e duas juntas prismáticas até ao elo terminal (RPP) (Figura 11). São de cinemática simples e a combinação do movimento rotacional com os prismáticos resulta num espaço de trabalho à sua volta com a forma de um cilindro. Relativamente ao robô cartesiano, o robô cilíndrico apresenta a possibilidade de conseguir trabalhar à sua volta. É uma vantagem para aplicações com várias zonas de trabalho distintas como, por exemplo, numa aplicação para soldar peças, na qual podem ser criadas duas zonas de soldadura: quando o robô estiver a soldar uma peça numa zona, na outra zona está a ser retirada outra peça já soldada e a ser preparada uma outra para soldar. Desta forma consegue-se otimizar os tempos de paragem do robô, enquanto se retira uma peça já soldada ou se introduz uma

nova peça para soldar. Estes robôs podem ser utilizados para aplicações de manipulação, movimentação e soldadura. À semelhança do robô cartesiano, o cilíndrico também apresenta limitações na orientação do terminal. Para as minimizar, pode ser adicionado um mecanismo com uma junta rotacional, tornando possível efetuar uma orientação em “x” do novo elemento terminal após a junta.

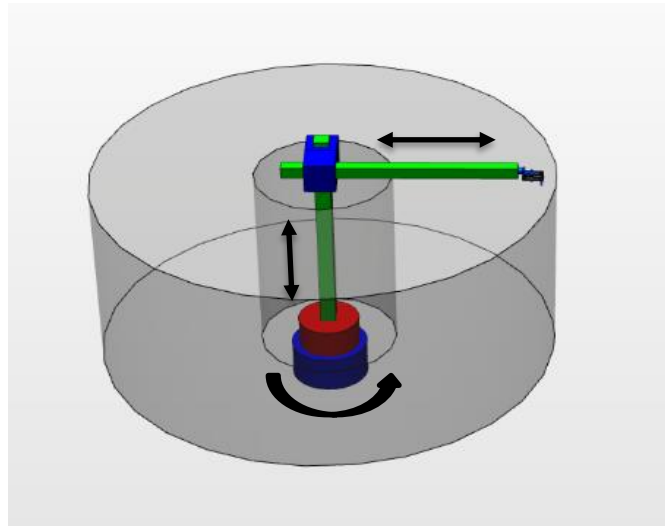


Figura 11 - Exemplo de robô cilíndrico e do seu espaço de trabalho.

Os robôs de coordenadas polares apresentam uma configuração com duas juntas rotacionais e uma prismática (RRP)(Figura 12). Este tipo de robô tem uma cinemática um pouco mais complexa e o seu espaço de trabalho tem a forma de uma esfera. O facto de ter duas juntas rotacionais permite que consiga alcançar posições acima do robô. Este tipo de robô, em comparação com o cilíndrico, apresenta um volume de trabalho maior, consegue pegar ou manipular objetos do chão ou até mesmo acima dele, mas é mais limitado na movimentação de objetos no eixo “z”: por exemplo, não consegue pegar num objeto e fazer um movimento linear em “z”, sem alterar a orientação do objeto. Normalmente, é utilizado para montagem ou movimentação de peças.

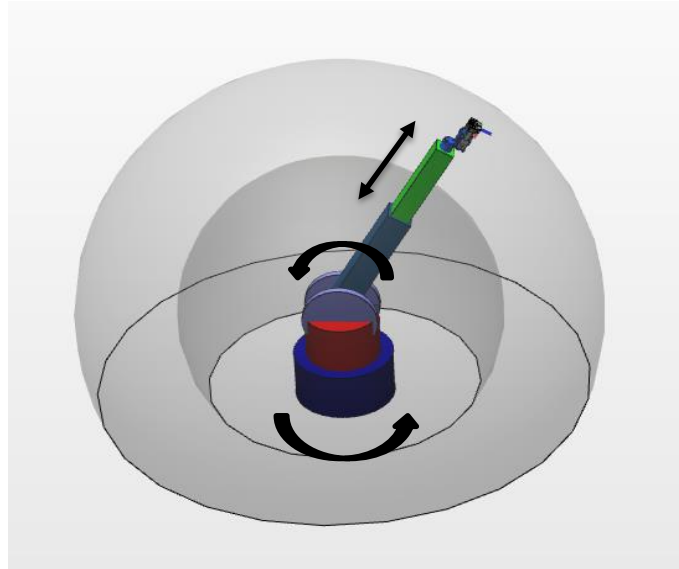


Figura 12 - Exemplo de robô esférico e do seu espaço de trabalho.

Os robôs SCARA apresentam uma configuração de duas juntas rotativas paralelas e uma junta prismática(Figura 13). Estes robôs têm uma área de trabalho muito parecida com um cilindro. Os robôs deste tipo são pequenos e de fácil integração. São rápidos a efetuar movimentos e é muito comum serem utilizados em aplicações de montagem de movimentação de peças. Alguns exemplos poderão ser uma aplicação para apertar parafusos ou a inserção de componentes eletrônicos numa PCB (placa de circuito impresso)

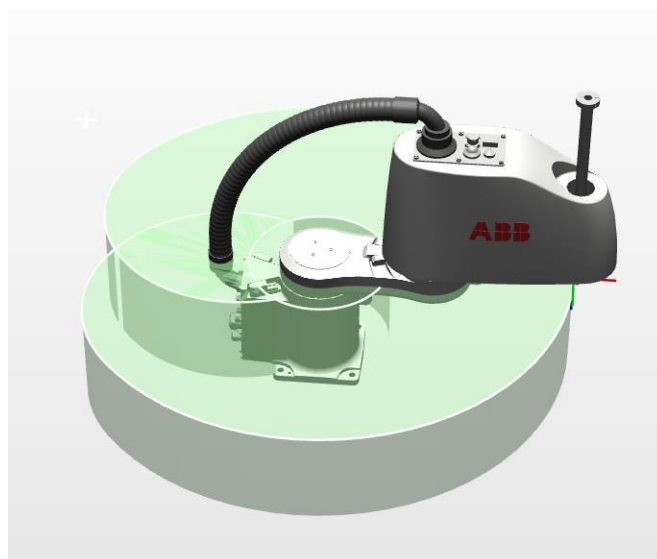


Figura 13 - Exemplo de robô SCARA e do seu espaço de trabalho.

Por fim, os robôs antropomórficos apresentam uma configuração de três juntas rotativas (RRR)(Figura 14). Para além destas três juntas, o mais usual neste tipo de robôs é existir um punho com mais 3 juntas rotativas. Com esta configuração, é possível efetuar posicionamento e orientação do terminal no espaço “xyz”, o que resulta num espaço de trabalho maior quando comparado com os robôs antes referidos. Estes têm uma cinemática complexa e podem ser utilizados em aplicações como soldadura, maquinação, pintura, manipulação e movimentação de cargas, paletização, polimento, inspeção e testes.

Este tipo de robô é o que se mais assemelha a um braço humano. O braço humano é constituído por um ombro com 3 juntas rotativas, um cotovelo com mais uma junta rotativa e, por fim, um punho com mais um conjunto de 3 juntas rotativas. Sendo assim, o braço humano tem um total de 7 graus de liberdade, o que faz dele um sistema redundante, não existindo singularidades [7]. As singularidades acontecem quando o robô apresenta uma configuração tal que seria necessário impor velocidades infinitamente altas, numa ou mais juntas, para manter determinadas velocidades do terminal do punho, quando este se movimenta numa trajetória.

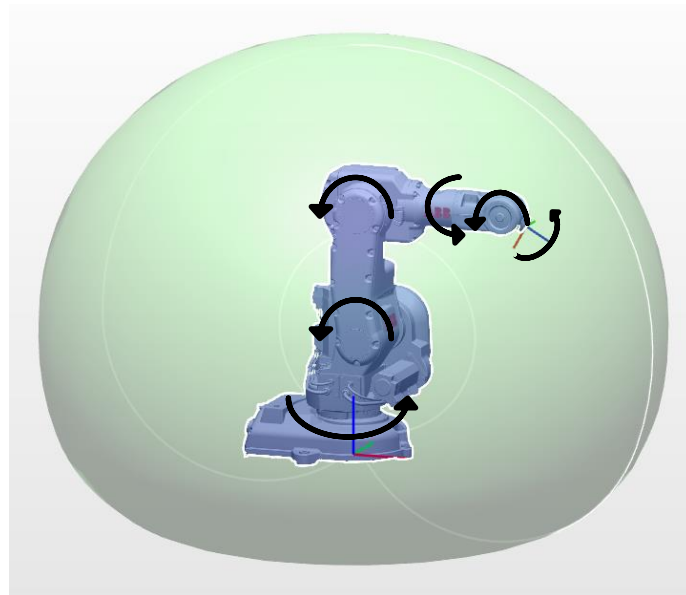


Figura 14 - Exemplo de robô antropomórfico e do seu espaço de trabalho.

Dum modo geral, os tipos de robôs industriais anteriormente referidos podem ser utilizados para os mais variados tipos de aplicações, tais como: movimentação de peças, paletização, soldadura, montagem de peças, pintura, polimento, inspeção de peças ou

equipamentos, teste e maquinação, entre outras. A seleção do tipo de robô para determinada aplicação depende de vários fatores como, por exemplo, o *payload* (*carga útil que pode suportar*), o espaço de trabalho envolvente na aplicação, o seu alcance ou o número de graus de liberdade necessários para efetuar a tarefa,

3.3 Cinemática

Neste capítulo será feito um breve resumo de como é constituída a cinemática de um robô antropomórfico industrial. Não é o propósito deste relatório efetuar desenvolvimentos matemáticos que descrevam a cinemática de um robô, mas sim, de uma forma muito resumida, dar a conhecer os aspetos da cinemática de um robô.

Um robô é constituído por elos conectados por juntas prismáticas ou rotacionais. Cada par junta-elos constitui um grau de liberdade. Assim, para um manipulador com N graus de liberdade, temos N pares juntas-elos, onde o primeiro elo é a base de sustentação do robô (sistema de coordenadas inerciais fixo) e no seu último elo é incorporada a ferramenta de trabalho ou garra. De um modo geral, num robô de seis graus de liberdade, os três primeiros são responsáveis pelo posicionamento da sua ferramenta no espaço de tarefas e os restantes graus de liberdade são responsáveis pela sua orientação. Assim, a primeira conexão ocorre entre o primeiro elo e o suporte da base e o último elo representa o ponto de conexão entre o elemento terminal do robô e a ferramenta. Na maioria das aplicações industriais, a programação de tarefas do robô é realizada por aprendizagem, consistindo na movimentação do robô até a posição alvo e depois gravar a sua posição atual. Este procedimento é efetuado individualmente para cada uma das posições alvo desejadas. Assim sendo, a programação de trajetórias de um robô torna-se mais simples, não necessitando de um conhecimento do modelo matemático, sendo a fase de aprendizagem, basicamente, uma operação de armazenamento de uma sequência de incrementos necessários para que o conjunto de variáveis articulares determine um posicionamento, especificado a partir de um perfil de trajetórias fornecido (robô controlado a partir do sistema de coordenadas de juntas). Como um robô é controlado através das suas variáveis articulares, a realização do controlo de posição em relação ao sistema de coordenadas cartesianas implicará o desenvolvimento de metodologias para transformação de coordenadas. A transformação de coordenadas polares para cartesianas

é normalmente realizada em tempo real, pelo controlador do robô. A partir do conjunto de variáveis articulares serão obtidas a posição e a orientação da sua ferramenta.

Um sistema articular pode ser representado matematicamente através de n corpos móveis C_i ($i = 1, 2, \dots, n$) e de um corpo C_0 fixo, interligados por n articulações, formando uma estrutura em cadeia. Para representar a situação relativa dos vários corpos da cadeia, é fixado a cada elemento C_i um referencial R . Podemos relacionar um determinado referencial $R_{i+1} (O_{i+1}, x_{i+1}, y_{i+1}, z_{i+1})$ com o seu anterior $R_i (O_i, x_i, y_i, z_i)$, como também com o sistema de coordenadas de origem da base (Figura 15) através da equação(1), onde $A_{i,i+1}$ representa as matrizes de transformação homogênea de rotação e L_i o vetor de translação de uma origem à outra. $A_{i,i+1}$ é resultante do produto matricial global entre as diversas matrizes de transformações homogêneas relacionadas com rotações ou translações sucessivas das diferentes articulações (equação (2)).

$$O_{i+1} = O_i + A_{i,i+1} * L_i \quad (1)$$

$$A_{i,i+1} = A_{1,2} * A_{2,3} * \dots * A_{i,i+1} \quad (2)$$

sendo,

$$A_{i,i+1} = \begin{bmatrix} Nx_o & Sx_o & Ax_o \\ Ny_o & Sy_o & Ay_o \\ Nz_o & Sz_o & Az_o \end{bmatrix} \quad (3)$$

Qualquer rotação no espaço pode ser decomposta num grupo de rotações elementares ao longo dos eixos X , Y e Z . A matriz de rotação elementar usada na equação de transformação é associada com a rotação elementar do referencial correspondente ao seu anterior. Este procedimento matemático pode ser estendido para toda a extensão do modelo. Assim sendo, a matriz de orientação de um ponto de interesse pode ser obtida pela equação (2). Consequentemente, o posicionamento completo de um corpo rígido no espaço poderá ser facilmente obtido através da equação(1) que fornece o seu vetor posição, sendo que a equação (3) representa a matriz de orientação associada, podendo ser expressa através de componentes angulares associadas às três direções de rotação

correspondentes aos eixos de referência do sistema de coordenadas (por exemplo, Roll, Pitch, Yaw - RPY ou quaternions).

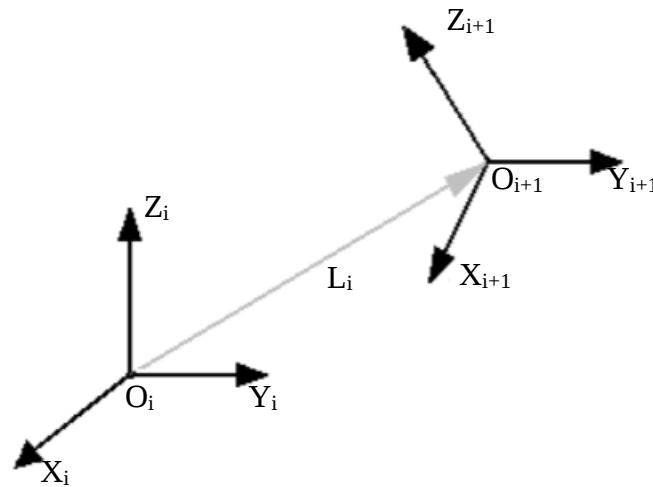


Figura 15 - Sistema de referência

Nas diversas aplicações industriais, um robô pode ser controlado e programado a partir do sistema de coordenadas associadas à sua ferramenta. É muito mais natural expressarmos o deslocamento absoluto do elemento terminal de um robô que considerarmos a variação das suas coordenadas articulares, embora a malha de controle de uma junta robótica seja estabelecida a partir da comparação de grandezas articulares, tornando-se necessário a realização de uma transformação geométrica apropriada para o estabelecimento da correspondência entre as variáveis articulares θ_i e as coordenadas absolutas do elemento terminal X_i . A Figura 16 apresenta o *input* e *output* da transformação direta de coordenadas para um robô com N graus de liberdade.

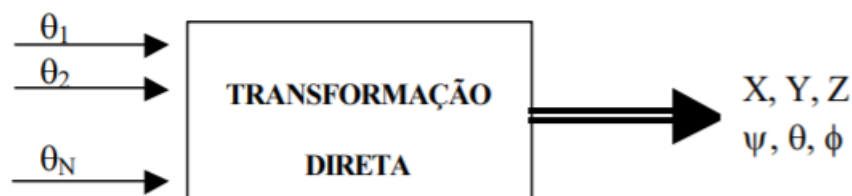


Figura 16 - Transformação direta de coordenadas.

A operação que realiza a correspondência entre esses dois espaços é chamada de transformação de coordenadas. A transposição direta de coordenadas apresenta uma

solução única, o mesmo não acontecendo com o problema inverso, onde manipuladores com um número de graus de liberdade superior a três podem conduzir a soluções múltiplas.

O modelo geométrico de um robô expressa a posição e a orientação do seu elemento terminal em relação a um sistema de coordenadas fixo à base do robô (Figura 17), em função das suas coordenadas generalizadas (coordenadas angulares, no caso de juntas rotacionais).

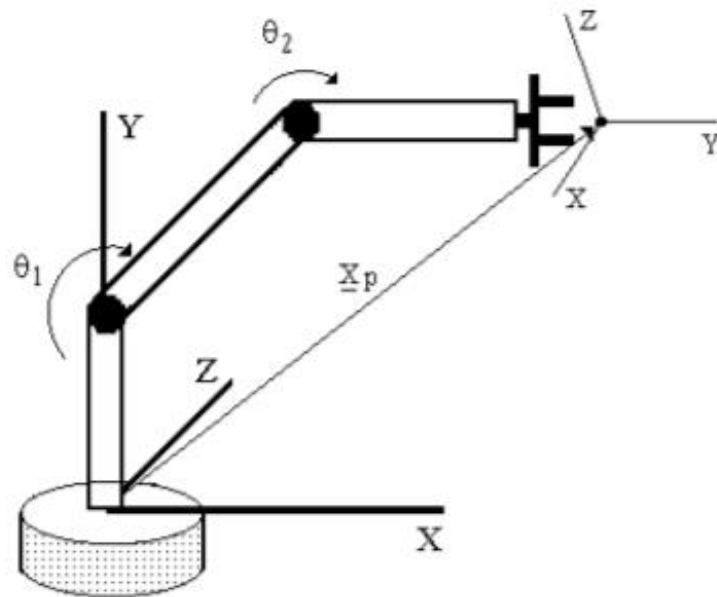


Figura 17 - Representação de um sistema de coordenadas de um robô

A evolução no tempo das coordenadas das juntas de um robô representa o modelo cinemático de um sistema articulado no espaço tridimensional. A notação de *Denavit-Hartenberg* (DH) é uma ferramenta utilizada para sistematizar a descrição cinemática de sistemas mecânicos articulados com N graus de liberdade [7]

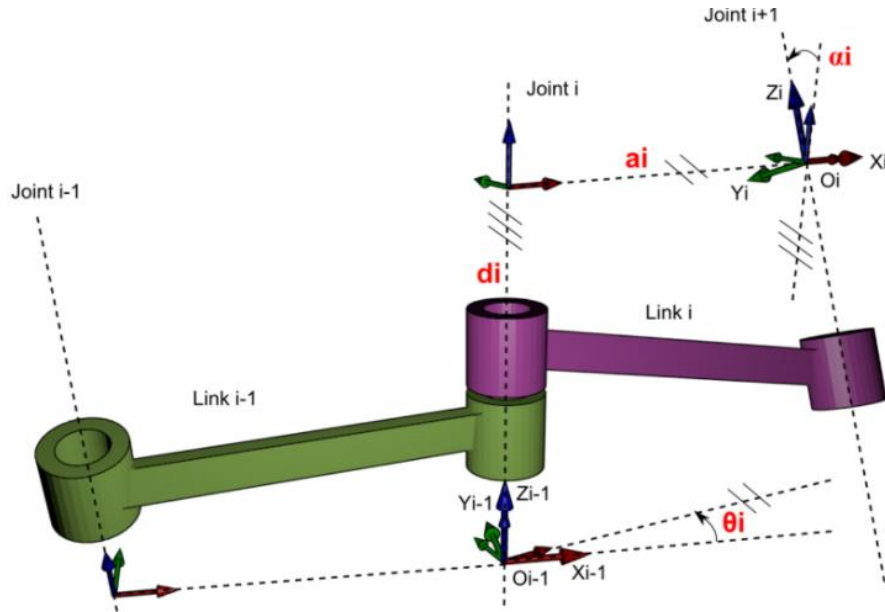


Figura 18 - Notação de Denavit-Hartenberg(DH) [7]

Na Figura 18 podemos visualizar dois elos conectados por uma junta que tem duas superfícies deslizantes, uma sobre a outra que permanecem em contacto. Um eixo de uma junta estabelece a conexão de dois elos. Estes eixos de juntas devem ter duas normais neles conectadas, uma para cada um dos elos. A posição relativa destes dois elos conectados (elo i-1 e elo i) é dada por d_i , que é a distância medida ao longo do eixo da junta entre as suas normais. O ângulo de junta θ_i entre as normais é medido num plano normal ao eixo da junta. Assim, d_i e θ_i podem ser chamados, respetivamente, distância e ângulo entre elos adjacentes. Eles determinam a posição relativa de elos vizinhos. Um elo i poderá estar conectado, no máximo, a dois outros elos (elo i-1 e elo i+1). Assim, dois eixos de junta são estabelecidos em ambos os terminais de conexão. O significado dos elos, do ponto de vista cinemático, é que eles mantêm uma configuração fixa entre as suas juntas, que podem ser caracterizadas por dois parâmetros: a_i e α_i . O parâmetro a_i é a menor distância medida ao longo da normal comum entre os eixos de junta (isto é, os eixos z_{i-1} e z_i para as juntas i e i+1, respetivamente). Assim, a_i e α_i , podem ser chamados, respetivamente, comprimento e ângulo de twist (torção) do elo i. Eles determinam a estrutura do elo i. Assim sendo, quatro parâmetros, a_i , α_i , d_i e θ_i , são associados a cada elo do manipulador.

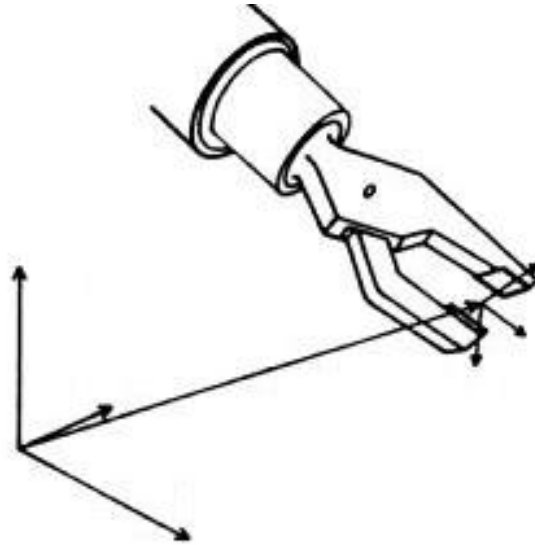


Figura 19 - Configuração do elemento terminal de um robô

A orientação de um sistema de coordenadas (por exemplo, referencial de uma ferramenta de trabalho) pode ser descrita como uma matriz de rotação que descreve a direção dos eixos do sistema de coordenadas em relação a um sistema de referência (Figura 19). A expressão (4) mostra que os eixos do sistema de coordenadas rotativo (n , s , a), vetores, podem ser expressos em relação ao sistema de coordenadas de referência através de componentes nas direções X_o , Y_o e Z_o . Estes três vetores, podem ser dispostos em colunas numa matriz rotacional designada matriz de orientação (4):

$$T = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \quad (4)$$

Esta matriz é constituída por seis parâmetros, componentes dos vetores de orientação, dificultando a realização de operações matemáticas. Consequentemente, nas aplicações industriais, a matriz de orientação espacial da ferramenta de um robô deverá ser expressa de uma forma mais concisa, ou seja, através de três ângulos *RPY* (Roll, Pitch e Yaw) ou a partir de quatro parâmetros - quatérnios.

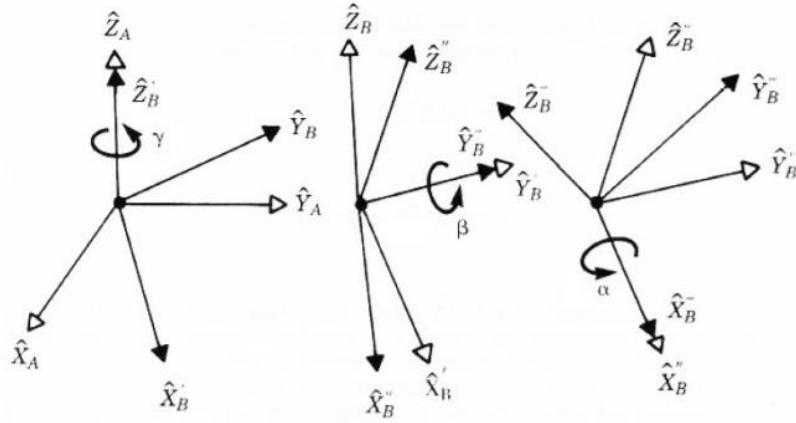


Figura 20 – Exemplo de rotações elementares, Roll, Pitch e Yaw [27].

Os ângulos Roll Pitch e Yaw (RPY) podem ser obtidos a partir de três rotações elementares γ , β , α em torno dos eixos Z, Y, X relativamente a configuração dos três últimos elos rotativos de um robô (Figura 20).

Outra forma bastante utilizada para a resolução do problema de orientação e cinemática inversa de robôs é a utilização de quatérniões. A partir da utilização da álgebra de quatérniões, a matriz de orientação da ferramenta de um manipulador (4), expressa em relação ao sistema de coordenadas da base, passa a ser expressa a partir de quatro parâmetros (q_1 , q_2 , q_3 e q_4).

$$\begin{aligned}
 q_1 &= \frac{1}{2} \sqrt{n_x + s_y + a_z + 1} \\
 q_2 &= \frac{1}{2} \sqrt{n_x - s_y - a_z + 1}, \text{ com sinal de } q_2 = (s_z - a_y) \\
 q_3 &= \frac{1}{2} \sqrt{s_x - n_y - a_z + 1}, \text{ com sinal de } q_3 = (a_z - n_y) \\
 q_4 &= \frac{1}{2} \sqrt{a_x - n_y - s_z + 1}, \text{ com sinal de } q_4 = (n_z - s_y)
 \end{aligned}
 \tag{5}$$

Ao mesmo tempo a orientação deverá ser normalizada, ou seja:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1
 \tag{6}$$

A principal vantagem desta representação é que a utilização de 4 parâmetros permitirá a obtenção de soluções únicas, implicando assim um número menor de manipulações computacionais.

O cálculo da cinemática inversa de um robô tem como objetivo, através da posição e da orientação da ferramenta em relação à base, determinar os ângulos das juntas que originam a posição e orientação da ferramenta. É um método mais complexo que o da cinemática direta, pelo que há vários fatores a ter em consideração, tais como o ponto ter de estar dentro do espaço de trabalho do robô, a existência de várias configurações do robô para atingir o mesmo ponto (Figura 21), a limitação da orientação para um determinado ponto no espaço de trabalho e singularidades, entre outros fatores. Resumindo, a solução pode não existir e, se existir, ela pode não ser única, dependendo da geometria do robô e da configuração dentro do seu espaço de trabalho. Por se tratar de um sistema de equações não-lineares, devemos preocupar-nos com o método de solução, com a existência da solução e com o número de soluções possíveis.

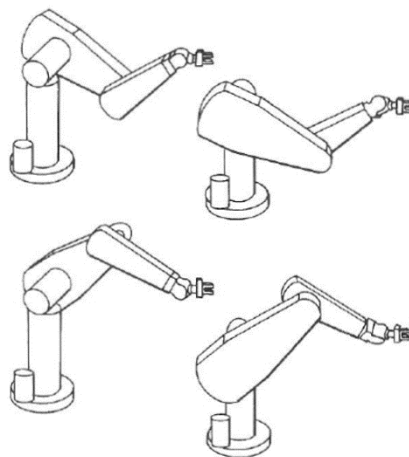


Figura 21 - Exemplo ilustrativo de um robô com quatro configurações diferentes para atingir o mesmo ponto no espaço de trabalho.

A necessidade da obtenção de referências em coordenadas e juntas, correspondentes a tarefas definidas no espaço cartesiano, é expressa matematicamente utilizando a cinemática inversa diferencial, isto é:

$$\theta = f^{-1}(x)$$

(7)

o que resulta no problema da inversão da matriz Jacobiana. Tal inversão não ocorre nos chamados pontos singulares onde o Jacobiano é nulo.

A função f é não linear e composta pela soma de produtos de senos e cossenos das coordenadas generalizadas (translações ou rotações elementares). Por isso, a sua inversão é, em geral, não trivial. Como f é não linear não se pode garantir a existência e/ou a unicidade de uma função inversa f^{-1} . No caso geral, só se pode determinar o número máximo de prováveis soluções. Os métodos de solução do problema da inversão do modelo geométrico são:

- Métodos analíticos, que conduzem à obtenção de todas as soluções. Estes métodos não são gerais, isto é, a inversão analítica não é trivial e, além disso, não há garantia de que seja possível fazê-la para um robô qualquer. Os métodos analíticos são adequados para robôs simples, isto é, aqueles que possuem um grande número de parâmetros de *Denavit-Hartenberg* nulos.
- Métodos numéricos iterativos, que convergem para uma solução possível entre todas as existentes. São de caráter geral e, com o atual desenvolvimento dos microcomputadores, a sua utilização em tempo real já é viável.

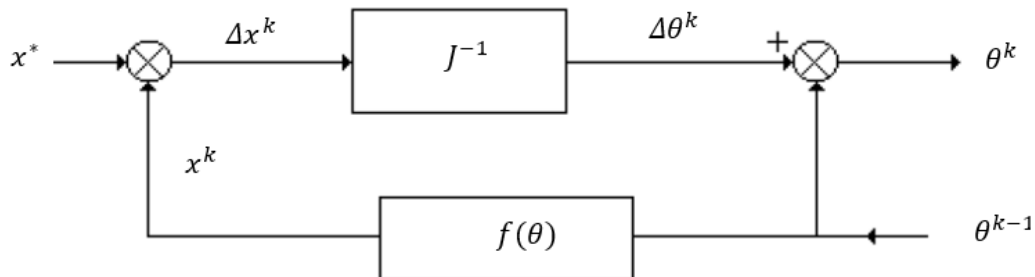


Figura 22 - Problema da cinemática inversa: determinar uma configuração θ correspondente a uma posição desejada x

Existem diversos métodos numéricos iterativos: um exemplo é o método recursivo, representado na Figura 22, que utiliza o cálculo do modelo geométrico direto e a matriz jacobiana inversa.

3.4 Tipos de linguagem de programação

Os robôs industriais são mecanismos automatizados que têm por finalidade efetuar uma tarefa previamente definida. Esses mecanismos necessitam de um controlador previamente programado que faz com que o robô consiga efetuar a tarefa com sucesso (Figura 23). O programa deve ser capaz de adaptar-se às variações de tarefas e também ser suficientemente flexível para permitir uma sequência dinâmica de operações.

A programação de um robô envolve a especificação de um conjunto de pontos no espaço que devem ser percorridos, sendo eventualmente definidas as trajetórias entre os pontos especificados assim como a definição da interação dos movimentos do robô com outros equipamentos, através da existência de capacidades sensoriais e da possibilidade de modificação da execução do programa.



Figura 23 - 1 Robô, 2 controlador, 3 teach pendant(consola) e 4 ligações.

Existem dois métodos de programação de um robô: *online* e *offline*.

O método *online* consiste na programação do controlador do robô utilizando um *software* de desenvolvimento através de um protocolo de comunicação, sendo o mais comum o Ethernet *IP*, podendo assim efetuar-se a programação diretamente no controlador. Para além de se poder programar o controlador através do método anteriormente descrito, existe sempre a opção de o poder programar utilizando a consola do próprio controlador. No caso de se criar um programa utilizando a consola, em que o robô tenha de executar uma trajetória, basicamente deve mover-se a ferramenta do robô até à posição alvo, recorrendo à consola de programação do robô, e gravar a posição. Este procedimento

deve-se efetuar para todas as posições alvo até que se consiga a trajetória pretendida. Obviamente, um programa para um robô não é só uma simples trajetória para efetuar uma determinada tarefa: o programa pode ser tão complexo quanto o exigir o tipo de aplicação. Este processo passa por várias etapas, tais como identificar os pontos que o robô deve alcançar, criar trajetórias favoráveis e fluidas entre pontos, criar todas as tarefas e rotinas que têm de ser efetuadas para as quais o robô foi idealizado e, por fim, testar todo o fluxo do programa e se necessário corrigir ou melhorar.

No método *offline*, a programação é desenvolvida sem a necessidade da presença do robô, recorrendo a um computador e um *software* de desenvolvimento específico. É possível efetuar toda a programação das tarefas do robô, ou até mesmo a configuração do mesmo, e posteriormente descarregar para o controlador.

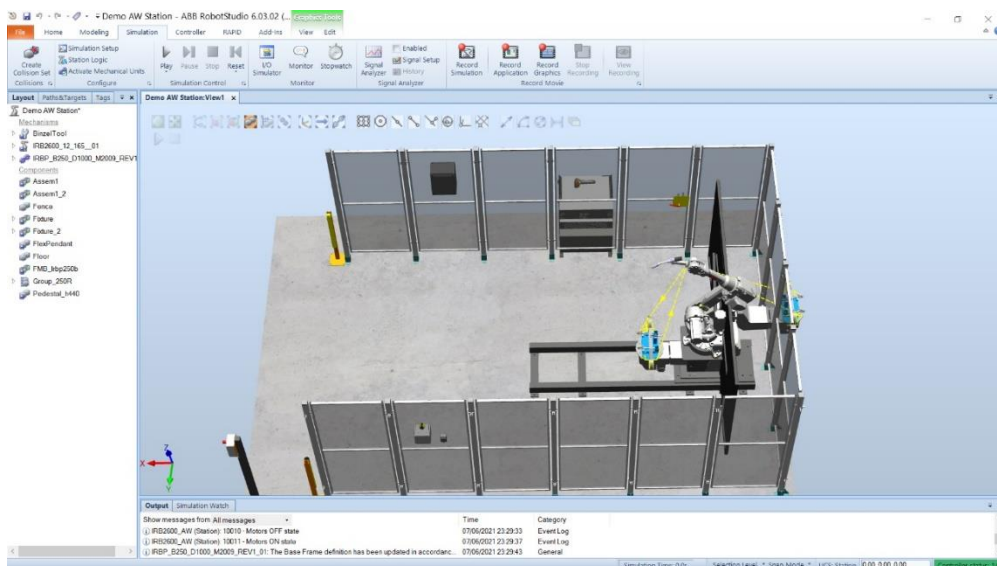


Figura 24 - Exemplo do simulador RobotStudio da ABB

A maioria dos fabricantes de robôs criou o seu próprio *software* de desenvolvimento e simulação para configuração e programação de robôs, sendo uma mais-valia para a fase do projeto em que é possível desenvolver a aplicação sem a necessidade de adquirir o robô (Figura 24).

