

isec

Engenharia

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA

Aplicações de Robótica Industrial – Estágio na Fluidotronica

Relatório de Estágio de Natureza Profissional para a obtenção do
grau de Mestre em Engenharia Eletrotécnica

Especialização em Automação e Comunicações em Sistemas
Industriais

Autor

Rúben Filipe Tavares Oliveira

Orientador

Doutor Victor Daniel Neto dos Santos

Supervisores na empresa Fluidotronica

Eng. Hélder Almeida da Silva

Eng. Luís Marques Silva

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA
DE COIMBRA

Coimbra, Novembro 2025

RESUMO

O presente relatório de estágio descreve o desenvolvimento e a integração de diversas soluções de automação e robótica aplicadas a ambientes industriais, realizadas na empresa Fluidotronica. O trabalho foi desenvolvido no âmbito do estágio curricular do Mestrado em Engenharia Eletrotécnica (MEE) e incidiu na utilização do software RobotStudio para a simulação de movimentos, programação e detecção de colisões em células robotizadas industriais, bem como na implementação de soluções envolvendo robôs colaborativos e sistemas de visão artificial.

Foram analisadas três células de produção pertencentes à indústria automóvel, cada uma com diferentes níveis de funcionalidade e complexidade. As células PRJ 1308 e PRJ 1311 executam tarefas semelhantes, nomeadamente a manipulação de componentes e a colocação de clips em peças plásticas do veículo em produção. Estas células são compostas por robôs da marca ABB, que foram programados e simulados no ambiente virtual RobotStudio.

A célula PRJ 1309, por sua vez, apresenta uma maior diversidade de operações, incluindo aquecimento, aplicação de fita, inserção de esponjas, tabs e clips, o que exige uma coordenação rigorosa entre os vários elementos do sistema.

No âmbito do estágio, foi também desenvolvido um projeto de soldadura a laser para um cliente da empresa, recorrendo a um robô colaborativo da marca JAKA. O principal desafio deste projeto consistiu na adaptação e parametrização do robô para realizar soldaduras em peças cilíndricas, de acordo com critérios técnicos específicos.

Além disso, foi implementada uma solução de alimentação automatizada de máquinas CNC, recorrendo a robôs colaborativos para a conceção e manipulação de peças metálicas em grande escala. A implementação incluiu também a parametrização de um sistema de visão artificial, que permitia detetar e localizar com precisão as peças durante os processos robotizados.

O trabalho desenvolvido durante o estágio evidencia a importância da simulação e da integração de tecnologias emergentes na automação industrial, nomeadamente soluções robóticas, que contribuem de forma significativa para aumentar a eficiência, a segurança e a flexibilidade dos processos produtivos.

Palavras-chave: Automação, Simulação, Cobots, Sistemas de Visão, Programação

ABSTRACT

The present internship report describes the development and integration of various automation and robotics solutions applied to industrial environments, carried out at the company Fluidotronica. The work was developed within the scope of the curricular internship of the Master's in Electrical Engineering and focused on the use of RobotStudio software for motion simulation, programming, and collision detection in industrial robotic cells, as well as on the implementation of solutions involving collaborative robots and artificial vision systems.

Three production cells belonging to the automotive industry were analyzed, each with different levels of functionality and complexity. Cells PRJ 1308 and PRJ 1311 perform similar tasks, namely the handling of components and the placement of clips on plastic parts of the vehicle in production. These cells are composed of ABB robots, which were programmed and simulated in the RobotStudio virtual environment.

Cell PRJ 1309, in turn, presents a greater diversity of operations, including heating, tape application, sponge insertion, tabs, and clips, which requires strict coordination between the various elements of the system.

Within the scope of the internship, a laser welding project was also developed for a client of the company, using a collaborative robot from the brand JAKA. The main challenge of this project was the adaptation and parameterization of the robot to perform welds on cylindrical parts, according to specific technical criteria.

In addition, an automated feeding solution for CNC machines was implemented, using collaborative robots for the design and manipulation of large-scale metal parts. The implementation also included the parameterization of an artificial vision system, which allowed the precise detection and location of the parts during the robotic processes.

The work developed during the internship highlights the importance of simulation and the integration of emerging technologies in industrial automation, particularly robotic solutions, which significantly contribute to increasing the efficiency, safety, and flexibility of production processes.

Keywords: Automation, Simulation, Cobots, Vision, Programming

EPÍGRAFE

Não se trata de bater com a cabeça na parede, mas de encontrar a porta com os olhos.

Werner von Siemens

O engenheiro acredita que, se algo pode ser imaginado, pode ser construído.

Henry Petroski

Para cada problema complexo, existe uma solução clara, simples... e errada. Cabe ao engenheiro encontrar a correta.

Henry Louis Mencken

DEDICATÓRIA

Dedico esta tese a todo o esforço, perseverança e dedicação investidos ao longo deste percurso. Cada momento de superação, cada desafio vencido e cada hora de trabalho refletido nestas páginas representam não apenas um objetivo alcançado, mas o testemunho do compromisso pessoal com o conhecimento, a evolução e a excelência.

A todos os que, direta ou indiretamente, fizeram parte desta caminhada — esta conquista também é vossa.

AGRADECIMENTOS

Gostaria de expressar, em primeiro lugar, a minha mais profunda gratidão aos meus pais, pelo constante apoio moral, emocional e financeiro ao longo de toda esta jornada académica. Sem a sua presença incondicional, este percurso não teria sido possível.

Em segundo lugar, agradeço aos meus amigos e colegas, cujo companheirismo, incentivo e partilha de conhecimentos foram essenciais em momentos de desafio e superação.

Não poderia deixar de reconhecer e agradecer aos professores e orientadores que, com dedicação e profissionalismo, contribuíram para a minha formação técnica e científica. Os seus ensinamentos, conselhos e exigência académica foram fundamentais para o meu crescimento enquanto estudante e futuro profissional da área.

Quero ainda manifestar o meu sincero agradecimento à empresa Fluidotronica, pela oportunidade de realizar o meu estágio curricular. Um agradecimento especial aos responsáveis da empresa, pela confiança depositada em mim, e a todos os colaboradores, pelo acolhimento caloroso e pelo ambiente de trabalho positivo, marcado pela entreajuda, profissionalismo e espírito de equipa, que contribuíram para o meu desenvolvimento pessoal e técnico ao longo dessa experiência.

Por fim, expresso o meu reconhecimento ao Instituto Superior de Engenharia de Coimbra (ISEC), por disponibilizar as condições necessárias à realização deste estágio, designadamente o acesso a recursos tecnológicos e bibliográficos e o ambiente académico que promove o pensamento crítico, a inovação e a excelência na formação em engenharia.

ÍNDICE

Índice	vi
1 Introdução	1
1.1 Enquadramento	2
1.2 Empresa de acolhimento	3
1.2.1 História	4
1.2.2 Serviços	5
1.3 Objetivos	5
1.4 Plano de Estágio	6
1.4.1 Tarefas realizadas	6
1.4.2 Diagrama de Gantt	6
1.5 Estrutura do Relatório	6
2 Estado de arte e conceitos	8
2.1 Tipos de robôs	8
2.1.1 Robôs articulados	8
2.1.2 Robôs cartesianos	9
2.1.3 Robôs cilíndricos	10
2.1.4 Robôs polares	10
2.1.5 Robôs SCARA	11
2.1.6 Robôs delta	11
2.1.7 Robôs colaborativos	12
2.2 Tipos de juntas	12
2.2.1 Junta rotacional	12
2.2.2 Junta prismática	13
2.2.3 Juntas esféricas	13
2.3 Tipos de estrutura cinemática	14
2.3.1 Esférica (RRP)	14
2.3.2 Cartesiana (PPP)	14
2.3.3 Cilíndrica (RPP)	15
2.4 Principais partes de um robô	15
2.4.1 Controlador	15
2.4.2 Esqueleto do robô	16

Aplicações de Robótica Industrial – Estágio na Fluidotronica

2.4.3	Atuadores	16
2.4.4	Sensores	17
2.4.5	Manipulador	17
2.5	Tipos de ferramentas	18
2.5.1	Configuração de ferramentas	19
2.6	Linguagens e ambientes de programação	20
3	Desenvolvimento de soluções robóticas para aplicações industriais	22
3.1	Célula PRJ 1308	24
3.1.1	Layout e descrição de funcionamento	24
3.1.2	Calibração de ferramentas	30
3.1.3	Definição de <i>Work Objects</i>	35
3.1.4	Programação	39
3.1.5	Alterações da célula	40
3.1.6	Colisões	42
3.1.7	Problemas detetados na célula	42
3.2	Célula PRJ 1309	43
3.2.1	Layout e descrição de funcionamento	43
3.2.2	Robô número 1	45
3.2.3	Robô número 2	53
3.2.4	Robô número 3	56
3.2.5	Robô número 4	59
3.2.6	Colisões	61
3.3	Célula PRJ 1311	63
3.3.1	Layout e descrição de funcionamento	63
3.3.2	Robô número 1	64
3.3.3	Robô número 2	66
3.3.4	Alterações das ferramentas	67
4	Projetos realizados com recurso a robôs colaborativos	69
4.1	Soldadura a laser usando robô colaborativo	69
4.1.1	Descrição	69
4.1.2	Robô utilizado	70
4.1.3	Primeira Proposta	71
4.1.4	Segunda Proposta	72
4.2	Alimentador de CNC usando Robô colaborativo	73
4.2.1	Configurações do robô	77
4.2.2	Fluxogramas de funcionamento	79
4.3	Parametrização de um sistema de visão para um sistema robotizado de deteção de peças	79
4.3.1	Parametrização da câmara	82

4.3.2	Resultados	90
4.4	Conformadora de caixas usando um robô colaborativo	91
5	Conclusão	94

ÍNDICE DE TABELAS

2.1	Linguagens de programação das principais marcas de robôs [26]	20
2.2	Principais softwares de programação segundo as marcas	21

ÍNDICE DE FIGURAS

1.1	Instalações da empresa Fluidotronica	3
1.2	Serviços disponibilizados pela empresa Fluidotronica	5
2.1	Robô da ABB, o IRB 1300-10/1.15[7]	9
2.2	Robô cartesiano da marca Campetella, modelo CO1-CO2 X-Series[9]	9
2.3	Exemplo de robôs cilíndricos[11]	10
2.4	Robô do tipo polar da marca Okura, modelo A1600 II e respetiva área de trabalho[10]	10
2.5	Robô do tipo SCARA e sua área de trabalho[10]	11
2.6	Robô do tipo delta da ABB, modelo IRB 360 FlexPicker[13]	11
2.7	Robô colaborativo da marca JAKA, modelo Zu 12[14]	12
2.8	Junta rotacional[16]	12
2.9	Juntas prismáticas[18]	13
2.10	Juntas esféricas[19]	13
2.11	Controlador IRC5 da ABB[23]	16
2.12	Exemplo de um manipulador	18
2.13	Diagrama de <i>payload</i> do robô GP25 da Yaskawa[25]	19
3.1	Peça de exemplo 1	22
3.2	Peça exemplo 2	23
3.3	Exemplo da ferramenta-estrela de 8 dedos	23
3.4	<i>Layout</i> da célula PRJ 1308	25
3.5	Representação da célula 1308 por áreas	26
3.6	Fluxograma do funcionamento geral da célula 1308	26
3.7	Carregador automático de clips	27
3.8	Dispensador de clips	27
3.9	Processo de inserção ou remoção de clips	28
3.10	Fluxograma do ciclo de carregamento de clips	29
3.11	Origem do TCP do acoplamento do robô nº 1 da célula 1308	30
3.12	Origem do TCP da garra do robô nº 1 da célula 1308	31
3.13	Origem do TCP da ferramenta de colocação de brackets do robô nº 1 da célula 1308	31
3.14	Origem do TCP da ferramenta "Estrela de 8 dedos" do robô nº 2 da célula 1308	32

3.15	Origem do TCP de calibração da ferramenta "Estrela de 8 dedos" do robô nº2 da célula 1308	32
3.16	Origem dos TCPs da ferramenta "Estrela de 20 dedos" e do acoplamento do robô nº 3 da célula 1308	33
3.17	Origem do TCP de calibração da "Estrela de 20 dedos" do robô nº 3	34
3.18	Localização dos novos TCPs da "Estrela de 20 dedos" do robô nº 3	34
3.19	"Pinos" das bases para o segundo e o terceiro robô da célula 1308	36
3.20	Definição do <i>user frame</i> na consola da ABB	36
3.21	Definição do <i>work object</i> pelo método de três pontos na consola da ABB	37
3.22	Representação da origem dos <i>work objects</i> dos robôs nº 2 e 3 para a base da Skoda da célula 1308	37
3.23	Variação da orientação dos <i>work objects</i> consoante a rotação do equipamento relativamente à origem do robô	38
3.24	Localização dos "pinos" e o <i>work object</i> do alimentador de clips da célula 1308	38
3.25	Zona de proximidade entre robô número dois e três	41
3.26	Zona de proximidade entre o robô três e os cilindros	41
3.27	Colisão da ferramenta com a base da célula 1308	42
3.28	Layout da célula PRJ 1309	43
3.29	Representação da célula 1309 por áreas	44
3.30	Fluxograma de funcionamento geral da célula 1309	44
3.31	Base para peças da Volkswagen	45
3.32	Base para peças da Skoda	45
3.33	Dispensador de esponjas	46
3.34	Ferramenta de manipulação de peças	47
3.35	Registo do TCP do primeiro robô da célula 1309	47
3.36	Localização do TCP do primeiro robô da célula 1309	48
3.37	Ferramenta responsável pela captura, manipulação e colocação das esponjas circulares	48
3.38	Localização do TCP do primeiro robô responsável pela captura, manipulação e colocação de esponjas da célula 1309	49
3.39	Alimentador automático de esponjas	50
3.40	Caixa mágica	54
3.41	Acoplamento de ferramentas	54
3.42	Registo de TCPs do segundo robô da célula 1309	55
3.43	Localização dos TPCs do segundo robô da célula 1309	55
3.44	Ferramenta Estrela de 9 dedos	56
3.45	Registo de TCPs do terceiro robô da célula 1309	56
3.46	Localização do TCP do acoplamento do terceiro robô	57
3.47	Localização do TCP da ferramenta-estrela 9 dedos do terceiro robô	57

3.48	Cinemática de inserção de clips	58
3.49	Ferramenta de aplicação de tabs e de esponjas	59
3.50	Registo de TCP's do quarto robô da célula 1309	60
3.51	Localização dos TCPs "Tool_Tabs1"do quarto robô	60
3.52	Localização dos TCPs "ToolVac_1"e "ToolVac_2"do quarto robô	61
3.53	Colisão durante a inserção de fita na célula 1309	61
3.54	Representação da ferramenta da primeira versão e da última	62
3.55	Colisão na aplicação de tabs na célula 1309	62
3.56	Layout da célula PRJ 1311	63
3.57	Representação da célula 1311 por áreas	64
3.58	Fluxograma de funcionamento geral da célula 1311	64
3.59	Localização do TCP do acoplamento do primeiro robô da célula 1311	65
3.60	Localização do TCP da "Estrela de 10 dedos" do primeiro robô da célula 1311	65
3.61	Localização dos novos TCPs da ferramenta "Estrela de 10 dedos" do robô nº 1 da célula 1311	66
3.62	Localização do TCP do acoplamento do segundo robô da célula 1311	67
3.63	Localização dos novos TCPs da ferramenta "Estrela de 16 dedos" do robô nº 2 da célula 1311	67
3.64	Alteração da ferramenta do robô número 2 da célula 1311	68
4.1	Exemplo do ângulo incidência do laser na peça	70
4.2	Robô colaborativo JAKA ZU-12	70
4.3	Exemplo de programação dos robôs JAKA	71
4.4	Inicializações do programa da segunda proposta	72
4.5	Representação 2D do círculo	73
4.6	Ilustração das mudanças na peça por fases	73
4.7	Alimentador de peças	74
4.8	Gabarito número 1	75
4.9	Gabarito número 2	75
4.10	Fluxograma do funcionamento geral da máquina em um ciclo de operação	76
4.11	Estrutura do alimentador da CNC	76
4.12	Tabuleiro com os quatro <i>blisters</i>	77
4.13	Definição de <i>user frames</i>	78
4.14	Configurações da carga da ferramenta	78
4.15	Configuração do programa de arranque	79
4.16	Layout do projeto de visão	80
4.17	Calha de armazenamento de peças	81
4.18	Silo e Base	82
4.19	Endereço IP da câmara	83

4.20	Configuração das ligações da rede	83
4.21	Configuração da LAN 2 do controlador do robô	84
4.22	Ambiente de parametrização da câmara	84
4.23	Parâmetros de calibração	85
4.24	Algoritmo de detecção de peças	86
4.25	Parâmetros do bloco de captura de figura	86
4.26	Parâmetros do bloco de reconhecimento de formas	87
4.27	Parâmetros de comunicação	88
4.28	Ponto de calibração da peça	88
4.29	Deteção de uma peça	89
4.30	Coordenadas da peça de calibração	90
4.31	Resultado da detecção de peças	90
4.32	Estrutura da conformadora	91
4.33	Botoneira de seleção de modos de funcionamento	92

LISTA DE ABREVIATURAS, ACRÓNIMOS E SIGLAS

ABB	<i>Asea Brown Boveri</i>
CNC	<i>Computer Numerical Control</i>
COBOT	<i>Collaborative Robot</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
EMAF	Exposição Internacional de Máquinas e Equipamentos para a Indústria
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IO	<i>Input/Output</i>
IP	<i>Internet Protocol</i>
IPC	Instituto Politécnico de Coimbra
IR	<i>Infrared</i>
ISEC	Instituto Superior de Engenharia de Coimbra
LCD	<i>Liquid Crystal Display</i>
LIDAR	<i>LIght Detection And Ranging</i>
MEE	Mestrado em Engenharia Eletrotécnica
PLC	<i>Programmable Logic Controller</i>
PPP	<i>Prismatic Prismatic Prismatic</i>
RPP	<i>Revolute Prismatic Prismatic</i>
RRP	<i>Revolute Revolute Prismatic</i>
SCARA	<i>Selective Compliance Assembly Robot Arm</i>
TCP	<i>Tool Center Point</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
VW	<i>Volkswagen</i>

1 INTRODUÇÃO

A crescente automatização dos processos industriais, observada nas últimas décadas, impulsionou significativamente o desenvolvimento e a implementação de soluções robóticas industriais avançadas, com o objetivo de aumentar a produtividade, a precisão e a flexibilidade das linhas de produção. Neste contexto, o presente relatório de estágio curricular do mestrado descreve a participação do autor em vários projetos de natureza industrial, com ênfase na programação, simulação e integração de sistemas robotizados. Para tal, foram utilizadas tecnologias de ponta, nomeadamente o software RobotStudio, robôs colaborativos e sistemas de visão artificial, com vista à otimização e modernização dos processos produtivos dos clientes da empresa de acolhimento do estágio.

Ao longo do estágio, foram desenvolvidas três células de produção automatizadas para um cliente da indústria automóvel. Nas duas primeiras células (PRJ 1308 e PRJ 1311), o foco principal incidiu sobretudo na simulação dos movimentos robóticos, na programação das sequências de operação e na deteção de colisões entre os diferentes elementos do sistema, recorrendo ao ambiente virtual RobotStudio, da Asea Brown Boveri (ABB). Estas duas células partilham funcionalidades semelhantes, nomeadamente a manipulação de peças e a inserção de componentes, como clips e brackets, utilizando robôs industriais da marca ABB.

A célula PRJ 1308 é composta por três robôs ABB, programados para realizar tarefas como a manipulação de peças, a colocação de brackets e a inserção de clips. Já a célula PRJ 1311, apesar de mais simples na sua composição, executa igualmente tarefas de manipulação e inserção de clips, com foco na otimização do espaço de trabalho e na coordenação dos movimentos dos robôs, para evitar colisões.

Por outro lado, a célula PRJ 1309 apresenta uma maior complexidade, envolvendo diversas etapas do processo produtivo, nomeadamente: manipulação de peças, aquecimento da área de aplicação da fita, colocação da fita adesiva, aplicação das esponjas, inserção dos clips e colocação dos tabs. Esta célula evidencia a integração de múltiplos sistemas e operações sequenciais, requerendo uma lógica de programação robusta e rigorosa para garantir a fluidez e a segurança da operação.

Além das células anteriormente mencionadas, foi também desenvolvido um projeto inovador de soldadura a laser com um robô colaborativo. O principal desafio deste projeto consistiu na integração de um robô colaborativo fornecido pelo cliente num processo automatizado de soldadura de peças cilíndricas, cumprindo requisitos técni-

cos específicos. O projeto em questão destaca-se pela sua complexidade técnica, pela natureza colaborativa do robô e pelo tipo de soldadura envolvido, que exige extrema precisão e controle.

Outro contributo relevante deste trabalho foi o desenvolvimento de um sistema de alimentação automatizada de máquinas *Computer Numerical Control* (CNC) utilizando um robô colaborativo, com o objetivo de reduzir o tempo de inatividade da máquina e otimizar o fluxo de produção.

Por fim, foi realizada a parametrização de um sistema de visão artificial, integrado num sistema robotizado de deteção de peças, que permite identificar e localizar com precisão os componentes a manipular, aumentando a flexibilidade e autonomia do processo automatizado.

Foi também implementada uma conformadora de caixas com recurso a um robô colaborativo. Para este efeito foi desenvolvido e ajustado um sistema de automação que permite formar caixas de forma eficiente, segura e repetitiva, com potencial aplicação em linhas de embalagem ou logística.

O presente relatório de estágio visa, documentar todas estas atividades, detalhando os métodos utilizados, os desafios enfrentados e os resultados obtidos, contribuindo assim para o desenvolvimento de soluções de automação mais inteligentes, seguras e eficientes na indústria moderna.

1.1 Enquadramento

Embora a robótica não seja um tema recente, foi nas últimas décadas que se assistiu a uma evolução significativa nesta área. A sua origem remonta a 1954, com a criação do primeiro robô industrial, o "Unimate", que marcou o início da automação nas linhas de produção[1].

Com o avanço da tecnologia e as exigências crescentes do mercado, tornou-se essencial encontrar soluções que respondessem a critérios rigorosos, como a redução do tempo de fabrico, o aumento da qualidade e a flexibilidade na produção. Estes desafios impulsionaram o investimento no desenvolvimento de novas soluções robóticas.

Os robôs têm a vantagem de manterem elevados níveis de produtividade e qualidade em tarefas que anteriormente exigiam a intervenção de vários operadores, reduzindo o risco de erro humano devido à falta de atenção ou à fadiga. Contudo, o aumento do investimento em células robotizadas também levanta importantes dilemas, como a necessidade de equilibrar os ganhos de eficiência com a preservação dos postos de trabalho, perante o crescimento do uso da automação[2].

A automação e a robótica contribuem para o aumento da produtividade, a melhoria da qualidade dos produtos e a criação de condições de trabalho mais seguras e ergo-

nómicas. Ao automatizar tarefas repetitivas e que exigem elevada precisão, estas tecnologias reduzem os erros humanos e promovem uma produção mais eficiente. Além disso, ao substituírem os trabalhadores em tarefas perigosas, exaustivas ou repetitivas, contribuem para uma melhoria significativa da segurança e da ergonomia no ambiente de trabalho. No entanto, um dos grandes desafios atuais da robótica consiste no desenvolvimento de ferramentas eficazes para a manipulação de peças, sobretudo para ambientes complexos e variáveis.

No que toca ao futuro da robótica prevê-se um cenário promissor e em constante evolução. Com os avanços na inteligência artificial, machine learning e sensores avançados, os robôs tornar-se-ão cada vez mais autónomos, adaptáveis e capazes de colaborar de forma segura com os seres humanos.

1.2 Empresa de acolhimento

A Fluidotronica [3] foi a empresa na qual realizei o estágio constante do plano do curso de mestrado em Engenharia Eletrotécnica, ramo de Automação e Comunicações em Sistemas Industriais. Fundada em 2004 e situada em Oliveira de Azeméis, a Fluidotronica é uma empresa que se dedica à automação e à robótica, tendo como objetivo oferecer produtos e serviços de excelência que não só atendam, mas superem as expectativas dos seus clientes. Essa é a base de todas as suas ações e decisões.



Figura 1.1: Instalações da empresa Fluidotronica

A Fluidotronica dedica-se à criação de soluções "chave na mão", contando com o estudo do processo, desenvolvimento de projetos mecânicos e elétricos e elaboração de processos desde a maquinação, programação e eletrificação até à montagem final e testes. A empresa está focada principalmente no desenvolvimento de soluções para a indústria automóvel, mas também para as indústrias alimentar e farmacêutica.

Para finalizar, refira-se que a Fluidotronica é portadora da certificação PME Líder, um selo de reputação de empresas criado pelo IAPMEI para distinguir o mérito das PME

nacionais com desempenhos superiores. Este selo é atribuído em parceria com o Turismo de Portugal e um conjunto de bancos parceiros, tendo por base a avaliação das melhores notações de *rating* e indicadores económico-financeiros.

1.2.1 História

Fundada em 2004, a Fluidotronica iniciou a sua atividade com uma forte rede de parcerias estratégicas com empresas de renome, nomeadamente a MiniTec, a HepcoMotion, a Fipa e a Stoger Automation. Apenas um ano depois, a empresa expandiu as suas operações, estabelecendo uma colaboração com a Bahr Modultechnik, e mudou-se para as instalações em Vilar, Oliveira de Azeméis. Este marco refletiu a ambição da empresa em crescer e oferecer soluções cada vez mais inovadoras.

Ao longo dos anos, a Fluidotronica consolidou a sua posição no mercado através de novas parcerias, com a Weiss em 2007 e com a Omnitrack em 2010, com vista a expandir a sua oferta de produtos e serviços. A empresa destacou-se também pela qualidade e excelência dos seus processos, tendo sido distinguida pela primeira vez com o estatuto de PME Excelência em 2013, um reconhecimento que se repetiria nos anos seguintes.

A década de 2010 ficou marcada por investimentos em infraestruturas e qualidade. Em 2015, a Fluidotronica obteve o certificado do Sistema de Gestão da Qualidade e, em 2017, deu um passo significativo com a mudança para novas instalações na Zona Industrial de Ouriçosa, em Oliveira de Azeméis, aumentando a sua capacidade de produção e de inovação. Em 2018, a empresa ampliou a sua atividade ao diversificar os seus serviços com a inclusão da impressão 3D, tendo também recebido a certificação ISO 14001:2015, que reconhece o seu compromisso com a gestão ambiental.

A busca constante pela inovação foi reafirmada em 2019, ano em que a empresa obteve a certificação no âmbito do Sistema de Gestão da Investigação, Desenvolvimento e Inovação. Este movimento demonstrou a importância que a Fluidotronica atribui ao desenvolvimento contínuo de soluções tecnológicas. Em 2022, a parceria com a JAKA ampliou ainda mais o portfólio de soluções oferecidas pela empresa.

A excelência e o reconhecimento contínuo da empresa são evidenciados pelo facto de ter sido distinguida com o estatuto de PME Líder em vários anos, nomeadamente em 2010, 2011, 2012, 2013, 2015 e 2018. Além disso, a presença regular na Exposição Internacional de Máquinas e Equipamentos para a Indústria (EMAF) reforçou a posição da empresa como um dos principais *players* do setor.

Com um percurso marcado pela inovação, qualidade e crescimento, a Fluidotronica consolidou-se como uma referência no setor industrial, procurando sempre oferecer soluções avançadas e sustentáveis aos seus clientes. O Anexo A ilustra a evolução da empresa desde a sua fundação.

1.2.2 Serviços

A Fluidotronica é uma empresa que oferece diferentes produtos "chave na mão", os quais estão categorizados na figura 1.2:

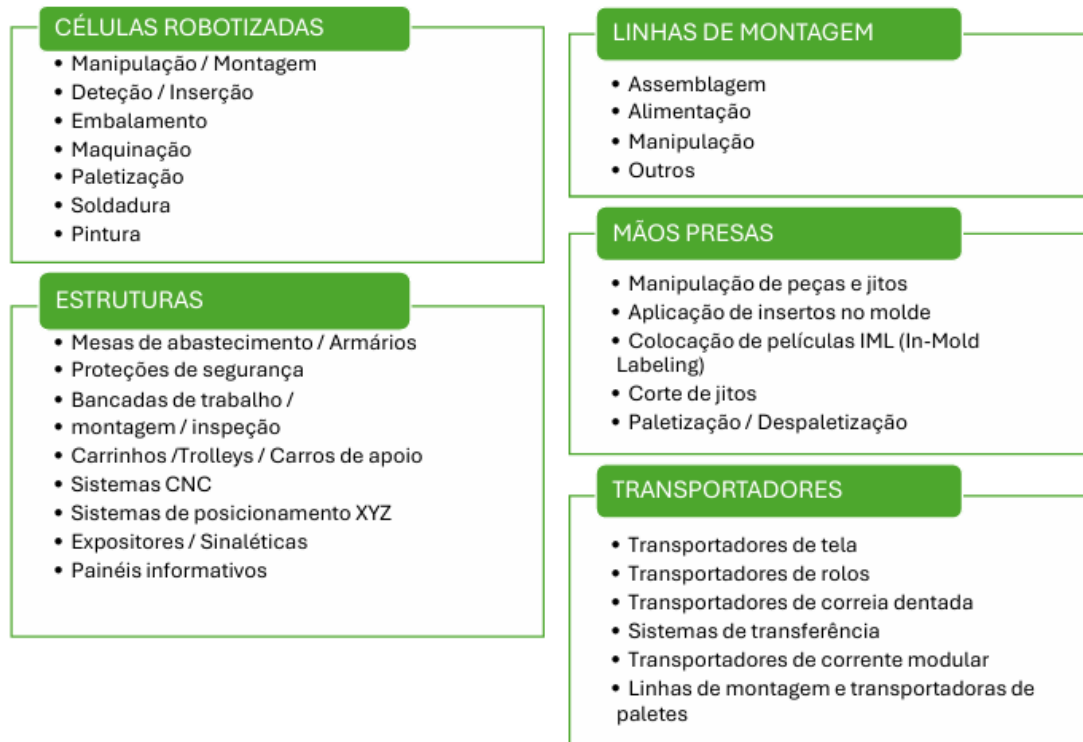


Figura 1.2: Serviços disponibilizados pela empresa Fluidotronica

1.3 Objetivos

O estágio compreendeu a realização de um conjunto de tarefas relacionadas com o desenvolvimento de soluções proprietárias da empresa Fluidotrónica na área da robótica industrial, em colaboração com a respetiva equipa de desenvolvimento. As tarefas realizadas tiveram suporte teórico na formação ministrada no curso de mestrado em Engenharia Eletrotécnica do Instituto Superior de Engenharia de Coimbra (ISEC).

Durante o estágio, foram alcançados os seguintes objetivos específicos:

- Conhecer a organização e o funcionamento de uma empresa;
- Desenvolver competências transversais (aptidões interpessoais);
- Adquirir conhecimentos técnicos na área da robótica industrial;
- Conceber, implementar e testar soluções proprietárias de robótica industrial, nomeadamente para clientes da indústria automóvel.

1.4 Plano de Estágio

O estágio teve início em 16 de outubro de 2024 e terminou a 27 de junho de 2025.

1.4.1 Tarefas realizadas

Entre as tarefas realizadas, destacam-se:

- Introdução à programação de robôs colaborativos (COBOTS);
- Simulação de trajetórias e identificação de colisões;
- Configuração e programação de diversos manipuladores de acordo com as funcionalidades a desempenhar: Montagem de fitas, clips e tabs;
- Configuração e programação de COBOTS para realizar soldaduras em diferentes peças;
- Configuração da câmara de visão;
- Estabelecimento de comunicação entre sistemas de visão e robôs;
- Interpretação e aplicação de sistemas de segurança;
- Realização de ensaios do projeto.

1.4.2 Diagrama de Gantt

Para garantir uma visão clara e organizada das atividades desenvolvidas ao longo do estágio na empresa, foi elaborado um diagrama de Gantt que apresenta a distribuição temporal das principais tarefas realizadas. Este planeamento permitiu acompanhar a evolução do trabalho, definir prioridades e assegurar o cumprimento dos prazos estabelecidos. No diagrama, encontram-se representadas as fases essenciais do estágio, desde a integração inicial na equipa até à execução dos projetos atribuídos, proporcionando uma leitura simples e objetiva da gestão do tempo e das etapas concluídas. O respetivo diagrama encontra-se no Anexo J.

1.5 Estrutura do Relatório

O segundo capítulo apresenta o estado da arte e os conceitos abordados, nomeadamente os conceitos teóricos relativos aos tipos de robôs, à arquitetura, aos componentes, aos tipos de ferramentas e às linguagens de programação.

O terceiro capítulo aborda a minha participação no projeto da Indea, que consiste em três células robotizadas, para as quais efetuei a simulação das referidas células.

O quarto capítulo aborda uma série de pequenos projetos que consistem na programação de robôs colaborativos para executar funções específicas.

No último capítulo, são apresentadas as considerações finais e conclusões dos diferentes projetos desenvolvidos durante o estágio.

2 ESTADO DE ARTE E CONCEITOS

No presente capítulo, serão abordados os tipos de robôs existentes no mercado, bem como algumas das suas linguagens de programação e os seus componentes.

2.1 Tipos de robôs

Ao longo dos anos, foram criados vários tipos de robôs com diferentes arquiteturas, adaptados a tarefas e espaços diferentes. Assim, temos os seguintes tipos de robôs[4]:

- Articulados;
- Cartesianos;
- Cilíndricos;
- Polares;
- *Selective Compliance Assembly Robot Arm* (SCARA);
- Delta;
- Colaborativos.