Com este tipo de *software*, não só é possível desenvolver o programa e testar como também é possível a importação de todo o *layout* envolvente ao robô, sendo fácil a perceção de todos os pontos que o robô terá de atingir, trajetórias que terá de efetuar e, se necessário, alterar o *layout* de modo a melhorar todo o fluxo de trajetórias para que o robô realize as tarefas com sucesso. Com este tipo de método é possível programar mesmo sem

ter o robô, o que dá uma enorme vantagem: por exemplo, no caso da aquisição de um robô novo, a sua entrega pode demorar semanas e, enquanto se espera pela chegada do mesmo, é possível desenvolver a aplicação e se necessário melhorar o *layout*; uma outra vantagem verifica-se quando há a necessidade de efetuar alterações num robô que esteja em funcionamento – basta utilizar um *backup* do robô que está em funcionamento, introduzir no simulador, efetuar as alterações necessárias, testar e depois descarregar no controlador real. Desta forma, não é necessária a paragem do robô por longos períodos de tempo, evitando grandes perdas de produção.

Cada fabricante tem a sua própria linguagem de programação, sendo que algumas são muito semelhantes a linguagens já conhecidas, tais como C, Pascal ou *assembly*.

Tabela 1 - Alguns exemplos de fabricantes e suas linguagens de programação

Fabricante	Linguagem
ABB	RAPID
KUKA	KRL
MOTOMAN	INFORM
FANUC	KAREL, RG
MITSUBISHI	MRL
UNIMATION, ADEPT	VAL, VALII
IBM	AML
BOSH	BAPS

De um modo geral, todas as linguagens são semelhantes em alguns aspetos principais, tais como:

- declaração de variáveis;
- tipos de movimentos (lineares, ponto a ponto ou circulares);
- teste e condições;
- ciclos;
- funções e procedimentos, entre outros.

4. Desenvolvimento do Estágio

Este estágio envolveu vários projetos, todos eles centrados na programação de robôs antropomórficos, com o objetivo de realizarem determinadas tarefas. Estes robôs estão inseridos em ilhas, em conjunto com mais equipamentos e devidamente vedadas para o efeito. Os trabalhos, tais como pesquisa, realização do relatório, desenvolvimento de programação dos robôs e comissionamento, foram realizados de forma alternada, devido aos tempos de desenvolvimento, construção de equipamentos e disponibilidades dos clientes.

Este estágio teve uma duração de 1458 horas, em que 226 foram destinadas a pesquisa, 312 para elaborar o relatório e as restantes 920 foram dedicadas aos projetos desenvolvidos.

4.1 Programação de um paletizador de caixas

Nos dias de hoje, muitas das empresas que criam os seus produtos têm, de alguma forma, que armazená-los em grandes quantidades, de forma organizada e standard, para posteriormente serem transportados até ao seu destino, quer se trate de um distribuidor ou de um consumidor final. Uma dessas formas é paletizar o produto, ou seja, dispô-lo organizado geometricamente de forma a encaixar dentro das dimensões de uma palete de madeira, plástico ou cartão, para depois ser transportado em grandes quantidades. Sendo um processo repetitivo, exaustivo e por vezes pesado para um operador, a maioria das empresas está cada vez mais a automatizar os processos de paletização, utilizando robôs para o efeito, diminuindo assim o esforço humano, otimizando tempo e evitando gastos desnecessários.

4.1.1 Objetivo

Foi proposto o desenvolvimento de um projeto que consiste em programar um robô antropomórfico de 6 graus de liberdade, inserido numa ilha robotizada. Este robô tem como principal objetivo paletizar caixas numa palete de cartão. Para além desta tarefa,

existem outras sub-tarefas que complementam todo o processo de paletização, tais como a colocação da paleta de cartão, a colocação de uma etiqueta em cada caixa e a colocação de um cartão entre níveis consecutivos da paletização.

Para que o robô consiga realizar o objetivo pretendido, vai ser equipado com uma ferramenta com duas valências distintas: a primeira, e principal, é uma garra para pegar nas caixas; a segunda é composta por um sistema de ventosas para pegar nas paletes de cartão vazias e nos cartões.

O robô está inserido numa ilha controlada por um *PLC* que, por sua vez, troca informação com o robô a fim de realizar o objetivo do projeto. A ilha é composta pelo robô, por uma mesa alimentadora de caixas, por várias mesas de transporte de paletes, por uma região para colocação de uma pilha de paletes de cartão, a qual passará a chamar-se armazém de paletes, por uma região para colocação de cartões, a qual passará a chamar-se armazém de cartões, e por um dispensador de etiquetas.

A mesa alimentadora de caixas, como o próprio nome indica, tem como função alimentar a ilha com as caixas para paletizar.

As mesas de transporte têm como finalidade efetuar o transporte da paleta finalizada para fora da ilha. É diretamente numa dessas mesas que o robô vai realizar a paletização.

O armazém de paletes é um local delimitado com uma capacidade de armazenar aproximadamente 14 paletes de cartão empilhadas. Sempre que o robô inicia uma paletização, é a este armazém que vai retirar paletes vazias.

O armazém de cartões é uma estrutura com uma capacidade de armazenar aproximadamente 30 cartões. Dependendo do tipo de formação da paletização, pode ser necessária a colocação de um cartão sempre que o robô preenche completamente um nível de caixas na paleta.

Por último, o alimentador de etiquetas (vinhetas) tem como função dispensar uma etiqueta sempre que o robô lhe apresente uma caixa, colocando então a etiqueta na caixa.

Duma forma mais detalhada, o robô tem que executar as seguintes tarefas:

1. pegar numa paleta de cartão vazia e colocar no tapete de paletização;
2. pegar numa caixa existente na mesa de entrega;
3. passar a caixa no dispensador de etiquetas para colocação da etiqueta;
4. colocar a caixa em cima da paleta até completar o nível (repete-se a sequência das tarefas 2 a 4 até o nível estar completo);

5. pegar num cartão e colocar em cima de cada nível completo da paleta (repete-se a sequência 2 a 5 até a paleta estar completa);
6. após terminada a paleta, é transferida para a próxima fase da linha através do tapete da mesa, ficando o robô pronto para iniciar uma nova paletização.

Na Figura 25, podemos ver representado um fluxograma geral, e muito resumido, das tarefas que o robô realiza para atingir o seu objetivo.

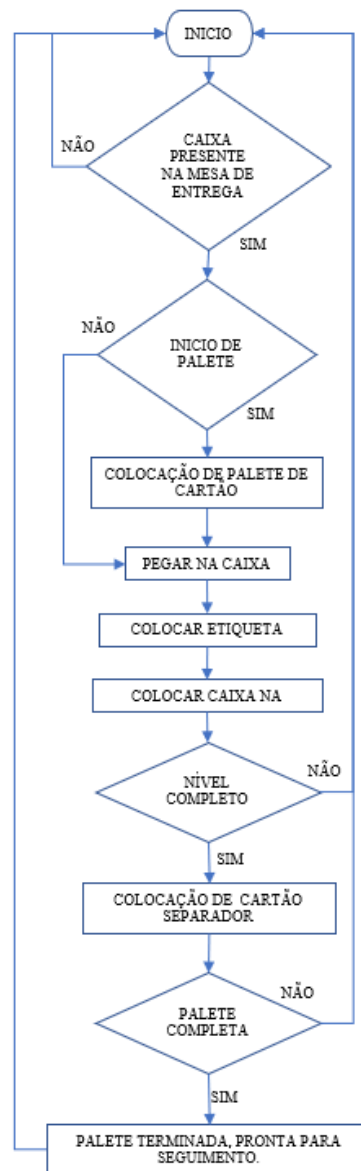


Figura 25 - Fluxograma geral de funcionamento.

Este projeto foi elaborado em várias etapas: desenvolvimento do projeto no simulador *RobotStudio* do fabricante *ABB*, no qual é inserido o *layout* e a ferramenta em 3D;

parametrização do robô, assim como dos sinais a serem trocados com o *PLC* que comanda a ilha; declaração e parametrização da ferramenta a utilizar pelo robô; declaração e parametrização das bases de trabalho; desenvolvimento de todo o código em linguagem *rapid*; criação de receitas de produto; efetuar testes no simulador; efetuar testes com o robô real nas instalações da EXSEPI; efetuar, por fim, o comissionamento do robô no cliente.

4.1.2 Ferramenta

A ferramenta é um equipamento montado na extremidade do robô e foi desenvolvida para executar duas funções distintas: a primeira consiste em pegar numa caixa e a segunda em pegar numa paleta de cartão ou num cartão. Em ambas as funções, a ferramenta deve conseguir transportar os objetos anteriormente referidos, sem que os mesmos se danifiquem ou se soltem.

Para pegar na caixa é utilizado um sistema do tipo garra, que é constituído por uma parte fixa, ou estrutura da ferramenta, e uma parte móvel, identificadas na Figura 26. A parte móvel é constituída por uma junta prismática, constituída por um mecanismo que desliza em guias lineares, e utiliza, como fonte de energia, um cilindro pneumático em série para efetuar esse movimento linear. O cilindro pneumático exerce a força necessária para pegar na caixa e transportá-la em movimento sem que se solte.

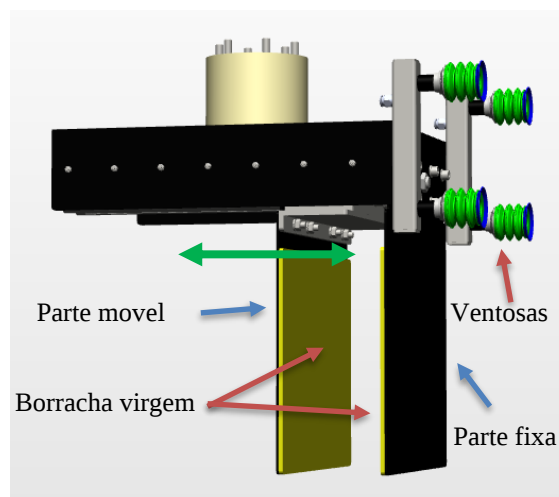


Figura 26 - Ferramenta para paletizador de caixas

Uma vez que a ferramenta é construída em metal e a caixa a transportar é de cartão e com o objetivo de aumentar a força de atrito entre o metal e o cartão, as faces da ferramenta

que estão em contato com a caixa têm colada borracha virgem. Com esta solução, ao aumentar a força de atrito entre o metal e o cartão (caixa) pode-se diminuir a força exercida pelo cilindro pneumático na parte móvel da ferramenta. Sendo assim, consegue-se reduzir o risco de se danificar a caixa por esmagamento e diminuir o risco de a caixa se soltar da ferramenta quando o robô estiver a efetuar movimentos.

Para além da garra, a ferramenta está ainda equipada com quatro ventosas, como se pode verificar na Figura 26, instaladas na sua estrutura. É com estas ventosas que a ferramenta vai pegar numa palete ou num cartão, utilizando um sistema de geração de vácuo do tipo *venturi* [8].

Após o departamento de desenvolvimento definir a ferramenta, há que inseri-la no simulador e acoplar à extremidade do robô, com a finalidade de conseguir visualizar e identificar o espaço necessário para a movimentar em segurança assim como detetar possíveis singularidades que possam vir a existir.

No *RobotStudio* foi adicionado o modelo da ferramenta em 3D sendo necessário efetuar algum trabalho de modelação, representado na Figura 27, tendo sido separadas as partes fixas das partes móveis e criada a junta linear para que fosse possível obter um modelo da ferramenta com o qual se consiga simular a abertura e o fecho da pinça.

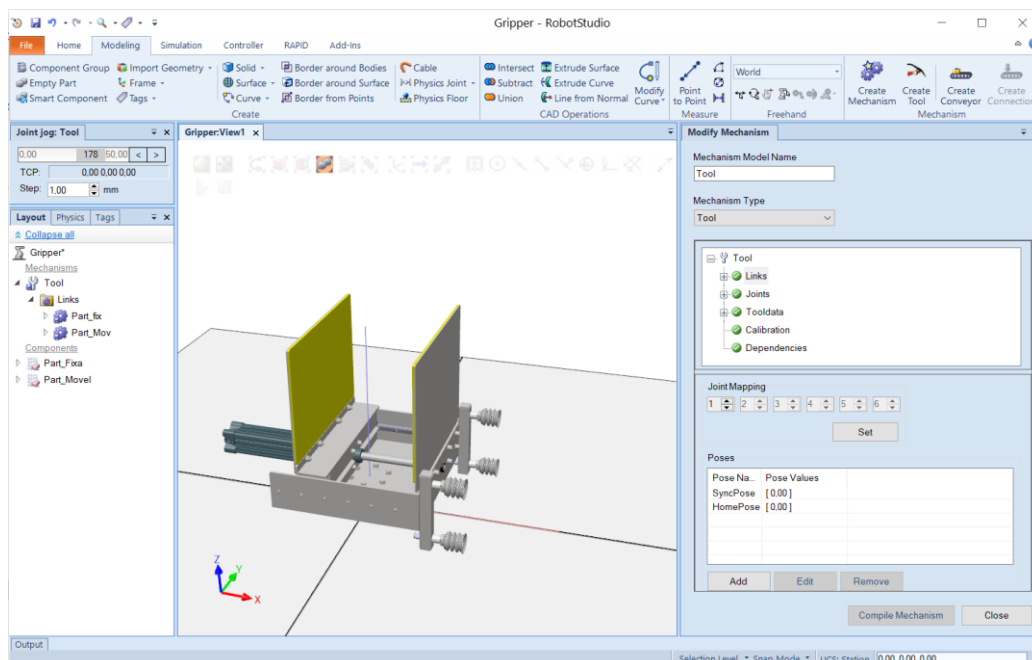


Figura 27 - Modelo da ferramenta em 3D

Foi corrigido o ponto de origem do modelo da ferramenta, pois este não se encontrava na posição e no ângulo desejado. Este ponto é utilizado como referência para efetuar a

ligação entre a ferramenta e o robô (Figura 28). É ele que determina a posição e a rotação em relação ao elo terminal do robô. Por uma questão de facilitar e diminuir os movimentos que o robô tem de efetuar, posiciona-se a ferramenta em relação ao terminal do eixo 6 do robô, de forma que seja a posição mais natural e comum entre posições alvo. Também é necessário ter em conta os ângulos de rotação das juntas do robô e a colocação da ferramenta numa posição de repouso. Os eixos do robô devem estar o mais próximos possível de zero, dando assim liberdade para se poder movimentar a ferramenta em qualquer eixo.

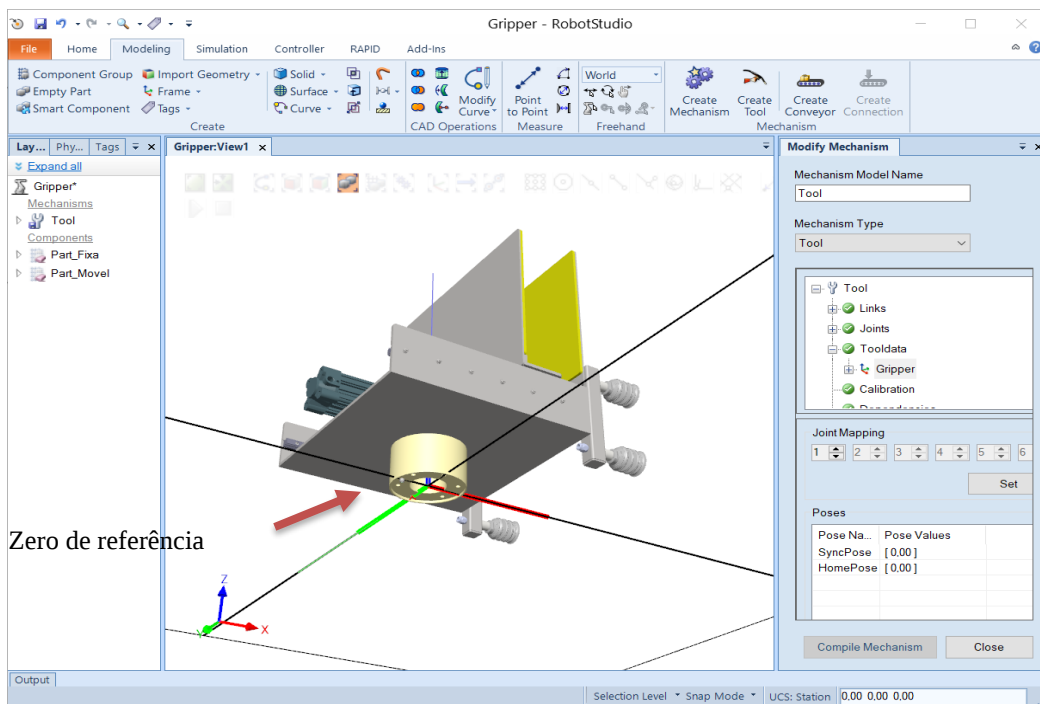


Figura 28 - Coordenadas de referência do zero do modelo 3D da ferramenta

Acoplou-se a ferramenta ao terminal do robô e criou-se os “TCPs”, que neste caso vão ser dois: um para o sistema de garra e outro para o sistema das ventosas. É a partir de um destes “TCP’s” e dos pontos de origem e de destino, que o controlador do robô efetua o cálculo da cinemática inversa, no caso de um movimento ponto a ponto, ou da cinemática direta, num movimento de juntas, para calcular as trajetórias para alcançar o ponto de destino, como referido no capítulo 3.3.

Na Figura 29, estão representados os dois “TCP’s”. Cada uma das cores corresponde a um eixo no sistema de coordenadas cartesianas: o vermelho corresponde ao eixo x, o verde ao eixo y e o azul ao eixo z.

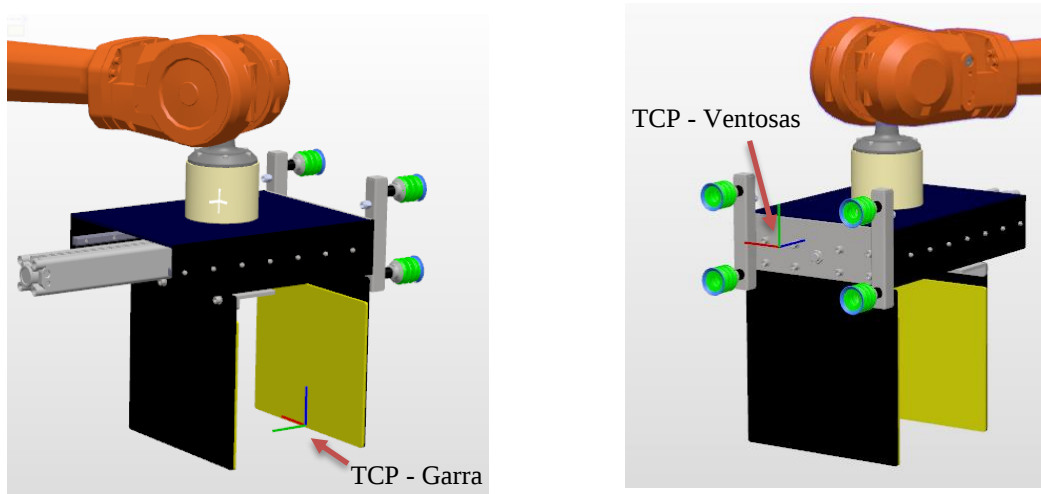


Figura 29 - Identificação dos TCP's: à esquerda a garra e à direita as ventosas

Os “TCP's” têm de ser definidos de tal forma que sejam de fácil identificação e que sejam acessíveis para que, por sua vez, seja fácil a calibração da ferramenta e a criação de posições alvo seja intuitiva.

Regra geral, o ‘TCP’ é criado em relação a uma parte fixa da ferramenta e, sendo um ponto de referência, tem de ser fácil de identificar, o que não aconteceria se fosse referenciado a uma parte móvel da ferramenta.

4.1.3 Layout

A criação do *layout*, é das tarefas iniciais mais importantes: é através dele que é definida toda a disposição dos equipamentos existentes na ilha robotizada. Uma má escolha da colocação dos equipamentos e do restante envolvente pode inviabilizar o bom funcionamento da ilha para o objetivo proposto. Todo o equipamento existente na ilha tem de estar em correta disposição, de forma a não obstruir as trajetórias do robô ou a ficar no alcance do mesmo, nomeadamente a zona de paletização, a mesa de entrega de caixas, o armazém para paletes e cartão e o dispensador de etiquetas.

A ilha robotizada tem como principais equipamentos, um robô da ABB da família IRB4600, uma mesa alimentadora de caixas, duas mesas com tapete, numa das quais se vai efetuar a paletização, um armazém para paletes de cartão, um armazém para cartões e um dispensador de etiquetas.

No *layout* inicial (Figura 30) estão representados quase todos os equipamentos e objetos, faltando aqui a mesa de entrega de caixas e a impressora de etiquetas. A falta destes equipamentos deve-se ao facto de, nesta fase, o restante da linha ainda estar em desenvolvimento. O que se sabia, era que a mesa de entrega de caixas iria estar à esquerda do robô, a uma altura de aproximadamente 800mm, ficando por definir a sua posição em x, y em relação ao robô. Já a etiquetadora, seria colocada na melhor posição possível em relação ao robô, de forma que o mesmo conseguisse passar com uma caixa pelo dispensador de etiquetas, com o objetivo de a etiqueta ficar colada num dos lados.

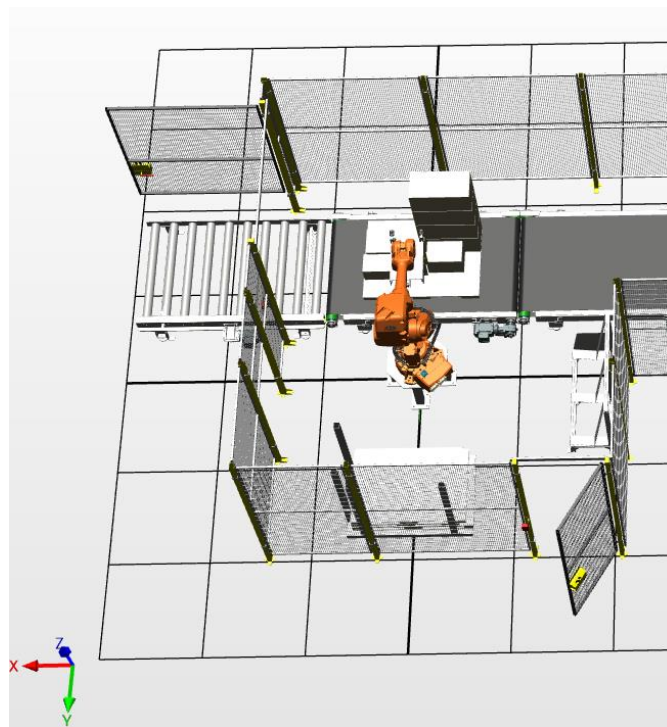


Figura 30 - Layout inicial da célula robotizada.

Todo este processo de correção dos equipamentos e objetos na ilha robotizada foi efetuado de forma interativa e sem recurso a cálculos exaustivos. O simulador *RobotStudio* permite que consigamos verificar todas as trajetórias, posicionamento e orientação que o robô terá de efetuar sem recorrer a cálculos, sendo um pouco uma tarefa de tentativa erro, o que será sempre um pouco mais rápido do que ter de calcular matrizes de translação e rotação para cada um dos pontos ou trajetória do robô.

O processo começa por inserir um modelo, em “*CAD 3D*”, da implantação da ilha robotizada no simulador *RobotStudio*, inicialmente criado pela equipa de projeto e

desenvolvimento existente na EXSEPI. Pretende-se, através do simulador, conseguir visualizar e verificar as futuras trajetórias do robô, com o intuito de efetuar possíveis correções na disposição quer do robô quer dos equipamentos envolventes na ilha.

Este processo tem como objetivo verificar que todas as posições alvo do robô estejam dentro do seu alcance e orientação, podendo assim realizar as tarefas necessárias para efetuar a paletização sem qualquer tipo de limitações.

Como não existia o modelo da mesa alimentadora da ilha, recorreu-se aos modelos de mesas já existentes no *RobotStudio* e foi colocada uma mesa na posição que era expectável. Para a etiquetadora, como não existia nenhum modelo disponível, foi criado um objeto, meramente representativo, recorrendo a formas geométricas básicas, existentes na secção de modelagem do *RobotStudio*.

Obviamente, este robô foi projetado e programado para que futuramente pudesse paletizar caixas de outras dimensões. Neste projeto foi definida a necessidade de operar com 6 formatos de caixas distintos.

Tabela 2 - Dados correspondentes a cada paletização.

Produto	Caixas por nível	Nº de níveis	Caixas por paleta	Dimensões das caixas	Dimensões da paleta completa	
Produto A	4	5	20	560x375x170	1140x780x950	
Produto B	4	2	12	595x295x275	1200x630x940	
Produto C	4	5	20	560x375x170	1140x780x950	
Produto D	6	4	24	345x345x230	1080x730x990	
Produto E	8	5	40	520x200x170	1070x830x950	
Produto F	10	6	60	575x152x140	1180x800x920	

O passo seguinte foi a criação das possíveis paletizações em desenho 3D. Foram facultados os dados técnicos das dimensões das caixas, das paletes de cartão e dos cartões e o *layout* para as 6 formações distintas, com a disposição das caixas em relação à paleta, o número de caixas por nível e o número de níveis. Com estes dados, apresentados na Tabela 2, foram criadas, na secção de modelagem do *RobotStudio*, as 6 paletizações e inseridas no *layout* da célula.

Com todos os equipamentos e objetos definidos e a inserção da ferramenta no robô, como se representa na Figura 31, é possível efetuar a verificação e a correção da posição do robô ou dos equipamentos em relação ao meio envolvente, de forma a poder efetuar todas as trajetórias necessárias, tais como pegar nas paletes de cartão, pegar no cartão, pegar na

caixa, passar com as caixas na etiquetadora e por fim conseguir colocar a caixa na paleta, sem limitações de posição e orientação.

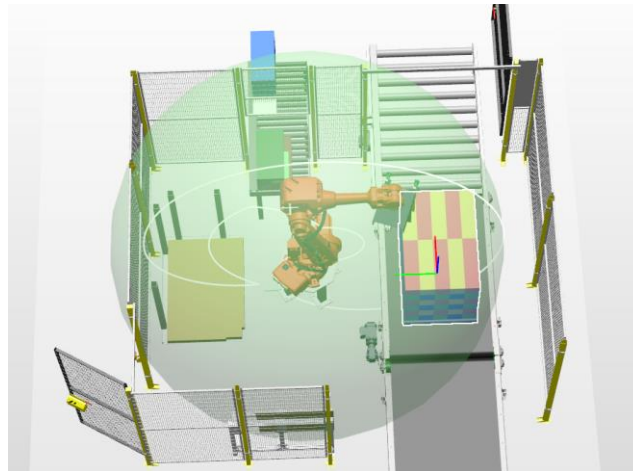


Figura 31 - Volume de trabalho do robô

Após uma primeira análise, foi possível perceber que existiam alguns constrangimentos com o *layout* inicial e foi necessário reposicionar o robô em relação ao meio envolvente. A mesa de alimentação não estava suficientemente dentro da ilha, de modo que a ferramenta conseguisse pegar numa caixa sem colidir com a vedação da ilha. Para corrigir este problema, a mesa foi deslocada em direção ao robô, criando espaço suficiente para que a ferramenta consiga pegar numa caixa sem colidir com a vedação, ver Figura 32.

O principal objetivo desta aplicação é efetuar uma paletização mediante as características exigidas para um determinado produto, características essas que podem ser a dimensão da caixa a paletizar, a dimensão da paleta de cartão e do cartão, a disposição das caixas na paleta, o número de caixas por nível e o número de níveis. Verificando algumas destas características, facilmente podemos deduzir que o robô não pode ter limitações na colocação e orientação das paletes, cartões e caixas na paleta. Através da Tabela 2, verifica-se que as dimensões máximas das paletizações podem ir até 1200mm de comprimento, 830mm de largura e 990mm de altura. O robô tem de conseguir efetuar a colocação de caixas dentro desse espaço de trabalho, de forma que consiga efetuar movimentos de aproximação e rotação ao ponto final de entrega da caixa na paleta. Para isso, a região de paletização tem de estar ao alcance do robô de forma que consiga efetuar as trajetórias para deixar a caixa na paleta sem que a ferramenta vá colidir com os demais equipamentos ou objetos dentro da célula robotizada.

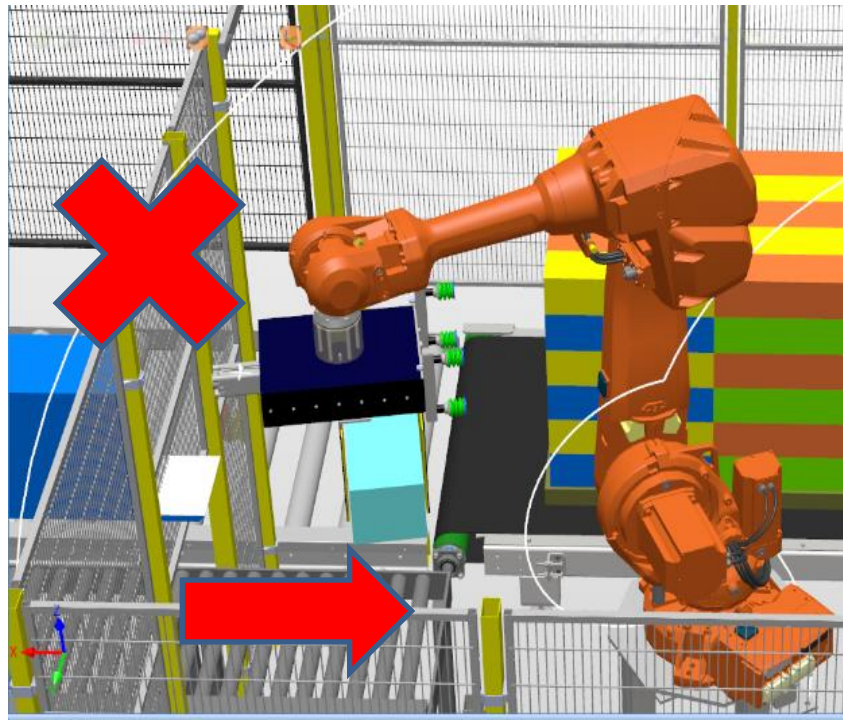


Figura 32 - Robô na área da mesa de alimentação, já corrigida.

Após a análise anteriormente descrita afastou-se o robô da mesa de paletização (Figura 33), para que fosse possível colocar a caixa em vários ângulos na paletização e não colidisse com o robô após a colocação das caixas mais próximas do mesmo.

Um dos equipamentos importantes existente dentro da célula é o dispensador de etiquetas, que é responsável pela colocação de etiquetas nas caixas a paletizar. Uma das tarefas do robô é, após recolher a caixa da mesa, passar com a caixa pelo dispensador para a colocação de etiqueta na mesma. Pensando na trajetória anteriormente descrita, o dispensador foi posicionado estrategicamente acima da mesa de entrega de caixas, para estar mais próximo da trajetória efetuada pelo robô após a recolha da caixa, evitando assim um aumento do tempo de ciclo na paletização de cada caixa que, por consequência, aumenta o tempo de ciclo total da paletização, caso o dispensador estivesse noutra localização dentro da ilha.

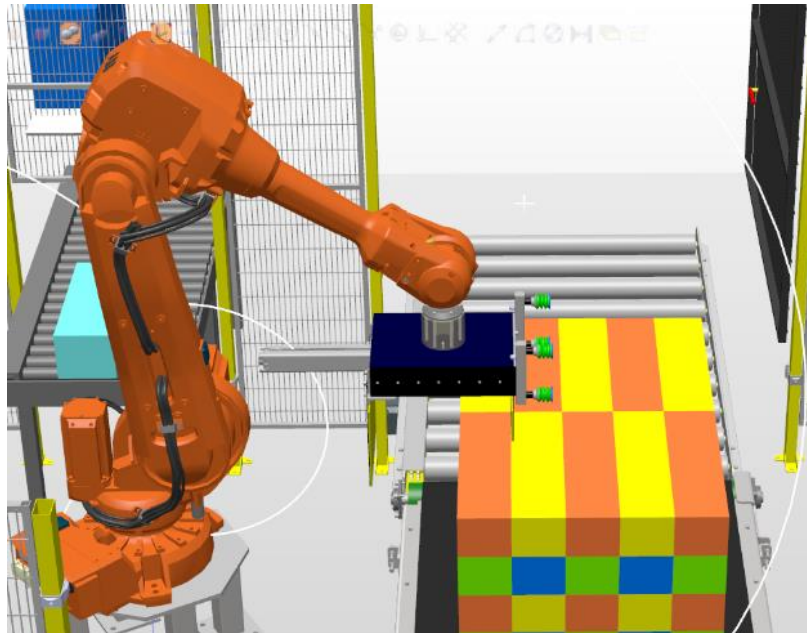


Figura 33 - Correção da posição do robô em relação à mesa de paletização.

Por fim, as posições do armazém de paletes de cartão e do armazém de cartões também foram individualmente verificadas e corrigidas, para que estivessem dentro do alcance do robô e para que este se consiga posicionar e orientar sem qualquer limitação.

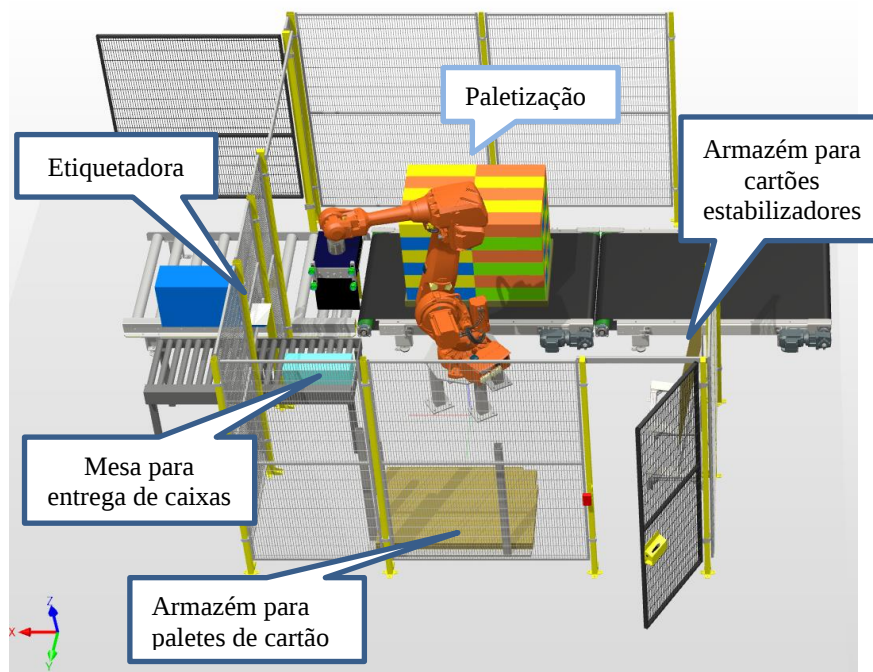


Figura 34 - Layout da célula robotizada, com todos os equipamentos.