2.1.1 Robôs articulados

Os robôs articulados são robôs que utilizam um sistema de juntas semelhante ao dos membros do ser humano. Os seus eixos rotativos conferem-lhes um maior grau de liberdade, permitindo uma maior precisão, repetibilidade e flexibilidade. Podem ter um número de juntas compreendido entre quatro e sete[5]. A figura 2.1[6] representa um robô articulado da marca KUKA.



Figura 2.1: Robô da ABB, o IRB 1300-10/1.15[7]

São principalmente aplicados em funções como o manuseamento de cargas, a soldadura, a pintura, a paletização, a alimentação de máquinas e a montagem de equipamentos.

2.1.2 Robôs cartesianos

Os robôs cartesianos diferem tanto nas aplicações como no esquema de juntas utilizado. Ao contrário do modelo anterior, que era composto por três juntas lineares prismáticas, formando o X, Y e o Z. Movem-se com a ajuda de servomotores ou motores de passo, que podem estar acoplados a correias, ou a um veio sem fim. Esta arquitetura permite elaborar trajetórias de grande precisão, aplicáveis a funções como a maquinação, o manuseamento de materiais, a colocação e a retirada de peças (*pick and place*) ou o corte[8]. A figura 2.2 ilustra um robô cartesiano com um sistema de veio sem fim:



Figura 2.2: Robô cartesiano da marca Campetella, modelo CO1-CO2 X-Series[9]

2.1.3 Robôs cilíndricos

Os robôs cilíndricos são facilmente identificáveis pela sua área de trabalho, que neste caso, tem a forma de um cilindro. A sua construção consiste num eixo rotacional e dois eixos prismáticos. A figura 2.3 ilustra dois exemplos de robôs cilíndricos segundo[10].



Figura 2.3: Exemplo de robôs cilíndricos[11]

2.1.4 Robôs polares

Os robôs polares têm uma área de trabalho com a forma de uma esfera e utilizam um sistema de coordenadas polares. Quanto à sua constituição mecânica, estes robôs utilizam uma junta de torção que liga a duas juntas rotativas[10], conforme ilustrado na figura 2.4.

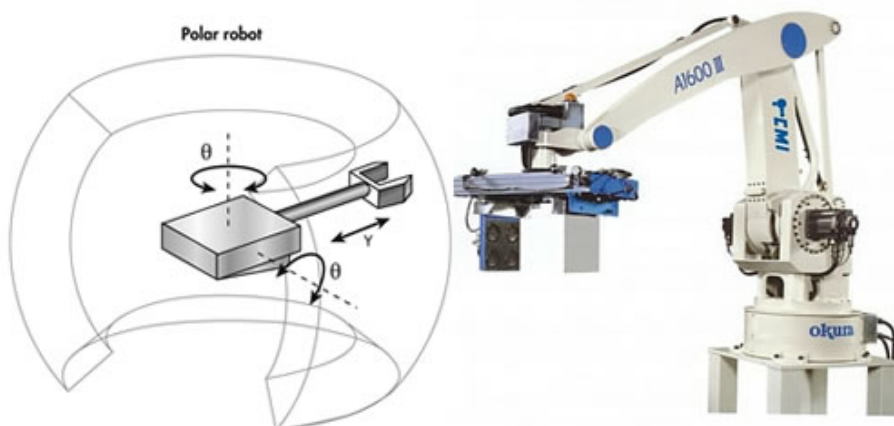


Figura 2.4: Robô do tipo polar da marca Okura, modelo A1600 II e respetiva área de trabalho[10]

2.1.5 Robôs SCARA

Os robôs SCARA, constituídos por quatro eixos, são os mais utilizados para operações de *pick and place*, devido à sua área de trabalho em forma de anel horizontal, que é ideal para movimentos repetitivos em espaços reduzidos. São constituídos por duas ou três juntas rotacionais e um eixo vertical[10]. A figura 2.5 mostra um robô SCARA e a respetiva área de trabalho.



Figura 2.5: Robô do tipo SCARA e sua área de trabalho[10]

2.1.6 Robôs delta

Os robôs delta são concebidos para serem rápidos e precisos, sendo os indicados para operações de *pick and place*. São constituídos por três ou quatro braços paralelos entre si, o que lhes confere uma baixa inercia, permitindo atingir velocidades superiores às dos outros robôs[12]. A figura 2.6 mostra um exemplo de um robô do tipo delta.



Figura 2.6: Robô do tipo delta da ABB, modelo IRB 360 FlexPicker[13]

2.1.7 Robôs colaborativos

Os robôs colaborativos não passam de robôs articulados, que foram concebidos para trabalhar ao lado dos seres humanos, ao contrário dos robôs clássicos, que estão confinados em jaulas. Como trabalham em colaboração com os seres humanos, estes robôs operam a velocidades mais reduzidas e possuem mecanismos de paragem mais sensíveis em caso de colisão. A figura 2.7 ilustra um robô colaborativo.



Figura 2.7: Robô colaborativo da marca JAKA, modelo Zu 12[14]

2.2 Tipos de juntas

2.2.1 Junta rotacional

As juntas rotacionais ou revolucionárias consistem num elo que roda em torno de um eixo. Este tipo de movimentos são realizados com o eixo de rotação perpendicular ao seu eixo[15].

Na figura 2.8 está representada uma junta rotacional.

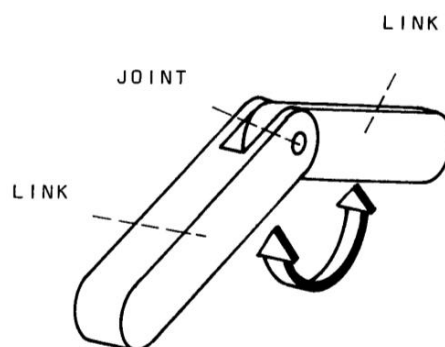


Figura 2.8: Junta rotacional[16]

2.2.2 Junta prismática

As juntas prismáticas permitem que dois elos deslizem entre si de forma linear. Estas juntas podem ser perpendiculares ou deslocarem-se em sentidos opostos[17]. A figura 2.9 apresenta dois exemplos de juntas prismáticas.

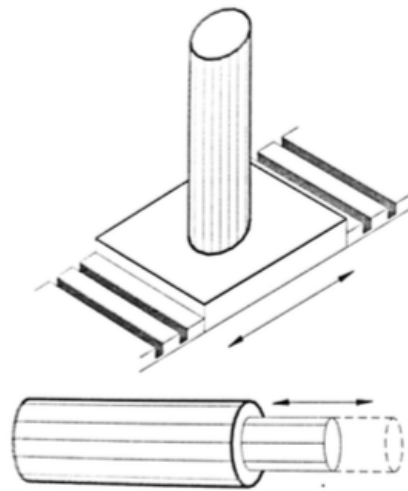


Figura 2.9: Juntas prismáticas[18]

2.2.3 Juntas esféricas

As juntas esféricas permitem uma movimentação nos três eixos, o que equivale a três graus de liberdade, como ilustra a figura 2.10. São semelhantes a um pulso humano, com a limitação de uma amplitude de movimento mais reduzida.

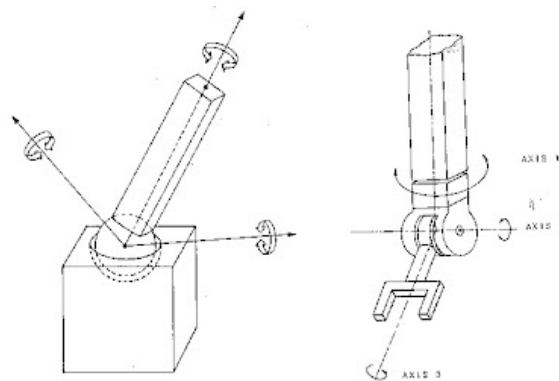


Figura 2.10: Juntas esféricas[19]

2.3 Tipos de estrutura cinemática

Em robótica, a estrutura de cinemática de um robô refere-se à forma como os seus elos e articulações estão organizados para permitir movimentos no espaço. Esta estrutura define os movimentos que o robô pode realizar, a sua área de trabalho e a sua aplicação ideal. As três configurações mais comuns são: a esférica (RRP), a cartesiana (PPP) e a cilíndrica (RPP), cada uma com características distintas.

A nomenclatura da estrutura depende sempre do número de juntas e do respetivo tipo, subdivididas em:

- R - *Revolute*;
- P - *Prismatic*.

2.3.1 Esférica (RRP)

A configuração esférica é composta por duas juntas rotativas (R), seguidas de uma junta prismática (P), que permite o movimento linear. Esta estrutura permite que o robô se mova em torno de um ponto central e se estenda radialmente a partir deste, de forma semelhante ao movimento de um braço humano que gira no ombro e estica o braço para a frente[20]. Os movimentos possíveis desta estrutura são:

- Rotação horizontal;
- Rotação vertical;
- Extensão linear.

A sua área de trabalho tem a forma de um segmento de esfera.

Principais aplicações: ideal para tarefas que exigem acesso a uma área ampla a partir de um ponto fixo, como a pintura e a soldadura.

2.3.2 Cartesiana (PPP)

A estrutura cartesiana é constituída por três juntas prismáticas (P), que permitem movimentos lineares ao longo dos eixos X, Y e Z. Este tipo de robô move-se apenas em linhas retas e é conhecido pela sua simplicidade e precisão[20]. Os movimentos desta estrutura são os seguintes:

- Translação linear ao longo de três eixos ortogonais;

Possui uma área de trabalho em forma de paralelepípedo (retangular) e tem como principais aplicações:

- Impressões 3D;
- Sistemas CNC;

- Sistemas de armazenamento de caixas;
- Processos de montagem que exigem alta precisão.

2.3.3 Cilíndrica (RPP)

A estrutura cilíndrica (RPP) combina uma junta rotativa (R) com duas juntas prismáticas (PP). Tal permite ao robô girar em torno de um eixo, mover-se verticalmente e estender-se para a frente ou para trás [20]. Esta estrutura permite os seguintes movimentos:

- Rotação em torno do eixo base;
- Movimento linear vertical e radial.

A área de trabalho tem a forma de um cilindro. As suas principais aplicações são:

- Tarefas como o empacotamento;
- Manuseamento de materiais;
- Montagem em linhas de produção.

2.4 Principais partes de um robô

Os robôs são sistemas compostos por vários componentes que trabalham em conjunto para executar tarefas. São constituídos por diversas partes que desempenham funções específicas, desde a deslocação até à interação com o ambiente. Por conseguinte, é necessário compreender as partes constituintes dos robôs [21].

- Controlador;
- Esqueleto;
- Atuadores;
- Sensores;
- Manipulador.

2.4.1 Controlador

O controlador é a unidade responsável pelo processamento dos vários tipos de dados do robô. A nível de hardware, o controlador possui uma unidade de processamento, sistemas de memória volátil e não volátil, sistemas de alimentação e conversão de energia, uma interface de I/O e sistemas de comunicação[22]. A figura 2.11 apresenta o controlador da "IRC 5" da ABB.



Figura 2.11: Controlador IRC5 da ABB[23]

2.4.2 Esqueleto do robô

Tal como no corpo humano, o esqueleto de um robô é a estrutura física que suporta os seus componentes e possibilita a sua movimentação. O esqueleto do robô suporta os atuadores, os sensores e os seus sistemas eletrónicos e pode ser fabricado em diversos materiais, como metal ou plástico, consoante a sua finalidade. Além disso, o seu design pode variar de acordo com as necessidades, podendo incluir articulações, eixos e suportes específicos para garantir estabilidade e funcionalidade[22].

2.4.3 Atuadores

Os atuadores são componentes fundamentais responsáveis pela execução de movimentos e pela interação do sistema com o ambiente externo. Eles convertem energia (geralmente elétrica, pneumática ou hidráulica) em movimento mecânico, permitindo que o sistema realize ações específicas, como deslocamento, rotação ou aplicação de força. Os atuadores podem variar amplamente consoante o tipo de energia utilizada e a aplicação pretendida. Entre os tipos mais comuns, destacam-se os seguintes[22]:

- Cilindros pneumáticos: utilizam ar comprimido para gerar um movimento linear e são amplamente utilizados em sistemas que requeiram alta velocidade e uma força moderada;
- Cilindros hidráulicos: utilizam um fluido pressurizado para gerar movimentos com grande força, sendo ideais para aplicações que requeiram elevados níveis de

potência, como em maquinaria pesada;

- Motores elétricos: convertem energia elétrica em movimento rotacional ou linear. Estes atuadores são essenciais em sistemas automatizados e robóticos, que exijam uma grande precisão e controlo.

Além destes, existem outros tipos de atuadores, como os motores de passo, os atuadores térmicos e os motores lineares, cada um com características e aplicações específicas, que lhes conferem flexibilidade e adaptabilidade a diferentes necessidades industriais e robóticas.

2.4.4 Sensores

Em relação aos sensores, o robô pode ser equipado com sensores internos e externos. Os sensores internos fornecem informações sobre os parâmetros do próprio robô, por exemplo, o ângulo de uma junta do manipulador, a força aplicada por um motor numa junta ou a inércia de um movimento, entre outros.

Os sensores externos compreendem todos os que lidam com a "observação" do mundo exterior. Alguns exemplos de sensores externos são:

- Sensores de proximidade: detetam a presença ou ausência de objetos próximos, sem contacto físico;
- Sensores de força: medem a força aplicada numa área específica, geralmente em juntas ou superfícies;
- Sensores de distância: medem a distância entre o sensor e um objeto. Podem funcionar com diferentes tecnologias, como *Light Detection And Ranging* (LIDAR), infravermelho (IR ou *infrared*), ou radar;
- Sensores de ultrassons: usam ondas sonoras de alta frequência para medir a distância até um objeto. O sensor emite um pulso ultrassónico e mede o tempo que o eco demora a retornar;
- Outros.

2.4.5 Manipulador

O manipulador é a componente que permite a movimentação do robô, sendo normalmente composto por segmentos articulados e por um *end effector*, que pode ser uma garra, uma ventosa ou uma ferramenta específica para a realização de tarefas. A figura 2.12 ilustra a constituição do manipulador, segundo [24].

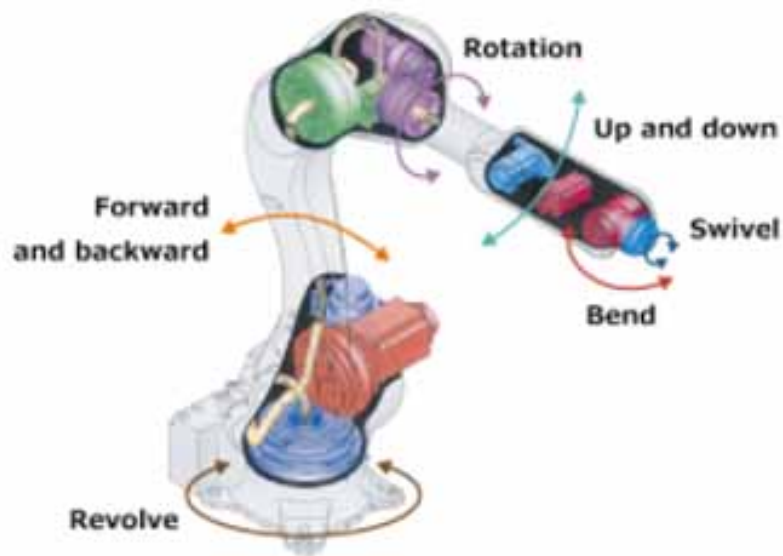


Figura 2.12: Exemplo de um manipulador

2.5 Tipos de ferramentas

Uma das partes mais complicadas da manipulação ou da realização de tarefas por parte de um robô é a escolha da ferramenta adequada.

A escolha da ferramenta adequada pode fazer a diferença entre ter de se escolher um robô com uma capacidade inferior ou superior. Parâmetros como o peso e o centro de massa influenciam a escolha do robô. Em regra, um robô que, por exemplo, suporta um peso de cinco quilos não significa que possa pegar numa ferramenta de cinco quilos. Quanto maior for a distância entre o flange do robô e o seu centro de massa, menor será o valor do peso que este suporta.

Um dos grandes desafios na criação da ferramenta é torná-la o mais leve possível e com o centro de massa o mais próximo possível do flange do robô, de modo a permitir ao cliente poupar dinheiro na compra do robô, no consumo de energia, na manutenção, no valor da ferramenta e no tempo de programação, visto que uma ferramenta maior pode obrigar o programador a gastar mais tempo, se o robô tiver de trabalhar num local com espaço reduzido.

A figura 2.13 mostra a influência da capacidade máxima de um robô no aumento do centro de massa em relação ao flange.