Após a verificação da posição de todos os equipamentos e objetos dentro da célula, chegámos ao resultado final do *layout*, representado na Figura 34. É a partir deste *layout* final que se vai começar a criar todas as trajetórias e bases de trabalho, para que o robô efetue todas as trajetórias a fim de alcançar os objetivos pretendidos, sem limitações de movimentação, com os demais equipamentos e objetos dentro da ilha robotizada.

4.1.4 Bases de trabalho

Uma base de trabalho, é um sistema de coordenadas com base nas coordenadas do referencial do mundo do robô. Ao criar uma base de trabalho, todos os pontos a ela associados, são relativos à mesma, ou seja, se for efetuada uma deslocação da base, todos os pontos a ela associados se deslocam no mesmo sentido e orientação. Por norma, as bases são definidas em pontos de referência a equipamentos ou mesas em que o robô tenha de efetuar tarefas.

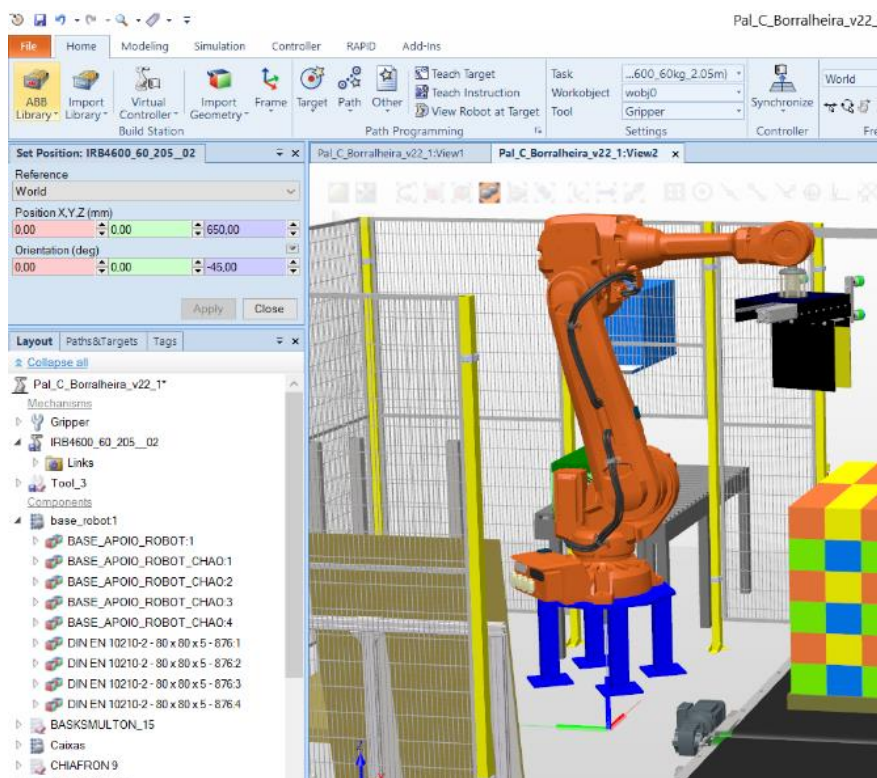


Figura 35 - Posicionamento do robô em relação à sua coordenada mundo.

Neste projeto utiliza-se um robô IRB4600, do fabricante ABB, em que a base de trabalho mundo é definida como 'wobj0'. O robô, para que tenha um melhor aproveitamento do

espaço de trabalho, foi colocado em cima de um pedestal, elevando-o assim a uma altura de 650mm (Figura 35). Desta forma, ao parametrizar a altura do robô em relação ao solo, essa translação será tida em conta nos cálculos das trajetórias e posições alvo.

Nesta aplicação para paletização de caixas foram criadas 4 bases de trabalho: uma para a paletização, à qual se deu o nome ‘Paletização’; uma para a mesa de alimentação de caixas, a qual se nomeou ‘Mesa_Entrada’; uma para o armazém de paletes de cartão, nomeada ‘Paleta_Cartão’; uma para o armazém de cartões, com o nome ‘Cartão_Intercalar’.

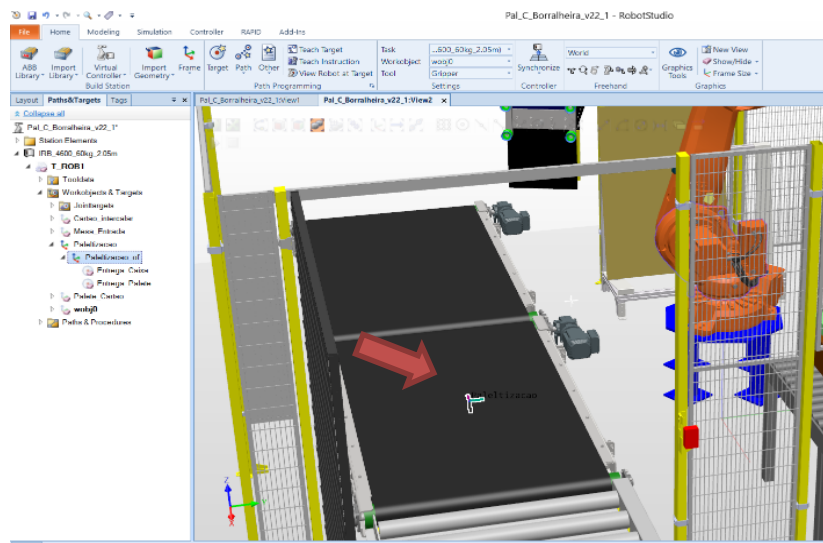


Figura 36 - Localização da base de trabalho para paletização.

A base da paletização foi colocada no centro do tapete da linha de transporte da paleta (Figura 36). Este local foi escolhido estrategicamente pelo facto de este tapete dar seguimento à paleta finalizada e por estar ao centro do tapete transportador. Um dos exemplos da simplicidade de operações a efetuar nesta base de trabalho, é a colocação da paleta de cartão centrada na paletização: basta pegar na paleta pelo centro e deixá-la no ponto (0,0) das coordenadas xy em relação à base de paletização. Uma vez que existem paletes de vários tamanhos, desta forma só é necessário efetuar cálculos para pegar na paleta de cartão ao centro, evitando efetuar cálculos de desvios, no eixo xy, para se poder centrar a colocação da paleta na base de paletização. Associados a esta base, estão os pontos de entrega de paletes, cartão e caixas.

A base de trabalho da mesa de alimentação de caixas foi criada em relação ao centro da mesa, junto à face mais próxima do robô (Figura 37). Como as caixas vêm centradas na mesa, esta base foi criada estrategicamente para ter um ponto comum, qualquer que seja a largura da caixa. Desta forma, para a ferramenta do robô poder pegar na caixa, basta efetuar um deslocamento de metade da sua largura para que fique centrada com a caixa.

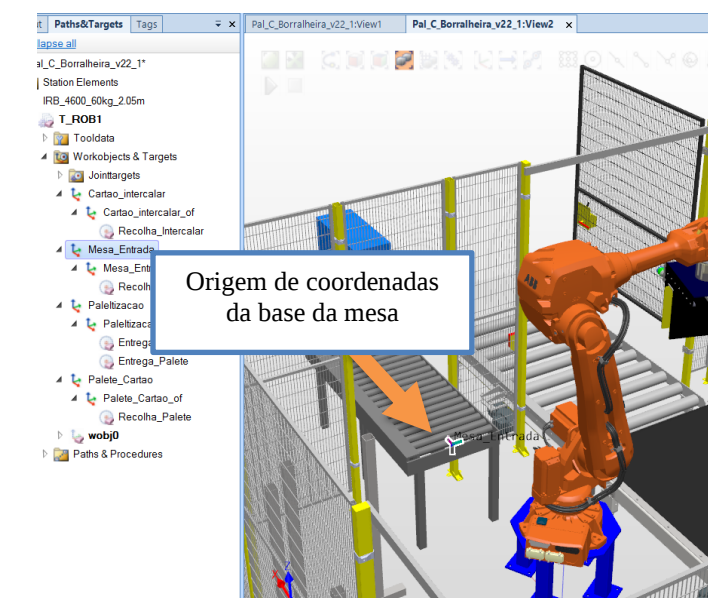


Figura 37 - Localização da base de trabalho da mesa de alimentação.

Associadas a esta base estão as posições de aproximação e recolha das caixas.

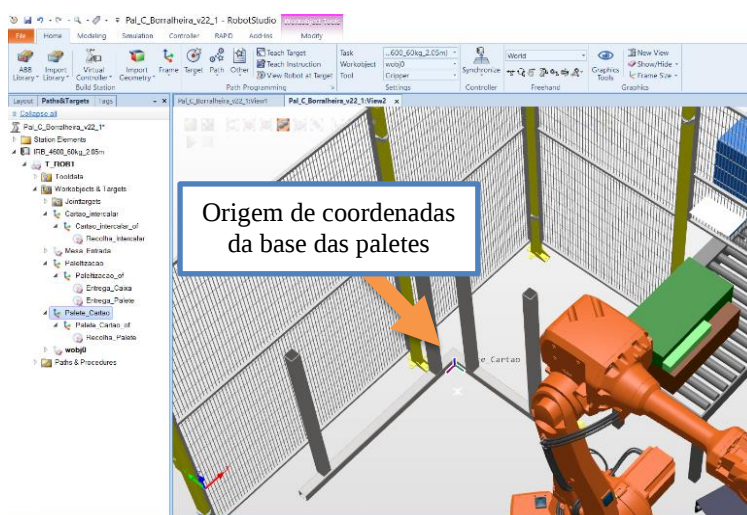


Figura 38 - Localização da base de trabalho do armazém de paletes

A base do armazém de paletes foi definida junto ao suporte de apoio das paletes de cartão (Figura 38). Este suporte serve de referência para mais facilmente o operador colocar as

paletes de cartão na posição certa em relação à base de trabalho. Desta forma a base de trabalho fica sempre referenciada a um dos cantos da paleta de cartão. Sendo assim, para definir a posição alvo de recolha no centro da paleta, basta dividir o comprimento e a largura da paleta por dois. Associados a esta base estão os pontos de aproximação e recolha de paletes de cartão.

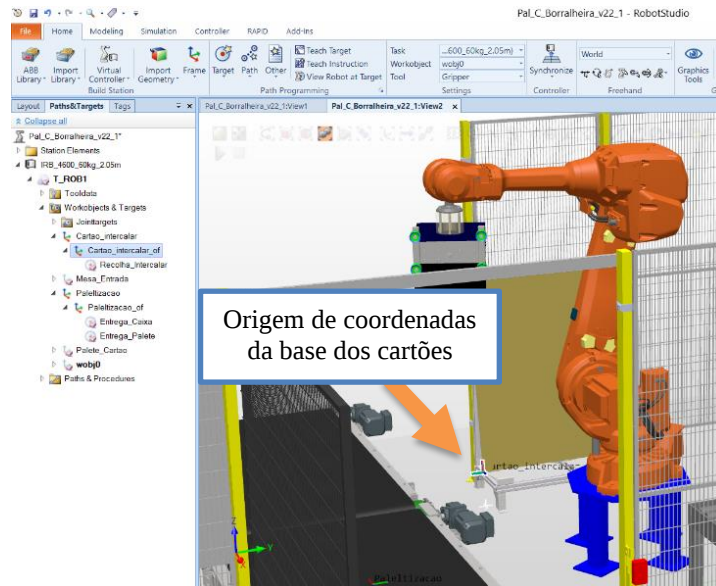


Figura 39 - Localização da base de trabalho do armazém de cartões

Por fim, vem a base dos cartões estabilizadores. Esta base foi definida no canto inferior esquerdo do suporte para a colocação dos cartões. Com a mesma coerência com que foi criada a base do armazém de paletes, esta base também fica num dos cantos do cartão, ver Figura 39, facilitando os cálculos para determinar a posição alvo do centro de cada cartão. Associados a esta base estão os pontos de aproximação e recolha dos cartões.

4.1.5 Posições alvo e trajetórias

Nesta secção são explicadas, de uma forma simples, as principais posições e trajetórias definidas para o robô realizar as tarefas, sendo que as restantes posições não serão abordadas devido ao facto, mesmo não sendo dispensáveis, de não serem da mesma relevância para este relatório.

4.1.5.1 Posição home

Começamos pela posição mais importante, a *home* (casa, ou posição inicial). A posição *home*, assim designada no programa, é onde inicia e termina qualquer tarefa que o robô tenha de executar (Figura 40). Esta posição no espaço cartesiano, deve ser sempre definida como posição de eixos do robô. Com este tipo de definição é garantido que, sempre que o robô se posiciona na *home*, a sua configuração de eixos é sempre a mesma, o que poderia não acontecer se estivesse definida como posição cartesiana (x,y,z) e a verificação da configuração do robô estivesse desativada (parâmetros confJ ou confL).

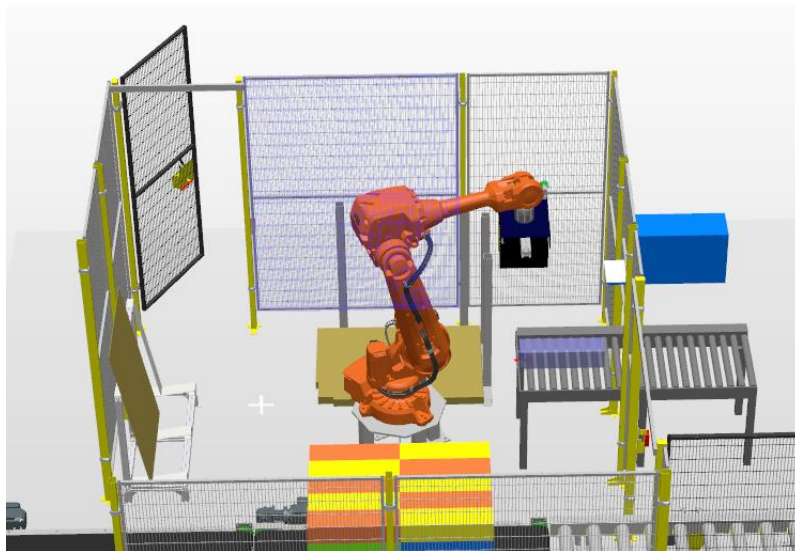


Figura 40 - Robô na posição home.

Neste caso, o robô poderia não ficar na configuração correta e, quando estivesse a iniciar o primeiro movimento de uma nova tarefa, poderia entrar em erro de configuração ao iniciar o movimento ou, ao efetuar o movimento fazendo a correção da configuração, poderia resultar numa colisão com os objetos mais próximos.

4.1.5.2 Posição de recolha de caixa

A posição alvo de recolha da caixa é uma das posições mais importantes e é através dela que se define a forma como a ferramenta do robô pode pegar na caixa. Para fazer a formação de uma paletização, nem sempre as caixas são colocadas na palete com a mesma

orientação, o que por vezes pode resultar num problema físico quando se entrega a caixa na paleta, mas, com esta capacidade de se poder alterar a orientação de pegar na caixa, facilita o processo de paletização.

As caixas, antes de serem entregues na ilha, são submetidas a um processo de construção, enchimento e selagem, ao qual não se dará ênfase pelo facto de não ser o foco deste relatório. Nesse processo, as caixas são transportadas por mesas com sistemas mecânicos desenvolvidos para tal. A mesa alimentadora da ilha é a última dessas mesas transportadoras, sendo que as caixas são deslocadas de mesa para mesa, sempre ao centro, como exemplificado na Figura 41.

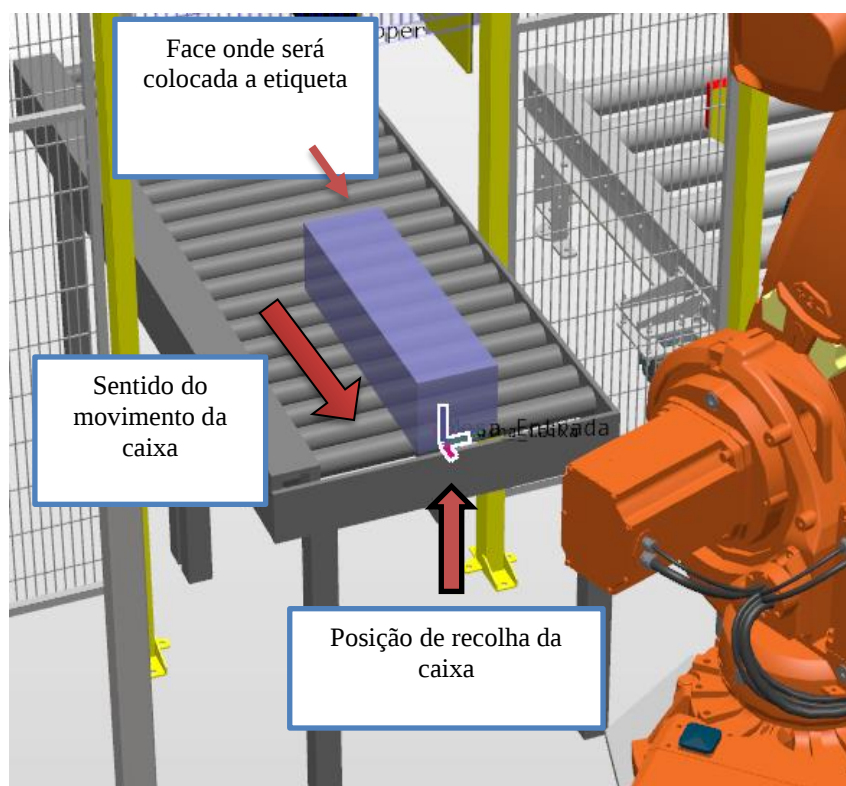


Figura 41 - Exemplo de entrega da caixa na ilha e posição de recolha da caixa.

A localização e orientação da recolha foi escolhida mediante os seguintes critérios: a posição tem de ser comum a todas as caixas dos produtos definidos; quando a ferramenta do robô pegar numa caixa, não pode interferir com objetos ou equipamentos; a orientação tem de ser o mais concordante possível com a posição de entrega da caixa na paleta, evitando rotações desnecessárias da ferramenta. Esta posição foi nomeada “Recolha_caixa”.

4.1.5.3 Posição de entrega de caixa

A posição alvo de entrega da caixa tem como objetivo posicionar a caixa na paleta, mediante a posição e orientação. Esta posição está inicialmente localizada no centro da mesa de paletização, na posição (0,0,0) no sistema de coordenadas “xyz” da base de paletização (Figura 42).

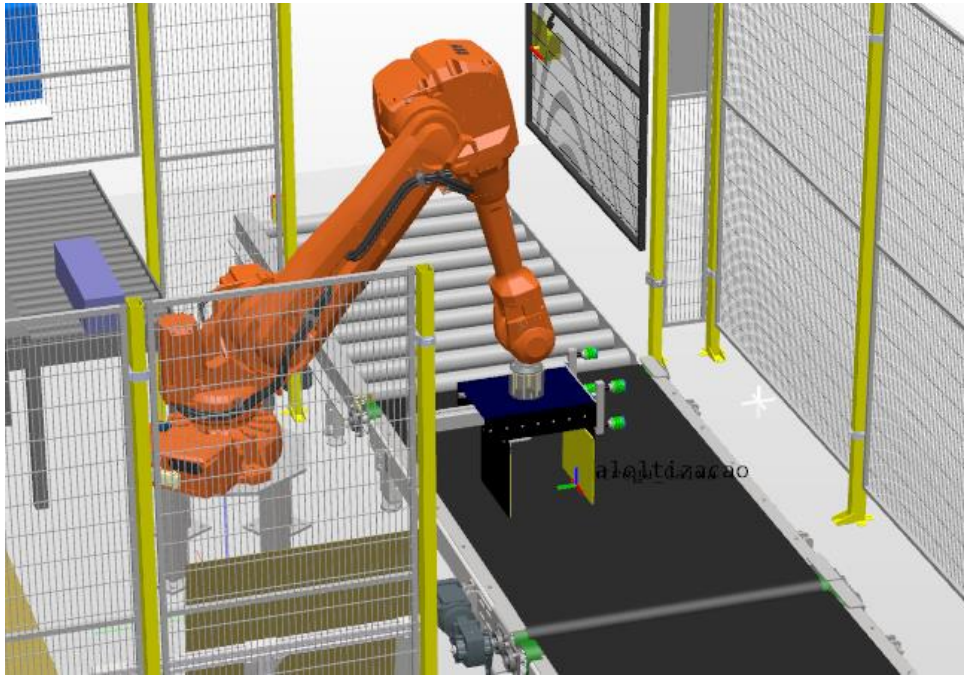


Figura 42 - Representação da posição de entrega da caixa na paleta

Sempre que o robô vai entregar uma caixa na paletização, a posição de entrega é modificada de forma que a caixa seja colocada na posição e com a orientação correta na paleta, indo ao encontro da correta formação de caixas na paleta, de acordo com o especificado pelo cliente para cada tipo de produto a paletizar.

4.1.5.4 Posição de recolha de paleta de cartão

A posição de recolha de paletes está localizada na posição (500,600,0), no sistema de coordenadas “xyz”, da base do armazém de paletes. A posição de recolha foi colocada estrategicamente nesta posição, em relação à base do armazém de paletes, pelos seguintes critérios: facilitar a correção da sua posição mediante a dimensão da paleta a ser recolhida

pela ferramenta que, neste caso, é a pinça de ventosas; garantir que a posição, quando definida, não origine uma colisão da ferramenta com o suporte de auxílio à colocação de paletes pelo operador no armazém (Figura 43).

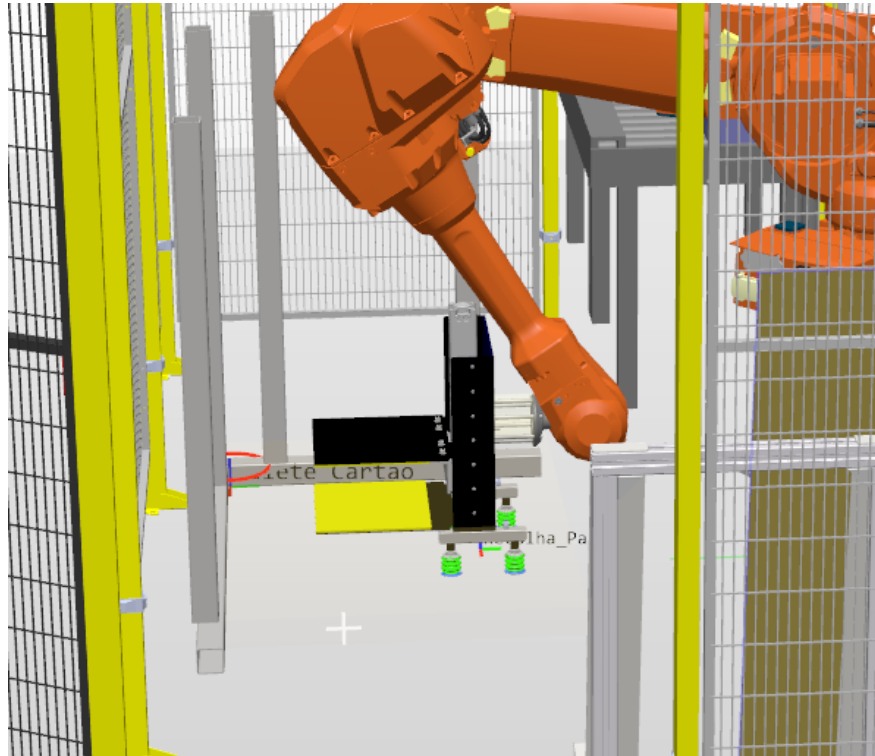


Figura 43 - Representação da posição de recolha de paletes.

Para que a ferramenta consiga recolher a paleta no seu centro, basta efetuar o cálculo do desvio entre o início da paleta e o seu centro, adicionando esse desvio à posição alvo de recolha.

4.1.5.5 Posição de recolha de cartões

A posição de recolha de cartões fica localizada, nas coordenadas “xyz”, na posição (500,-500,0), no sistema da base do armazém de cartões (Figura 44).

A escolha da posição de recolha de cartões em relação à base do armazém de cartões, obedece aos seguintes critérios: facilitar a correção da sua posição mediante a dimensão do cartão a ser recolhido pela ferramenta; por segurança, garantir que quando definida a sua posição, esta não origine uma colisão do conjunto robô e ferramenta com o suporte de colocação de cartões.

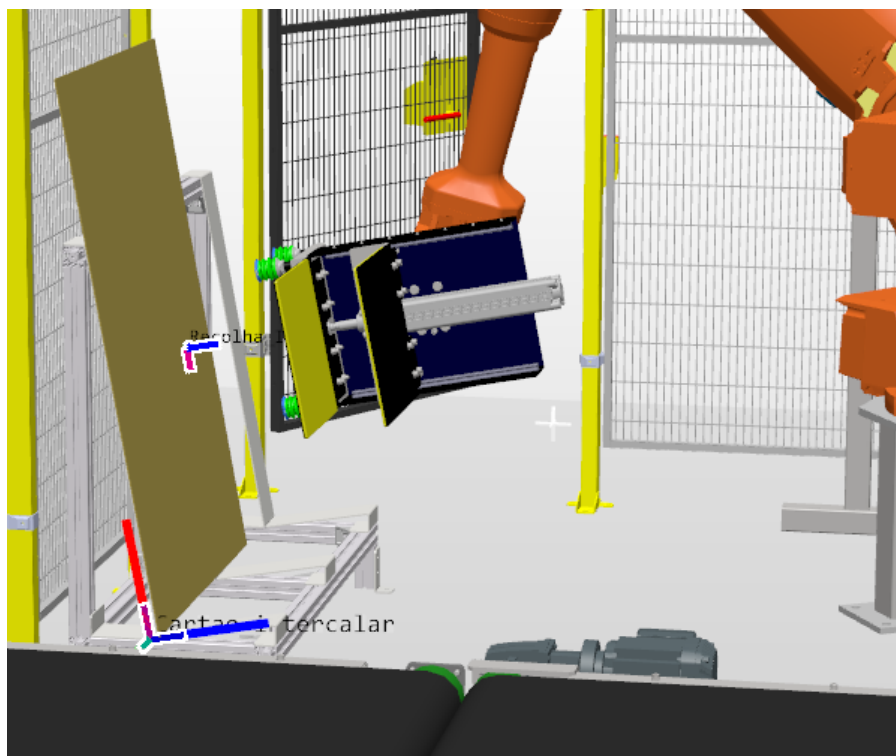


Figura 44 - Representação da posição de recolha de cartões.

4.1.6 Desenvolvimento da aplicação para paletização de caixas

O objetivo deste projeto é desenvolver uma aplicação em que um robô paletize caixas. Para isso é necessário reunir informação das características da paleta a ser formada. Essa informação, de uma forma resumida, é a seguinte: as dimensões da caixa; a orientação na paleta; o número de caixas por plano; o número de planos, entre outras informações.

A ilha é composta por: um robô antropomórfico IRB4600 da ABB que tem a capacidade de movimentar cargas até 60Kg, com alcance até 2050mm; um transportador de rolos, para fazer chegar caixas ao alcance do robô; uma mesa para a formação e transporte da paleta depois de finalizada; um armazém de paletes de cartão; um armazém de cartões intercalares; uma etiquetadora.

O processo de paletização é iniciado quando for detetada a primeira caixa na mesa de rolos. De seguida, o robô vai buscar uma paleta de cartão ao armazém e coloca-a na linha de transporte de paletes. A partir daqui, começa a pegar nas caixas que vão chegando na mesa de rolos e coloca-as na paleta, na posição e orientação correta, até formar o primeiro plano. Após o plano estar formado, o robô vai buscar um cartão ao armazém e coloca-o na paleta. Todo este processo repete-se até formar o número de planos necessário para

completar a paleta, a qual é então transportada para a mesa seguinte, ficando a mesa de paletização livre para dar início à formação de uma nova paleta.

Uma das características desta aplicação é poder paletizar vários formatos de caixas. Obviamente que a dimensão máxima da caixa a ser paletizada tem de estar dentro das dimensões para as quais a ferramenta foi desenvolvida. Há também a possibilidade de a orientação das caixas na paleta ser variável, ou seja, podem ser colocadas em vários ângulos, nomeadamente 0, 90, 180 e -90 graus, dependendo da formação do produto a paletizar. Estes ângulos estão diretamente relacionados com a reorganização das caixas na paleta e a visibilidade da etiqueta presente na caixa, pois esta tem de ficar visível no exterior da paleta, para que desta forma, seja mais fácil a identificação do produto.

Para que o robô consiga realizar a paletização com sucesso, foi criada uma receita para cada tipo de caixa a ser paletizada, na qual estão todos os dados necessários à formação da paleta e de modo que a aplicação desenvolvida interprete e processe esses dados e execute as várias tarefas até formar a paleta desejada.

Para além de toda a informação para se poder realizar a paletização, o robô necessita de reunir informação do que o rodeia, neste caso através de sensores, para depois processar esses sinais e atuar ou realizar tarefas em conformidade. Este tema será abordado na próxima secção.

4.1.6.1 Sinais com PLC, sensores e atuadores

Para executar todas as tarefas, o robô necessita de informação, seja sensorial ou recebida de outros equipamentos, que permita ao controlador tomar decisões, tais como parar o robô em caso de emergência, selecionar uma receita, iniciar um ciclo de paletização, atuar o cilindro pneumático, ligar sistema de vácuo ou mesmo pedir uma etiqueta, entre outros. Através do esquema elétrico, fornecido pelo departamento de desenvolvimento, verifica-se as conexões existentes e onde estão conectadas, para que se configure no controlador do robô as mesmas conexões com os nomes coincidentes ao esquema.

O controlador do robô vem equipado com entradas e saídas do sistema, nas quais estão interligados o circuito de emergência da ilha (estas ligações não serão abordadas, uma vez que não são controladas pela aplicação desenvolvida), um módulo ‘DSQC 1030’ da ABB (Figura 45), o qual passamos a chamar de módulo local, já configurado com 16

entradas e 16 saídas digitais e comunicação com equipamentos exteriores através de ethernet/ip que, neste caso, vai comunicar com o *PLC* da ilha.



Figura 45 - Módulo DSQC 1030 de entradas e saídas digitais.

No módulo ‘DSQC 1030’ as entradas digitais estão ligadas à parte sensorial que é composta por sensores de deteção da existência de paletes e de cartões nos respetivos armazéns, sensores de deteção de ferramenta(pinça) aberta ou fechada, vacuostato para deteção de presença de palete ou cartão nas ventosas e pressostato para deteção de caixa na ferramenta (Figura 46). De seguida será efetuado um breve resumo da função de cada um dos sensores.

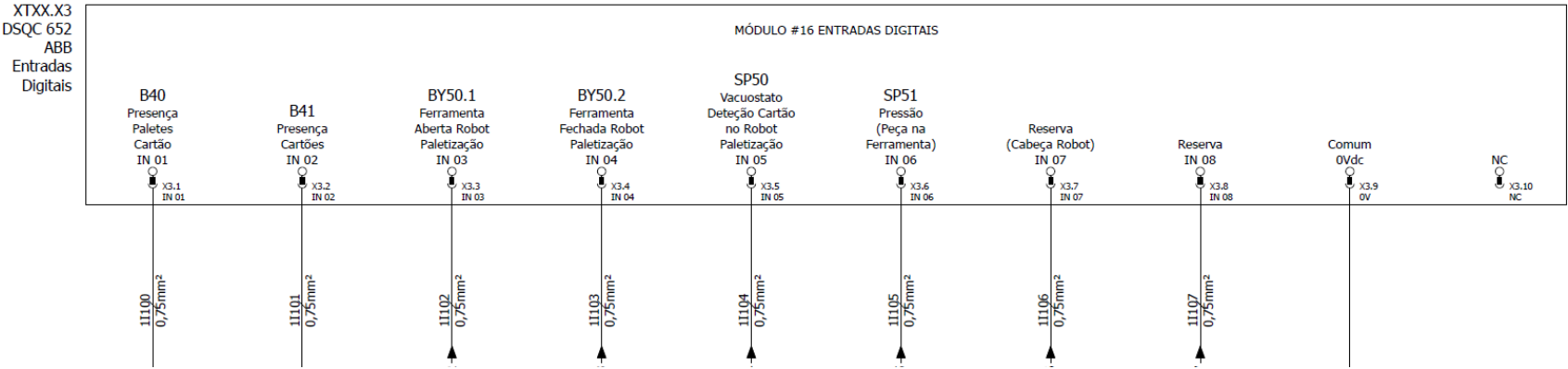


Figura 46 - Esquema de entradas digitais.

O sensor de deteção de existência de cartões, como já referido, está instalado no armazém destinado à colocação de paletes. Este sensor tem como objetivo, alertar para a falta de paletes no armazém (Figura 47).



Figura 47 - Representação do sensor de presença de paletes no armazém.

Os sensores de deteção do estado da pinça da ferramenta estão instalados no cilindro pneumático que efetua o movimento de abertura ou fecho da pinça. Quando o cilindro pneumático está recolhido o sensor de pinça fechada é atuado e quando o cilindro pneumático está completamente avançado o sensor de pinça aberta é atuado (Figura 48).

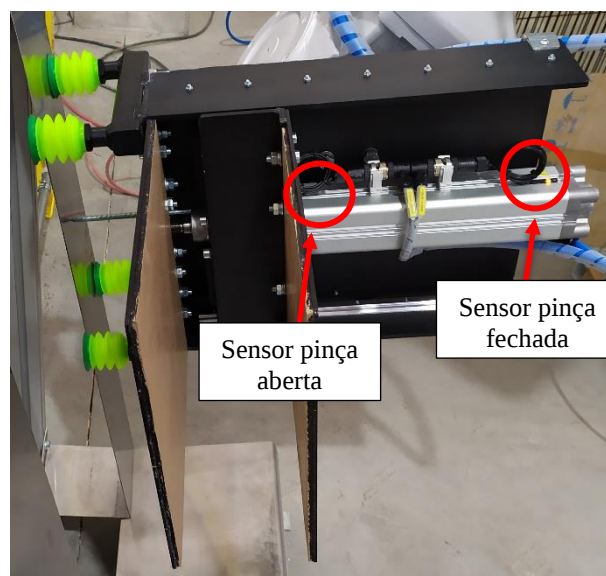


Figura 48 - Sensores de abertura e fecho da pinça.

O sensor de vácuo, está instalado na entrada da válvula venturi, no gerador de vácuo, (Figura 49). Este vacuostato tem como objetivo detetar uma depressão nas ventosas montadas na ferramenta. Desta forma confirma-se que as ventosas agarraram algum objeto que, neste caso, pode ser uma paleta ou um cartão.

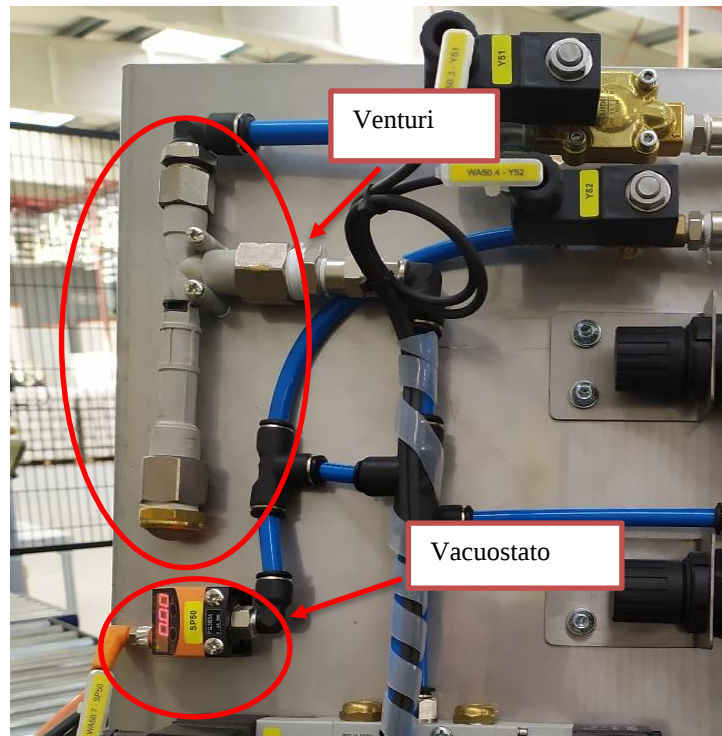


Figura 49 - Sistema de vácuo(venturi) e vacuostato.

Por último vem o sensor de pressão. Este sensor está instalado à saída da electroválvula que efetua o recuo do cilindro pneumático, ou seja, quando a pinça está a fechar (Figura 50). Quando a pinça vai pegar numa caixa, os sensores de abertura ou fecho da pinça não estão atuados, mas é necessário saber se a pinça pegou em algo, neste caso, uma caixa. O pressostato vai ativar um sinal mediante a pressão pré-programada: é desta forma que se deteta a presença de uma caixa na pinça.

A electroválvula para abertura e fecho da pinça é composta por duas bobines. Quando uma das bobines é acionada, o cilindro pneumático abre a pinça e quando a outra bobine é acionada, fecha a pinça (Figura 50). As bobines nunca podem ser acionadas em simultâneo: se isso acontecer, pode acionar o cilindro pneumático para uma posição indesejada ou mesmo não efetuar qualquer movimento.

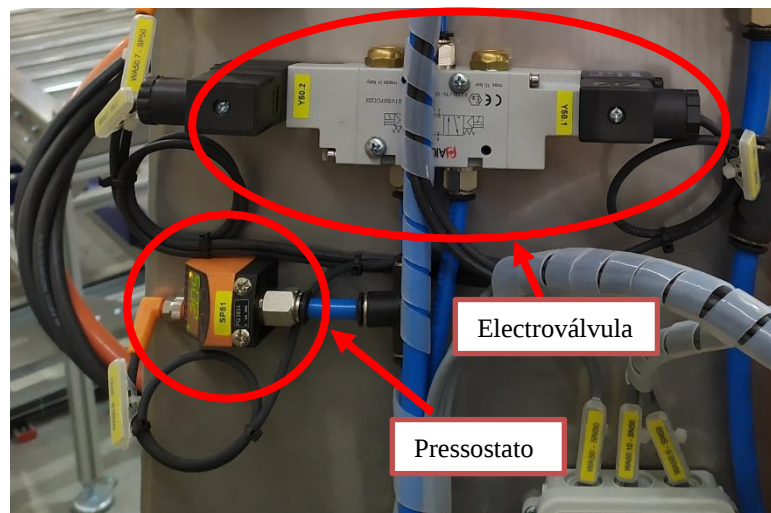


Figura 50 - Pressostato e electroválvula para abertura e fecho da pinça.

A electroválvula do sistema de vácuo está instalada à entrada da válvula venturi, que gera o vácuo para as ventosas pegarem numa paleta ou num cartão (Figura 51).

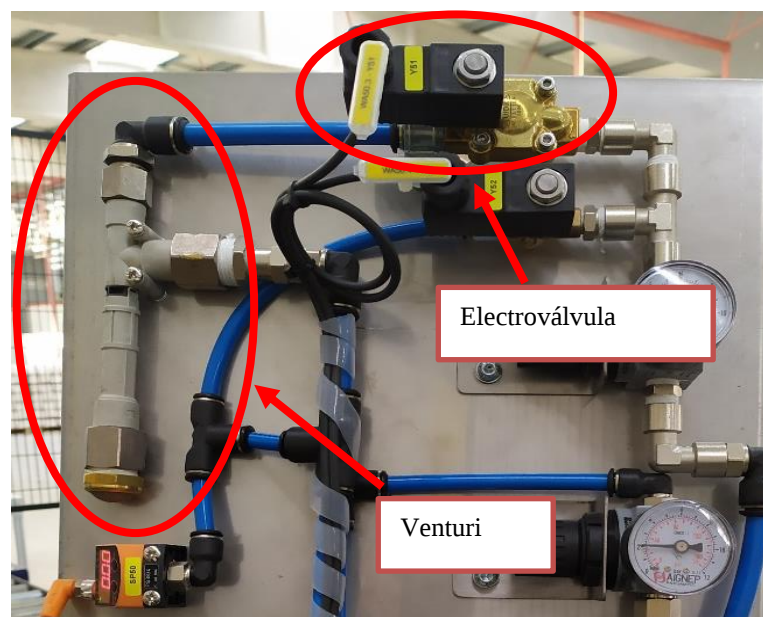


Figura 51 - Electroválvula para acionar sistema de vácuo venturi.

Nas saídas digitais do módulo local, estão ligados os atuadores, que são a electroválvula de abertura e fecho da pinça, a electroválvula do injetor de vácuo (venturi) e ativação do laser de medição de distância (Figura 52).

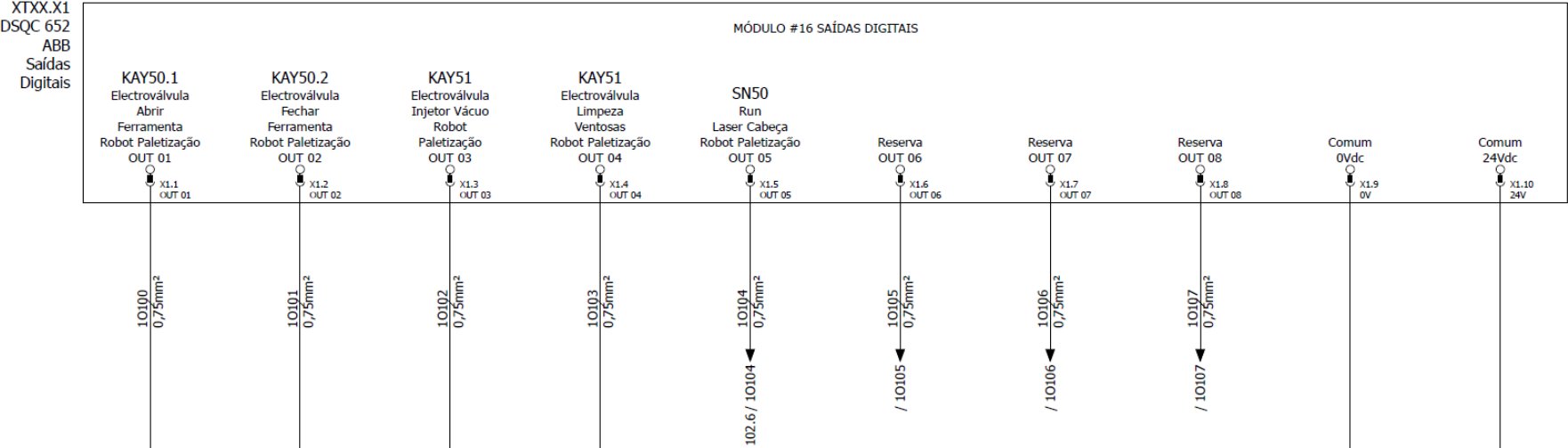


Figura 52 - Esquema das saídas digitais.

Na ferramenta está instalado um sensor laser que mede a distância entre um objeto e a ferramenta. Este laser tem dois propósitos: o primeiro consiste em medir a distância a que se encontra o cartão ou palete, quando os for recolher aos armazéns (Figura 53); o segundo, detetar a presença de um cartão na ferramenta, neste caso nas ventosas.

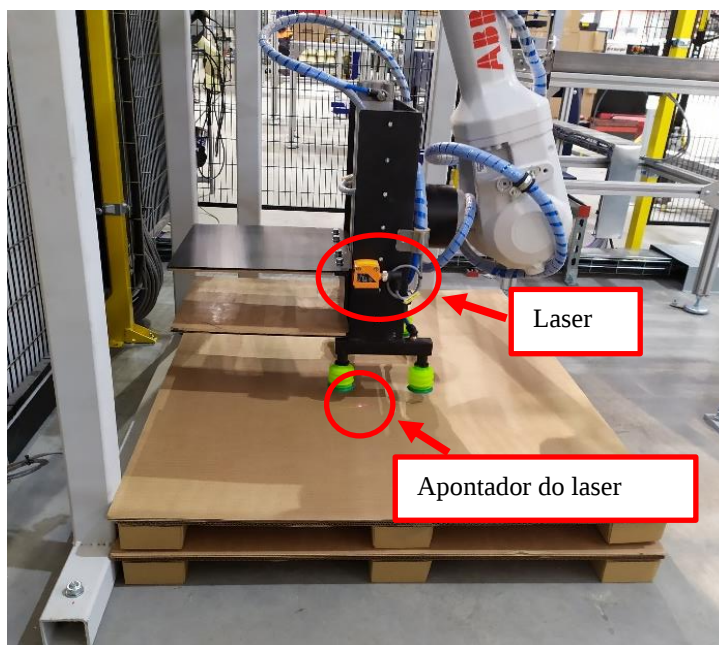


Figura 53 - Laser para medição de distância.

O conjunto completo dos sinais de entradas e saídas do módulo pode ser consultado na Tabela A1 do Anexo A.

A comunicação com o *PLC* da ilha é feita utilizando o protocolo ethernet/ip: trata-se de um protocolo de camada de transporte, segundo o modelo *OSI* [9], para aplicações industriais [10].

Esta comunicação foi configurada no controlador do robô, começando por se criar uma comunicação industrial, à qual se atribuiu o nome de “EtherNetIP_Anybus”, com o IP “192.168.2.11” (Figura 54). Este *IP* foi atribuído para que estivesse dentro da gama definida para os equipamentos que englobam a ilha.

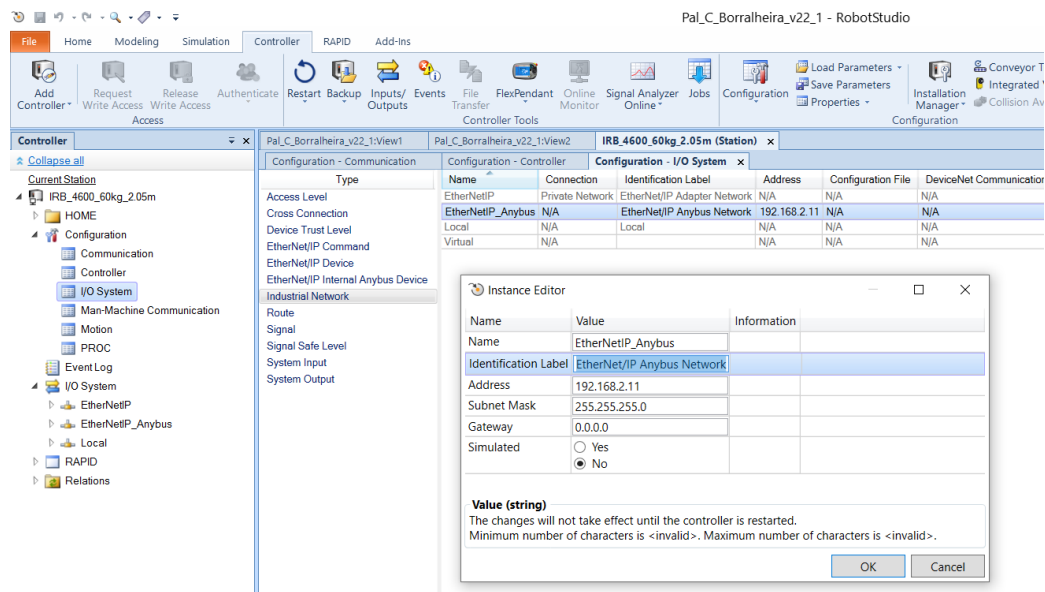


Figura 54 - Configuração de uma rede ethernet/ip para comunicar com o PLC.

Depois de criada a comunicação, é necessário definir o número de bytes a serem enviados e recebidos entre o robô e o PLC. Foram definidos 64 bytes para entrada de dados e 64 bytes para a saída de dados (Figura 55). O número de bytes a serem transmitidos está sobredimensionado para que futuramente, caso seja necessário, se possa adicionar sinais para troca de informação com o 'PLC'.

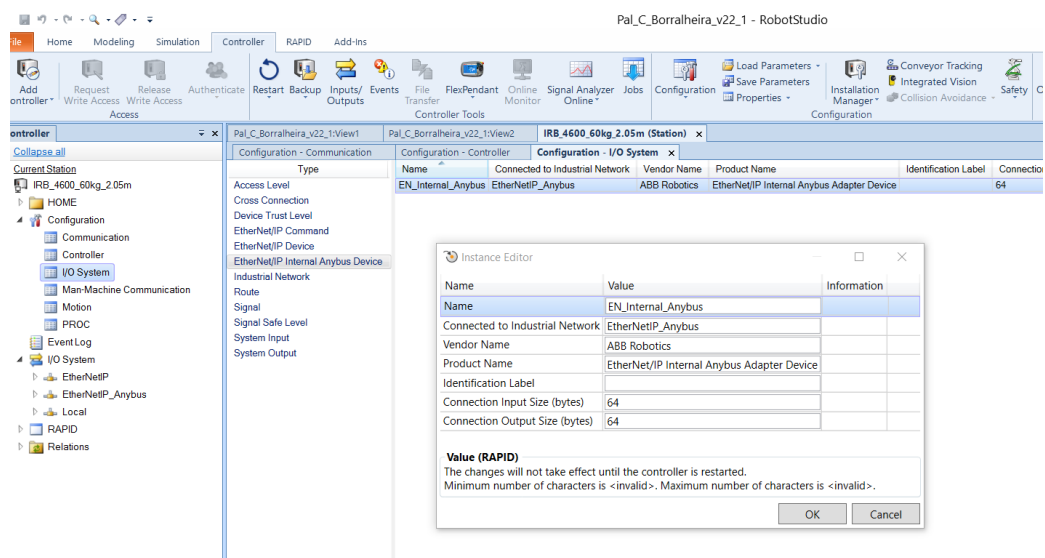


Figura 55 - Declaração do número de bytes de entrada e saída por via da comunicação ethernet/ip.

Depois de configurar toda a comunicação com o PLC, foi necessário criar e configurar toda a informação necessária para o PLC interagir com o robô. Esta informação pode ser

enviada como sinais digitais(bits), grupos de sinais digitais (bytes) e sinais analógicos. Toda a informação referente aos sinais foi criada ao mesmo tempo que se desenvolvia a aplicação. No início do desenvolvimento, não era claro qual a informação que era necessário trocar com o PLC para que a aplicação desempenhasse todas as tarefas desejadas.

Tabela 3 - Exemplo de dados que são trocados entre o PLC e o controlador do robô.

ABB_to_PLC	PLC_to_ABB
EN_DO00_MotOn	EN_DI00_CmdMtOn
EN_DO01_Auto	EN_DI01_StartMain
EN_DO02_Cycle	EN_DI02_Start
EN_DO03_Emer	EN_DI03_ResetStop
EN_DO10_SEARCH_NOT_FIND	EN_DI10_EXCHANGE_PROD
EN_DO13_Palette_concluida	EN_DI11_INIT_PAL
EN_DO14_Imp_Etiqueta	EN_DI12_Palette_concluida_ACK
EN_DO16_WZ_HOME	EN_DI13_Coloc_Etiqueta
EN_DO17_WZ_PALETTE	EN_DI14_M_Func_Pal_Base

Na Tabela 3, estão representados alguns dos dados a serem trocados com o PLC. A tabela completa pode ser consultada nos Anexo B e C.

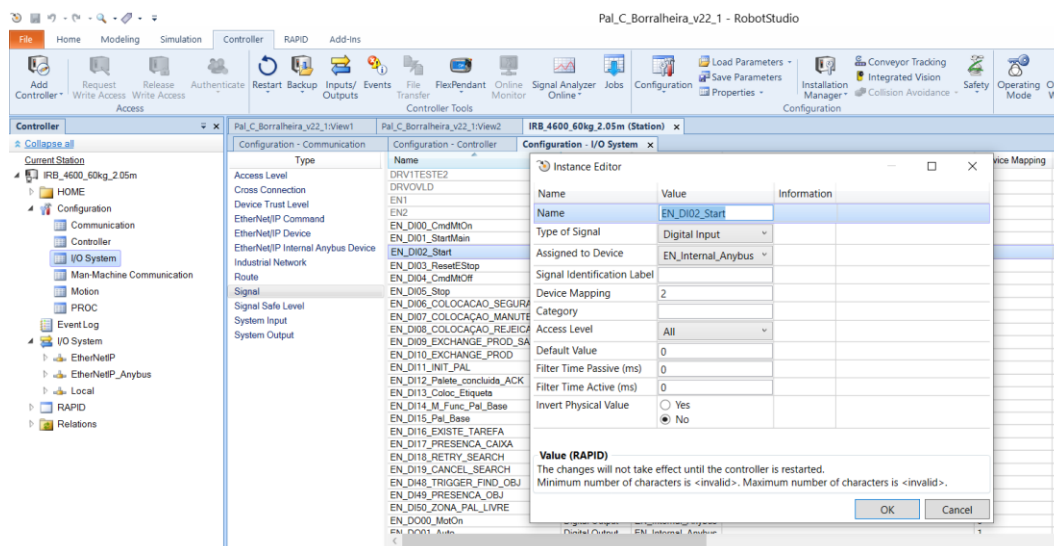


Figura 56 - Exemplo da configuração de um sinal.

O passo seguinte foi a configuração dos sinais no controlador. Essa configuração é feita individualmente para cada sinal, tendo como principais parâmetros o nome, o tipo de sinal, o dispositivo de comunicação e o mapeamento (Figura 56).

4.1.7 Programação da aplicação de paletização de caixas

A programação da aplicação é composta por vários módulos que podem conter programas ou funções. No caso dos programas, são inseridas as instruções que o controlador terá de executar e que podem ser a declaração de variáveis, os ciclos, as condições, os movimentos e até chamar outros programas ou funções. Os programas podem ter parâmetros de entrada e não devolvem parâmetros. Já as funções, podem fazer o mesmo que os programas, mas devolvem sempre um parâmetro. Normalmente são utilizadas para efetuar cálculos ou tarefas que não exigem controlo de movimento do robô.

Esta aplicação é dividida em vários módulos que contêm programas ou funções que desempenham tarefas específicas, de forma a simplificar a interpretação de cada tarefa que o robô terá de executar. Na Figura 25 da secção 4.1.1, está representado um fluxograma generalizado das tarefas do programa desenvolvido. Na realidade, toda a aplicação desenvolvida é muito mais complexa, tendo procedimentos e tarefas a ser realizadas que serão abordados nos parágrafos seguintes.

Dum modo geral e mais simplista, sempre que chega uma nova caixa à mesa de entrega para ser paletizada o robô verifica se é o início da paletização.

No caso ser o início da paletização, o robô executa a tarefa de recolher uma paleta de cartão no armazém e coloca-a na mesa onde é feita a paletização, seguindo então para a tarefa seguinte.

No caso da paletização já ter sido iniciada, o robô passa para as tarefas seguintes, que consistem em recolher a caixa existente na mesa de entrega, passar com a caixa no dispensador de etiquetas e colocá-la na paleta, executando estas tarefas até esse nível estar completo. Assim que o nível estiver completo, o robô verifica se existe a necessidade de colocar um cartão. Se sim, o robô recolhe um cartão no armazém e coloca-o em cima do nível completo. De seguida, o robô verifica se a paleta está completa. Se sim, dá como terminada a paletização ao *PLC* da ilha, para que desta forma se dê seguimento à paleta concluída para fora da ilha e se inicie um novo processo de paletização. No caso de a paleta não estar concluída, o robô vai construir o próximo nível até este estar completo e repete este processo até a paleta estar completa, para depois ser dada como concluída, como já referido anteriormente.

O controlador do robô pode funcionar em modo de ciclo único ou de ciclo contínuo. Em ambos os casos, o controlador executa um programa denominado ‘MAIN’. No modo de ciclo único o controlador só executa o programa uma única vez. Para o executar novamente é necessário indicar ao controlador um novo ciclo, utilizando um sinal de início de ciclo. Já no modo de ciclo contínuo, o controlador executa o programa ‘MAIN’ infinitamente, até que seja dada a indicação ao controlador para terminar, utilizando um sinal de paragem de ciclo.

É no programa ‘MAIN’ que o robô faz a sua inicialização, troca de receita, verifica todas as condições para se efetuar a paletização e só depois é que é chamado o programa para efetuar a paletização, denominado por ‘Prog_1’.

Nas Figura 57 e Figura 58 está representado o fluxograma geral do programa ‘MAIN’, dividido em duas partes. Este programa é sempre executado quando se inicia o arranque do robô. Efetua várias verificações ao estado das variáveis e os sensores dão informação relativa ao estado de toda a ilha, por exemplo, se o robô está na posição inicial ‘HOME’, se é para trocar a receita, se é para reinicializar a paletização, se a mesa de paletização está livre ou não. O robô toma uma decisão e executa determinada tarefa após processar toda a informação anteriormente referida. Esta tarefa pode ser a colocação do robô numa posição inicial, executar outro programa, como pode ser simplesmente a reinicialização de variáveis.

Após o robô terminar a tarefa, inicia novamente o programa ‘MAIN’ para dar início a um novo ciclo.

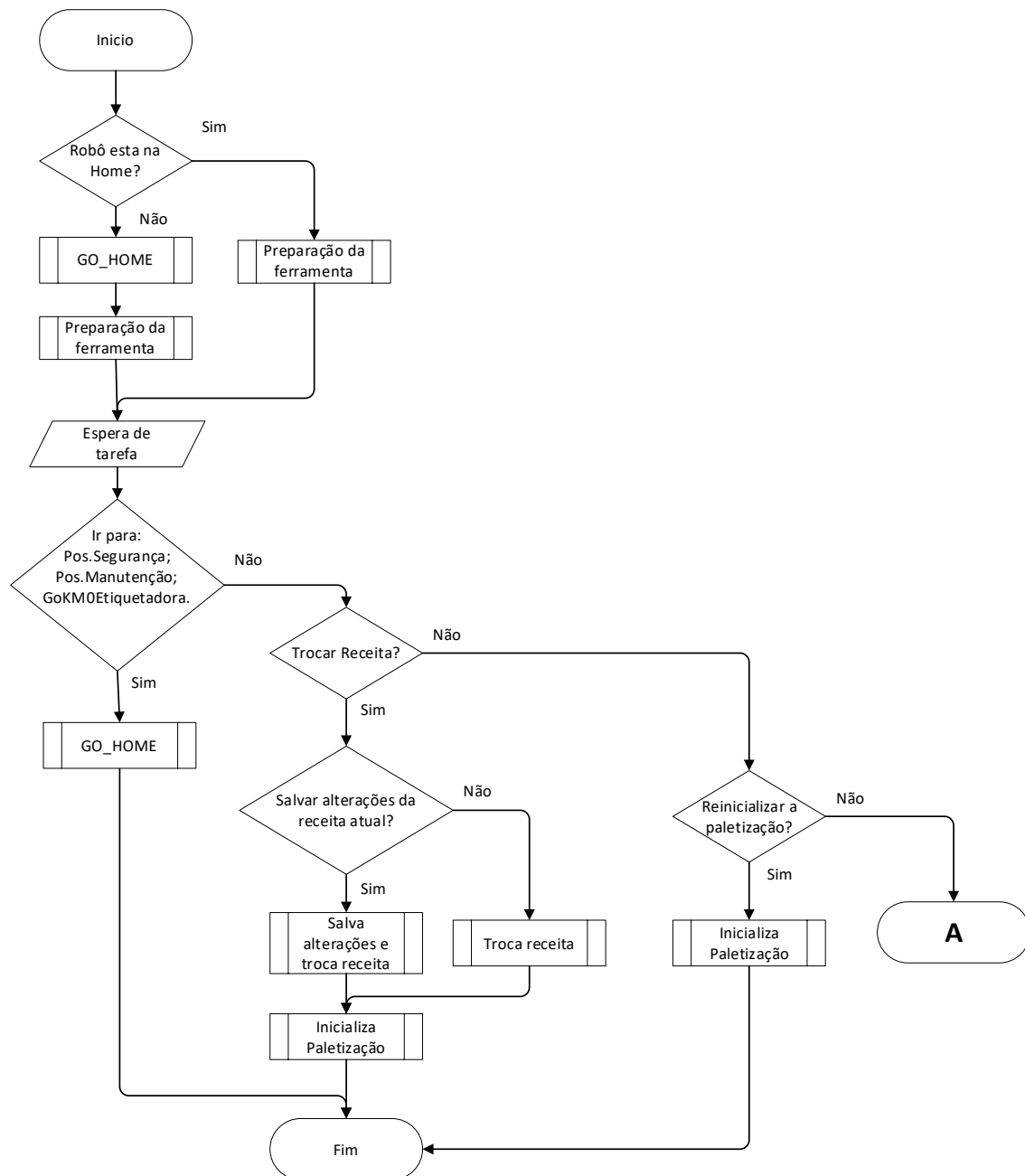


Figura 57 - Fluxograma do programa 'MAIN', primeira parte.

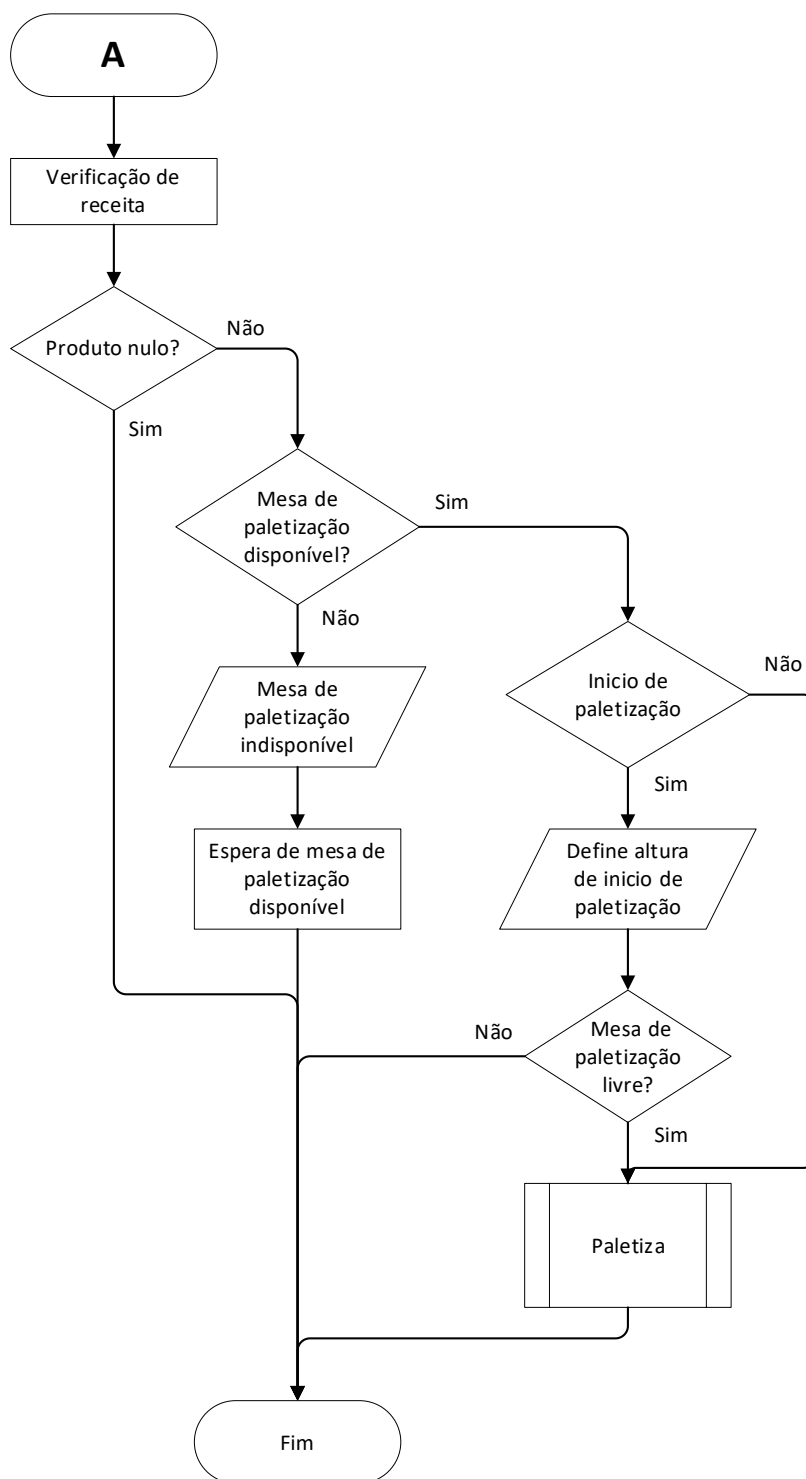


Figura 58 - Fluxograma do programa 'MAIN', segunda parte

Explica-se de seguida, de uma forma geral, as condições para realizar cada tarefa que o robô pode vir a executar no programa ‘MAIN’.

Uma das primeiras condições é verificar se o robô está na posição inicial ‘HOME’: só depois é que o robô passa a verificar as condições seguintes (Figura 59). Se o robô não se encontrar na posição ‘HOME’, vai executar o subprograma ‘GoHome’, no qual vai reposicionar-se, numa velocidade mais reduzida, nessa posição.

```

25 IF EN_DO16_WZ_HOME=0 OR Go_Manutencao=1 THEN
26     GoHome;
27     ! Caso o robo de posição desconhecida, espera 2s
28     IF...ENDIF
33     TRY_CLOSE_GRIPPER1:
34     fechar_gripper;
35     IF Flag_f_fc_gripper GOTO TRY_CLOSE_GRIPPER1;
36     desligar_vacuo;
37 ELSE
38     IF Go_Manutencao=1 GOTO NEXT1;
39     TRY_CLOSE_GRIPPER2:
40     IF...ENDIF
45     IF Flag_f_fc_gripper GOTO TRY_CLOSE_GRIPPER2;
46     NEXT1:
47     desligar_vacuo;
48     Reset EN_DO08_POS_DESC;
49 ENDIF

```

Figura 59 - Condição que verifica se o robô está na posição 'HOME'.

Após o robô estar na posição ‘HOME’, é preparada a ferramenta para iniciar uma nova tarefa. Neste caso, a ferramenta deve ter a pinça fechada e o vácuo das ventosas desligado. Uma vez que o robô esteja na posição ‘HOME’ e a ferramenta preparada, estão reunidas todas as condições para executar uma nova tarefa, ficando à espera do sinal ‘EN_DI16_EXISTE_TAREFA’ do autómato que controla toda a ilha (Figura 60).

```

56 ! ESPERA DE INICIO DE TAREFA
57 task_msg msg98;
58 SetDO EN_DO06_AtMain,1;
59 WaitTime 0.5;
60 WaitDI EN_DI16_EXISTE_TAREFA,1\MaxTime:=1000\TimeFlag:=flag_overflow_time;

```

Figura 60 - Condição de espera de nova tarefa.

O PLC pode indicar ao robô, através de sinais para ele enviados, que tipo de tarefa tem de executar. Nesta aplicação pode-se executar os seguintes tarefas: posicionar o robô numa posição de manutenção ou executar o KM0 da etiquetadora (ponto de referência para posicionar a etiquetadora); troca de receita; reinicialização da paletização; no caso

de não acontecer nenhuma das opções anteriores, o robô vai paletizar. Explica-se, de seguida, cada uma destas tarefas.

Ao receber a informação da existência de uma tarefa, a primeira condição a ser verificada é se existe a necessidade de colocar o robô na posição de segurança, que neste caso é a posição ‘Home’, na posição de manutenção ou mesmo na posição de *KM0* do dispensador de etiquetas, de acordo com o código apresentado na Figura 61.

```

70 |           ! VERIFICA NECESSIDADE DE IR PARA POSICAO DE SEGURANÇA OU MANUTENÇÃO
71 | IF EN_DI06_COLOCACAO_SEGURANCA=1 OR Go_Mantencao=1 OR GO_KM0_Etiquetadora=1 THEN
72 |     task_msg msg1;
73 |     GoHome;

```

Figura 61 - Verificação da necessidade de colocar o robô na posição de segurança, de manutenção ou *KM0* do dispensador de etiquetas.