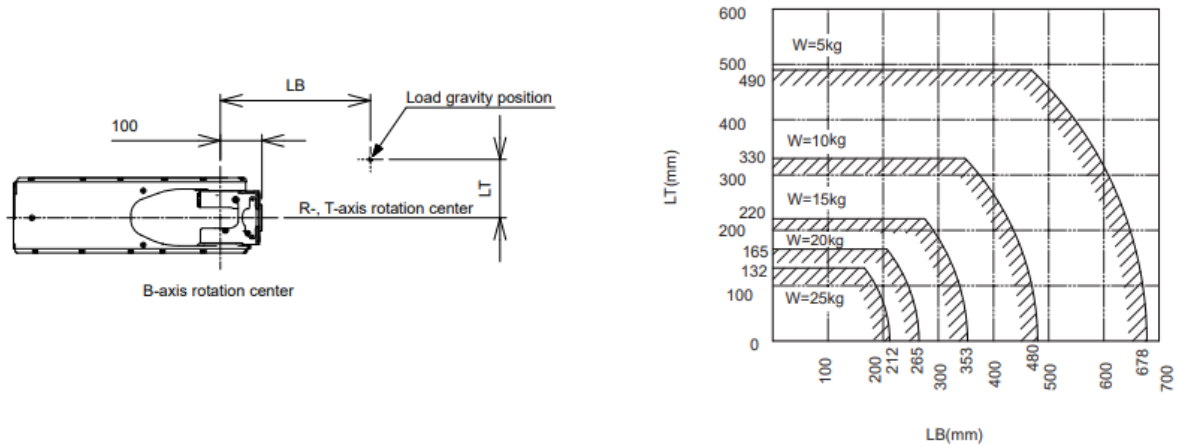


Figura 2.13: Diagrama de *payload* do robô GP25 da Yaskawa[25]

É possível encontrar vários tipos de ferramentas:

- Pinças de vácuo ou ventosas;
- Pinças pneumáticas;
- Pinças elétricas;
- Pinças magnéticas;
- Tocha de soldadura.

2.5.1 Configuração de ferramentas

A eficiência e a precisão com que o robô manipulador executa tarefas que dependem diretamente da forma como este é configurado para interagir com o ambiente. Entre essas configurações, refira-se as associadas às ferramentas acopladas ao robô: garras, tochas, sensores ou outros atuadores. Estas configurações, não só determinam a forma como o robô manipula objetos ou realiza operações, como também influenciam a segurança, o desempenho e a capacidade de adaptação a diferentes aplicações. A calibração das ferramentas permite:

- Marcação e ajuste de pontos diretamente com base no sistema de coordenadas da ferramenta. Essa abordagem pode representar uma mais-valia significativa, especialmente em termos de facilidade de operação e de redução do tempo gasto em ajustes;
- Todos os pontos previamente programados devem ser reajustados automaticamente, caso a ferramenta tenha de ser substituída;
- Considerar eventuais desvios entre a simulação e a realidade, dado que a ferramenta pode apresentar variações devido a tolerâncias de fabrico;
- Ajustar o binário e otimizar a elaboração das trajetórias do robô, tornando-as o

mais suaves possível, dado que se conhece o peso, a inércia e o centro de massa da ferramenta, o que contribui para o aumento da vida útil do robô.

2.6 Linguagens e ambientes de programação

Uma linguagem de programação de robôs é um conjunto de regras e instruções que permitem aos programadores controlar o comportamento, a lógica e a comunicação dos robôs para realizar tarefas específicas. As linguagens de programação têm a sua própria sintaxe e semântica, ou seja, um conjunto de regras que definem como os comandos e as estruturas devem ser organizados e interpretados para que o controlador entenda e execute as tarefas.

As linguagens de programação dos robôs dependem diretamente da marca do fabricante, sendo que cada fabricante tem a sua própria sintaxe. Na tabela 2.1 estão representadas as linguagens de programação usadas pelos principais fabricantes de robôs:

Tabela 2.1: Linguagens de programação das principais marcas de robôs [26]

Fabricante	Nome da Linguagem
KUKA	KRL
ABB	RAPID
YASKAWA	INFORM
FANUC	KAREL ou Teach Pendant

Porém, existem exceções, alguns fabricantes têm linguagens comuns que se aplicam geralmente a robôs colaborativos, os quais podem utilizar linguagens como o *python* ou *blockley*. Na robótica, existem vários métodos de programação [26]:

- *Online* - este método consiste na programação diretamente no robô, ou seja, quando está em operação, normalmente recorrendo a consolas que vêm incorporadas no controlador do robô.
- *Offline* - consiste na programação de robôs em ambientes virtuais, recorrendo a um computador. Esta abordagem permite simular as tarefas a executar, sendo o programa resultante posteriormente transferido para a memória do robô.
- Programação por demonstração - permite "ensinar" o robô a realizar tarefas por imitação de movimentos.

Um ambiente de programação de robótica é um conjunto de ferramentas e plataformas que permitem aos programadores criar e simular o comportamento dos robôs antes da sua implementação no mundo real. Isto permite a definição das ações e a configuração dos sensores e atuadores, simulando os movimentos do robô num ambiente virtual, antes da sua colocação em operação.

Na tabela 2.2 apresenta os diferentes ambientes de programação de diversos fabricantes.

Tabela 2.2: Principais softwares de programação segundo as marcas

Fabricante	Software
KUKA	KukaSim
ABB	RobotStudio
YASKAWA	MotoSim
FANUC	RoboGuide

3 DESENVOLVIMENTO DE SOLUÇÕES ROBÓTICAS PARA APLICAÇÕES INDUSTRIAIS

Este capítulo aborda o projeto principal do estágio curricular. O projeto em questão inclui várias células robotizadas, que serão divididas em partes, com as seguintes designações:

- Célula "PRJ 1308";
- Célula "PRJ 1309";
- Célula "PRJ 1311".

Todas as células deste projeto têm algo em comum, sendo que o principal objetivo compreende a inserção de clips e fita e tabs, em conjuntos de peças laterais dos frisos dos guarda-lamas de carros das marcas Volkswagen e Skoda. No entanto, as peças a serem trabalhadas são diferentes de célula para célula.

Por razões de sigilo, não é possível apresentar imagens das peças, dado que os modelos em produção ainda não estão disponíveis no mercado. Por conseguinte, serão utilizadas imagens de peças idênticas ou de modelos anteriores.

As figuras 3.1 e 3.2 mostram a aplicação de fita¹, de clips² e tabs³.

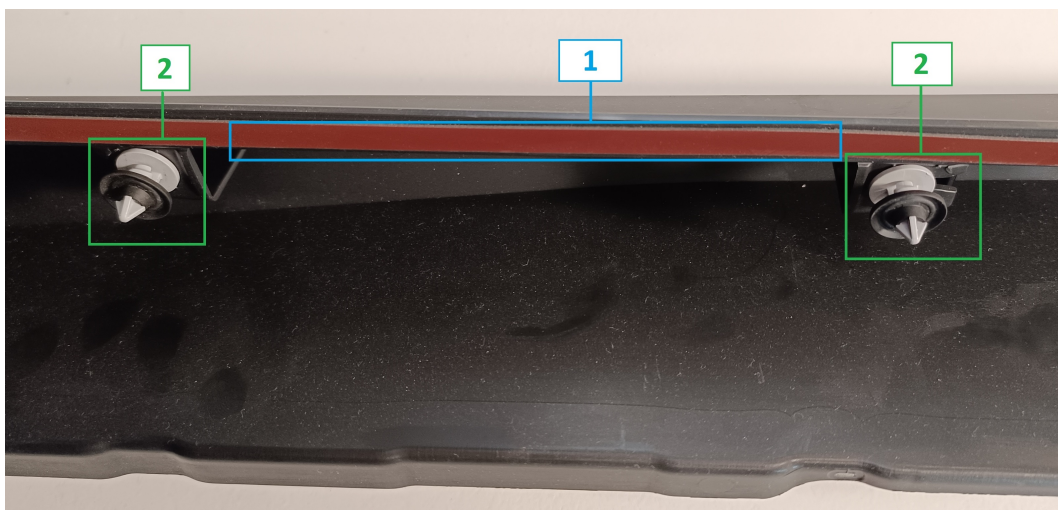


Figura 3.1: Peça de exemplo 1

¹A fita é um elemento que serve para uma maior fixação da peça ao veículo.

²Um clip é um elemento de fixação rápida de plástico, usado para unir peças sem parafusos.

³Um tab é uma pequena extensão saliente da peça plástica, usada para fixação parcial.

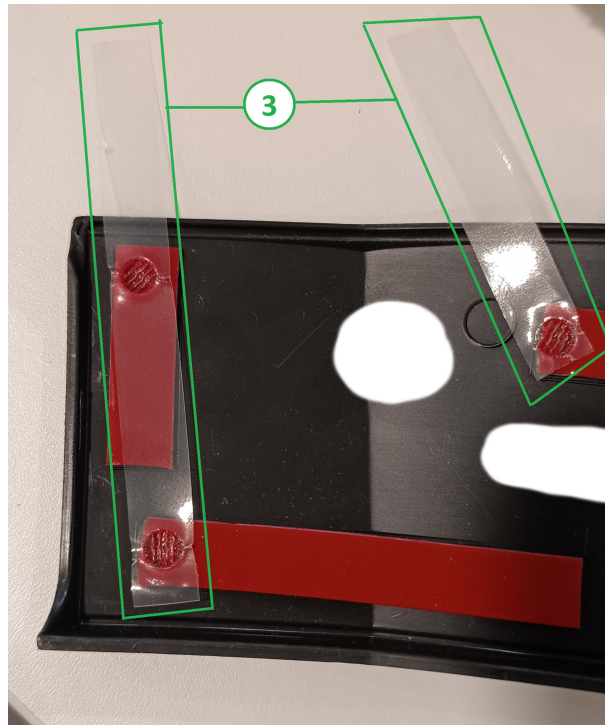


Figura 3.2: Peça exemplo 2

Legenda das figuras 3.1 e 3.2:

1. Fita;
2. Clips;
3. Tabs.

Para implementar o processo automatizado de inserção de clips foi necessário desenvolver uma ferramenta proprietária, designada por "estrela de 8 dedos", presente na figura 3.3.

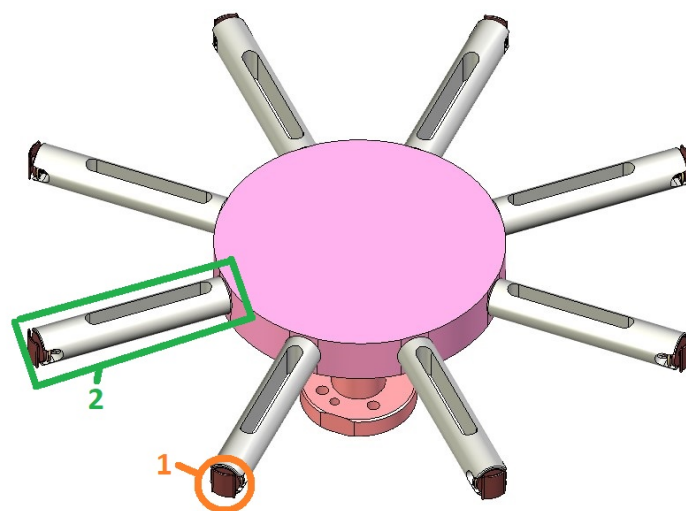


Figura 3.3: Exemplo da ferramenta-estrela de 8 dedos

Legenda da figura anterior:

1. Clip
2. Dedo

Esta ferramenta permite colocar clips nas peças e quanto maior for o número de dedos, maior será a sua capacidade de entrega de clips. O número de dedos está diretamente relacionado com a quantidade de clips que uma peça necessita, sendo que este valor varia de célula para célula.

A realização deste projeto compreende a execução das seguintes etapas:

- Carregamento dos ficheiros 3D;
- Criação dos controladores virtuais dos robôs na célula;
- Definição dos dados das ferramentas dos robôs;
- Criação dos *work objects* (em ABB ou *user frames* noutras marcas);
- Criação dos pontos das rotas;
- Programação do ficheiro RAPID e simulação;
- Carregamento dos ficheiros RAPID para o controlador;
- Ajuste dos pontos;
- Realização de testes.

3.1 Célula PRJ 1308

Este subcapítulo apresenta uma das células da linha de produção simuladas pelo autor, na ferramenta de programação e simulação RobotStudio, com o objetivo de programar o funcionamento dos robôs e detetar possíveis colisões.

3.1.1 Layout e descrição de funcionamento

A célula "PRJ 1308" possui três robôs da ABB os quais são responsáveis pelas seguintes tarefas:

- Manipulação de peças;
- Inserção de brackets;
- Inserção de clips.

Para além desses robôs, a célula inclui também um alimentador automático de clips, um dispensador de clips, um tapete de entrada e um tapete de saída. A figura seguinte representa o *layout* da célula PRJ 1308.

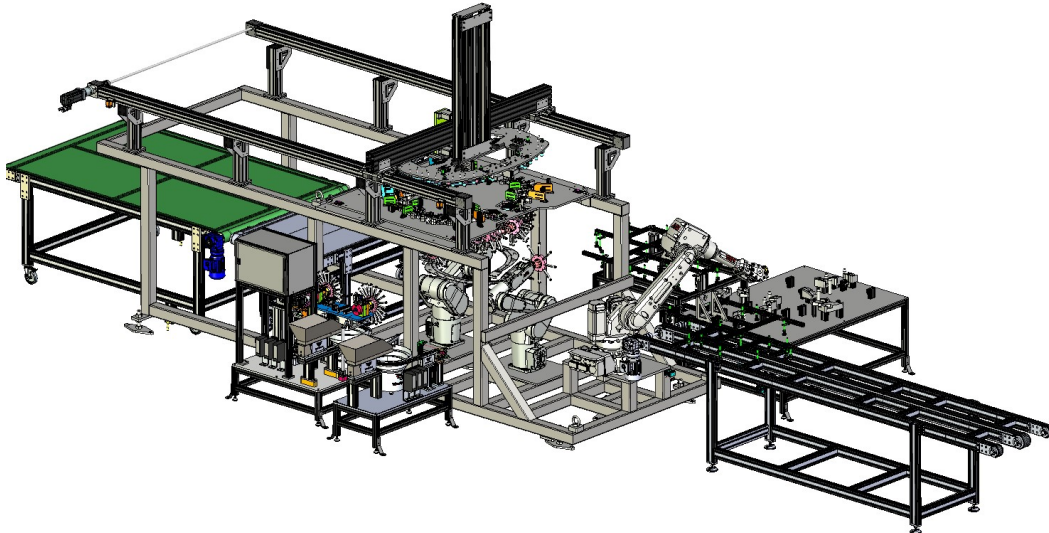


Figura 3.4: *Layout* da célula PRJ 1308

De seguida, serão explicadas as tarefas efetuadas por cada um dos robôs no processo de montagem que se pretende implementar com esta solução.

O primeiro robô, situado ao lado do tapete de entrada, possui duas ferramentas disponíveis: uma para a manipulação de peças (garra ventosa) e outra para a aplicação de brackets (garra acionada por um cilindro pneumático).

Se o robô não tiver qualquer ferramenta acoplada, este terá de se deslocar ao posto da ferramenta e acoplar a garra ventosa que irá manusear as peças, caso estas estejam presentes no tapete de entrada ou se houver um pedido para ir buscar peças à mesa de entrada manual. Posteriormente, as peças serão transportadas para a base, onde mais tarde serão prensadas pelo manipulador. Após concluir esta tarefa, o robô irá buscar a segunda ferramenta, responsável pela aplicação dos brackets e procederá à sua aplicação nas peças.

Os robôs dois e três trabalham simultaneamente na inserção de clips, porém, existem diferenças entre eles, tanto na ferramenta utilizada, como na forma de aplicação. O segundo robô trabalha apenas com uma ferramenta, denominada "estrela de oito dedos", enquanto o terceiro robô pode trabalhar com duas estrelas, que serão carregadas com clips, provenientes de um alimentador automático. Por sua vez, o segundo robô terá de fazer o carregamento por si próprio. O carregamento da estrela, consiste em colocar clips nos "dedos da ferramenta", processo esse que está exemplificado na figura 3.3.

O segundo robô aplicará os clips diretamente na peça final e nos brackets que já foram encaixados pelo primeiro robô na referida peça. A utilização de dois robôs para aplicar os clips tem como objetivo reduzir o tempo de ciclo. A figura 3.5 representa as áreas da célula, abordadas no texto anterior.

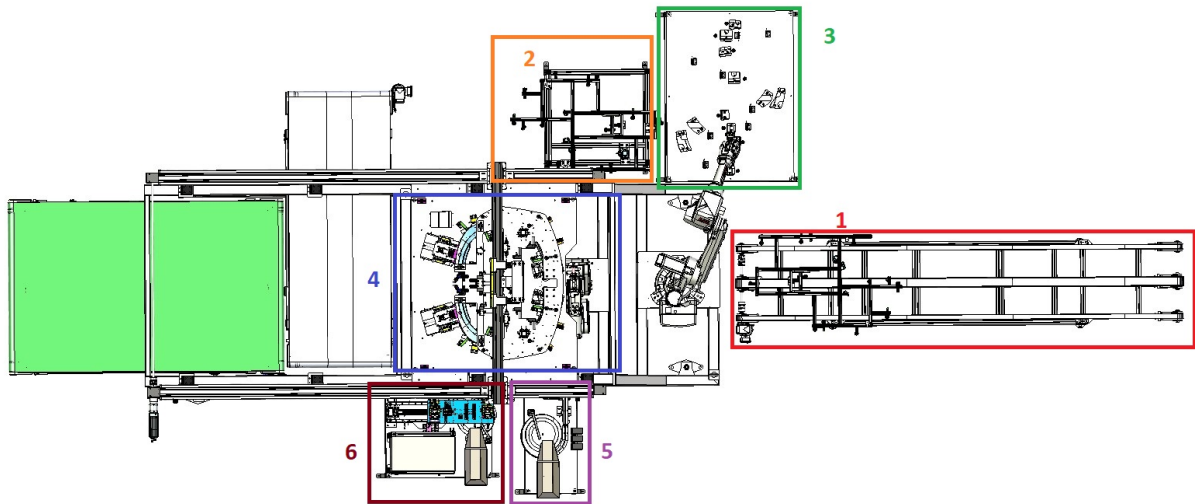


Figura 3.5: Representação da célula 1308 por áreas

Legenda da figura 3.5

1. Tapete de entrada de peças;
2. Zona de ferramentas;
3. Mesa de entrada manual de peças;
4. Base para aplicação das peças;
5. Dispensador de clips;
6. Máquina de alimentação automática das estrelas.

Na figura 3.6 está representado um fluxograma do funcionamento geral da célula.

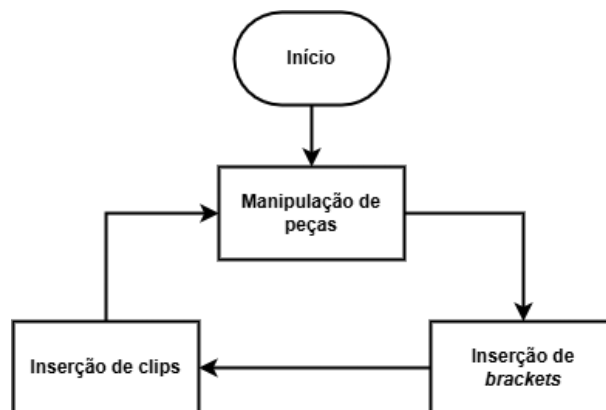


Figura 3.6: Fluxograma do funcionamento geral da célula 1308

Como foi referido, o carregador automático de clips que está representado na figura 3.7 é responsável pelo carregamento de clips numa estrela. Esta máquina permite reduzir o tempo do ciclo, uma vez que possibilita o carregamento de clips numa segunda estrela, enquanto o robô insere os clips disponíveis nas peças da estrela que está acoplada

ao robô. Quando a estrela do robô já não tiver clips disponíveis, a máquina e o robô trocarão de ferramentas, permitindo a aplicação de clips em contínuo.

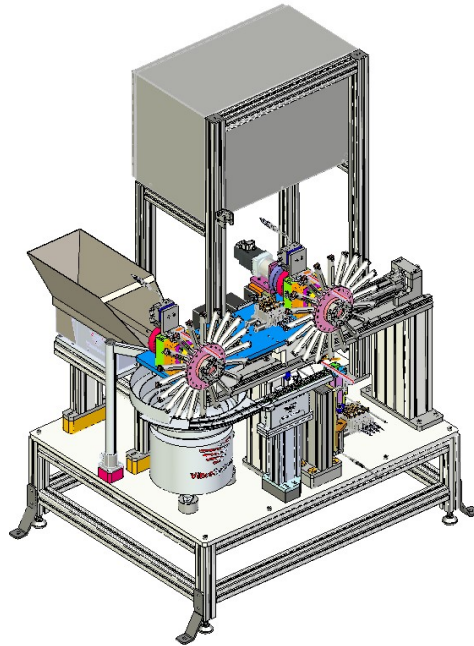


Figura 3.7: Carregador automático de clips

O dispensador de clips representado na figura 3.8 é responsável por fornecer clips ao robô para que este os carregue individualmente na estrela. O referido dispensador, que funciona por ação vibratória, consegue entregar os clips na posição e com a cadência solicitadas.

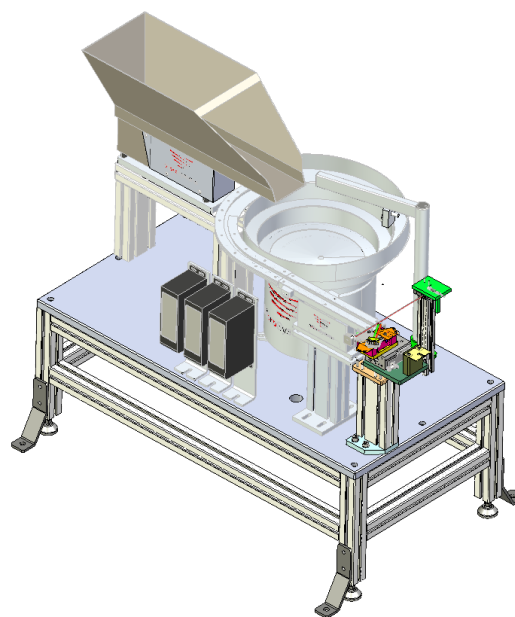


Figura 3.8: Dispensador de clips

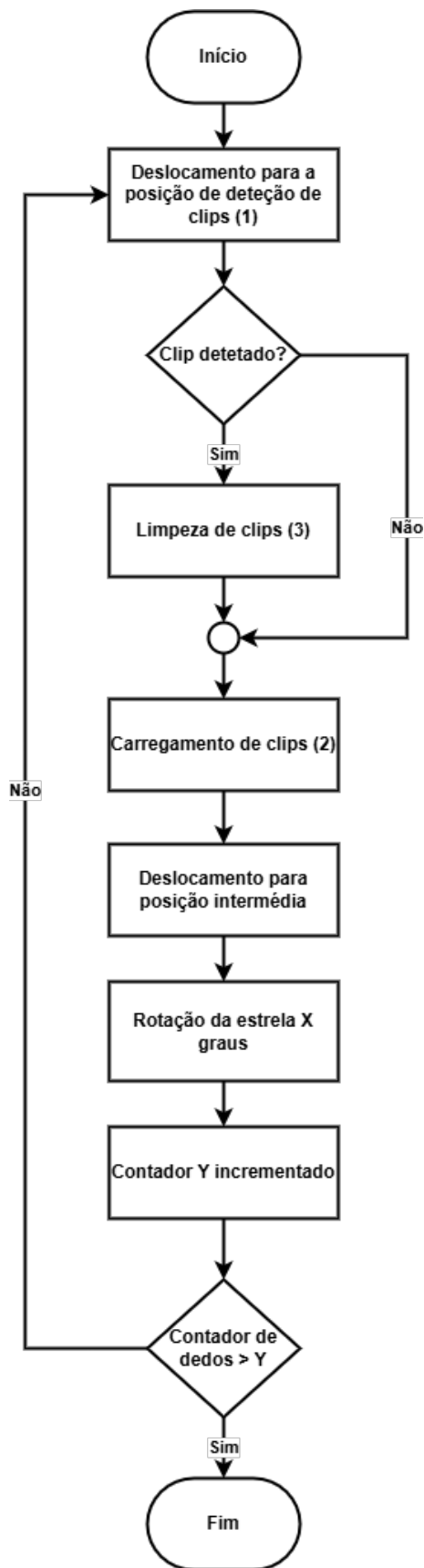


Figura 3.10: Fluxograma do ciclo de carregamento de clips

Sendo que X é dado pela seguinte fórmula:

$$X = \text{graus atuais do eixo 6} + \text{contador Y} * \frac{\text{número total de dedos}}{360}$$

E Y está dependente do número máximo de dedos que a estrela pode conter.

3.1.2 Calibração de ferramentas

A calibração correta do *Tool Center Point* (TCP) é essencial para garantir a precisão e a repetibilidade dos movimentos do robô, sobretudo em operações que envolvam a troca de ferramentas ou acoplamentos. Um TCP bem definido permite que o sistema compreenda com exatidão a posição e a orientação do ponto de atuação da ferramenta em relação ao flange do robô.

No caso dos robôs que utilizam um sistema de acoplamento, é necessário definir um TCP específico para o próprio acoplamento, uma vez que o robô será operado temporariamente sem ferramentas. Para garantir a consistência dos movimentos e facilitar o processo de alinhamento e acoplamento, definiu-se o TCP com a orientação do eixo Z sempre no sentido oposto ao flange, conforme ilustrado na figura 3.11.

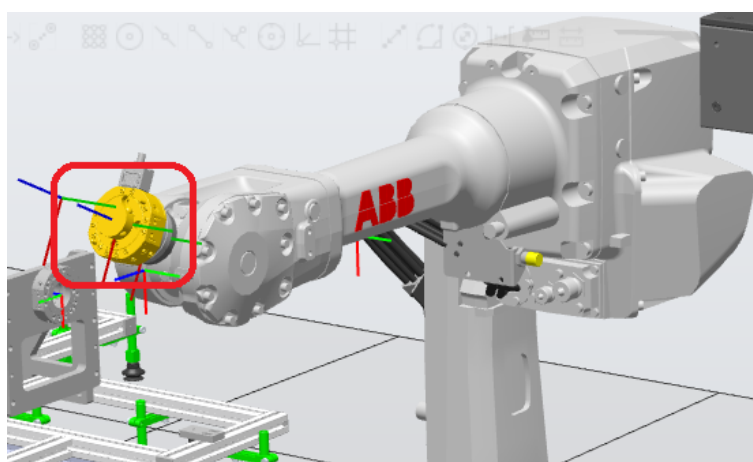


Figura 3.11: Origem do TCP do acoplamento do robô nº 1 da célula 1308

A primeira ferramenta deste robô é uma garra ventosa, responsável pela manipulação das peças. Neste caso particular, a posição do TCP é definida num ponto algures no centro da garra, como se pode observar na figura 3.12.

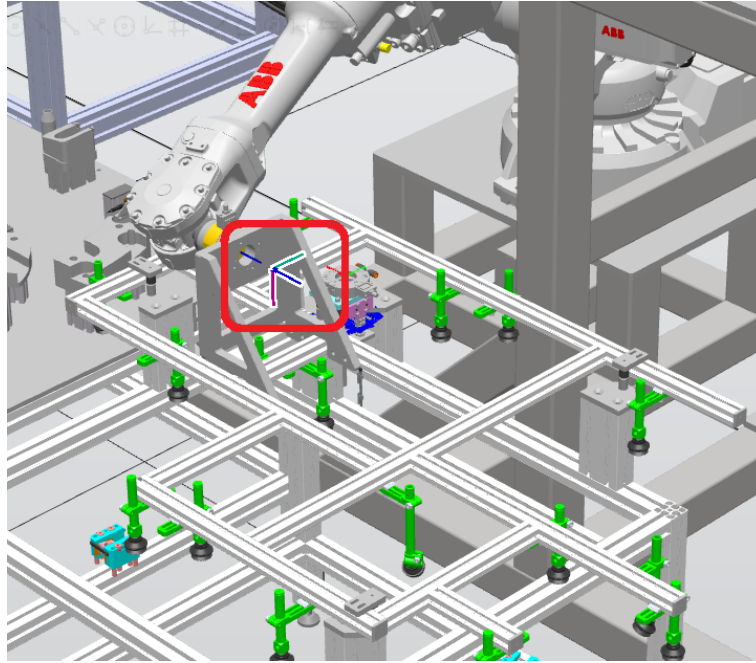


Figura 3.12: Origem do TCP da garra do robô nº 1 da célula 1308

O primeiro robô contará também com uma ferramenta adicional, responsável pela manipulação dos brackets. Assim, será novamente necessário configurar o respectivo TCP, com o eixo Z perpendicular ao bracket, conforme ilustrado na figura 3.13.

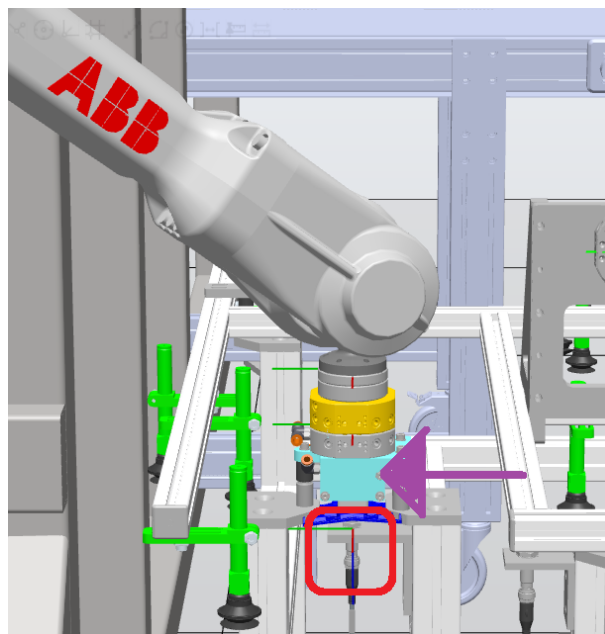


Figura 3.13: Origem do TCP da ferramenta de colocação de brackets do robô nº 1 da célula 1308

O segundo robô está equipado com uma estrela de oito dedos e não tem qualquer sistema de acoplamento. Por essa razão, foram definidas duas ferramentas distintas

para permitir a manipulação correta dos objetos, conforme ilustrado nas figuras 3.14 e 3.15.

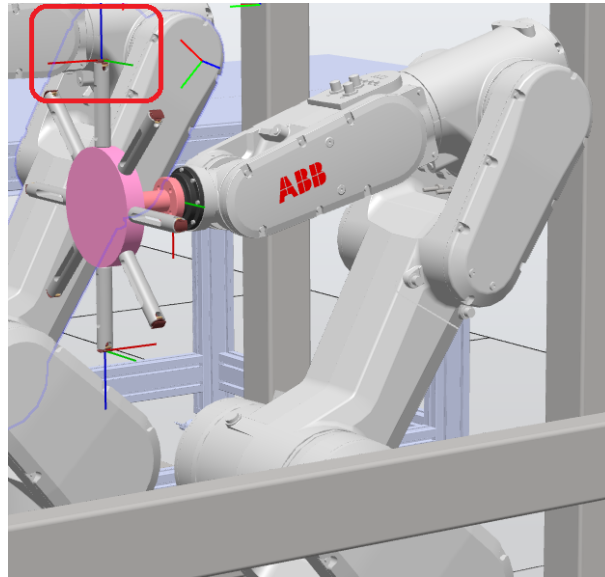


Figura 3.14: Origem do TCP da ferramenta "Estrela de 8 dedos" do robô nº 2 da célula 1308

O TCP de calibração permite que haja uma referência para afinação dos *work objects*. Este assunto será abordado mais à frente no presente relatório.

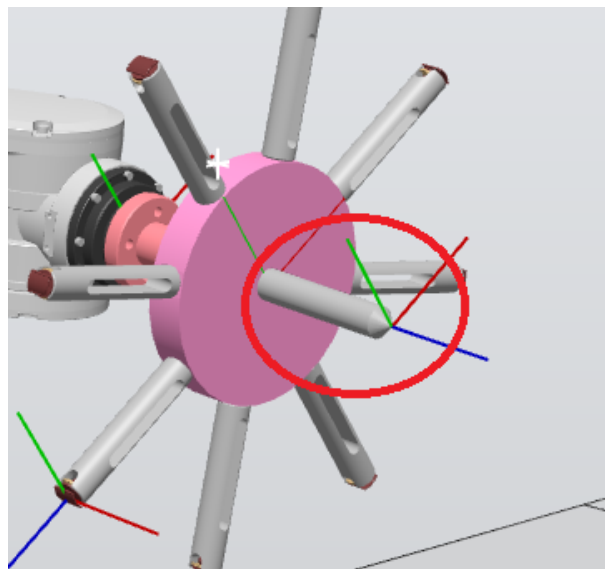


Figura 3.15: Origem do TCP de calibração da ferramenta "Estrela de 8 dedos" do robô nº2 da célula 1308

Importa referir que o "pino" presente na figura anterior, que se encontra dentro de uma elipse, permite definir os *work objects* com maior precisão. Este "pino", que está acoplado à estrela por meio de uma rosca, é posteriormente retirado para evitar colisões com outros elementos presentes no cenário.

No que diz respeito ao terceiro robô, este está equipado com um carregador automático de clips, com capacidade para suportar até duas estrelas em simultâneo. Quanto às ferramentas, o terceiro robô dispõe de três opções distintas:

- Ferramenta estrela de vinte dedos n.º 1;
- Ferramenta estrela de vinte dedos n.º 2;
- Ferramenta de calibração;
- Acoplamento.