A colocação do robô na posição de manutenção e *KM0* (quilómetro zero) da etiquetadora, resume-se à execução do subprograma ‘GoHome’. Dentro deste, é executado um outro subprograma chamado ‘Go_manutenção’, ou seja, o programa ‘GoHome’ coloca o robô na posição ‘Home’ e só depois é que executa o subprograma ‘Go_manutenção’. Desta forma garante-se que quando o robô iniciar a trajetória para a posição de manutenção, segue sempre a mesma trajetória. A posição de manutenção é utilizada para que um técnico possa efetuar manutenção na ferramenta do robô, sem ter de posicionar o robô, de forma manual, numa posição em que se consiga efetuar essa operação.

```

237 | IF Go_Mantencao=1 THEN
238 |     Reset Go_Mantencao;
239 |     aux_joint_atual:=CJointT();
240 |     aux_joint_destino:=aux_joint_atual;
241 |     aux_joint_destino.robax.rax_1:=175;
242 |     SetGO EN_G096_111_MOV_INICIADO,2;
243 |     MoveAbsJ aux_joint_destino,vGoSeg,z50,Gripper\WObj:=wobj0;
244 |     MoveAbsJ J_Mantenção,vGoSeg,z50,tool0\WObj:=wobj0;
245 |     WaitRob\InPos;
246 |     UI_ASK_MANUT;
247 |     MoveAbsJ aux_joint_destino,vGoSeg,z50,Gripper\WObj:=wobj0;
248 |     MoveAbsJ Home\NoE0ffs,vGoSeg,z50,tool0\WObj:=wobj0;
249 |     SetGO EN_G0112_127_MOV_CONCLUIDO,2;
250 | ELSEIF GO_KM0_Etiquetadora=1 THEN
251 |     KM0_Etiq;
252 |     GOTO INICIO;
253 | ENDIF

```

Figura 62 - Opção de o robô ir para a posição de manutenção, ou efetuar o *KM0* da etiquetadora.

Na Figura 62 está representado o trecho de código que verifica o sinal ‘Go_Mantencao’. Quando ativado o sinal ‘Go_Mantencao’ com o valor lógico “1”, o robô vai para a

posição de manutenção e vai aparecer uma mensagem na consola para informar que ele já se encontra nessa posição (Figura 63).

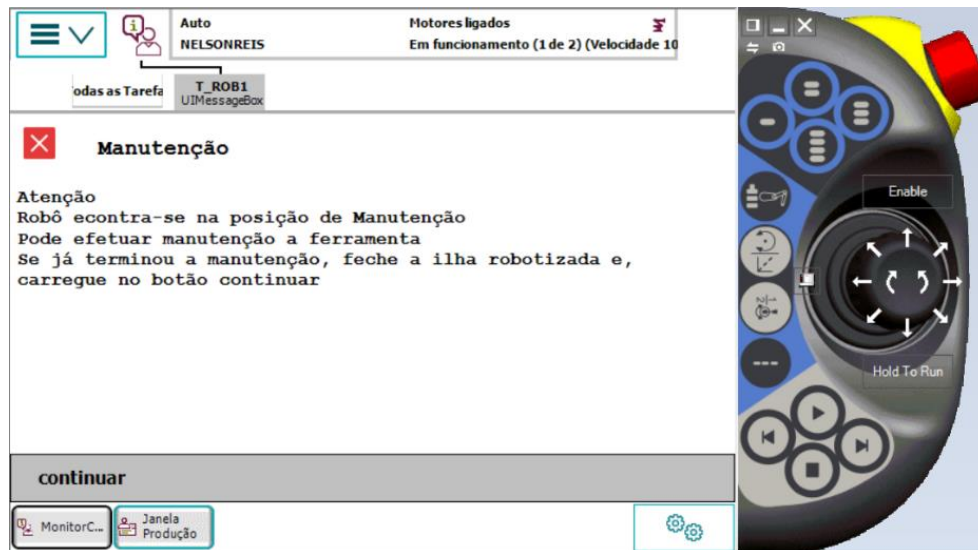


Figura 63 - Representação da mensagem que confirma que o robô está na posição de manutenção.

Nesta fase, o técnico pode entrar na ilha e efetuar a manutenção à ferramenta. Quando terminar, deve sair de ilha, arrancar com o sistema e só depois confirmar, na consola do robô, que concluiu a ação de manutenção. Desta forma, o robô vai novamente para a posição 'Home' e fica à espera de nova tarefa.

O subprograma 'KM0_Etiq' coloca a ferramenta numa posição próxima do dispensador de etiquetas para que desta forma seja fácil de reposicionar o dispensador de etiquetas. A ferramenta, estando na posição *KM0*, serve de referência para ajustar a posição do dispensador de etiquetas.

```
3 PROC KM0_Etiq()  
4     Reset GO_KM0_Etiquetadora;  
5     fechar_gripper;  
6     UI_ASK_KM0_1;  
7     MoveL RelTool(KM0_Etiquetadora,0,400,-100),vGoSeg,fine,Gripper\WObj:=wobj0;  
8     MoveL RelTool(KM0_Etiquetadora,0,30,-100),vGoSeg,fine,Gripper\WObj:=wobj0;  
9     MoveL KM0_Etiquetadora,vGoSeg,fine,Gripper\WObj:=wobj0;  
10    UI_ASK_KM0_2;  
11    MoveL RelTool(KM0_Etiquetadora,0,400,0),vGoSeg,fine,Gripper\WObj:=wobj0;  
12 ENDPROC
```

Figura 64 - Subprograma 'KM0_Etiq'.

No caso do sinal ‘GO_KM0_Etiquetadora’ assumir o nível lógico “1”, o robô vai executar o subprograma ‘KM0_Etiq’ (Figura 64). Ao executar este programa, vai aparecer na consola do robô uma mensagem a alertar para que não haja eventuais objetos próximos da posição ‘KM0_Etiquetadora’ com os quais a ferramenta do robô possa colidir. Para que o robô inicie a trajetória para a posição ‘KM0_Etiquetadora’, o técnico terá de efetuar a confirmação na consola do robô (Figura 65).

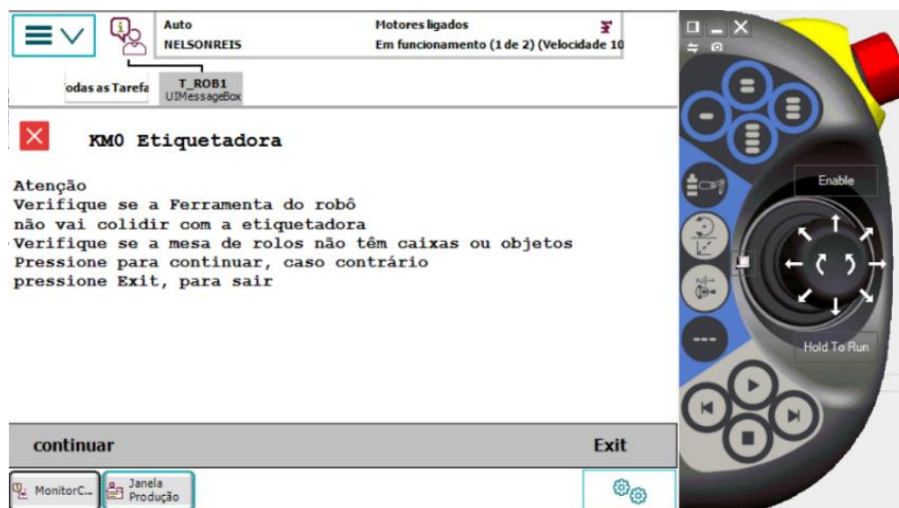


Figura 65 - Mensagem de alerta e confirmação de início de trajetória para a posição 'KM0_Etiquetadora' do dispensador de etiquetas.

Após a chegada do robô à posição 'KM0_Etiquetadora', é apresentada uma nova mensagem a informar já se encontra nessa posição. Para além desta informação, a mensagem dá indicações de como se deve posicionar o dispensador de etiquetas em relação à ferramenta. Nesta fase, o técnico deve entrar dentro da ilha e deve posicionar o dispensador de etiquetas da forma indicada pela mensagem apresentada na consola do robô. Quando o técnico terminar de posicionar o dispensador, deve sair da ilha e confirmar na consola, pressionando a opção 'continuar'. Só depois deve arrancar com o sistema (Figura 66). Desta forma, o robô vai para a posição 'Home' e fica à espera de nova tarefa.

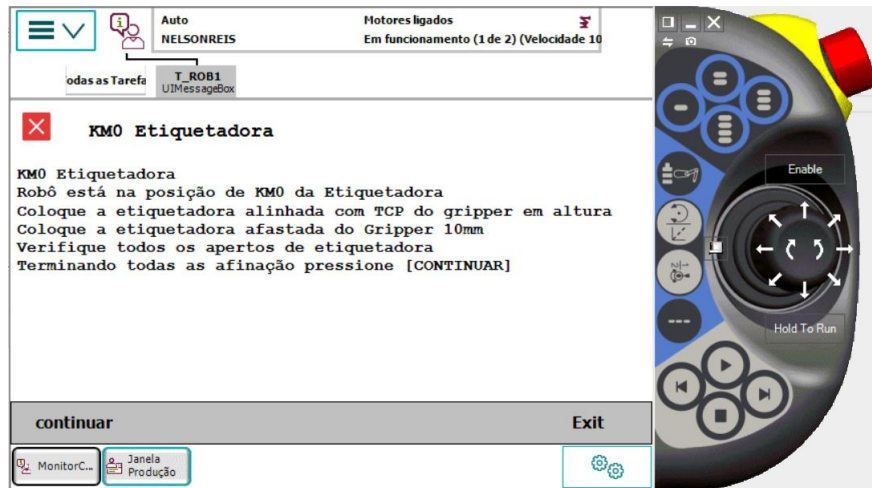


Figura 66 - Mensagem a informar que o robô se encontra na posição 'KM0_Etiquetadora' e indicação do procedimento para colocar o dispensador de etiquetas na posição correta.

No programa 'MAIN', quando o robô recebe uma nova tarefa e o sinal 'EN_DI09_EXCHANGE_PROD_SAVE' ou o sinal 'EN_DI10_EXCHANGE_PROD' estiver a um, o robô troca a receita (Figura 67).

Como se pode constatar, há dois sinais diferentes para trocar a receita, pois existem duas opções de troca de receita. Em ambos os casos, é executada a função 'change_prod()', recebendo como parâmetros o número da receita que se pretende utilizar na paletização e salvando a receita existente.

```

75      ELSEIF EN_DI09_EXCHANGE_PROD_SAVE=1 OR EN_DI10_EXCHANGE_PROD=1 THEN
76          ! Troca de receita
77          Reset EN_D033_EXCHANGE_PROD_NOK;
78          Reset EN_D032_EXCHANGE_PROD_OK;
79          ! Salva receita com alterações no ficheiro de origem e troca receita
80          IF EN_DI09_EXCHANGE_PROD_SAVE=1 THEN
81              IF change_prod(\save,EN_GI80_95_N_RECEITA) THEN
82                  SetDO EN_D032_EXCHANGE_PROD_OK,1;
83              ELSE
84                  SetDO EN_D033_EXCHANGE_PROD_NOK,1;
85              ENDIF
86              ! Troca receita sem salvar alterações
87              SetGO EN_G0208_223_N_Receita,no_receita(file_name);
88          ELSEIF EN_DI10_EXCHANGE_PROD=1 THEN
89              IF change_prod(EN_GI80_95_N_RECEITA) THEN
90                  PulseDO\PLength:=1,EN_D032_EXCHANGE_PROD_OK;
91              ELSE
92                  PulseDO\PLength:=1, EN_D033_EXCHANGE_PROD_NOK;
93              ENDIF
94              SetGO EN_G0208_223_N_Receita,no_receita(file_name);
95          ENDIF
96          IF EN_D033_EXCHANGE_PROD_NOK=0 AND EN_D032_EXCHANGE_PROD_OK=1 THEN
97              init_pal;
98              GoHome;
99          ENDIF
100          ! Verifica se os dados da receita estão corretos
101          Check_receita;

```

Figura 67 – Opção de o robô trocar de receita.

Se o sinal ‘EN_DI09_EXCHANGE_PROD_SAVE’ estiver a ”1”, o robô guarda a receita num ficheiro já existente e depois carrega o ficheiro correspondente à nova receita. Desta forma, todas as alterações que se efetuam na receita serão guardadas. Quando o sinal ‘EN_DI10_EXCHANGE_PROD’ estiver a ”1”, o robô vai eliminar a receita existente e carrega o ficheiro correspondente à nova receita. Em ambos os casos, se a receita carregada contiver erros, o robô carrega uma receita nula padrão. Esta receita tem toda a estrutura de uma receita normal, mas com valores nulos. Desta forma, ao ler esses valores, o robô não inicia a paletização.

O nome receita, é um termo utilizado pela empresa EXSEPI. Não é mais do que um conjunto de parâmetros que serão interpretados pelo robô, inseridos num ficheiro e que podem ser modificados para se obter um resultado desejado, neste caso a paletização de caixas de uma determinada dimensão. Para cada formato de caixa diferente a ser paletizado existe um ficheiro parametrizado de acordo com a formação da paleta pretendida.

A escolha da receita é feita pelo operador através de um *HMI (Human Machine Interface)*, existente na parte exterior da ilha. Este *HMI* foi programado pelo responsável pela automação desta ilha, não sendo o objetivo deste relatório explicar o funcionamento do mesmo. Em todo o relatório só serão abordadas as páginas que interagem com o robô, com o objetivo de tornar mais clara essa interação.

O operador escolhe a designação da receita desejada na lista de receitas, cujo menú se apresenta na Figura 68, utilizando os botões mais (“+ +”) e menos (“- -”). Depois de encontrada a receita, o operador carrega no botão “LER RECEITA PARA VERIFICAÇÃO” e só depois é que pressiona o botão “ATIVAR RECEITA PARA PRODUÇÃO”. Por defeito, a receita atual no robô é salva e só depois é carregada uma nova receita, não existindo a opção para o operador carregar uma nova receita sem salvar a receita existente no robô.

LOGIN		SELEÇÃO DE RECEITA		VOLTAR	SAIR
LISTA DE RECEITAS					
--	++				
Nº receita	11	Designação	SKOGSVINBAR 12		
LER RECEITA PARA VERIFICAÇÃO		GRAVAR RECEITA EM VERIFICAÇÃO		GRAVAR RECEITA EM PRODUÇÃO	
RECEITA EM VERIFICAÇÃO					
Nº receita	11	Designação lida	SKOGSVINBAR 12		
		Designação editada	SKOGSVINBAR 12		
COND. ACTIVACÃO		ACTIVAR RECEITA PARA PRODUÇÃO	Receita Robot OK FALHA	EDIÇÃO RECEITA	
RECEITA EM PRODUÇÃO					
Nº receita	11	Designação lida	SKOGSVINBAR 12		
		Designação editada	SKOGSVINBAR 12		

Figura 68 - Página da lista de receitas no HMI.

O ficheiro da receita contém os dados relativos à paletização, nomeadamente: o nome do produto a ser paletizado; o nome da receita; as características da paletização; as dimensões da paleta de cartão; as dimensões do cartão e a sua posição; as dimensões, a posição e a orientação das caixas; as velocidades definidas para as mais variadas tarefas, entre outros. As estruturas das receitas são sempre iguais. Desta forma, os nomes das variáveis no programa do robô são comuns à estrutura das receitas. Se não utilizássemos este tipo de ficheiro com receita, teriam de ser criados programas individuais para cada produto a ser paletizado, o que levaria a criar variáveis diferentes para cada produto, resultando num trabalho muito exaustivo e complexo.

Para criar uma nova receita, basta copiar o ficheiro de uma receita existente e depois alterar o seu nome e os valores das variáveis, para que quando o robô for interpretar a receita consiga paletizar o produto da forma desejada.

```

1  MODULE receita_6(SYSMODULE)
2      ! Nome do ficheiro
3      CONST string file_name:="receita_6.sys";
4
5      ! Nome do produto
6      CONST string produto:="CHIAFRON 9";
7
8      ! Numero da receita
9      PERS num num_receita:=6;
10
11     ! Declara??o de caracteristicas de paletiza??o
12     CONST caracticas_paletizacao data_paletizacao:=[TRUE,FALSE,6,4,20];
13
14     ! Declara??o das caracteristicas caixa
15     CONST caracticas_caixa data_caixa:=[350,345,235];
16
17     ! Declara??o de dados das paletes
18     PERS paleta dados_paleta{2}:=[ [TRUE,1070,730,68,20,0,0,100,[2000,200,0,0],[200,100,0,0],[200,100,0,0],
19                                     [TRUE,1070,730,68,20,0,0,100,[2000,200,0,0],[200,100,0,0],[200,100,0,0],
20
21     ! Declara??o de dados do cart??o intercalar
22     PERS intercalar dados_intercalar{5}:=[ [1050,710,3,0,1,0,0,100,[2000,200,0,0],[1000,100,0,0],[500,100,0,0],
23                                             [1050,710,3,0,1,0,0,100,[2000,200,0,0],[1000,100,0,0],[500,100,0,0],
24                                             [1050,710,3,1,1,0,0,100,[2000,200,0,0],[1000,100,0,0],[500,100,0,0],
25                                             [1050,710,3,0,1,0,0,100,[2000,200,0,0],[1000,100,0,0],[500,100,0,0],
26                                             [1050,710,3,1,1,0,0,100,[2000,200,0,0],[1000,100,0,0],[500,100,0,0],
27
28     ! Declara??o de velocidades para recolha de caixa na mesa para plano impar
29     CONST V_rec_cx V_rec_cx1:=[ [1000,200,50,50],[500,300,50,50]];
30
31     ! Declara??o de velocidades para recolha de caixa na mesa para plano par
32     CONST V_rec_cx V_rec_cx2:=[ [1000,200,50,50],[500,300,50,50]];
33
34     ! Declara??o de velocidades para entrega de caixa na paletiza??o do plano impar
35     CONST V_ent_cx V_ent_cx1:=[ [500,100,50,50],[200,300,50,50]];
36
37     ! Declara??o de velocidades para entrega de caixa na paletiza??o do plano par
38     CONST V_ent_cx V_ent_cx2:=[ [500,100,50,50],[200,300,50,50]];
39

```

Figura 69 - Exemplo da estrutura de uma receita.

Na Figura 69 apresenta-se uma parte da estrutura da receita número seis. Pode-se visualizar algumas declarações de variáveis e a estrutura de dados que contém as variáveis. Sempre que o robô necessita de saber, por exemplo, a dimensão da caixa, é nas variáveis deste ficheiro que vai recolher essa informação.

Sempre que o robô carrega uma nova receita, ela é verificada para garantir que quando se cria ou altera uma receita os valores de cada variável estão dentro da gama de valores corretos, evitando possíveis falhas nos dados introduzidos no robô.

A reinicialização da paletização é outra tarefa do robô existente no programa 'MAIN'. Após o robô receber o sinal de que existe uma tarefa para executar, e nenhuma das condições explicadas anteriormente se verificarem, a condição seguinte a ser verificada é a necessidade da paletização ser reinicializada. Para tal, o sinal 'EN_DI11_INIT_PAL' tem de estar a "1" (Figura 70).

Após a verificação da condição anterior, é executado o subprograma 'init_pal', o qual é responsável por reinicializar as variáveis necessárias para que a paletização seja corretamente reinicializada bem como a expulsão de alguma paleta que esteja na mesa de paletização. Sempre que é reinicializada a paletização em curso, é enviado um sinal de confirmação ao PLC, 'EN_DO09_INI_ACK', informando que o robô terminou a reinicialização da paletização. Desta forma, o robô fica preparado para começar uma nova paletização.

```
106      ELSEIF EN_DI11_INIT_PAL=1 THEN
107          ! reinicializar paletização
108          init_pal;
109          PulseDO\PLength:=2,EN_DO09_INI_PAL_ACK;
```

Figura 70 - Opção do robô para reinicializar a paletização.

Na Figura 71 encontra-se representado na íntegra o subprograma para reinicializar a paletização. Observe-se que, a partir da linha 14 até à linha 44, o programa inicializa variáveis que estão associadas ao ponto atual em que se encontrava a paletização em curso, nomeadamente: o número da entrega da caixa(idx_cx); o número do plano(idx_plano); a altura atual a que a paleta se encontra(G_alt_paltizcao); as variáveis do tipo *flags*(bandeiras) que servem para controlo do programa. Na linha 45 é enviado o sinal 'EN_DO13_Paleta_concluída', que tem como objetivo informar o PLC que a paleta pode ser expulsa, ou seja, transportada para a mesa seguinte. Na linha 46, o robô fica à espera de uma confirmação de que o PLC recebeu o sinal de paleta concluída. Se o robô ficar à espera dessa confirmação por mais de 30 segundos, envia novamente o sinal de paleta concluída.

```

14  PROC init_pal()
15      ClkStop clk_ciclo_total;
16      smk_clk_ciclo_total:=ClkRead(clk_ciclo_total);
17      PDispOff;
18      Confl\On;
19      ! INICIALIZAÇÃO DE VARIÁVEIS DE FLUXO DO PROGRAMA
20      idx_cx:=1;
21      idx_plano:=1;
22      ! INICIALIZAÇÃO DE VARIÁVEIS DE PALETIZAÇÃO
23      G_alt_paltizcao:=0;
24      ! Flags de controlo das paletes
25      flag_entrega_P1_ok:=FALSE;
26      flag_entrega_P2_ok:=FALSE;
27      flag_cancel_r_PL:=FALSE;
28      ! Flags de controlo dos intercalares
29      flag_entrega_inter_ok:=FALSE;
30      flag_cancel_r_C:=FALSE;
31      ! Flags de controlo paletização
32      flag_entrega_cx_ok:=FALSE;
33      flag_fim_Pal:=FALSE;
34      ! Flags de controlo de dados
35      Flag_data_error:=FALSE;
36      ! Flags de controlo da ferramenta
37      Flag_f_Ab_gripper:=FALSE;
38      Flag_f_fc_gripper:=FALSE;
39      Flag_f_lg_vacu:=FALSE;
40      Flag_f_dl_vacu:=FALSE;
41      Flag_f_depress:=FALSE;
42      Flag_f_pg_caixa:=FALSE;
43      TRY_PALETE_CONCLUIDA:
44      Palete_concluida:=TRUE;
45      SetDO EN_D013_Palete_concluida,1;
46      WaitDI EN_DI12_Palete_concluida_ACK,1\MaxTime:=30\TimeFlag:=flag_palete_concluida;
47      IF flag_palete_concluida THEN
48          TPWrite "A espera de confirmação da saída de palete";
49          GOTO TRY_PALETE_CONCLUIDA;
50      ENDIF
51      ClkReset clk_ciclo_total;
52      ClkStart clk_ciclo_total;
53  ENDPROC

```

Figura 71 - Subprograma para reinicializar a paletização.

Após verificar todas as condições anteriores e nenhuma delas se encontrar ativa, o controlador efetua várias condições para poder paletizar. Na Figura 72 apresenta-se o código utilizado para efetuar essas mesmas condições: na linha 112 são verificados todos os dados da receita, executando o subprograma ‘Check_receita’; na linha 114, verifica-se se a receita existente é uma receita nula e, se for o caso, termina o programa ‘MAIN’, dando início a um novo ciclo do robô: na linha 115, verifica-se se a mesa de paletização está livre para paletizar; na linha 127, verifica-se se a mesa de paletização não está preparada para começar a paletização. Só depois de todas estas verificações é que o robô vai paletizar, executando o subprograma ‘Prog_1’.

```

110 ELSE
111     ! Verifica dados da receita
112     Check_receita;
113     ! verifica se a receita é nula
114     IF produto="NULL" GOTO FIM;
115     IF EN_DI14_M_Func_Pal_Base=0 THEN
116         ! Verificação de tapete livre para iniciar paltização
117         IF idx_cx=1 AND idx_plano=1 AND (NOT flag_entrega_P1_ok) AND EN_DI50_ZONA_PAL_LIVRE=0 THEN
118             !Zona de paletização ocupada, termina programa
119             Flag_wait_tapete_livre:=FALSE;
120             alarm_msg alarm16;
121             WaitDI EN_DI50_ZONA_PAL_LIVRE,1\MaxTime:=20\TimeFlag:=Flag_wait_tapete_livre;
122             IF Flag_wait_tapete_livre THEN
123                 Flag_wait_tapete_livre:=FALSE;
124                 GOTO FIM;
125             ENDIF
126         ENDIF
127     ELSEIF EN_DI14_M_Func_Pal_Base=1 THEN
128         ! Verificação altura de iniciar paltização
129         IF idx_cx=1 AND idx_plano=1 AND (NOT flag_entrega_P1_ok) THEN
130             G_alt_paltizcao:=160;
131         ENDIF
132     IF EN_DI50_ZONA_PAL_LIVRE=0 OR EN_DI15_Pal_Base=0 THEN
133         ! Paletização com palete de madeira
134         Flag_wait_tapete_ocupado:=FALSE;
135         alarm_msg alarm18;
136         ! Espera pelo sinal de zona de palete ocupada
137         WaitDI EN_DI50_ZONA_PAL_LIVRE,0\MaxTime:=20\TimeFlag:=Flag_wait_tapete_ocupado;
138         IF Flag_wait_tapete_ocupado THEN
139             Flag_wait_tapete_ocupado:=FALSE;
140             GOTO FIM;
141         ENDIF
142         ! Espera pelo sinal de presença de palete de madeira
143         WaitDI EN_DI15_Pal_Base,1\MaxTime:=20\TimeFlag:=Flag_wait_tapete_ocupado;
144         IF Flag_wait_tapete_ocupado THEN
145             Flag_wait_tapete_ocupado:=FALSE;
146             GOTO FIM;
147         ENDIF
148     ENDIF
149 ENDIF
150 Prog_1;
151 ENDIF

```

Figura 72 - Verificações antes do robô começar a paletizar.

O subprograma ‘Prog_1’ é executado quando todas as condições anteriormente descritas se verificarem, ou seja, estando o robô na posição ‘Home’ e tendo uma tarefa para executar que não seja ir para a posição de manutenção ou de segurança, trocar a receita ou reinicializar a paletização. Caso as condições anteriormente descritas não sejam verdadeiras, o robô passa a verificar se a mesa de paletização está disponível para paletizar e só depois é que dá início à paletização.

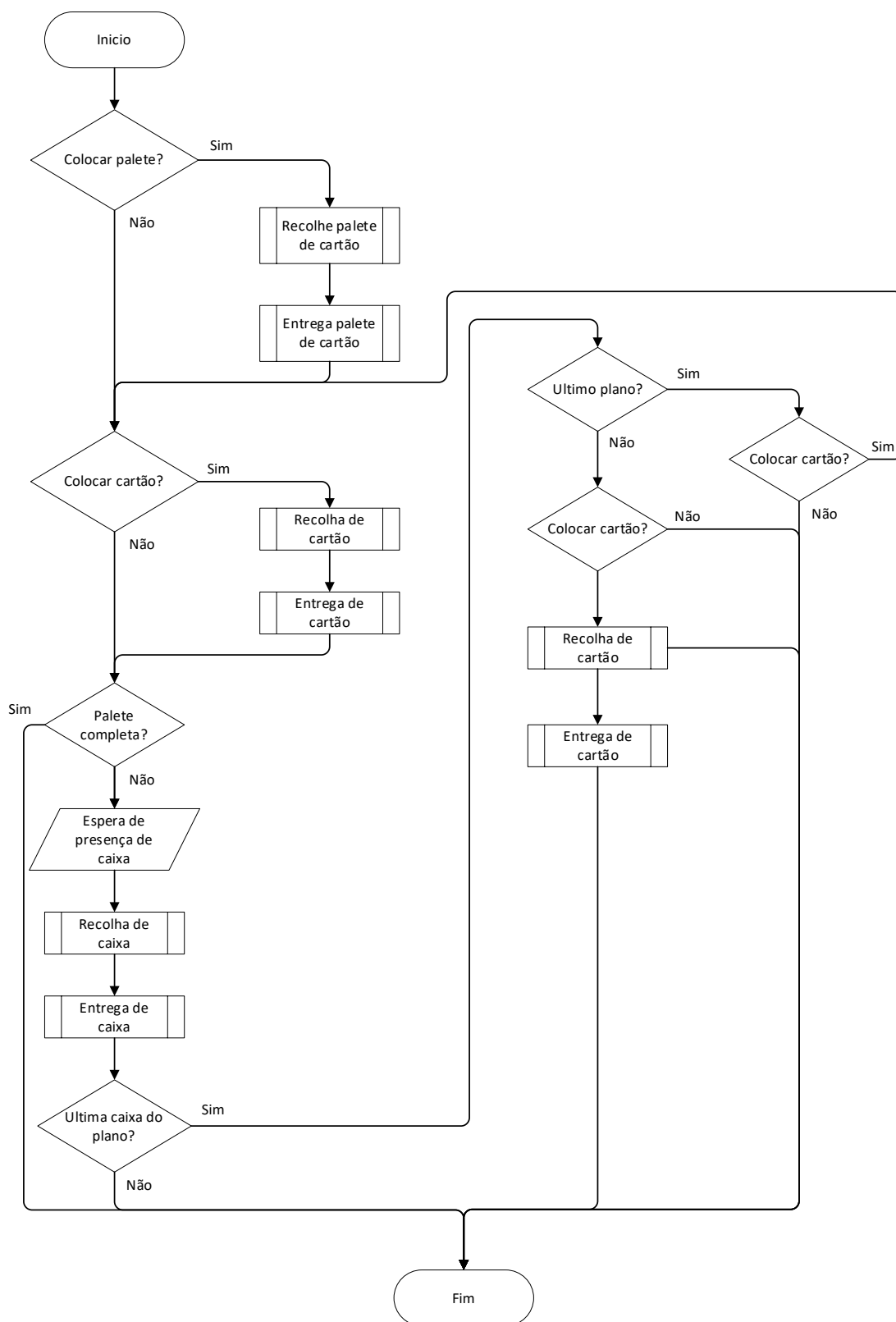


Figura 73 - Fluxograma geral do subprograma 'Prog_1'

Devido à complexidade do subprograma ‘Prog_1’, aborda-se as tarefas executadas apenas de uma forma geral. Na Figura 73 está uma representação simples do fluxograma do programa ‘Prog_1’.

Uma das primeiras condições a serem verificadas é a necessidade de colocação da palete de cartão. Nesta condição, verifica-se se a paletização se encontra no início e, se a receita ordena a colocação de uma paleta de cartão, o robô recolhe a paleta no armazém de paletes e entrega-a na mesa de paletização. Depois, é verificado se há a necessidade de colocar um cartão e se está no início de um plano/nível da paletização. Em caso afirmativo, o robô vai recolher o cartão ao armazém de cartões e entrega-o em cima da paleta ou em cima de cada plano. De seguida, fica à espera de uma caixa na mesa de entrega. Quando tiver a confirmação da presença da caixa, executa o subprograma ‘recolhe_cx’, que tem como objetivo recolher a caixa que está na mesa. Terminada a recolha da caixa, executa o subprograma ‘entrega_cx’, que tem como objetivo entregar a caixa na paleta. Terminada a entrega da caixa na paleta, o robô verifica se é a última caixa do plano. Se não for, termina o subprograma ‘Prog_1’ e por sua vez termina o programa ‘MAIN’ e o robô inicia novo ciclo.

No caso de ser a última caixa, o robô verifica se é o último plano. Se não for, verifica se é para colocar um cartão. Em caso afirmativo, recolhe o cartão no armazém de cartões, executando o subprograma ‘recolha_inter’ e depois entrega o cartão na paleta, executando o subprograma ‘entrega_inter’. Se não for para colocar cartão, o robô dá a paleta como concluída. Se for a última caixa e o último plano, existe a possibilidade de colocar um último cartão em cima da paleta e só depois é que a dá como concluída.

Sempre que a paleta é dada como concluída, o *PLC* avança com a paleta para a próxima mesa, ficando a mesa de paletização livre para dar início a uma nova paletização.

```

171 PROC Prog_1()
172     ! início da contagem de ciclo
173     ClkReset clk_ciclo;
174     ClkStart clk_ciclo;
175     ! Ativação de interrupções
176     Active_interrupts;
177     IF idx_cx>data_paletizacao.n_caixa_plano GOTO ENTREGA_OK;
178     PDispOff;
179     ! Verificação da necessidade de colocação de paletes
180     IF...ENDIF
226     COLOCA_INTER:
227     ! Necessidade de colocação de intercalar
228     IF...ENDIF
247     ! Termina Programa
248     IF...ENDIF
252     ! executa programa de paletização
253     WaitDI EN_DI17_PRESENCA_CAIXA,1\MaxTime:=10\TimeFlag:=Flag_f_not_caixa;
254     IF...ENDIF
258     ! Recolha de caixa na mesa de entrada
259     recolhe_cx idx_cx,idx_plano;
260     IF Flag_f_pg_caixa GOTO FIM;
261     ! Entrega de caixa na paleta
262     entrega_cx idx_cx;
263     ENTREGA_OK:
264     ! Verifica se é a ultima caixa do plano
265     IF...ENDIF
328     FIM:
329     ! Se estiver na posição home na paleta
330     IF...ENDIF
333     ! Expulsão de paleta
334     ! Inicializa paletização
335     ! Fim da contagem de ciclo
336     ClkStop clk_ciclo;
337     ClkStop clk_caixa;
338     ClkStop clk_intercalar;
339     ClkStop clk_paleta_cartao;
340     smk_clk_ciclo:=ClkRead(clk_ciclo);
341     smk_clk_caixa:=ClkRead(clk_caixa);
342     smk_clk_intercalar:=ClkRead(clk_intercalar);
343     smk_clk_paleta_cartao:=ClkRead(clk_paleta_cartao);
344     IF Flag_fim_Pal init_pal;
345     ENDPROC

```

Figura 74 - Subprograma 'Prog_1'.

Na Figura 74 está representada a parte do código do subprograma 'Prog_1' com a qual se explica, de uma forma simplista, o fluxo do código. Na linha 177 é verificado se a posição de entrega é superior ao número de entregas por plano. Caso tal se verifique, o código salta para a linha 263. Desta forma não recolhe nenhuma caixa, uma vez que o plano já está completo. Na linha 228 do código, é verificado se é necessário colocar um cartão. Esta condição só se verifica se for o início de um plano e se na receita estiver ativa a colocação do cartão nesse plano. Já na linha 248 é verificado se a contagem do plano é superior ao número de planos da paleta. Se esta condição se verificar, a paleta é dada como concluída. Após todas estas verificações, o programa fica à espera do sinal

‘EN_DI17_PRESENCA_CAIXA’, que significa a presença de uma caixa na mesa de entrega. Quando tal se verificar, o robô executa o subprograma ‘recolha_cx’, que tem como parâmetros de entrada o número da entrega e do plano atual. Logo a seguir à recolha, na linha 262 é executado o subprograma ‘entrega_cx’, que tem como parâmetro de entrada o número da entrega atual. Terminada a entrega na linha 265, é verificado se é a última entrega do plano, se é para colocar cartão e se é o último plano. Por fim, na linha 330 é verificado se o robô está na posição ‘home’. Se não estiver, efetua o movimento até essa posição e termina o programa, ficando assim preparado para um novo ciclo.