É importante salientar que ambas as estrelas são exatamente iguais. A lógica de funcionamento baseia-se na utilização de uma das estrelas disponíveis até que todos os clips tenham sido aplicados. No final do ciclo, o robô devolve a estrela vazia à máquina de carregamento automático. Nesse momento, a máquina já tem a outra estrela totalmente carregada e pronta para ser utilizada no ciclo seguinte. Por este motivo, o robô necessita de, pelo menos, duas configurações de ferramentas: uma para o acoplamento e outra para a estrela. A localização dos respetivos TCP está representada na figura 3.16.

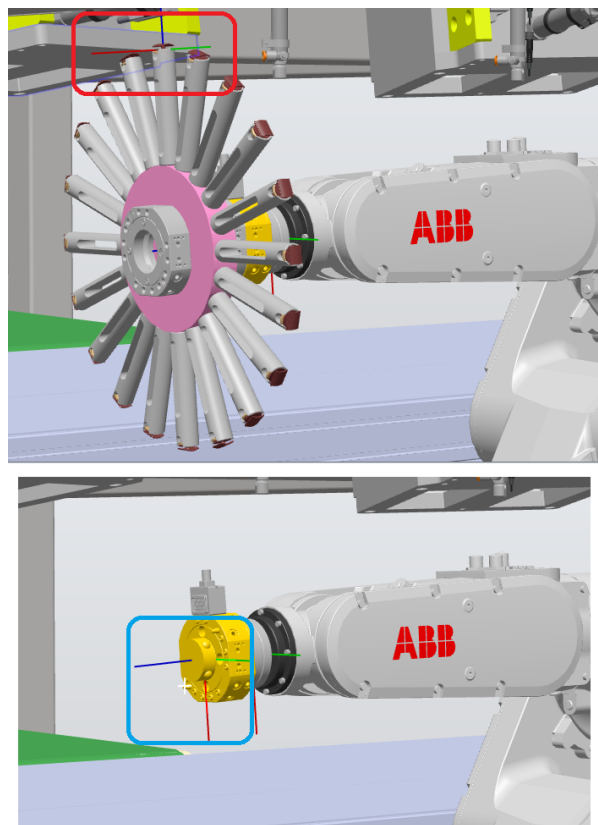


Figura 3.16: Origem dos TCPs da ferramenta "Estrela de 20 dedos" e do acoplamento do robô n.º 3 da célula 1308

No que diz respeito à orientação dos eixos dos TCPs das estrelas, o eixo "Z" foi definido de forma a apontar sempre na direção radial, colinear a um dedo específico. Esta

configuração foi adotada com o objetivo de uniformizar a abordagem em todas as células, garantindo que o sentido de aproximação durante o ajuste de pontos seja sempre realizado ao longo do eixo "Z+", facilitando assim a programação e a manutenção do sistema. Mais uma vez foi definido o TCP que corresponde à calibração dos *work objects* presentes na figura 3.17.

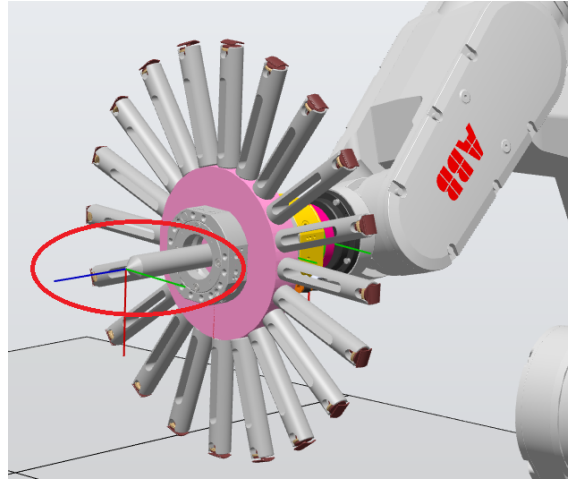


Figura 3.17: Origem do TCP de calibração da "Estrela de 20 dedos" do robô nº 3

Mais tarde, ponderou-se a colocação de um TCP em cada um dos dedos da estrela, facilitando dessa forma a afinação da inserção de clips. O RobotStudio possui funcionalidades que permitem definir os TCP de forma mais simples e rápida do que a que se obtém usando o método dos três pontos físicos, para a definição de um TCP na extremidade de cada dedo. A figura 3.18 representa os novos TCP associados a cada dedo.

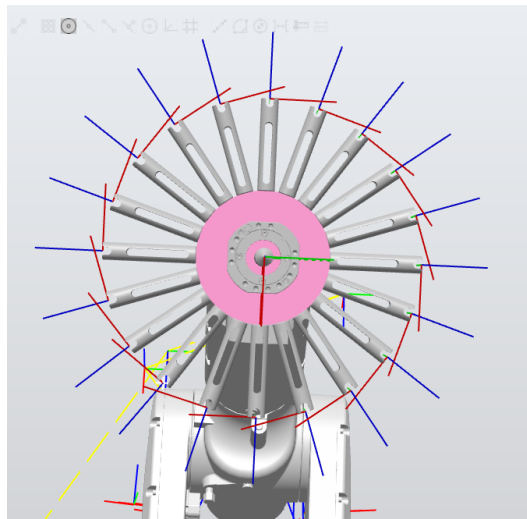


Figura 3.18: Localização dos novos TCPs da "Estrela de 20 dedos" do robô nº 3

Concluiu-se que este novo método é mais vantajoso, mas implica a calibração de um maior número de TCPs.

3.1.3 Definição de *Work Objects*

A definição dos *work objects* permite associar pontos a uma área de trabalho específica. A sua utilização estabelece uma ligação direta entre os pontos e a área, de forma que, quando esta muda de lugar, todos os pontos nela marcados são arrastados para a nova área de trabalho. Desta forma, é possível reduzir o tempo de calibração dos pontos. Durante o desenvolvimento de uma célula robótica, os seus periféricos (tapetes de entrada, alimentadores automáticos, etc.) podem sofrer pequenas alterações das suas localizações. Se cada periférico tiver a sua área de trabalho definida, basta apenas alterar a origem da respetiva área de trabalho para que todos os pontos nela marcados sejam atualizados caso contrário, é necessário ajustar cada um desses pontos. Cada área de trabalho terá uns "pinos" que estão localizados sempre no mesmo lugar com a finalidade de definir uma referência fixa para cada zona. É de salientar que, em ABB temos *user frames* e *work objects*, que são conceitos diferentes.

Um *user frame* está deslocado num dado sentido em relação ao sistema de orientação "mundo", este *user frame* está configurado para ser usado numa área específica. Por outro lado, o *work object* está associado a um *user frame*, isto significa que um *user frame* pode conter vários *work objects*[27]. Por exemplo, se considerarmos uma mesa com vários locais de trabalho, podemos supor que a mesa tem um *user frame* definido e que cada área de trabalho é um *work object* diretamente ligado à mesa. Se a mesa se deslocar, as áreas associadas (*work objects*) deslocar-se-ão com ela.

Cada robô terá os seus devidos *work objects* começando pelo primeiro robô. Este terá sete *work objects* para as respetivas áreas:

- Base das peças da marca Skoda;
- Base das peças da marca Volkswagen (VW);
- Zona de ferramentas;
- Alimentador de brackets;
- Alimentador de fitas colas dos brackets;
- Zona do tapete de entrada de peças;
- *Side Feeder*.

O segundo robô conta com quatro *work objects*:

- Base Skoda;
- Base Volkswagen;
- Área de carregamento de clips;
- Área de carregamento de *stickers*.

Para as bases das peças das marcas Skoda e Volkswagen, foi solicitado aos projetistas para inserir os "pinos" como demonstrado na figura 3.19.

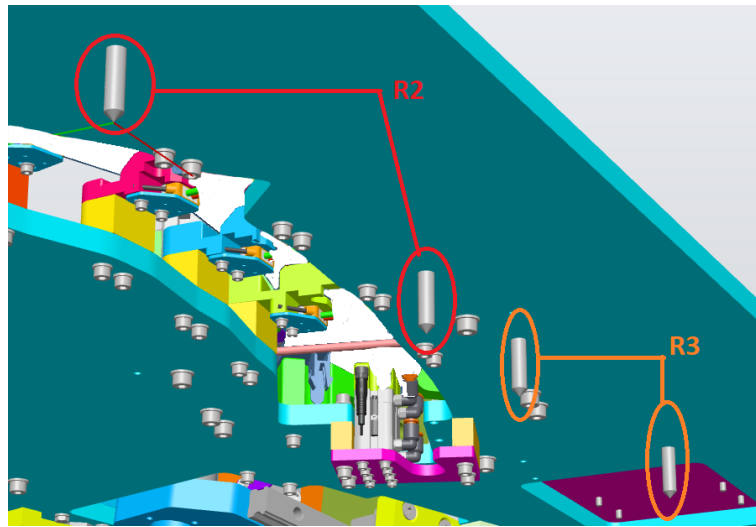


Figura 3.19: "Pinos" das bases para o segundo e o terceiro robô da célula 1308

O método utilizado para definir os *work objects* consiste na definição de três pontos num plano, sendo que é aqui que entram os "pinos" nas áreas. O processo consiste em posicionar a ferramenta de calibração o mais alinhada possível com o primeiro pino, que se tornará a origem. Em seguida, a ferramenta deve ser deslocada para o segundo pino, que se tornará a orientação "X". Por fim, a ferramenta de calibração é colocada novamente no pino que se tornará a orientação "Y". Desta forma, o controlador do robô calcula a orientação do plano e a respetiva cota em relação à sua origem. As figuras 3.20 e 3.21 ilustram o processo de definição dos *work objects*.

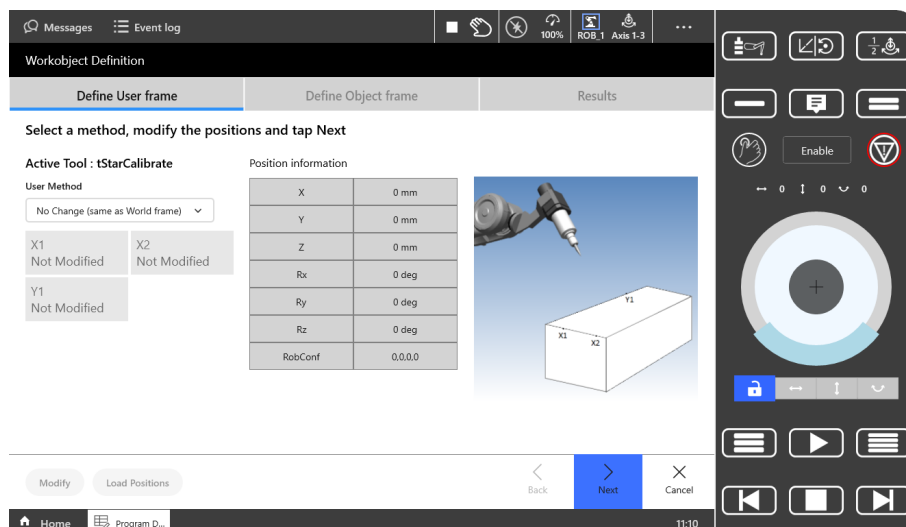


Figura 3.20: Definição do *user frame* na consola da ABB

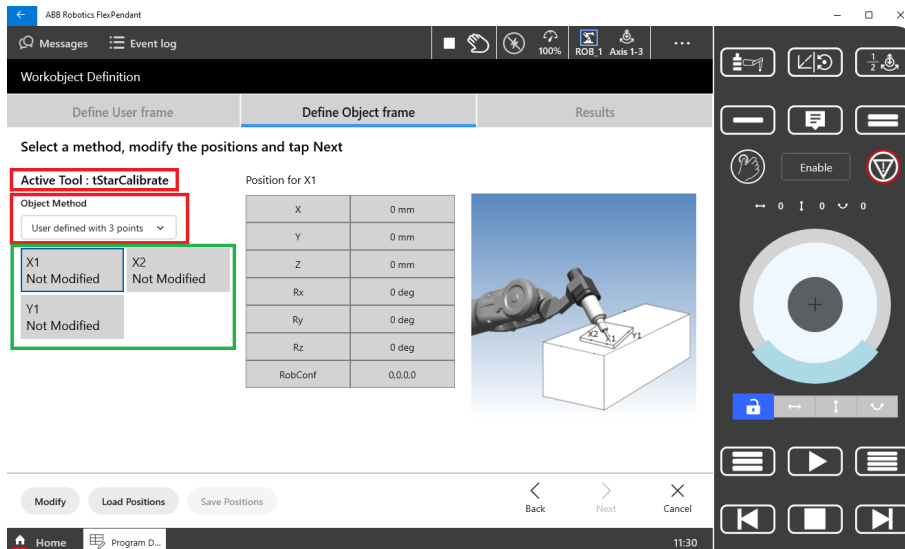


Figura 3.21: Definição do *work object* pelo método de três pontos na consola da ABB

Após a respetiva configuração, os *work objects* ficam representados em 3D como mostra a figura 3.22 para os *work objects* do segundo e terceiro robô na base da marca Skoda. Após a respetiva configuração no ambiente de programação, os *work objects* são representados graficamente no modelo tridimensional (3D) do sistema. A Figura 3.22 ilustra esta representação visual, especificamente para os *work objects* associados ao segundo e ao terceiro robôs posicionados na base da linha de montagem da Skoda.

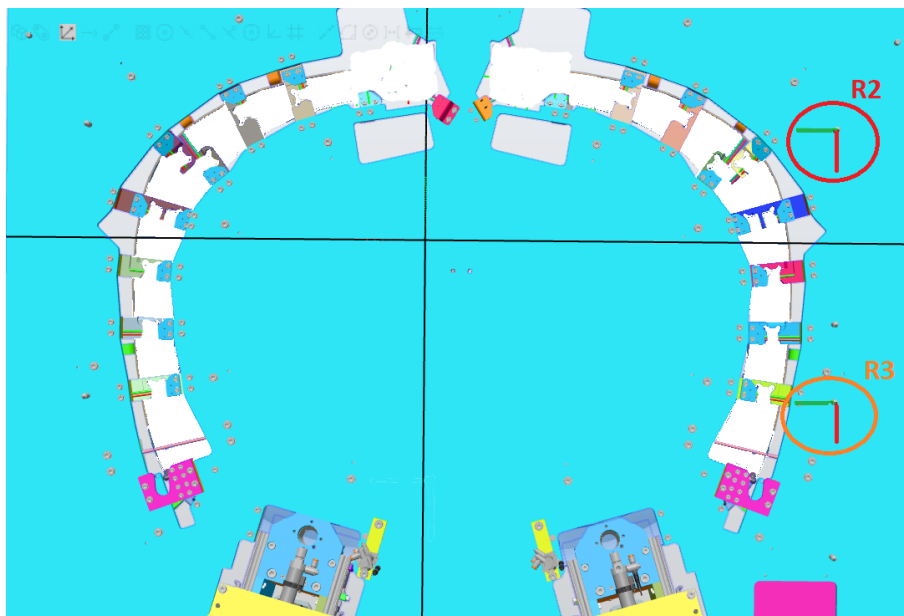


Figura 3.22: Representação da origem dos *work objects* dos robôs nº 2 e 3 para a base da Skoda da célula 1308

A posição do "X" e do "Y" de cada *work object* está diretamente relacionada com a rotação do robô. Se um equipamento rodar -90° em azimute em torno do eixo "Z" o sistema

de orientação do *work object* torna-se "X'", "Y'" e "Z'", caso o equipamento rodar 90° em azimute em torno do eixo "Z" o sistema de orientação do mesmo passa a ter a orientação "X''", "Y''" e "Z''", porém, a única diferença entre eles é a sua localização que varia de equipamento para equipamento. Na figura 3.23 está ilustrado como a orientação varia com a rotação do eixo do "Z".

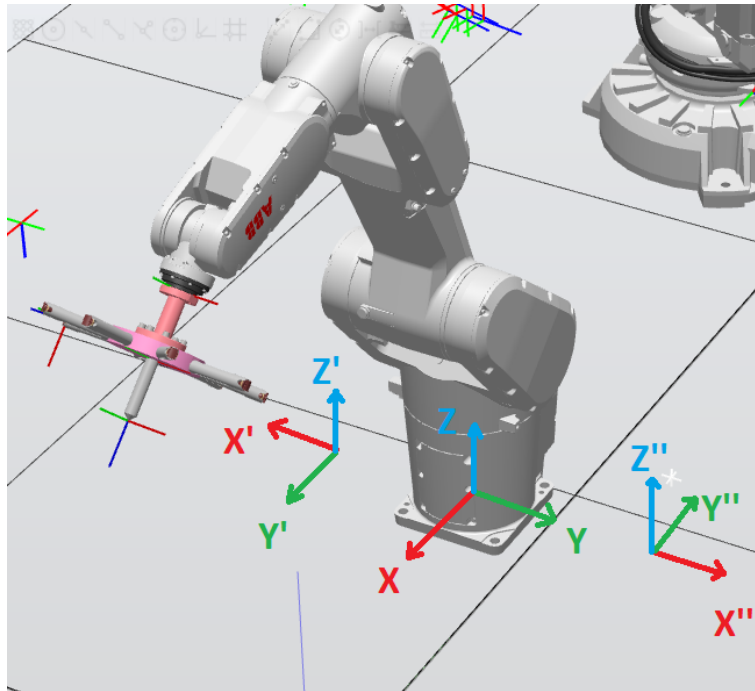


Figura 3.23: Variação da orientação dos *work objects* consoante a rotação do equipamento relativamente à origem do robô

Como o alimentador de clips está localizado à direita do robô da figura 3.23 toma a orientação "X'", "Y'" e "Z'". Na figura 3.24 está demonstrada a localização dos pinos e do *work object* para a alimentadora de clips.

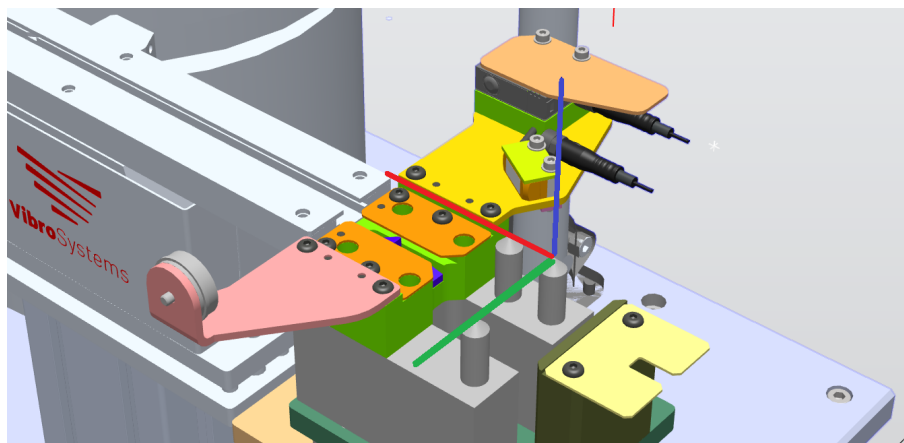


Figura 3.24: Localização dos "pinos" e o *work object* do alimentador de clips da célula 1308

Desta forma, torna-se mais fácil a interpretação dos sistemas de orientação e evita que estes estejam assinalados em todos os equipamentos.

3.1.4 Programação

Nesta fase, serão abordados alguns dos aspetos mais importantes do código desenvolvido. De forma a tornar o código idêntico entre as células da Indea, procurou-se aplicar o mesmo raciocínio sempre que possível. No que se refere à inserção de clips, a cinemática é idêntica à que será abordada na secção 3.2.4. Podem ser utilizados quatro ou cinco pontos, sendo que dois deles são marcados e os outros dois ou três são calculados com recurso a variáveis temporárias para armazenar a posição. O excerto de código seguinte ilustra a declaração das variáveis.

```

1      !Variaveisde posicao
2      VAR robtarget posAproximacao;
3      PERS robtarget posInsert;
4      PERS robtarget posEntradaClip;
5      PERS robtarget posSaidaClip;

```

O processo de inserção de clips é repetitivo podendo ser programado por um ou vários ciclos de forma a descrever os movimentos idênticos realizados. Para que isso seja possível, é obrigatório usar *arrays* temporários para guardar as posições e orientações dos diversos TCPs.

```

1      !Novas variaveis de posicao
2      PERS robtarget pos_Insert_Clip {18};
3      PERS robtarget pos_Aprox_Clip {18};

```

O *Programmable Logic Controller* (PLC) informa o robô sobre o modelo de veículo para o qual as peças anteriormente mencionadas estão a ser produzidas. Em seguida, é executada uma rotina que carrega os valores das posições das trajetórias de aproximação do robô e das posições onde será realizada a inserção dos clips para *arrays*.

```

1 PROC carregamentoPosicoes()
2     !versao=1 – Skoda
3     !versao=2 – VW
4     IF versao=1 THEN
5         !Posicoes de entrada de clips
6         pos_Aprox_Clip {1} :=POS_Aprox_Clip1_Skoda;
7         pos_Aprox_Clip {2} :=POS_Aprox_Clip2_Skoda;
8         ...
9     ELSE
10        ...
11    ENDIF
12 ENDPROC

```

O passo seguinte consiste na programação do procedimento de inserção dos clips. A parte inicial do programa inclui uma secção responsável pela calibração das posições para a inserção dos clips nas peças do veículo, que habitualmente não é executada. Este código só é executado se o operador o ativar manualmente, utilizando um comando da consola *teach pendant panel*.

Em seguida, há uma condição que permite ignorar as peças do lado direito ou esquerdo, caso se pretenda fabricar peças apenas de um dos lados. Dentro dessas condições, existe um ciclo que calcula os valores de uma posição, a rotação da estrela e o próprio movimento.

```

1 PROC Path_InsertClips(bool clipsRH, bool clipsLH)
2     GOTO inicio;
3     ! Calibracao manual das posicoes para insercao de clip nas pecas Skoda
4     MoveJ POS_Insert_Clip1_Skoda, v100, fine, ToolEstrela1 \WObj:=wobj0;
5     MoveJ POS_Insert_Clip2_Skoda, v100, fine, ToolEstrela1 \WObj:=wobj0;
6     ...
7     inicio:
8     IF clipsRH=TRUE THEN
9         FOR i FROM 1 TO 9 DO
10            !Calcula rotacao das posicoes
11            posEntradaClip:=calculaRot6Eixo(pos_Aprox_Clip{i}, TRUE, TRUE);
12            posSaidaClip:=calculaRot6Eixo(pos_Insert_Clip{i}, TRUE, TRUE);
13            posAproximacao:=calculaRot6Eixo(pos_Aprox_Clip{i}, FALSE, TRUE);
14            posInsert:=calculaRot6Eixo(pos_Insert_Clip{i}, FALSE, TRUE);
15            !Movimentacoes
16            MoveJ
17                posEntradaClip, velIntermedia, fine, ToolEstrela1 \WObj:=wobj0;
18            MoveL posAproximacao, velAproximacao, fine,
19                ToolEstrela1 \WObj:=wobj0;
20            MoveL posInsert, velAproximacao, fine,
21                ToolEstrela1 \WObj:=wobj0;
22            WaitTime 1;
23            MoveL
24                posSaidaClip, velAproximacao, fine, ToolEstrela1 \WObj:=wobj0;
25            contadorDedo:=contadorDedo+1;
26        ENDFOR
27    ENDFIF
28    ...
29 ENDPROC

```

3.1.5 Alterações da célula

Durante o desenvolvimento deste projeto, foi identificada uma dificuldade na inserção de clips por parte do robô n.º 3, decorrente da sobreposição das respetivas áreas de trabalho com as do robô n.º 2. Esta situação está representada na figura 3.25, estando a zona de conflito assinalada a vermelho com setas.

Constatou-se também que uma das juntas do robô n.º 3 se deslocava para uma zona muito próxima de um conjunto de cilindros pneumáticos, conforme ilustrado na figura 3.26, o que poderia potencialmente causar uma colisão.

Para mitigar estes problemas, foi proposta a rotação da base do robô n.º 3 em 180°. Após a atualização do *layout*, procedeu-se à simulação do novo cenário, tendo-se verificado uma melhoria significativa. Com a nova configuração, os robôs n.º 2 e n.º 3 passaram a dispor de mais espaço para as suas trajetórias, o que reduziu o risco de colisões e aumentou a segurança e a eficiência do sistema.

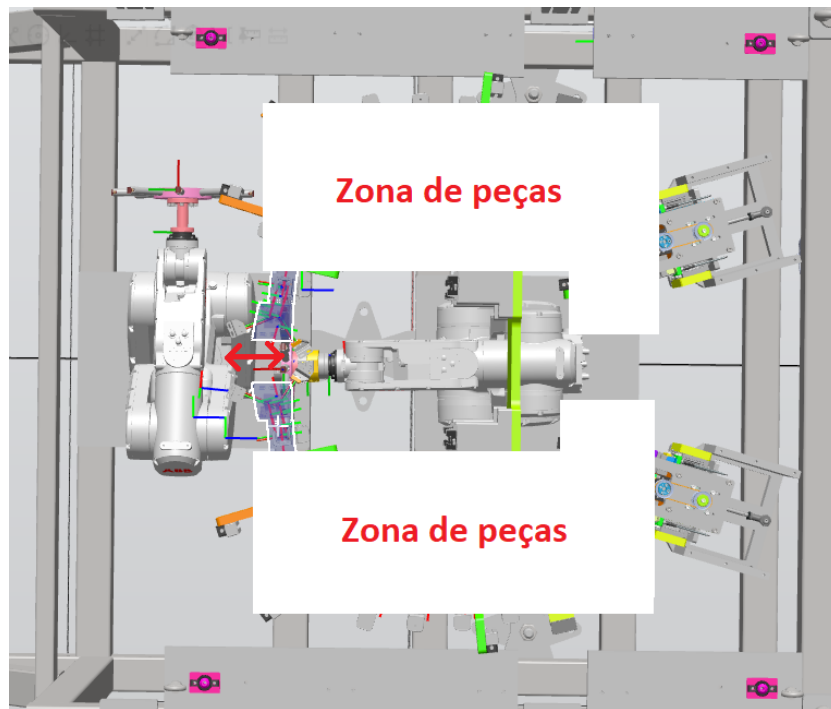


Figura 3.25: Zona de proximidade entre robô número dois e três

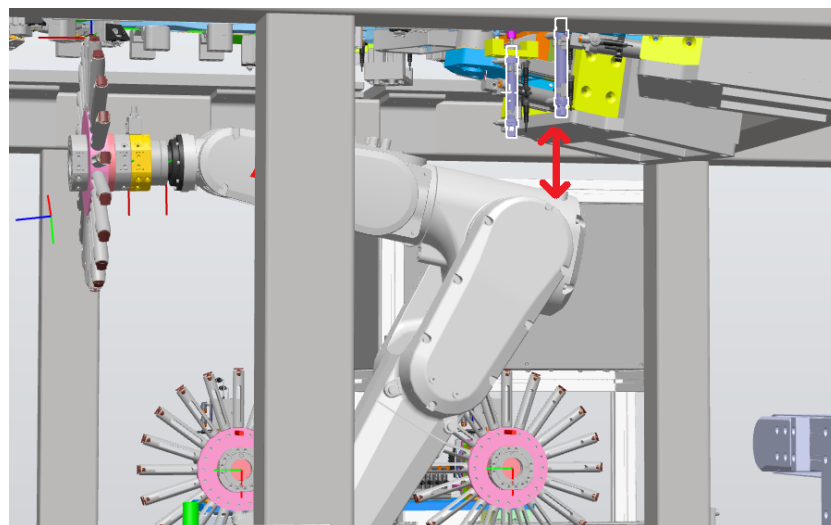


Figura 3.26: Zona de proximidade entre o robô três e os cilindros

3.1.6 Colisões

Nesta célula, foi detetada uma colisão entre a ferramenta e a base, tendo sido necessário efetuar alterações à ferramenta. A figura 3.27 assinala a referida colisão. Durante a análise do funcionamento da célula robótica, foi detetada uma colisão entre a ferramenta do robô e a estrutura da base. Este incidente evidenciou a necessidade de alterar o projeto da ferramenta, de modo a garantir a sua operação segura e eficiente dentro dos limites do espaço de trabalho.

A zona de colisão está assinalada na figura 3.27, o que permite visualizar claramente o ponto de interferência entre a ferramenta e a base. Esta identificação foi essencial para redefinir o modelo da ferramenta e para validar o novo trajeto operacional, minimizando o risco de contacto indesejado com a estrutura.

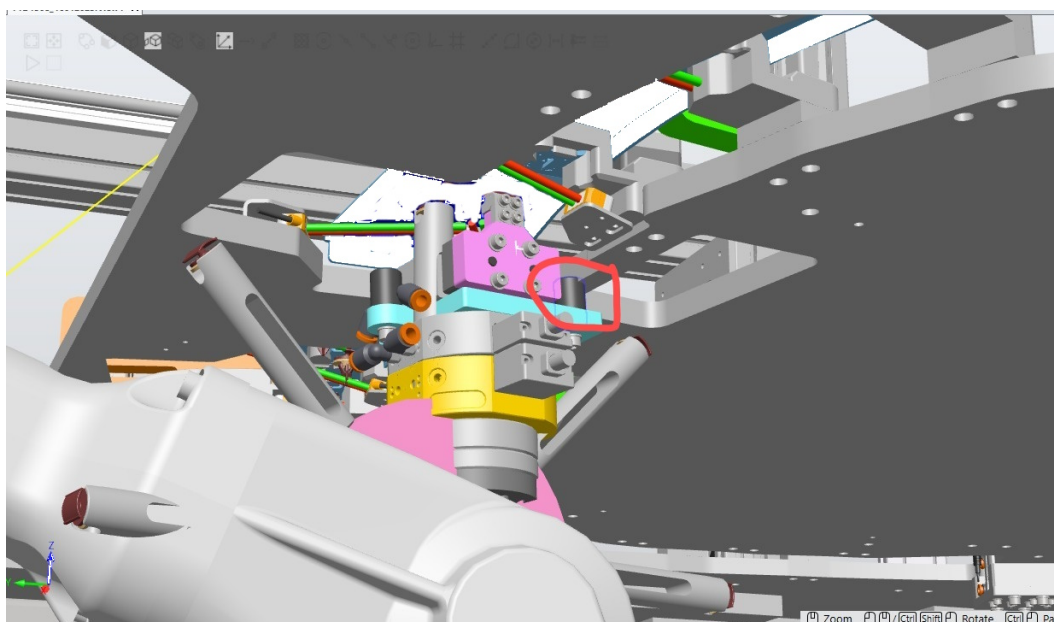


Figura 3.27: Colisão da ferramenta com a base da célula 1308

3.1.7 Problemas detetados na célula

Como referido anteriormente, esta célula pode trabalhar com duas versões de peças. No que se refere às peças da Skoda, não foram registados grandes problemas. Porém, com as peças da Volkswagen, verificou-se alguma dificuldade na colocação das peças na sua base, visto o primeiro robô não ter alcance suficiente para realizar essa tarefa, devido ao tamanho das peças. Estas eram ligeiramente maiores que as da Skoda, o que levou os projetistas a efetuarem alterações à estrutura da célula.

3.2 Célula PRJ 1309

Nesta secção, é abordado o trabalho de programação dos robôs de outra célula, no ambiente de simulação RobotStudio, com o objetivo de simular os respetivos movimentos e detetar possíveis colisões entre eles.

3.2.1 Layout e descrição de funcionamento

A célula "PRJ 1309" é responsável por inúmeras funções, nomeadamente:

- Manipulação de peças;
- Aquecimento da zona de colocação de fita;
- Colocação de fita;
- Colocação das esponjas;
- Colocação de clips;
- Colocação de tabs.

A figura 3.28 representa a célula PRJ 1309.

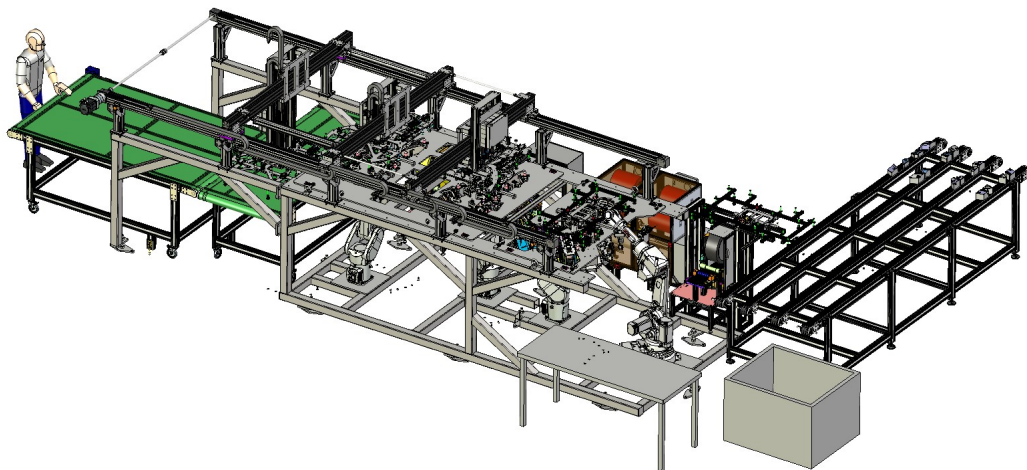


Figura 3.28: Layout da célula PRJ 1309

Na análise realizada, considerámos que a referida célula possui um conjunto de máquinas distintas. Assim, cada robô e cada periférico (um dispensador de clips, fitas e esponjas) constituirá uma máquina individual. Cada máquina desempenhará várias funções distintas que serão controladas por um ou vários PLC. A célula em análise possui onze máquinas, conforme demonstrado na figura 3.29, que estão numeradas de acordo com o sentido da entrada para a saída das peças.