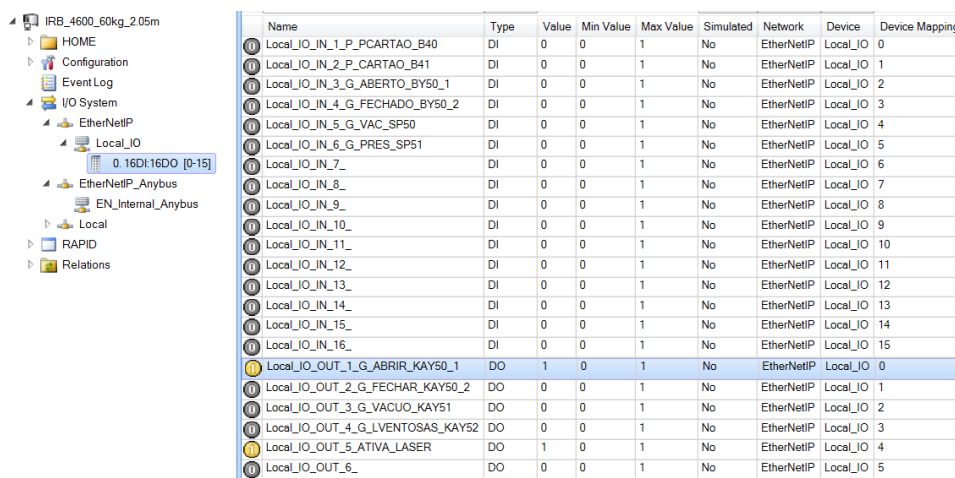
4.1.8 Ensaios e comissionamento

Nesta secção aborda-se, de uma forma geral, os ensaios efetuados com o robô já inserido na ilha com os demais equipamentos e o comissionamento da ilha robotizada ao cliente. Os ensaios têm como objetivo testar toda a programação efetuada no robô e verificar se todos os equipamentos estão a funcionar de acordo com o seu objetivo. No caso de ser identificado algum problema, em conjunto com as equipas envolvidas no projeto, nomeadamente a equipa de técnicos de instalação da ilha, a equipa de automação e a equipa de desenvolvimento de projetos, deverá encontrar-se a solução que melhor se adequa à resolução do problema encontrado. O comissionamento, não é mais que a instalação da ilha no cliente e a sua validação, ou seja, vai decorrendo à medida que se efetuam os ensaios e se vai confirmando o correto funcionamento da instalação, de acordo com o objetivo pretendido.

Nesta fase do projeto foi testado todo o programa no controlador real e feita a instalação do robô na ilha robotizada. Foram testadas todas as entradas e saídas locais do módulo instalado no controlador do robô, explicado na secção 4.1.6.1. De seguida foi testada toda a informação trocada com o *PLC* da ilha, também explicado na secção anteriormente referida. Por fim foi calibrada a ferramenta instalada no robô, e passou-se à calibração das bases de trabalho correspondentes à paletização, da mesa de entrega de caixas, do armazém de paletes e do armazém de cartões.

Na Figura 75 apresenta-se uma imagem do *software RobotStudio*, através do qual se podem visualizar todos os sinais das entradas e saídas do módulo local. Desta forma,

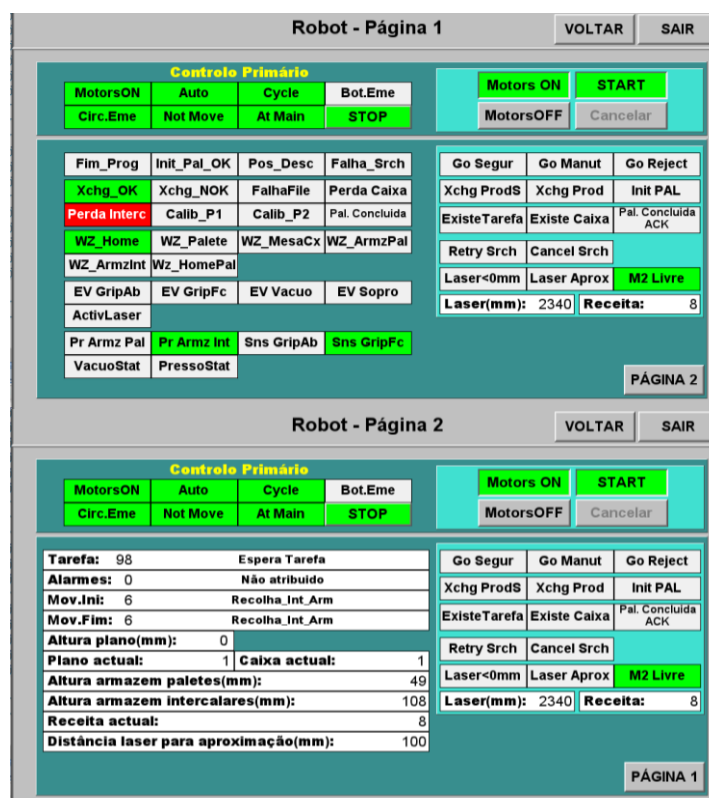
pode-se verificar se o robô está a receber todos os sinais referentes à parte sensorial aqui ligada. Através deste *software* também se pode efetuar a ativação das saídas, para ver se todos os atuadores estão a funcionar corretamente.



Name	Type	Value	Min Value	Max Value	Simulated	Network	Device	Device Mapping
Local_IO_IN_1_P_CARTAO_B40	DI	0	0	1	No	EtherNetIP	Local_IO	0
Local_IO_IN_2_P_CARTAO_B41	DI	0	0	1	No	EtherNetIP	Local_IO	1
Local_IO_IN_3_G_ABERTO_BY50_1	DI	0	0	1	No	EtherNetIP	Local_IO	2
Local_IO_IN_4_G_FECHADO_BY50_2	DI	0	0	1	No	EtherNetIP	Local_IO	3
Local_IO_IN_5_G_VAC_SP50	DI	0	0	1	No	EtherNetIP	Local_IO	4
Local_IO_IN_6_G_PRES_SP51	DI	0	0	1	No	EtherNetIP	Local_IO	5
Local_IO_IN_7_	DI	0	0	1	No	EtherNetIP	Local_IO	6
Local_IO_IN_8_	DI	0	0	1	No	EtherNetIP	Local_IO	7
Local_IO_IN_9_	DI	0	0	1	No	EtherNetIP	Local_IO	8
Local_IO_IN_10_	DI	0	0	1	No	EtherNetIP	Local_IO	9
Local_IO_IN_11_	DI	0	0	1	No	EtherNetIP	Local_IO	10
Local_IO_IN_12_	DI	0	0	1	No	EtherNetIP	Local_IO	11
Local_IO_IN_13_	DI	0	0	1	No	EtherNetIP	Local_IO	12
Local_IO_IN_14_	DI	0	0	1	No	EtherNetIP	Local_IO	13
Local_IO_IN_15_	DI	0	0	1	No	EtherNetIP	Local_IO	14
Local_IO_IN_16_	DI	0	0	1	No	EtherNetIP	Local_IO	15
Local_IO_OUT_1_G_ABRIR_KAY50_1	DO	1	0	1	No	EtherNetIP	Local_IO	0
Local_IO_OUT_2_G_FECHAR_KAY50_2	DO	0	0	1	No	EtherNetIP	Local_IO	1
Local_IO_OUT_3_G_VACUO_KAY51	DO	0	0	1	No	EtherNetIP	Local_IO	2
Local_IO_OUT_4_G_LVENTOSAS_KAY52	DO	0	0	1	No	EtherNetIP	Local_IO	3
Local_IO_OUT_5_ATIVA_LASER	DO	1	0	1	No	EtherNetIP	Local_IO	4
Local_IO_OUT_6_	DO	0	0	1	No	EtherNetIP	Local_IO	5

Figura 75 - Verificação de sinais do módulo local de entradas e saídas.

Também se pode verificar toda a comunicação com o *PLC*. Para isso, existem duas páginas (Figura 75) no *HMI* da ilha dedicadas à informação trocada (recebida e enviada) entre o *PLC* e o robô.



Robot - Página 1

Controlo Primário

MotorsON Auto Cycle Bot.Eme
Circ.Eme Not Move At Main STOP

Motors ON START
MotorsOFF Cancelar

Fim_Prog Init_Pal_OK Pos_Desc Falha_Srch
Xchg_OK Xchg_NOK FalhaFile Perda Caixa
Perda Interc Calib_P1 Calib_P2 Pal. Concluida
WZ_Home WZ_Palette WZ_MesaCx WZ_ArmzPal
WZ_ArmzInt WZ_HomePal
EV GripAb EV GripFc EV Vacuo EV Sopro
ActivLaser
Pr Armz Pal Pr Armz Int Sns GripAb Sns GripFc
VacuoStat PressoStat

Go Segur Go Manut Go Reject
Xchg Prods Xchg Prod Init PAL
ExisteTarefa Existe Caixa Pal. Concluida ACK
Retry Srch Cancel Srch
Laser<0mm Laser Aprox M2 Livre
Laser(mm): 2340 Receita: 8

PÁGINA 2

Robot - Página 2

Controlo Primário

MotorsON Auto Cycle Bot.Eme
Circ.Eme Not Move At Main STOP

Motors ON START
MotorsOFF Cancelar

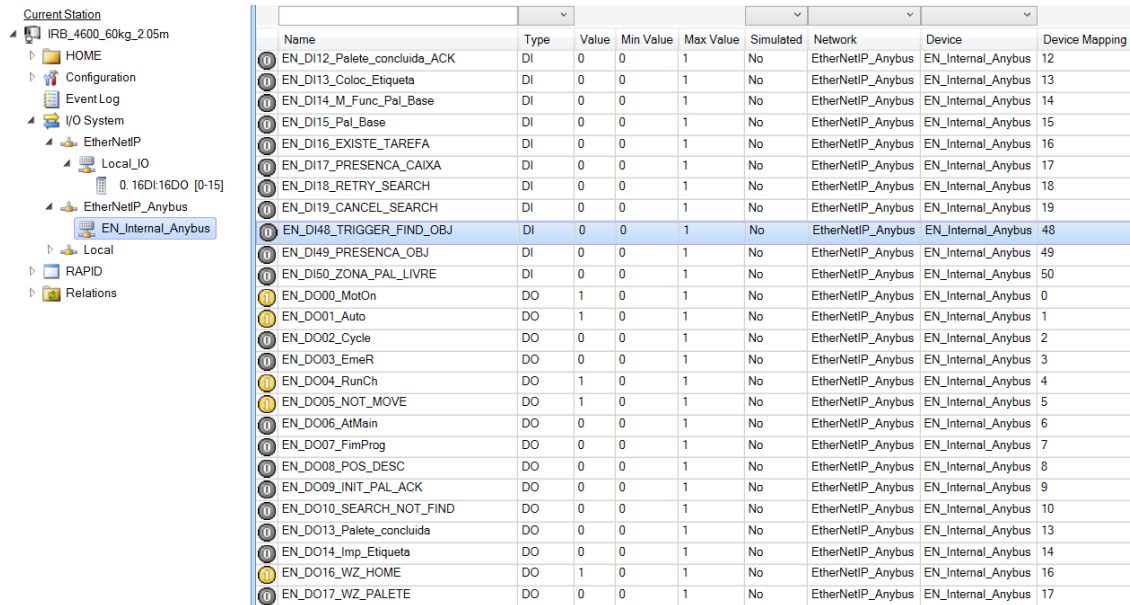
Tarefa: 98 Espera Tarefa
Alarmes: 0 Não atribuido
Mov.Ini: 6 Recolha_Int_Arm
Mov.Fim: 6 Recolha_Int_Arm
Altura plano(mm): 0
Plano actual: 1 Caixa actual: 1
Altura armazem paletes(mm): 49
Altura armazem intercalares(mm): 108
Receita actual: 8
Distância laser para aproximação(mm): 100

Go Segur Go Manut Go Reject
Xchg Prods Xchg Prod Init PAL
ExisteTarefa Existe Caixa Pal. Concluida ACK
Retry Srch Cancel Srch
Laser<0mm Laser Aprox M2 Livre
Laser(mm): 2340 Receita: 8

PÁGINA 1

Figura 76 - Páginas do HMI com a informação enviada ao PLC e recebida do robô.

Na Figura 77 encontra-se toda a informação trocada entre o PLC e o robô, sendo desta forma que se verifica a correspondência da informação entre os equipamentos.



Name	Type	Value	Min Value	Max Value	Simulated	Network	Device	Device Mapping
EN_DI12_Paleta_concluida_ACK	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	12
EN_DI13_Coloc_Etiqueta	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	13
EN_DI14_M_Func_Pal_Base	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	14
EN_DI15_Pal_Base	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	15
EN_DI16_EXISTE_TAREFA	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	16
EN_DI17_PRESENCIA_CAIXA	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	17
EN_DI18_RETRY_SEARCH	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	18
EN_DI19_CANCEL_SEARCH	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	19
EN_DI48_TRIGGER_FIND_OBJ	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	48
EN_DI49_PRESENCIA_OBJ	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	49
EN_DI50_ZONA_PAL_LIVRE	DI	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	50
EN_DO00_MotOn	DO	1	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	0
EN_DO01_Auto	DO	1	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	1
EN_DO02_Cycle	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	2
EN_DO03_EmerR	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	3
EN_DO04_RunCh	DO	1	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	4
EN_DO05_NOT_MOVE	DO	1	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	5
EN_DO06_AtMain	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	6
EN_DO07_FimProg	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	7
EN_DO08_POS_DESC	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	8
EN_DO09_INIT_PAL_ACK	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	9
EN_DO10_SEARCH_NOT_FIND	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	10
EN_DO13_Paleta_concluida	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	13
EN_DO14_Imp_Etiqueta	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	14
EN_DO16_WZ_HOME	DO	1	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	16
EN_DO17_WZ_PALETTE	DO	0	0	1	No	EtherNetIP_Anybus	EN_Internal_Anybus	17

Figura 77 – RobotStudio: verificação da informação trocada entre o PLC e o robô.

Uma vez terminada a verificação de todos os sinais do módulo local e da informação trocada entre o PLC e o robô, inicia-se a calibração da ferramenta, através do TCP da pinça e do TCP das ventosas. Para isso utiliza-se o método de “TCP&Z, X” para cada um dos TCP’s [11]. Este método consiste em posicionar o TCP da pinça ou das ventosas e gravar quatro ou mais posições do TCP num ponto fixo, mas com orientações diferentes (Figura 78).

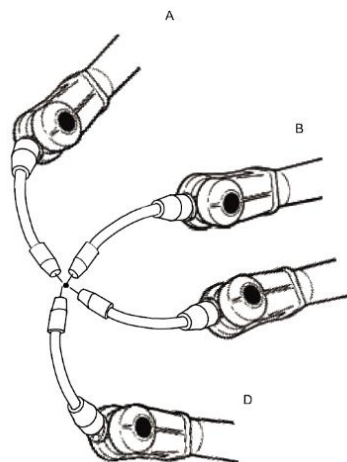


Figura 78 – Exemplo de método de calibração [11].

Desta forma o robô consegue definir o centro do *TCP*. Depois grava-se um ponto no afastamento em *Z* do *TCP* e um ponto no afastamento em *X*. Assim, o robô define a orientação do *TCP*. Este procedimento tem de ser efetuado tantas vezes quanto o número de *TCP*'s existentes: neste caso são dois, o do gripper e o das ventosas.

Depois de ter a ferramenta devidamente calibrada passa-se à calibração das bases de trabalho do robô. Para se poder efetuar uma correta calibração das bases foi colocado um parafuso na ferramenta, com o objetivo de a extremidade desse parafuso coincidir com o *TCP* das ventosas: à extremidade desse parafuso chama-se apontador. Assim, basta posicionar a ferramenta de forma que o apontador fique no ponto de origem da base a ser calibrada e depois verifica-se se existe alguma diferença entre a posição do apontador e o ponto de origem. Caso exista, procede-se à alteração da posição da base no referencial mundo do robô até estar coincidente com o apontador.

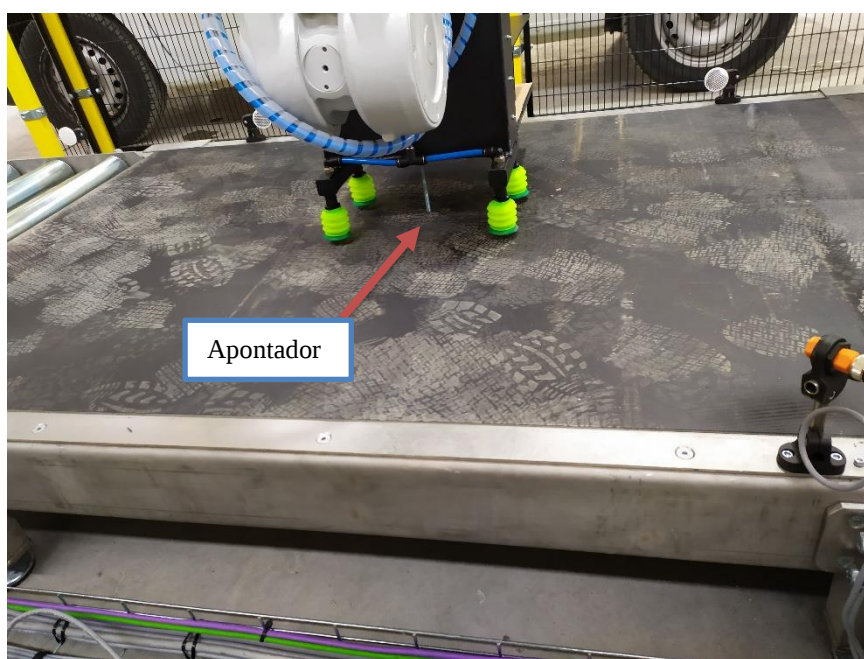


Figura 79 - Apontador posicionado no centro de origem da base de paletização.

Na Figura 79 apresenta-se o exemplo da calibração da base de trabalho da paletização, que fica no tapete de transporte da palete, como já foi referido no subcapítulo 4.1.4.

O mesmo procedimento é efetuado para as restantes bases de trabalho. Quando não é possível colocar o apontador no ponto de origem de coordenadas da base a ser calibrada, posiciona-se a ferramenta a uma distância conhecida. Desta forma efetua-se a comparação distância indicada no robô com a distância medida e no caso de os valores da posição não

estarem corretos, procede-se à correção da posição da base a ser calibrada em relação ao referencial mundo do robô.



Figura 80 – Apontador numa posição aproximada do ponto de origem da base de trabalho.

Na Figura 80 está representado o posicionamento do apontador em relação ao ponto de origem da base de trabalho do armazém de cartões. Neste caso é impossível colocar o apontador no ponto de origem da base, pois embateria com a estrutura metálica do armazém.

Após todas as bases estarem devidamente calibradas, passa-se à fase de testar e corrigir a paletização. Começa-se por testar uma das receitas, verificando se o robô efetua bem a recolha e a entrega da paleta vazia no tapete da paletização. Na Figura 81-(a), à esquerda, pode-se ver o robô a transportar uma paleta nas ventosas à saída do respetivo armazém. À direita (Figura 81-(b)) o robô deixa a paleta na mesa-tapete. Como já referido ao longo deste relatório, é nesta mesa que é feita a paletização das caixas e, após terminada, a paleta é transportada até uma máquina de envolver com filme.

Verificou-se que esta máquina não conseguia envolver filme abaixo de 130mm de altura. Uma vez que era necessário envolver toda a paleta de modo que o filme baixasse até

prender a paleta de cartão, e como esta só tinha 70mm de altura, optou-se por colocar duas paletes sobrepostas. Desta forma conseguiu-se obter 140mm de altura no início da paletização. Assim, o filme já conseguia apanhar a paleta de cartão. No final da linha, a paleta completa e envolta em filme é recolhida e armazenada e a paleta sobranete é reintroduzida no armazém de paletes do robô.

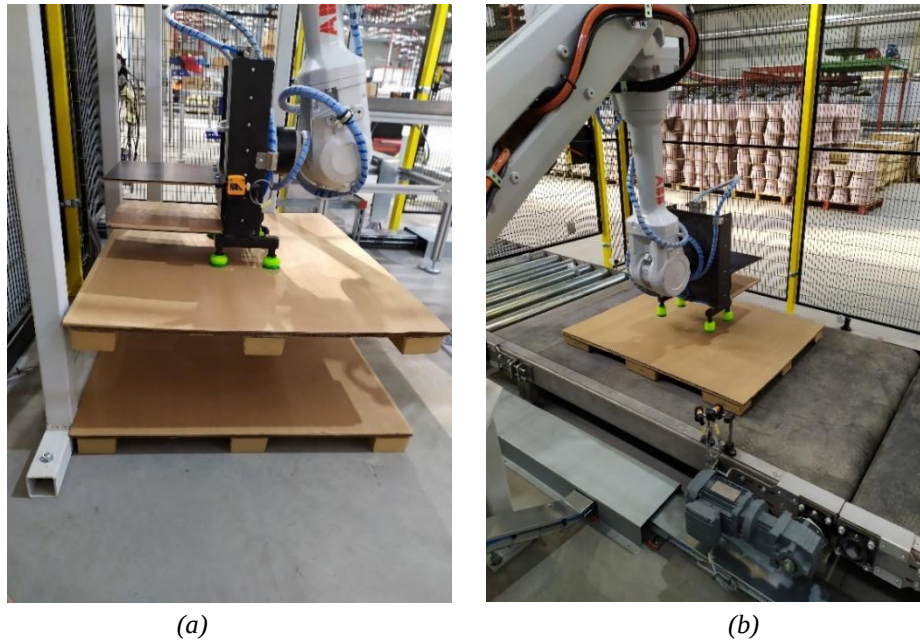
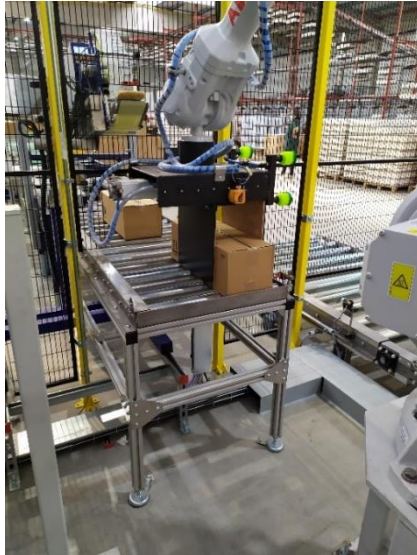
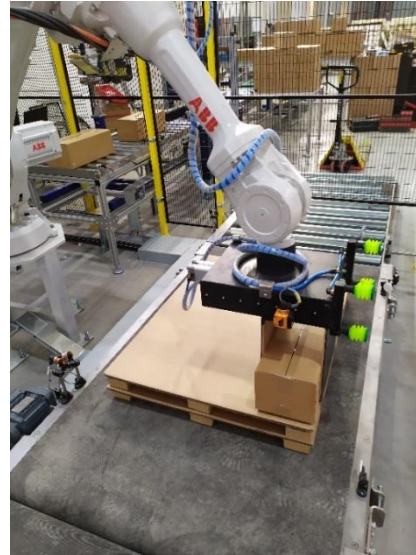


Figura 81 – À esquerda (a), o robô pega numa paleta de cartão e à direita (b) deixa a mesma paleta na mesa-tapete.

Passou-se então ao teste do processo de recolha de caixas na mesa de entrega e da sua colocação na paleta. Após alguns testes e correções, conseguiu-se que o robô pegasse na caixa corretamente (Figura 82-(a)) e que a deixasse na paleta como se pode ver na Figura 82-(b). O teste de colocação das caixas na paleta foi moroso e repetitivo, uma vez que se teve de testar todas as entregas do plano. Isto acontece porque, na realidade, as caixas têm pequenas diferenças nas dimensões e o comportamento da pinça também altera as condições de entrega (Figura 82).



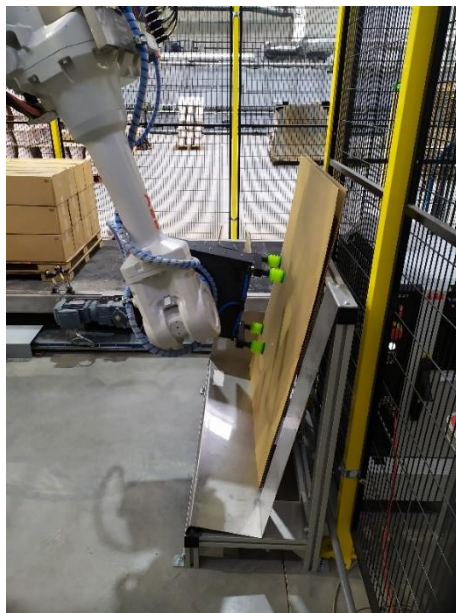
(a)



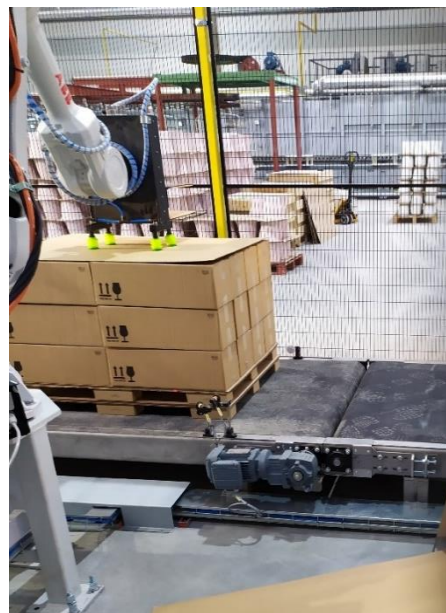
(b)

Figura 82 - À esquerda (a), o robô a pegar numa caixa. À direita (b), a entrega da caixa na paleta.

Tendo alguns planos completos, foi a vez de testar a recolha do cartão no armazém e a sua colocação acima do plano pretendido (Figura 83). Também não foi uma operação fácil uma vez que quando o robô pegava num cartão, e devido às propriedades do cartão, o vácuo que era gerado conseguia influenciar o cartão seguinte.



(a)



(b)

Figura 83 - À esquerda (a), o robô a pegar num cartão no armazém e à direita (b) a entregá-lo na paleta.

Foi necessário descer a quantidade de vácuo gerada e baixar a velocidade do robô ao recolher o cartão, para que quando o robô iniciasse a recolha houvesse tempo suficiente para que o segundo cartão se desprendesse e caísse no armazém, não alterando a sua posição.

Terminados os testes da colocação do cartão, completou-se a paleta e o robô deu-a como concluída, podendo ser enviada através dos tapetes das linhas para a envolvidora de filme (Figura 84).

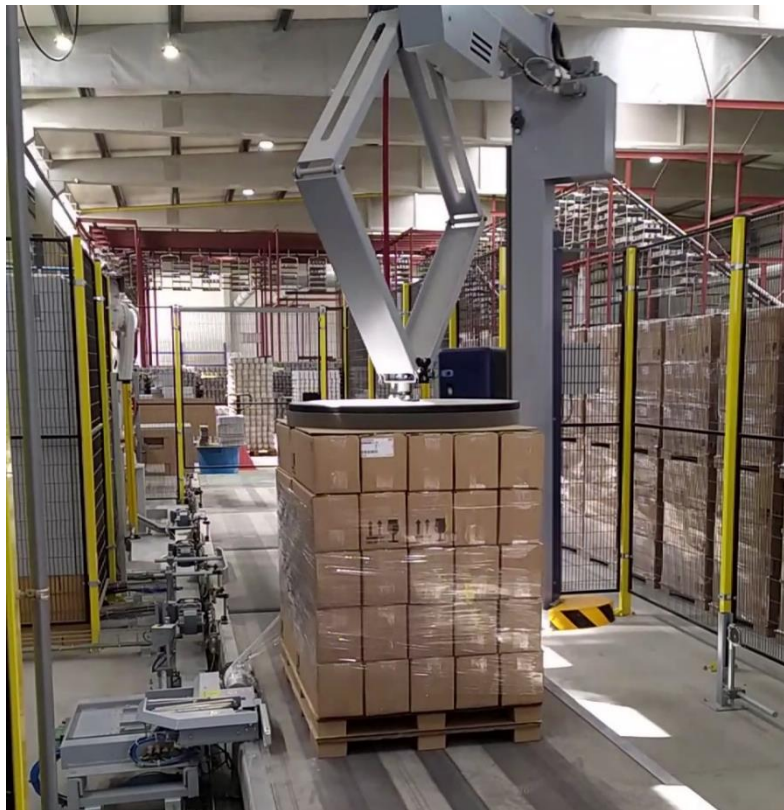


Figura 84 - Paleta na envolvidora de filme.

Por último, verificou-se a colocação de etiquetas nas caixas. Começou-se por ajustar a posição do dispensador de etiquetas. Colocando o robô na posição *KM0* da etiquetadora, afinou-se com mais precisão o dispensador de etiquetas de modo que ficasse alinhado com a parte inferior e ao centro da ferramenta. Este ponto coincide com o *TCP* da pinça que pega nas caixas (Figura 85).

Após afinado o dispensador de etiquetas testou-se a colocação de etiquetas na caixa. Foi necessário efetuar alguns ajustes da velocidade do robô ao passar no dispensador, assim como ajustar o tempo em que o robô efetuava o pedido de uma etiqueta. O dispensador

de etiquetas tinha um parâmetro de ajuste de velocidade, mas esse valor não tinha nenhuma correspondência com uma medida padrão, ou seja, foi um pouco de tentativa e erro até que se conseguisse ter uma sincronização de velocidade entre o robô e o dispensador para obter uma colocação de etiqueta correta.

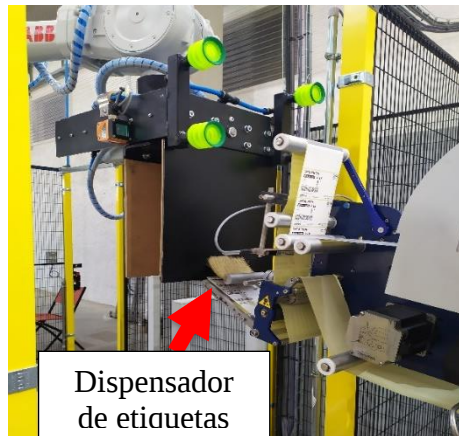


Figura 85 - Ajuste do dispensador de etiquetas: robô na posição KM0 da etiquetadora.

Foi também identificado um problema com o tempo de atraso entre o pedido da etiqueta e o momento em que era dispensada. Foi necessário o robô antecipar o pedido da etiqueta de forma que fosse dispensada na posição correta na caixa. Este ajuste também foi efetuado por tentativa e erro.

No final de todos os ajustes efetuados, conseguiu-se obter uma colocação correta da etiqueta na caixa (Figura 86).

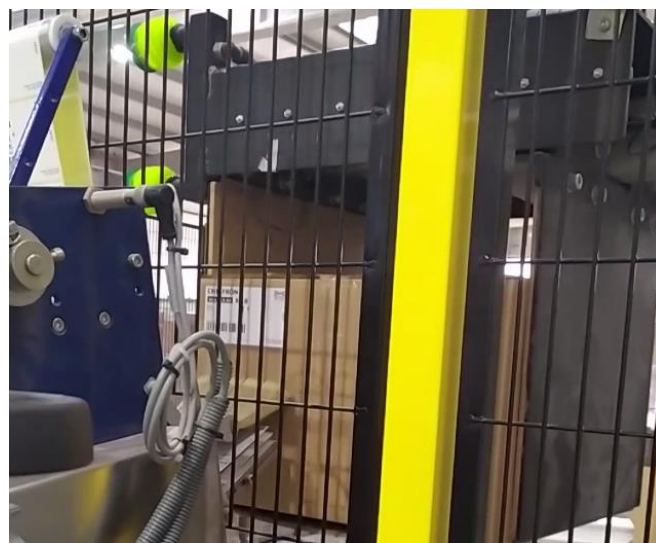


Figura 86 - Colocação de etiqueta na caixa.

Só depois de concluídos todos os testes e ajustes numa primeira receita é que foram criadas as restantes. Cada receita diz respeito a uma caixa, palete e cartão de diferentes dimensões. Também a orientação das caixas na paleta e a posição da etiqueta na caixa podem variar de receita para receita. Todos estes parâmetros são introduzidos na receita com o objetivo de o robô conseguir efetuar a recolha das caixas, a colocação da etiqueta, a paletização com as posições e orientações pretendidas e a colocação de cartões nos respetivos planos. Todas as receitas foram testadas individualmente, com o objetivo de ajustar as posições de recolha e de entrega de cada caixa na paleta e a posição da colocação da etiqueta na caixa.



Figura 87 - Exemplo de quatro paletizações diferentes.

Na Figura 87 encontra-se quatro paletes completas com caixas distintas. Como se pode observar, as caixas têm dimensões diferentes, bem como as paletes de cartão.

Estando todas as receitas parametrizadas e testadas, deu-se este trabalho por concluído. Foi realizado em 232 horas, das quais 144 foram destinadas ao desenvolvimento do programa para o robô e as restantes 88 a efetuar testes e comissionamento.

4.2 Programação de robô para colocação de etiquetas na paleta

A maioria das empresas que armazenam os seus produtos em paletes têm a necessidade de identificar cada um dos produtos armazenados. Com esta identificação torna-se mais fácil criar um rastreio de cada produto armazenado.

Neste projeto foi desenvolvida uma aplicação para colocação de etiquetas em paletes com garrafas de vidro. Uma vez que os métodos de configuração e programação do robô são muito semelhantes ao projeto anteriormente exposto, não se entrará em grandes detalhes e será apenas explicado em que consiste a aplicação, apresentando-se o resultado final.

4.2.1 Objetivo

O cliente que propôs este projeto produz garrafas de vidro e coloca-as empilhadas em paletes de madeira. Após as paletes estarem completas é colocada uma etiqueta, nas faces laterais da paleta, que contém informação que identifica o produto paletizado. Estas paletes podem ter uma altura que varia entre um metro e dois metros e quarenta centímetros. Normalmente, estas etiquetas são colocadas por um operador, o que implica que ele tenha de subir a uma plataforma ou escada para poder colocar a etiqueta na paleta, apresentando-se aqui um problema de segurança que convém evitar.

O objetivo deste trabalho consistiu na instalação de um robô que coloque uma etiqueta nas faces laterais da paleta. Para isso o robô tem de efetuar a recolha de uma etiqueta num tabuleiro, aplicar cola na etiqueta e depois colocá-la na paleta. Este robô tem a capacidade de poder colocar etiquetas nas quatro faces laterais da paleta. As etiquetas podem ser de formato A4 ou A5, podendo ser colocadas na paleta com orientação vertical ou horizontal, impressas por duas impressoras destinadas a esse efeito. As impressoras estão equipadas com dois tabuleiros cada, um para etiquetas normais e outro para etiquetas com uma *RF TAG* (etiqueta de radiofrequência). Para além da impressão na etiqueta existe a possibilidade da gravação de um código de barras numa *RF TAG*. Além das etiquetas anteriormente descritas, ainda existem três tabuleiros alimentados manualmente por um

operador com etiquetas já impressas. Estas etiquetas têm uma dimensão de aproximadamente dez centímetros de largura por cinco centímetros de altura, são coloridas e não têm qualquer informação do produto.

4.2.2 Layout

O *layout* do sistema consiste num robô IRB2600 da ABB, uma impressora para formato A4, uma impressora para formato A5, três tabuleiros pequenos, uma pistola para aplicar cola nas etiquetas, dois transportadores de paletes, uma mesa rotativa e um calcador, mecanismo para pressionar a paleta, enquanto o robô está a executar o processo de etiquetagem como se representa na Figura 88.

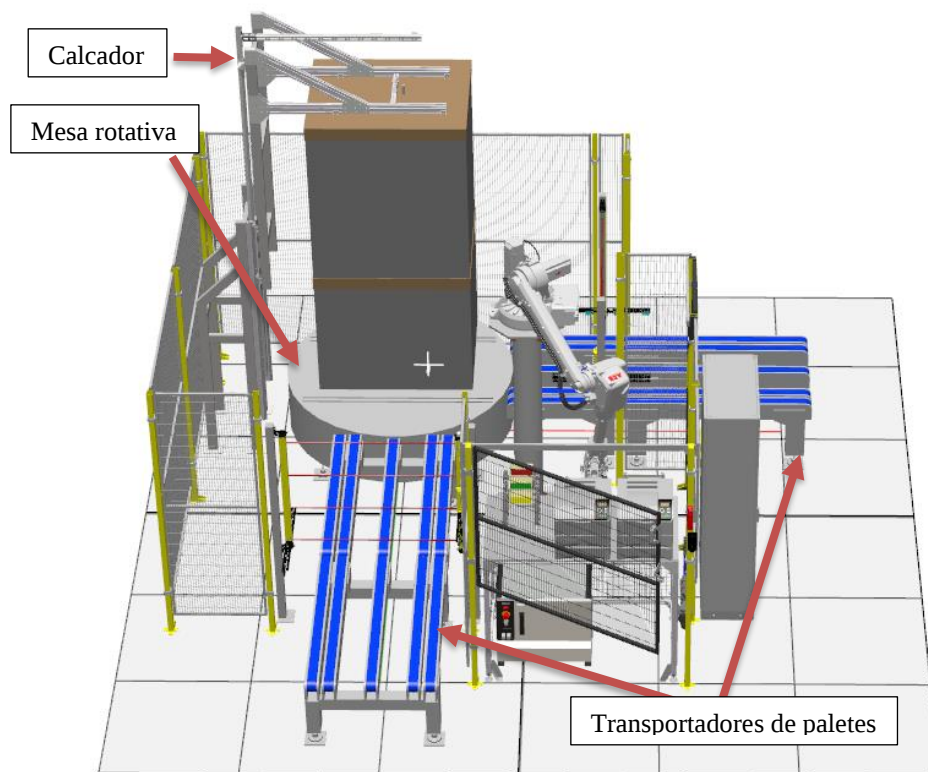


Figura 88 - Layout da implantação do robô na linha de etiquetagem.

Nesta Figura 88 podemos observar o calcador, que tem como objetivo pressionar um cartão que vem em cima da paleta, no sentido de cima para baixo, e segurar as garrafas quando o robô for colocar a etiqueta. De outra forma, como a paleta é muito instável, poderia suceder a queda de garrafas aquando da aplicação da etiqueta. A mesa rotativa

tem como objetivo orientar a paleta para o próximo transportador e também para que o robô consiga colocar etiquetas nas quatro faces da paleta. Os transportadores têm a finalidade de transportar a paleta para a mesa rotativa e dar seguimento à paleta no final, já devidamente etiquetada.

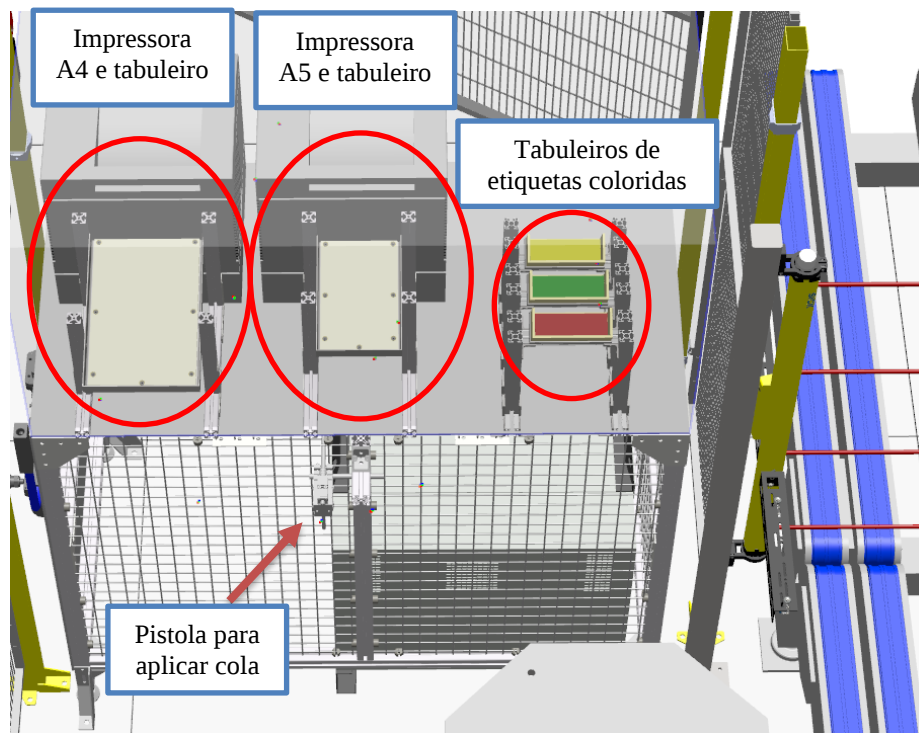


Figura 89 - Mesa das impressoras, tabuleiros e pistola de aplicação da cola.

Na Figura 89 está representada a mesa onde estão as impressoras, os tabuleiros e a pistola para aplicação da cola nas etiquetas. Dependendo da necessidade de cada paleta a ser etiquetada, o robô pode ir recolher uma etiqueta no tabuleiro para folhas A4, no tabuleiro para folhas A5, ou num dos três tabuleiros das etiquetas coloridas. Após o robô pegar numa das etiquetas, passa com ela em frente da pistola para aplicar cola. Neste *layout* ainda existe um tabuleiro adicional, o qual é chamado de rejeição e que não está aqui representado porque foi adicionado quando a ilha estava a ser instalada. Este tabuleiro tem como objetivo recolher etiquetas que poderão ser rejeitadas, ou porque não estão bem impressas, ou mesmo por o código de barras incorreto. As impressoras, quando recebem uma ordem de impressão, vão buscar os dados a uma zona de memória partilhada por todas as impressoras do cliente. Pode-se dar o caso de essa zona de memória não estar atualizada, podendo ser impressa uma etiqueta não correspondente à pretendida. Por esta razão, é feita uma inspeção do código de barras antes da etiqueta ser recolhida no

tabuleiro. Caso não seja o código de barras correto, a etiqueta é recolhida pelo robô e colocada no tabuleiro de rejeição.

4.2.3 Ferramenta

O robô foi equipado com uma ferramenta de trabalho capaz de recolher e transportar uma etiqueta por meio de ventosas e de efetuar a inspeção a um código de barras, utilizando uma camara fotográfica.

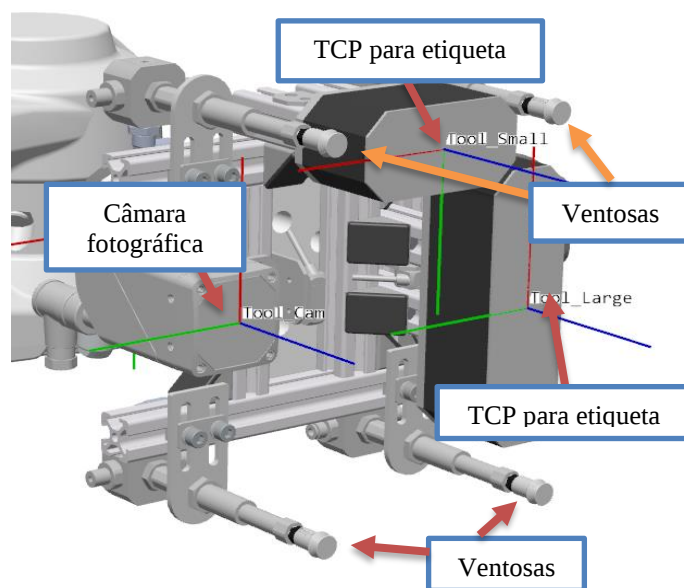


Figura 90 - Ferramenta para pegar, fotografar e colocar a etiqueta na paleta.

A ferramenta é composta por quatro ventosas para segurar a etiqueta, duas esponjas para garantir a colagem da etiqueta na paleta e uma câmara fotográfica.

Esta ferramenta vai ter três *TCP's*, um para as etiquetas em formato A4, um para o formato A5 e um para câmara fotográfica de identificação de código de barras (Figura 90). A câmara fotográfica é utilizada para identificar a etiqueta impressa no tabuleiro através do código de barras, com o intuito de evitar colar uma etiqueta que não corresponda à pretendida e verificar se a etiqueta impressa ficou colada na paleta, fazendo para isso uma nova inspeção após colar a etiqueta na paleta. Para tal, após aplicar uma etiqueta impressa, o robô posiciona a câmara de forma a conseguir identificar o código de barras da etiqueta colada na paleta. Um dos problemas que poderiam surgir era a ausência de cola numa etiqueta, o que impediria que ficasse colada na paleta. Assim, com

esta última inspeção efetuada pela câmara, garante-se que a paleta quando sai da ilha de etiquetagem tem as respetivas etiquetas. No caso de não ser detetada a etiqueta na paleta, o robô repete todo o processo da sua colocação.

Esta ferramenta está equipada com duas esponjas com a finalidade de pressionar de uma forma mais suave a etiqueta contra a paleta, garantindo que fica corretamente colada.

4.2.4 Bases de trabalho

Neste projeto foram criados dois referenciais de coordenadas, um para a mesa rotativa onde vai estar a paleta para colocar a etiqueta e outro para a mesa onde estão colocados os tabuleiros e a pistola para aplicação da cola na etiqueta.

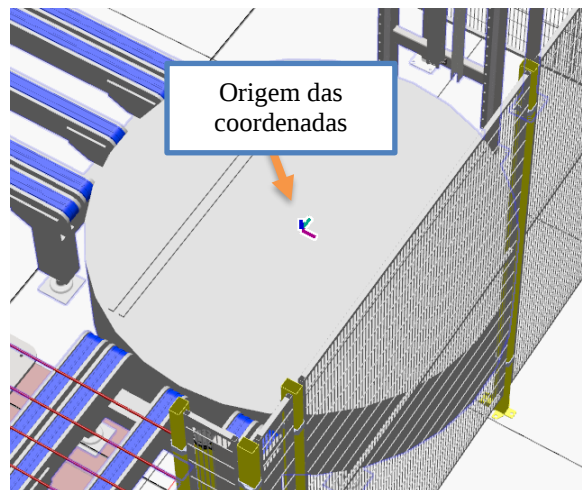


Figura 91 - Representação do sistema de coordenadas da base de trabalho de etiquetagem.

O ponto de origem do sistema de coordenadas do referencial da mesa rotativa foi posicionado no centro da mesa, sendo assim de fácil identificação e até para ser mais simples efetuar qualquer tipo de cálculo de posicionamento da etiqueta, seja qual for a orientação da paleta (Figura 91).

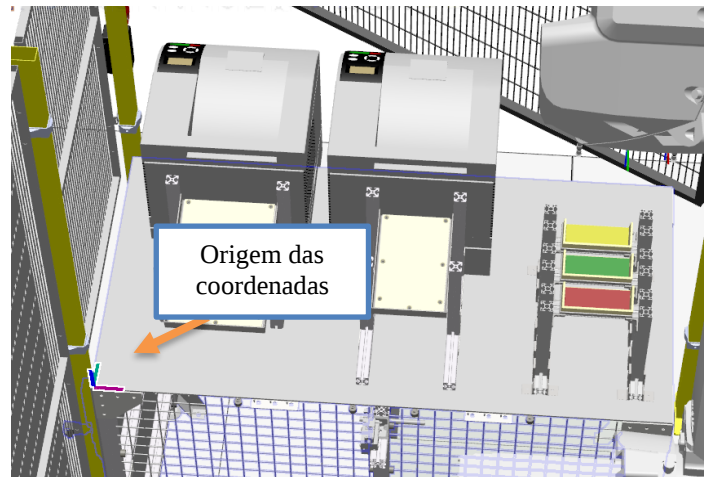


Figura 92 - Representação do sistema de coordenadas da mesa dos tabuleiros.

O ponto de origem do sistema do referencial da mesa dos tabuleiros foi posicionado num canto lateral, sendo de fácil identificação e mais intuitivo para criar qualquer ponto alvo nesse sistema de coordenadas (Figura 92).

4.2.5 Posições e trajetórias

Neste trabalho, tal como no anterior, foram criadas posições alvo. Não se aprofundará muito este tema, uma vez que não acrescenta nada de novo em relação ao trabalho anterior.

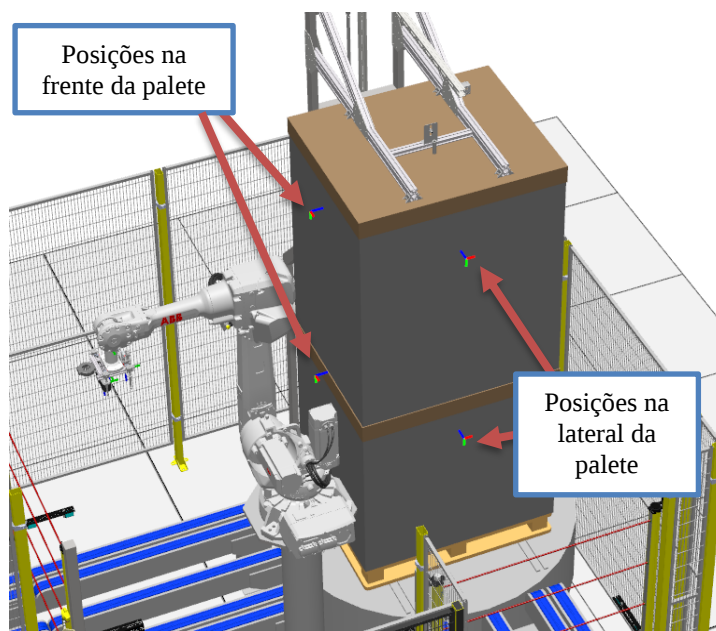


Figura 93 - Posições alvo para etiquetagem da paleta.

As principais posições alvo são para a ferramenta conseguir pegar nas etiquetas, aplicar cola nas etiquetas e aplicar as etiquetas na paleta.

Na Figura 93 estão representadas as posições alvo referentes à colocação de etiquetas na paleta. Como se pode observar, foram definidas posições na frente e na face lateral da paleta.

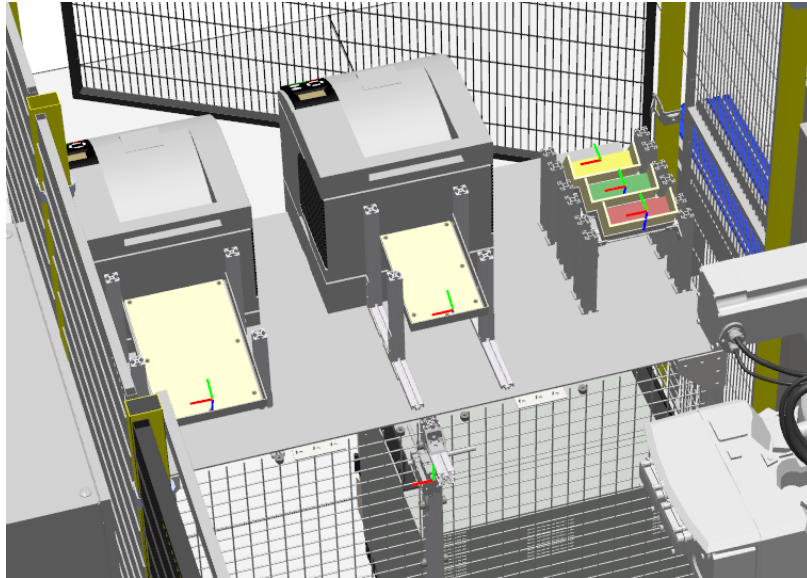


Figura 94 - Posições alvo para os tabuleiros e pistola de aplicar cola.

Desta forma é possível colocar etiquetas em duas faces da paleta e, ao rodar a mesa rotativa em 180°, é possível colocar as etiquetas nas outras duas faces da paleta.

Já na Figura 94 representam-se as posições alvo dos tabuleiros e da pistola de aplicar cola.

4.2.6 Desenvolvimento da aplicação

Como já foi descrito, este trabalho consiste em programar um robô IRB 2600 da ABB para que aplique etiquetas numa paleta com garrafas. Esta ilha robotizada está inserida numa linha de transporte de paletes com garrafas empilhadas, que podem ter origem noutras linhas de produção. Em resumo, ao chegar uma paleta à ilha robotizada, o PLC que gere a ilha recebe informação da base de dados do cliente relativa à paleta a ser etiquetada. Esta informação define o tipo de etiqueta a ser colocada, o formato (A4, A5 ou etiqueta colorida), a orientação (no caso de ser A4 ou A5), o número representado no código de barras, as faces da paleta em que leva a etiqueta e a sua posição. Existe ainda a

possibilidade de a paleta não levar nenhuma etiqueta. Neste caso a paleta passa pela ilha através dos transportadores sem parar para aplicar etiqueta.

Assim que o *PLC* recebe toda a informação relativa à colocação das etiquetas na paleta é dada a ordem ao robô para que este inicie o processo de aplicação de etiquetas. As impressoras são controladas pela base de dados do cliente e sempre que o robô inicia uma aplicação de etiqueta, o *PLC* pede em simultâneo a impressão de uma etiqueta. As etiquetas são iguais, independentemente da face da paleta em que são colocadas. Basicamente, o *PLC* pede uma etiqueta referente ao código de barras recebido. Assim que é impressa a etiqueta, o robô inicia todo o processo de etiquetagem começando por fazer um controlo da etiqueta impressa, comparando o código de barras impresso com o que foi transmitido pelo *PLC*. Para isso utiliza a câmara de leitura de código de barras existente na ferramenta. Se o código não corresponder, o robô retira a etiqueta e coloca-a num tabuleiro chamado lixo. Este tabuleiro só foi colocado na ilha depois de esta estar implantada no cliente. Após a recolha da etiqueta dos tabuleiros A4 ou A5 há a possibilidade de o robô colocar a etiqueta perto de um gravador de *RFID* (*radio frequency identification* – identificação por radiofrequência), gravando assim uma *RF TAG*. Esta gravação só é possível quando a etiqueta já vem equipada com *RF TAG*. As etiquetas coloridas não têm este dispositivo. Após este procedimento, o robô passa com a etiqueta em frente da pistola para aplicar cola e depois coloca-a numa das faces da paleta. Assim que estiver aplicada a etiqueta, o robô, no caso das etiquetas de formato A4 e A5, verifica novamente o código de barras. Desta forma é confirmado que a etiqueta ficou aplicada na paleta. No caso de a etiqueta não estar presente, o robô inicia novamente o processo de aplicação. O robô tem a capacidade de colocar etiquetas em duas faces da paleta e, sempre que é necessário colocar as etiquetas nas outras duas faces a mesa rotativa orienta a paleta. Tal como no projeto anterior, o robô executa um programa denominado ‘*MAIN*’ que está a ser percorrido em ciclo contínuo (Figura 95). De uma forma simples pode-se dizer que este programa se resume a duas tarefas: reposicionar o robô na posição ‘*HOME*’ e executar o subprograma ‘*Prog_ETQ*’, que consiste numa sequência de tarefas que o robô tem de efetuar para poder aplicar corretamente a etiqueta na paleta.

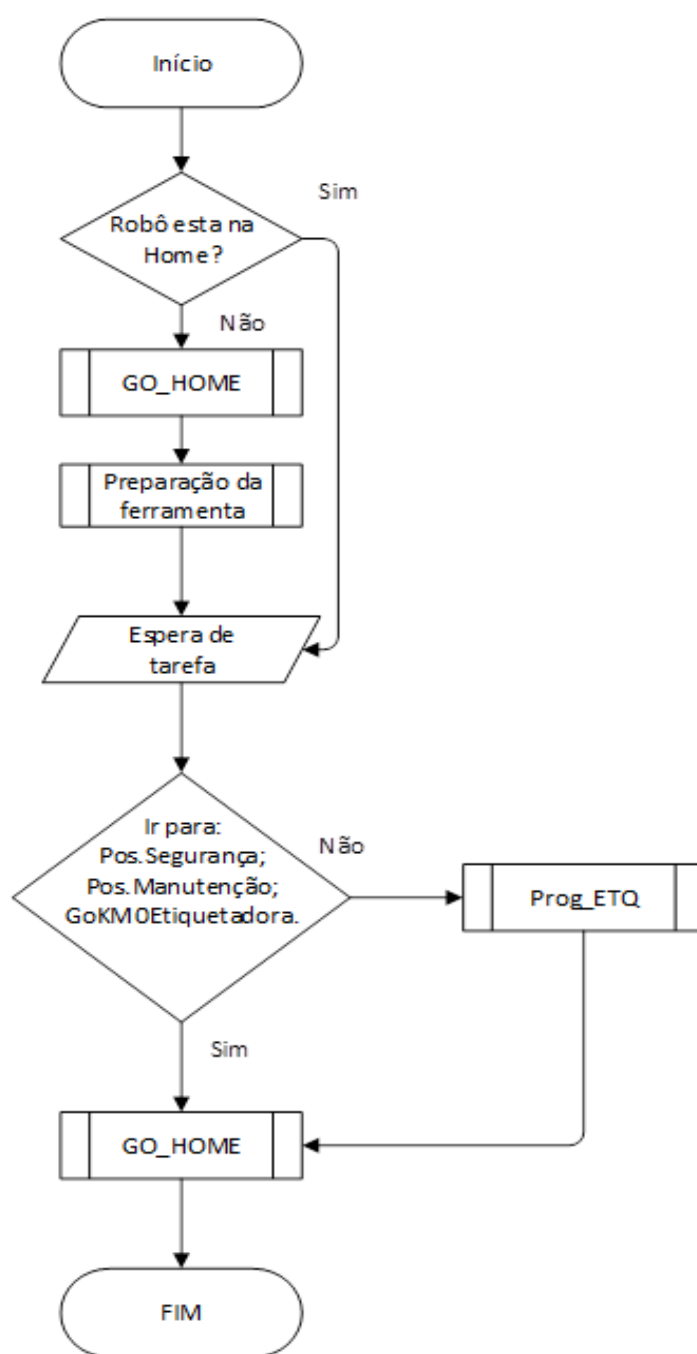


Figura 95 - Fluxograma do programa MAIN

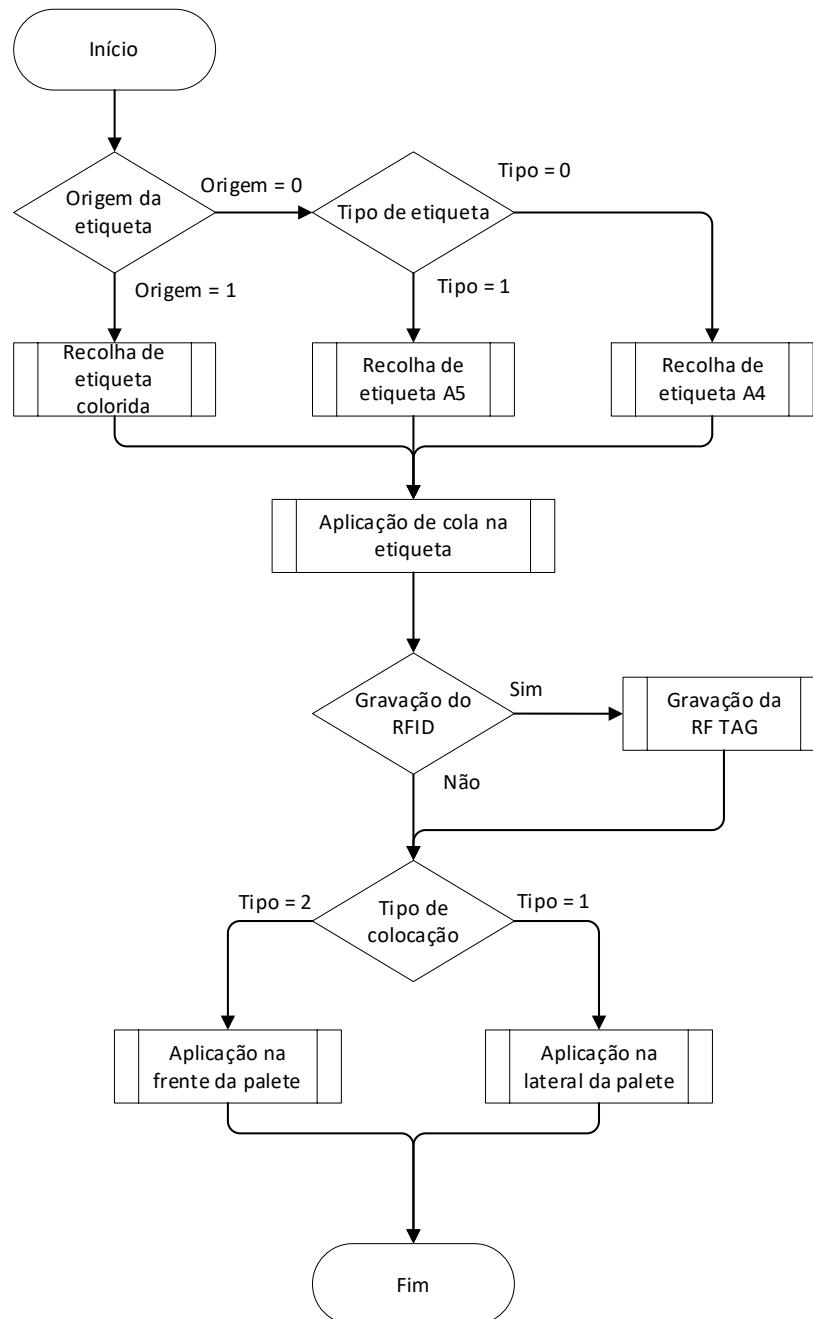


Figura 96 – Fluxograma do subprograma para aplicar etiqueta na paleta 'Prog_ETQ'.

Na Figura 96 está representado um fluxograma simplificado das tarefas que o robô realiza para aplicar a etiqueta na paleta ao executar este subprograma. O robô começa por verificar a origem da etiqueta, que pode vir dos tabuleiros de etiquetas impressas em A4 ou A5 ou dos tabuleiros das etiquetas coloridas. Recolhida a etiqueta, e no caso de ser uma etiqueta com *RF TAG*, o robô passa com a etiqueta num gravador de *RFID* e só depois é que a apresenta à pistola de cola, aplicando a cola ‘pulverizada’ na parte superior

da etiqueta, na forma de um traço na horizontal da folha. Por fim, dependendo do tipo de colocação, o robô coloca a etiqueta na frente ou na face lateral da paleta.

No caso de ser uma etiqueta impressa no formato A4 ou A5, a verificação do código de barras efetuada pela câmara está incluída na recolha da etiqueta nos tabuleiros e na aplicação de etiqueta na paleta, sendo esta verificação opcional.

Esta ilha trabalha de forma autónoma sem a necessidade de intervenção do operador. Só quando existe uma perda de etiqueta durante a trajetória do robô é que existe a intervenção do operador para tomar a decisão de repetir a colocação da etiqueta ou não.

4.2.7 Ensaios e comissionamento

Na fase de ensaios e comissionamento foram testados todos os sinais trocados com o *PLC* da ilha e os sinais no módulo interno de entradas e saídas, tal como no projeto anterior. Também foi calibrada a ferramenta e as bases de trabalho. Depois foram efetuados os testes de recolha, gravação da *RF TAG*, aplicação de cola e por fim a colocação da etiqueta na paleta.

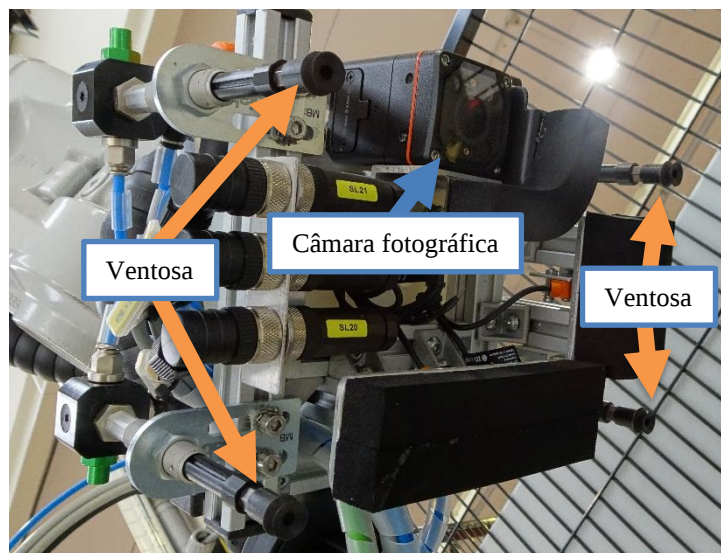


Figura 97 - Ferramenta aplicada no terminal do robô.

Na Figura 97 pode-se ver o aspeto final da ferramenta para pegar nas etiquetas, com a câmara de identificação de código de barras nela instalada, sendo muito semelhante ao modelo em 3D utilizado no simulador *RobotStudio*.

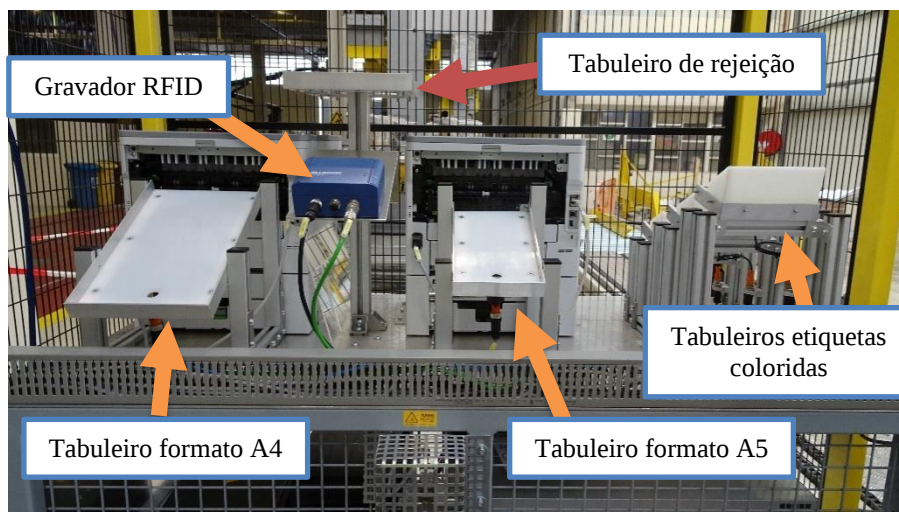


Figura 98 - Tabuleiros para etiquetas de formato A4 e A5, gravador *RFID* e tabuleiro de rejeição.

Na Figura 98 pode-se identificar a forma como ficaram dispostos os tabuleiros para etiquetas dos formatos A4, A5 e etiquetas coloridas. Também é visível o gravador *RFID*, que não estava representado no modelo 3D no simulador, assim como o tabuleiro para rejeição de etiquetas. Os tabuleiros destinados às etiquetas coloridas estão orientados para o lado exterior da ilha, de forma que o operador, abrindo uma porta, tem acesso às impressoras para efetuar a reposição de etiquetas nos respetivos tabuleiros.

Esta ilha está instalada numa linha de transporte que ainda estava em construção e, devido ao fato de ainda não estarem a serem encaminhadas paletes provenientes das linhas de produção, foi utilizada uma paleta modelo para se poder efetuar testes. Também a base de dados do cliente ainda não estava em comunicação com a ilha para poder dar informação acerca da paleta. Neste caso, foi necessário simular todos os dados da paleta como se tivessem sido recolhidos da base de dados. Utilizou-se um modelo de etiqueta A4 e A5 com um código de barras idêntico ao gerado pela base de dados para se poder efetuar os testes de identificação do código de barras pela câmara e de gravação da *RF TAG* no gravador *RFID*.

Também foi testada a aplicação de cola nas etiquetas A4, A5 e coloridas, assim como a sua colocação na paleta.

Na Figura 99 pode-se observar como fica a etiqueta numa face lateral da paleta. Como já foi referido no parágrafo anterior, foi utilizada uma paleta modelo para efetuar testes. A paleta que supostamente é para etiquetar não está plastificada e tem um cartão no topo.

Também é possível ver o calcador: obviamente que com a paleta que foi utilizada para testes não faz qualquer sentido a sua existência, uma vez que já tem uma película que garante a proteção e estabilidade da paleta, como já foi explicado anteriormente, a suposta paleta para etiquetar, é composta por garrafas empilhadas, contendo um cartão entre planos e na parte superior da paleta um cartão. Se não existisse o calcador para efetuar pressão sobre a paleta, o robô poderia derrubar garrafas ao colocar a etiqueta.

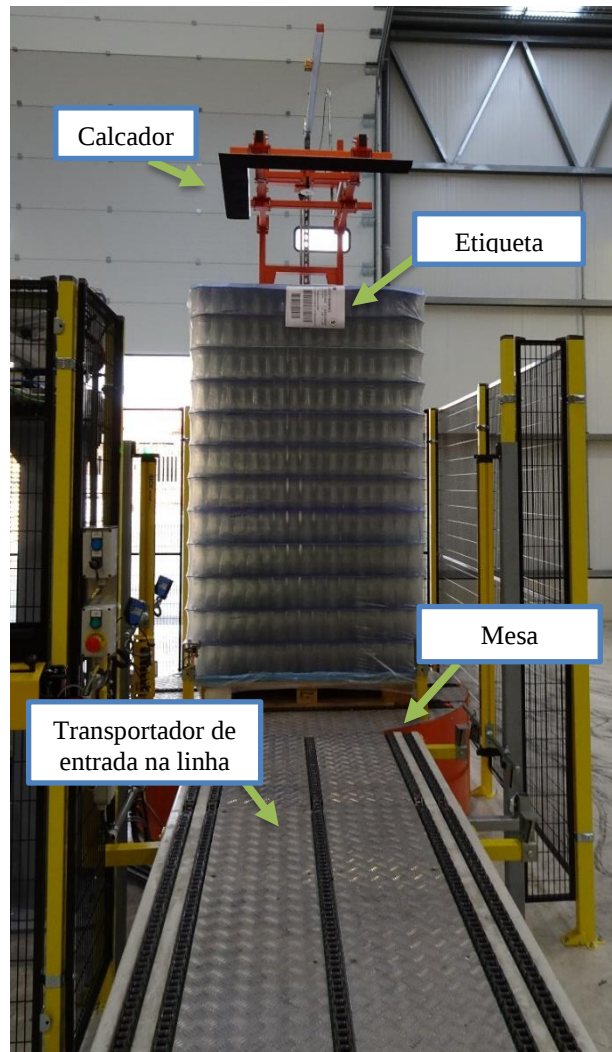


Figura 99 - Exemplo de como fica aplicada a etiqueta na paleta, neste caso na face lateral.

Também é possível visualizar o transportador da paleta para dentro da ilha e a mesa rotativa, na qual a paleta é rodada para conseguir colocar etiquetas em todas as faces e orientada para o transportador de saída da ilha.

Apesar de terem sido efetuados todos os testes simulando a informação que deveria ser recolhida da base de dados e não se utilizar o tipo de paleta que é suposto ser etiquetada,

conseguiu-se testar todas as operações que o robô tem de realizar, para quando a ilha receber paletes provenientes das linhas de produção estar completamente operacional. Este trabalho teve uma duração total de 112 horas, das quais 88 foram de desenvolvimento do programa e as restantes 56 destinadas a ensaios e comissionamento do robô.

4.3 Paletização de pasta de argila

O terceiro trabalho descrito no presente relatório foi a programação de um robô IRB460 da ABB, para que paletizasse cilindros ou paralelepípedos de pasta de argila em paletes de madeira que posteriormente seriam armazenadas e enviadas para clientes. A paletização desta pasta era, até aqui, feita por operadores, sendo um trabalho árduo e prejudicial para a saúde e que poderia causar lesões musculares e doenças relativas à coluna vertebral. Com o intuito de eliminar esse risco, foi proposto criar uma ilha robotizada que realizasse esse trabalho árduo.

Neste trabalho, o robô tem de realizar as seguintes tarefas: pegar numa paleta de madeira num armazém dentro da ilha e colocá-la na mesa de paletização; colocar a pasta na paleta até esta estar concluída. No caso da paletização de cilindros, a sua formação na paleta é idêntica a um prisma triangular. Já os paralelepípedos, aos quais o cliente dá o nome de lastras, são colocados lado a lado até formarem um plano. Dependendo da altura das lastras, a paleta pode ter dois ou três planos. A ilha está equipada com um aplicador que corta película com um determinado comprimento e a aplica imediatamente em cima da paleta de madeira ou entre planos. Esta película tem várias finalidades: uma delas é para a pasta não estar em contato direto com a paleta; outra, é para se tornar mais fácil retirar o produto da paleta, dado que com o passar do tempo a pasta fica colada.

Este robô também utiliza o mesmo método de receitas, tal como foi utilizado no primeiro projeto, em que o operador tem de escolher numa consola *HMI* o tipo de produto que quer paletizar.

Neste trabalho foram criadas 32 receitas, todas definidas para medidas de blocos de pasta pré-definidas. Em média, a parametrização e teste de cada receita demorou cerca de 12 horas. A pasta a paletizar pode ter um comprimento que varia entre 300mm e 800mm. No caso dos cilindros só existem comprimentos de 300, 400 e 800mm e o diâmetro pode ir

de 60mm a 200mm. As lastras podem ter entre 400 a 600mm de comprimento, uma altura de 150 a 250mm e uma espessura entre 30 a 45mm.

Na Figura 100 pode-se observar o *layout* da ilha, onde estão todos os equipamentos que a constituem, nomeadamente o robô IRB460, as mesas para recolha da pasta, a mesa para paletização, o dispensador de película e o armazém de paletes de madeira.

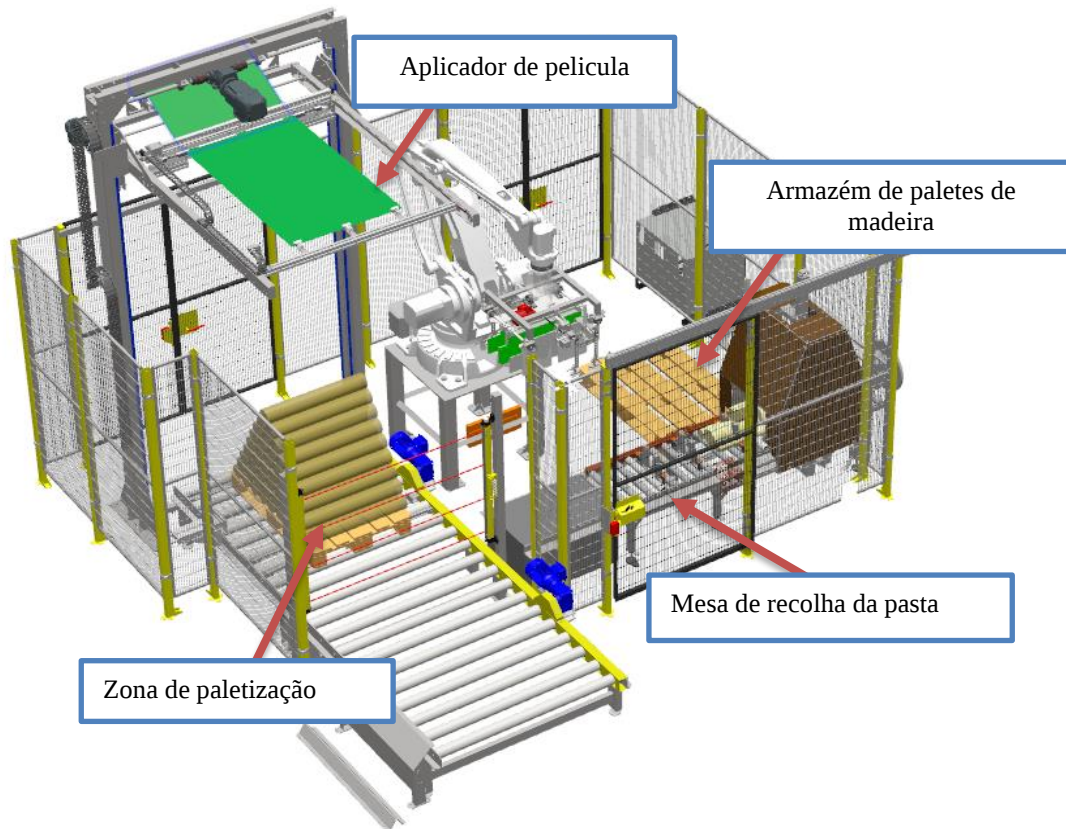


Figura 100 - Layout da ilha robotizada para paletização de pasta de argila.

Na paletização de cilindros, podem chegar ao robô através de uma mesa de transporte até três cilindros de uma só vez.

A ferramenta do robô foi desenvolvida para poder pegar numa paleta, em cilindros e nas lastras. Para isso utiliza uma pinça composta por duas chapas que se movem (Figura 101). Na pasta cilíndrica, a ferramenta tem a capacidade de pegar entre um a três cilindros de uma só vez, podendo da mesma forma libertar um, dois ou três cilindros de uma só vez. Na paletização de lastras o robô agarra em duas lastras que estão lado a lado.

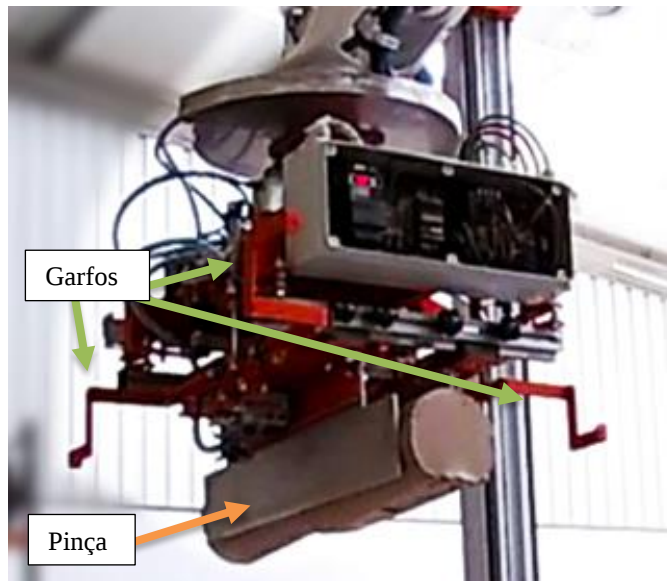


Figura 101 - Ferramenta para pegar na pasta e paletes.

A ferramenta utiliza um sistema de garfos para pegar lateralmente na paleta entre as régua de madeira inferiores e superiores.

Durante a fase de testes, verificou-se que a pasta escorregava da pinça quando se efetuavam movimentos com o robô. Para resolver este problema, as chapas da pinça foram alvo de alterações que incluíram, nomeadamente, a criação de uma ligeira curvatura na extremidade das chapas para se adaptar melhor ao formato cilíndrico da pasta.

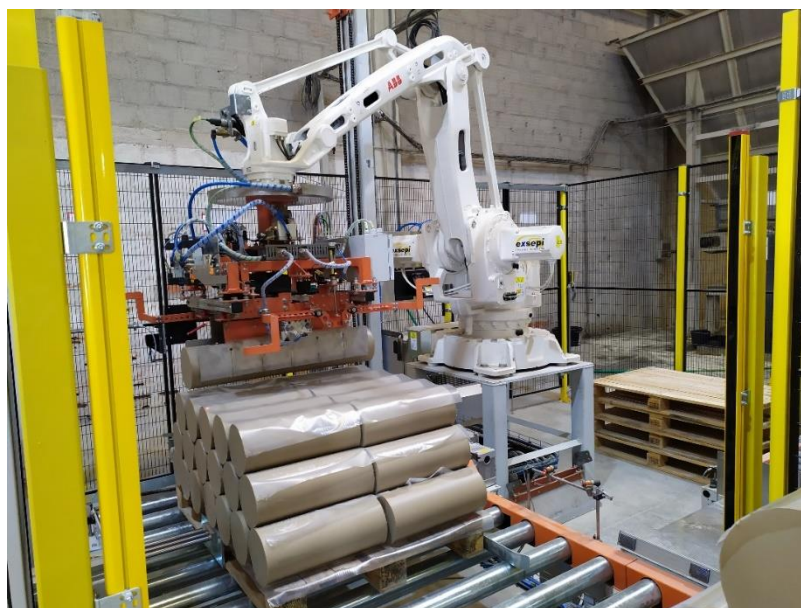


Figura 102 - Exemplo do robô a efetuar a paletização de pasta em formato de cilindro.

Neste projeto só foi testada a paletização de pasta em formato de cilindro, ficando por testar a pasta em formato de paralelepípedo, devido à indisponibilidade do cliente para efetuar a paragem da linha de produção.

Na Figura 102, pode-se observar o robô a efetuar uma paletização de pasta em formato de cilindro. É possível visualizar a película entre camadas e na base da paleta.

Este trabalho foi realizado em 520 horas, tendo o desenvolvimento do programa demorado 312 e os ensaios e comissionamento 208.

4.4 Alteração de ilha robotizada

Um cliente da indústria automóvel tinha uma ilha robotizada, na qual o robô recolhia veios de caixas de velocidades, os quais serão denominados como peças, de um tapete e os colocava em duas máquinas para efetuar acabamento. Uma das máquinas efetuava o acabamento à peça e dava seguimento para outra linha; a outra máquina realizava o acabamento à peça e devolvia-a para o robô a recolher e a colocar num carro com uma matriz de furos. Esse carro, quando cheio, tinha de ser retirado pelo operador para retirar as peças e depois tinha de ser colocado novamente na ilha.

A alteração a ser efetuada passou por adicionar um transportador, em substituição do carro, no qual o robô coloca as peças prontas para depois serem transferidas para outra linha. Com esta alteração ganhou-se tempo de ciclo da ilha, evitando assim a paragem da ilha para o operador ter de esvaziar o carro com peças. Ao adicionar esse transportador foi necessário alterar trajetórias para que o robô não colidisse com ele e adicionar um ponto de entrega de peça no transportador.



Figura 103 - Consola do controlador do robô.

Na ilha está instalado um robô KR15 do fabricante KUKA com um controlador KRC1 o qual, devido a ser já muito antigo, não foi possível programar utilizando um *software* de desenvolvimento no computador, tendo sido necessário programá-lo na sua própria consola (Figura 103).

Na ilha existe uma estrutura de alumínio debaixo da qual está o carro em que o robô coloca as peças maquinadas. Essa estrutura desce quando o operador efetua um pedido à ilha para retirar o carro com peças, com o objetivo de selar o acesso ao interior da ilha quando retirado o carro (Figura 104).

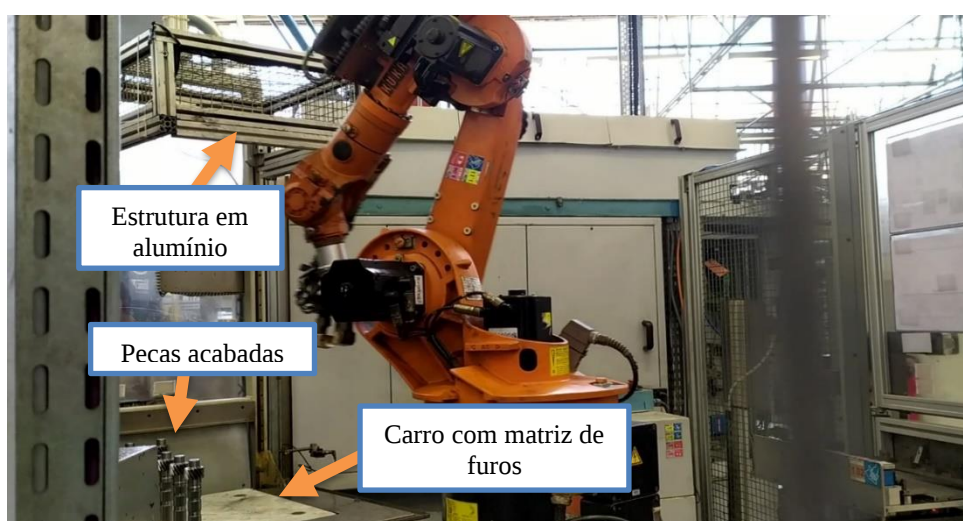


Figura 104 – Layout anterior: robô a passar na frente da estrutura de alumínio.

O novo transportador foi posicionado onde estava situada a estrutura de alumínio, tendo para isso sido alterada para se poder colocar o transportador. Devido ao pequeno alcance do robô, o transportador foi colocado próximo do robô para que este consiga deixar a peça no transportador. Por consequência, foi necessário alterar as trajetórias do robô quando passa nessa região.

Na Figura 105 pode-se ver como o transportador está instalado na ilha assim como o robô a executar uma trajetória muito mais afastado. Também é possível observar o transportador onde o robô coloca a peça já acabada. O transportador tem um suporte que se movimenta a todo o seu comprimento por meio de um sistema de motor e correia. O suporte é composto por três bases de apoio nas quais o robô coloca a peça acabada e, desta forma, a peça é transportada para a próxima ilha para ser posteriormente introduzida

numa nova máquina. Assim, não existe intervenção do operador para efetuar essa transferência da peça.



Figura 105 - Transportador inserido na ilha e robô na nova trajetória.

Após alterar as trajetórias do robô para que não embatesse no transfere, foram criadas as posições de aproximação e entrega da peça ao suporte no transportador (Figura 106). Neste trabalho também foram adicionados dois sinais a enviar ao PLC que gere a ilha, um para indicar que o robô está na região do transportador e o outro para indicar que deixou uma peça no suporte.

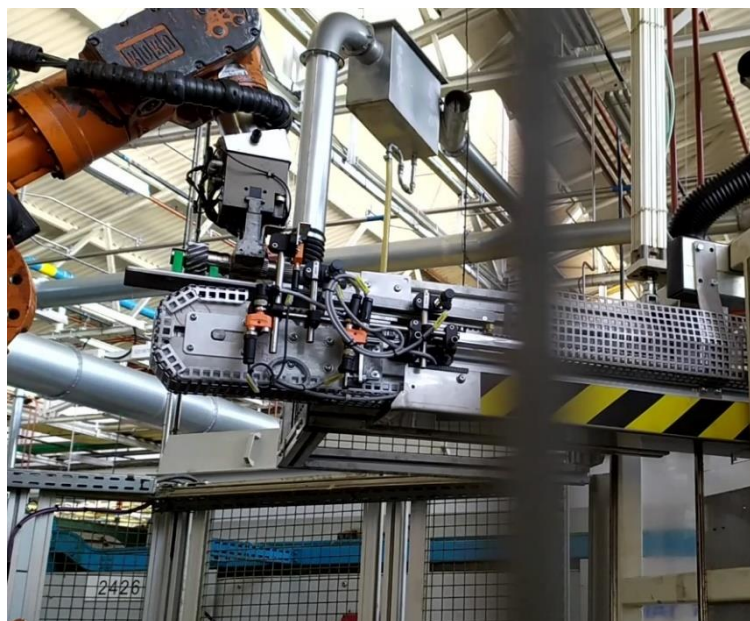


Figura 106 - Posição de deixar peça no suporte do transportador.

Sempre que o transportador estava livre e em posição para receber uma peça e a máquina tinha uma peça acabada para recolher, o PLC dava a ordem ao robô para recolher a peça na máquina e a entregar no transportador.

Este trabalho foi realizado durante paragens da ilha programadas, para que não existissem grandes quebras de produção, normalmente feitas às quintas-feiras de cada semana. O tempo de duração do trabalho foi de 56 horas.

5. Conclusão

O estágio realizado na EXSEPI foi um grande desafio, superado com a ajuda dos companheiros de trabalho e com o profissionalismo e rigor de uma grande equipa. Foram encontradas algumas dificuldades, tais como a compreensão de novas linguagens e da metodologia de programação utilizadas por cada um dos fabricantes de robôs e a compreensão dos meios de comunicação entre dispositivos e das suas configurações.

Todas as dificuldades foram ultrapassadas com os conhecimentos já adquiridos ao longo dos anos de estudo e da experiência profissional e com a ajuda do coordenador do estágio, que esteve sempre presente e disponível.

Sendo este estágio desenvolvido num ambiente industrial, pode-se constatar a importância que os robôs têm na produção para realizar tarefas como a movimentação de cargas pesadas e a realização de movimentos repetitivos com precisão e repetibilidade. Na execução dessas tarefas, um ser humano, ao longo de horas ou dias de trabalho, não conseguiria obter o mesmo nível de produção devido à fadiga física e até psicológica, podendo ainda originar a ocorrência de falhas humanas que poderiam ser prejudiciais e, em alguns casos, fatais para o operador. Também se verificou a desmistificação do conceito de que os robôs tiram o trabalho às pessoas. Pelo contrário, naquilo que foi presenciado, todos os robôs implementados tiveram como objetivo colmatar a falta de mão de obra, melhorar a produção e melhorar as condições de trabalho dos operadores, tendo estes sido realocados em tarefas menos pesadas, menos repetitivas e menos prejudiciais à sua saúde e ao seu bem-estar.

À medida que foi desenvolvido o primeiro projeto realizado, foram consolidados conhecimentos de modelação e simulação no *software RobotStudio*. Este *software* permitiu que se pudesse realizar todo o desenvolvimento do projeto, desde a implantação do robô na ilha robotizada através do modelo 3D até à simulação de toda a programação desenvolvida. Na fase de testes foi possível compreender os diferentes comportamentos entre um modelo simulado e o comportamento real de todo um sistema robótico, incluindo a ferramenta e seus componentes, nomeadamente os sistemas de vácuo e de garra ou pinça e adaptá-los de forma a melhor realizarem as tarefas para as quais foram desenvolvidos. Os testes realizados permitiram compreender o comportamento do sistema de vácuo utilizado, neste caso um sistema de geração de vácuo do tipo venturi, as

difficultades que o robô tinha em pegar e manipular uma paleta de cartão ou um cartão, sendo que este problema ficou resolvido testando várias velocidades e acelerações do robô até se obter um resultado aceitável. Na manipulação de caixas também foi regulada a força que a pinça da ferramenta exercia ao pegar na caixa. Para isso, foi colocada borracha nas paredes laterais da pinça com o objetivo de aumentar o atrito e de se poder diminuir a força exercida numa caixa podendo, assim, alterar a pressão de ar utilizada para movimentar o cilindro pneumático que, por sua vez, efetuava o movimento de abertura e fecho da pinça. Ainda existiu um problema com a colocação de etiquetas nas caixas que, devido à inexistência de uma medida de velocidade padronizada por parte da etiquetadora e respetivo dispensador de etiquetas, constituiu um desafio até se encontrar a velocidade de sincronismo entre o robô que passava com uma caixa para etiquetar e a velocidade do dispensador de etiquetas. Através do “clássico” método de tentativa e erro, obteve-se a correta colocação da etiqueta na caixa. A programação do controlador do robô foi muito desafiante, uma vez que foram criadas uma estrutura e uma metodologia de programação que permitiram reutilizar o mesmo processo noutros robôs, sem que para isso se tenha de efetuar grandes alterações à programação. Este método consistiu na criação de pequenos subprogramas que, ao interpretarem dados relativos ao produto a ser paletizado, contidos num ficheiro denominado receita, realizavam as tarefas necessárias para obter o resultado final desejado, que neste caso era a paletização de caixas em cima de uma paleta de cartão.

No segundo trabalho apresentado neste relatório desenvolveu-se um sistema robótico para etiquetagem de paletes com garrafas. Sendo um projeto mais simples relativamente ao anterior, não houve grandes dificuldades, mas existiu a preocupação de que na manipulação e colocação de etiquetas estas não se soltassem das ventosas e que fosse aplicada cola na etiqueta de forma adequada. Para isso foram criados mecanismos de deteção de perda de etiqueta, nomeadamente sensores de reflexão no objeto que detetavam a presença de uma etiqueta nas ventosas da ferramenta do robô, assim como também foram ajustadas a aceleração e a velocidade do robô para garantir que ao efetuar a movimentação das etiquetas, estas não se soltavam. Tal como aconteceu com o processo de etiquetagem do primeiro trabalho apresentado, também no processo de aplicação de cola nas etiquetas se detetou a existência de um atraso entre o sinal dado para a projeção da cola na etiqueta e a efetiva projeção da cola. Para resolver este problema foi antecipado o sinal que ordena a projeção de cola, bem como a ordem para terminar essa projeção.

Mais uma vez, foi pelo método de tentativa e erro que se conseguiu obter os resultados desejados.

No terceiro trabalho apresentado, a paletização de pasta de argila, existiram várias dificuldades no processo de pegar na pasta, movimentá-la e colocá-la na posição correta na paleta. Ao pegar na pasta, ela escorregava da ferramenta, mesmo com movimentos e acelerações reduzidos. Esta questão foi superada com a introdução de algumas alterações físicas na ferramenta, nomeadamente, a criação de uma pequena curvatura na extremidade da pinça, para que desta forma ‘abraçasse’ a pasta e esta não escorregasse. Ultrapassado o problema de movimentação da pasta, detetou-se uma nova dificuldade ao deixar a pasta na primeira camada na paleta. Como a pasta tinha um formato cilíndrico, rolava ao abrir a pinça, saindo assim da posição pretendida na paleta. Para resolver este problema teve de se jogar com a altura de deixar a pasta na paleta e a velocidade de abertura da pinça. Pelo fato de este projeto ser muito demorado no que respeita a testes, ficou por testar a paletização de lastras, devido à indisponibilidade do equipamento por parte do cliente, uma vez que o robô se encontrava a trabalhar na paletização de cilindros e para fazer os testes em falta teria de se fazer uma paragem programada.

No quarto e último trabalho, foi efetuada a alteração de uma ilha robotizada que estava em pleno funcionamento. Este trabalho, apesar de aparentemente ser muito simples, apresentou uma dificuldade suplementar pelo fato de se tratar de um controlador muito antigo, por ter de se programar o robô utilizando a sua própria consola e por ter sido um trabalho intermitente, ou seja, foi efetuado nas paragens da linha de produção, programadas para o efeito.0

Em resumo, todos os trabalhos realizados foram bastante desafiantes, tendo aprendido com cada um deles a resolver problemas que aparentemente não eram detetados em simulação. Com a experiência de cada projeto, foi desenvolvida a capacidade de programar qualquer tipo de robô, uma vez que o método de programação entre fabricantes de robôs é muito convergente. Foi também desenvolvida a capacidade de resolução de problemas, mesmo que sejam de outras áreas de competência. Foi desenvolvida uma grande capacidade de trabalho com equipas multidisciplinares e, frequentemente, com o cliente.

Fica a esperança de que no futuro apareçam novos desafios e que com a experiência obtida neste estágio, a contribuição pessoal e da equipa de trabalho seja resolutiva para a melhoria da EXSEPI. Fica também a consciência de que os robôs na indústria, para

além de aumentarem a produção e promoverem a diminuição do desperdício, sejam cada vez mais utilizados para auxiliarem na melhoria das condições de trabalho das pessoas, com o objetivo de salvaguardar a sua saúde e o seu bem-estar no local de trabalho.

Referências

- [1] EXSEPI, “Exsepi - Estudos e Projectos Industriais, Lda,” 2021. [Online]. Available: <https://exsepi.com/language/pt/empresa/>. [Acedido em 29 03 2021].
- [2] <https://www.bell-labs.com/timeline/#/1940/1/open/>, “The 40s,” NOKIA Bell Labs , [Online]. Available: <https://www.bell-labs.com/timeline/#/1940/1/open/>.
- [3] Wikipédia, “Karel Čapek,” 19 08 2020. [Online]. Available: https://pt.wikipedia.org/wiki/Karel_%C4%8Capek.
- [4] M. Akinbola, “Robotic Dystopias,” 3 Maio 2006. [Online]. Available: [http://academic.depauw.edu/aevans_web/HONR101-02/WebPages/Spring2006/Akinbola\(Mike\)/R.U.R.%20\(1921\).html](http://academic.depauw.edu/aevans_web/HONR101-02/WebPages/Spring2006/Akinbola(Mike)/R.U.R.%20(1921).html).
- [5] L. LEAL, “R.U.R. e a origem do termo “robô”,” Editora MADREPÈROLA, 23 06 2020. [Online]. Available: <https://www.editoramadreperola.com/r-u-r-e-a-origem-do-termo-robo/>. [Acedido em 29 03 2021].
- [6] Kawasaki Robotics, “From the Birth of Industrial Robots in the United States to Their Launch in Japan,” Kawasaki Robotics, 2018. [Online]. Available: https://robotics.kawasaki.com/en1/anniversary/history/history_01.html.
- [7] Wikipedia, “Parâmetros de Denavit-Hartenberg,” [Online]. Available: https://pt.wikipedia.org/wiki/Par%C3%A2metros_de_Denavit-Hartenberg. [Acedido em 12 2021].
- [8] Wikipedia, “Giovanni Battista Venturi,” 09 01 2021. [Online]. Available: https://en.wikipedia.org/wiki/Giovanni_Battista_Venturi. [Acedido em 05 04 2021].
- [9] Wikipédia, “Modelo OSI,” [Online]. Available: https://pt.wikipedia.org/wiki/Modelo_OSI#4_-_Camada_de_Transporte. [Acedido em 26 09 2021].
- [10] Siemon, “Siemon,” [Online]. Available: <https://www.siemon.com/br/whitepapers/ethernet-ip.asp>. [Acedido em 13 07 2021].
- [11] ABB, “Manual de operação, IRC5 com FlexPendant, ID: 3HAC16590-6,” ABB AB, Västerås Suécia, 2013.
- [12] Robotic Industries Association, “Robotics Online,” 2021. [Online]. Available: <https://www.robotics.org/Robotic-Resources>.
- [13] J. N. Pires, Automação Industrial, Av. Praia da Vitória, 14-A, 1000-247 Lisboa: ETEP, 2015.
- [14] P. Corke, Robotics, Vision and Control, Berlin Heidelberg: Springer, 2013.
- [15] J. Angeles, Fundamentals of Robotic Mechanical Systems, Switzerland: Springer, 2014.
- [16] J. A. M. F. d. Souza, “Automação e Robótica,” *Automação Industrial e Robótica*, pp. 1-11, 2013.
- [17] ABB, “Product specification IRB4600, ID 3HAC032885-001,” ABB, 2020.
- [18] ABB AB, “Technical reference manual, System parameters, ID: 3HAC050948-001,” ABB AB, Västerås Sweden, 2018.
- [19] ABB, “Technical reference manual, RAPID Instructions, Functions and Data types , ID: 3HAC 16581-1,” ABB AB, Västerås Sweden, 2010.
- [20] ABB, “Technical reference manual, RAPID kernel, ID: 3HAC050946-001,” ABB AB, Västerås Sweden, 2018.
- [21] M. Matjaž, B. Tadej, U. Aleš, L. Jadran, S. Aleš, M. Marko, R. Jure e Š. Sebastjan, Robotics, Switzerland: Springer, 2019.
- [22] A. Barrientos, L. F. Peñín, C. Balaguer e R. Aracil, Fundamentos de Robótica, segunda edição, Madrid: McGraw-Hill/INTERAMERICANA DE ESPANHA, S. A., 1997.
- [23] INTRACT, “INTRACT,” [Online]. Available: <https://www.intract.com.br/robo-cartesiano-serie-f9000n>. [Acedido em 03 05 2021].
- [24] Wikipédia, “John Bardeen,” [Online]. Available: https://en.wikipedia.org/wiki/John_Bardeen. [Acedido em 23 10 2021].
- [25] Wikipédia, “Walter Houser Brattain,” [Online]. Available: https://en.wikipedia.org/wiki/Walter_Houser_Brattain. [Acedido em 23 10 2021].

- [26] Wikipédia, “William Shockley,” [Online]. Available:
https://en.wikipedia.org/wiki/William_Shockley. [Acedido em 29 3 2021].
- [27] W. F. Lages, “Descrições e Transformações Espaciais,” 26 8 2019. [Online]. Available:
<http://www.ece.ufrgs.br/~fetter/eng10026/spatial.pdf>.

Anexos

Anexo A

Tabela A1- Sinais do modulo DSQC1030 de entradas e saídas.

OUT_ABB
Local_IO_OUT_1_G_ABRIR_KAY50_1
Local_IO_OUT_2_G_FECHAR_KAY50_2
Local_IO_OUT_3_G_VACUO_KAY51
Local_IO_OUT_4_G_LVENTOSAS_KAY52
Local_IO_OUT_5_ATIVA_LASER
IN_ABB
Local_IO_IN_1_P_PCARTAO_B40
Local_IO_IN_2_P_CARTAO_B41
Local_IO_IN_3_G_ABERTO_BY50_1
Local_IO_IN_4_G_FECHADO_BY50_2
Local_IO_IN_5_G_VAC_SP50
Local_IO_IN_6_G_PRES_SP51

Anexo B

Tabela A2 - Dados do controlador para PLC

ABB_to_PLC
EN_DO00_MotOn
EN_DO01_Auto
EN_DO02_Cycle
EN_DO03_EmeR
EN_DO04_RunCh
EN_DO05_NotMove
EN_DO06_AtMain
EN_DO07_FimProg
EN_DO08_POS_DESC
EN_DO09_INIT_PAL_ACK
EN_DO10_SEARCH_NOT_FIND
EN_DO13_Palete_concluida
EN_DO14_Imp_Etiqueta
EN_DO16_WZ_HOME
EN_DO17_WZ_PALETE
EN_DO18_WZ_MESA_R_CAIXA
EN_DO19_WZ_PALETE_CT
EN_DO20_WZ_ARM_CT
EN_DO21_WZ_HOME_PAL
EN_DO32_EXCHANGE_PROD_OK
EN_DO33_EXCHANGE_PROD_NOK
EN_DO34_FALHA_LOAD_FILE
EN_DO35_PERDA_CAIXA
EN_DO36_PERDA_CARTAO
EN_DO48_EV_A_GRIPPER_KAY50_1
EN_DO49_EV_F_GRIPPER_KAY50_2
EN_DO50_EV_VACUO_KAY51
EN_DO51_EV_LVACUO_KAY52
EN_DO52_ATIVA_LASER
EN_DO56_SNS_P_PCARTAO_B40

EN_DO57_SNS_P_CARTAO_B41
EN_DO58_SNS_G_ABERTO_BY50_1
EN_DO59_SNS_G_FECHADO_BY50_2
EN_DO60_SNS_G_VAC_SP50
EN_DO61_SNS_G_PRES_SP51
EN_GO64_79_TAREFA
EN_GO80_95_ALARMES
EN_GO96_111_MOV_INICIADO
EN_GO112_127_MOV_CONCLUIDO
EN_GO128_143_ALT_PLANO
EN_GO144_159_N_PLANO
EN_GO160_175_N_ENTREGA
EN_GO176_191_ALT_ARM_PAL_CARTAO
EN_GO192_207_ALT_ARM_CARTAO
EN_GO208_223_N_Receita
EN_GO224_239_Dist_aprox_tcp

Anexo C

Tabela A3 - Dados do PLC para o controlador do robô.

PLC_to_ABB
EN_DI00_CmdMtOn
EN_DI01_StartMain
EN_DI02_Start
EN_DI03_ResetStop
EN_DI04_CmdMtOff
EN_DI05_Stop
EN_DI06_GO_SEGUR
EN_DI07_GO_MANUT
EN_DI08_GO_REJECT
EN_DI09_EXCHANGE_PROD_SAVE
EN_DI10_EXCHANGE_PROD
EN_DI11_INIT_PAL
EN_DI12_Paleta_concluida_ACK
EN_DI13_Coloc_Etiqueta
EN_DI14_M_Func_Pal_Base
EN_DI15_Pal_Base
EN_DI16_EXISTE_TAREFA
EN_DI17_PRESENCA_CAIXA
EN_DI18_RETRY_SEARCH
EN_DI19_CANCEL_SEARCH
EN_DI48_TRIGGER_FIND_OBJ
EN_DI49_PRESENCA_OBJ
EN_DI50_ZONA_PAL_LIVRE
EN_GI64_79_MD_LASER
EN_GI80_95_N_RECEITA
EN_GI96_111_Dist_aprox_tcp