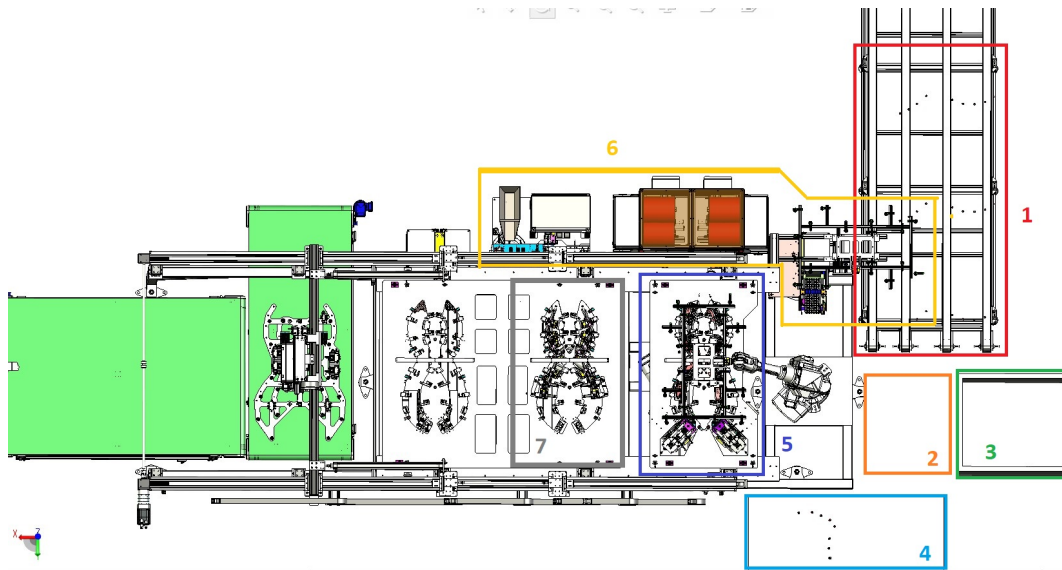


Figura 3.29: Representação da célula 1309 por áreas

Legenda da figura 3.29:

1. Entrada de peças;
2. Zona do plasma;
3. Zona de colocação de peças defeituosas;
4. Zona de entrada manual;
5. Zona de *place* na base;
6. Zona de ferramenta;
7. Base de inserção de clips.

Na zona seis da figura anterior, encontram-se um dispensador de clips, duas caixas mágicas e uma máquina dispensadora de esponjas. O processo geral está representado na figura 3.30.



Figura 3.30: Fluxograma de funcionamento geral da célula 1309

3.2.2 Robô número 1

As tarefas estão divididas entre os robôs. O primeiro robô é responsável por transportar a ferramenta até ao seu posto, caso ainda não o tenha feito. Em seguida, deverá realizar a manipulação das peças provenientes do tapete e encaminhando-as para a zona de plasma ou para o caixote de peças NOK (não OK), caso apresentem defeito. Se as peças estiverem em conformidade, o robô deverá colocá-las na primeira base, permitindo que o segundo robô execute a sua função. Além das peças provenientes do tapete de entrada, este robô também poderá manipular peças provenientes da mesa de entrada manual. As figuras 3.31 e 3.32 representam, respetivamente, as bases destinadas às peças das marcas de automóveis Volkswagen e Skoda.

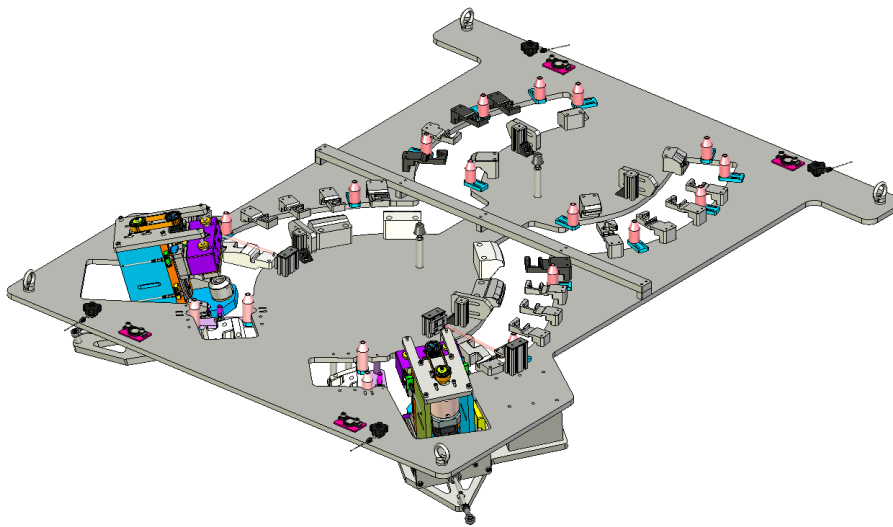


Figura 3.31: Base para peças da Volkswagen

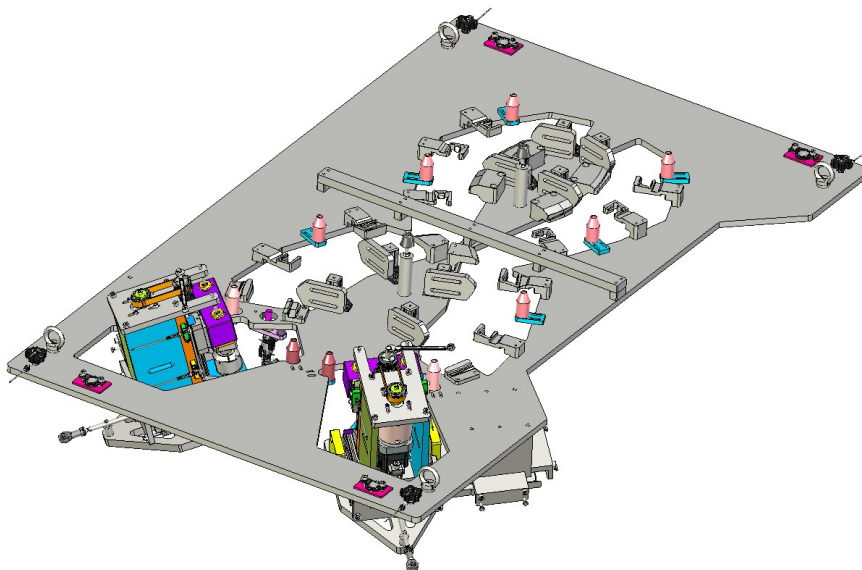


Figura 3.32: Base para peças da Skoda

A discrepância entre as bases obriga a que o robô tenha posições de "largar" diferentes para cada tipo de peças. De seguida, o robô deve desacoplar a ferramenta de manipulação e utilizar a ferramenta de manipulação de esponjas, que se encontra previamente aparafusada ao flange do robô. Em seguida, deverá deslocar-se até ao dispensador de esponjas e realizar a respetiva manipulação das esponjas para o local designado para as peças. A figura 3.33 representa o dispensador de esponjas.



Figura 3.33: Dispensador de esponjas

Calibração de ferramentas

O primeiro robô terá apenas duas ferramentas do tipo de ventosa. Uma será utilizada para manipular peças das marcas Volkswagen e Skoda e a outra servirá para aplicar esponjas circulares. A ferramenta de manipulação está representada na figura 3.34.

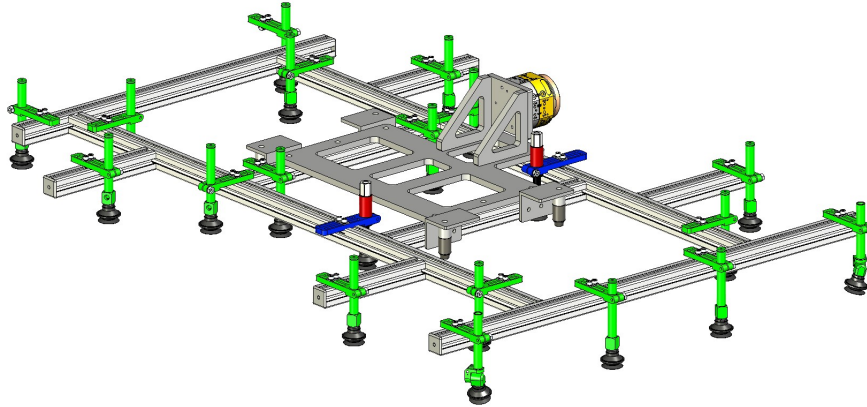


Figura 3.34: Ferramenta de manipulação de peças

No que concerne às configurações das ferramentas, o primeiro robô foi configurado com as ferramentas ilustradas na figura 3.35

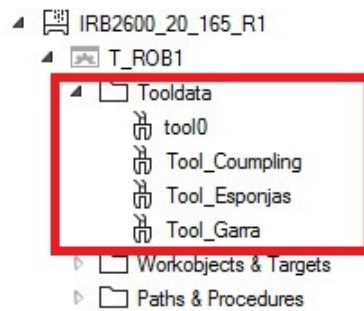


Figura 3.35: Registo do TCP do primeiro robô da célula 1309

A figura 3.36 representa a posição do TCP "Tool_Garra" do primeiro robô da célula 1309, responsável pela manipulação da peça.

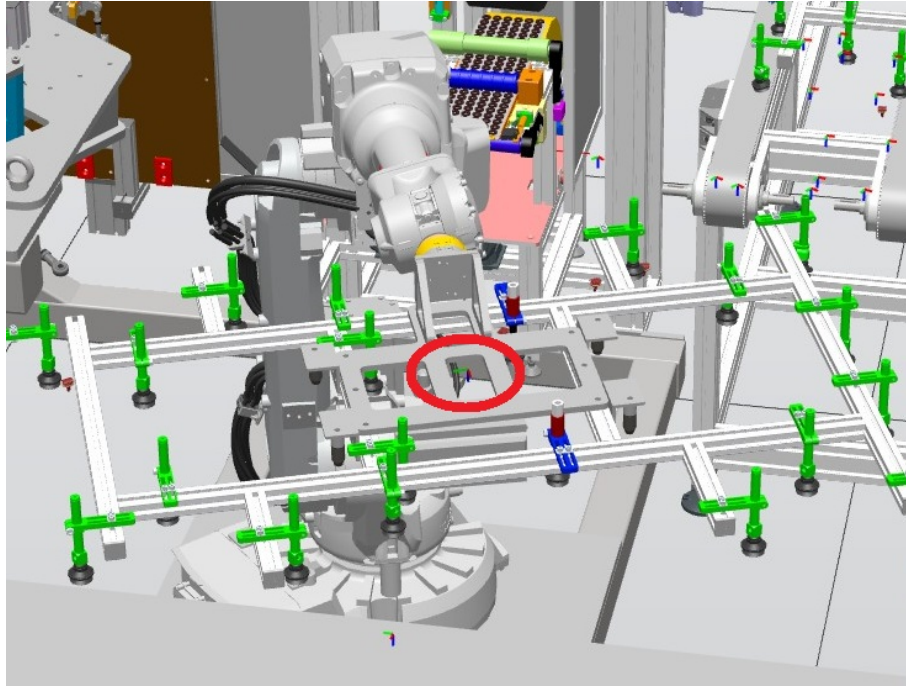


Figura 3.36: Localização do TCP do primeiro robô da célula 1309

A figura 3.37 mostra a ferramenta responsável pela captura, manipulação e colocação das esponjas circulares nas peças. É de salientar que esta ferramenta está aparafusada ao flange do robô, permanecendo fixa e não sendo removida durante o processo. No entanto, a ferramenta de manipulação é a única que utiliza o acoplamento, uma vez que, para proceder à colocação das esponjas no local adequado, é necessário que esta se encontre no pousa-ferramentas.

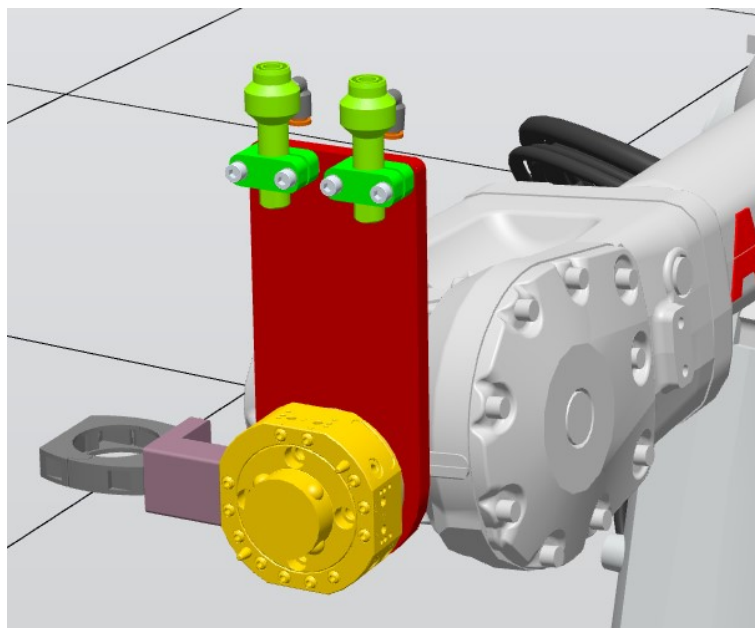


Figura 3.37: Ferramenta responsável pela captura, manipulação e colocação das esponjas circulares

A figura 3.38 mostra o TCP correspondente à ferramenta de colocação das esponjas circulares (assinalado a rosa) e, novamente, o TCP associado ao acoplamento (assinalado a azul).

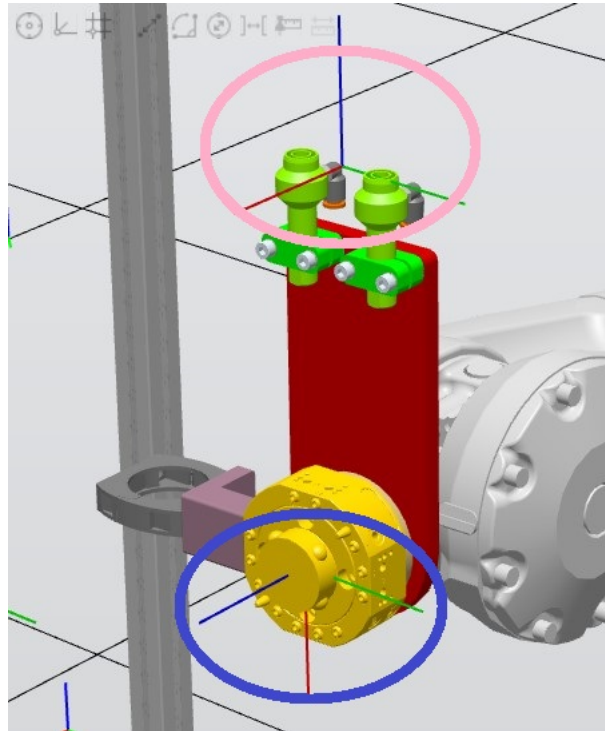


Figura 3.38: Localização do TCP do primeiro robô responsável pela captura, manipulação e colocação de esponjas da célula 1309

É importante salientar que, para a execução do processo de aplicação de esponjas, a ferramenta requer o auxílio de um periférico responsável pela dispensação automática das mesmas, conforme representado na figura 3.39.

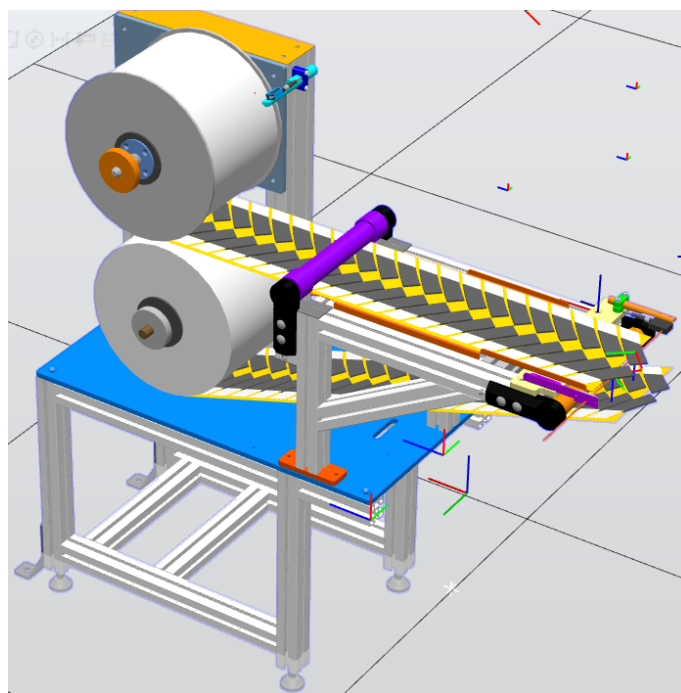


Figura 3.39: Alimentador automático de esponjas

Como referido anteriormente, são fabricadas quatro peças para o modelo de veículo em produção: duas peças para o lado direito do veículo e outras duas peças para o lado esquerdo. No entanto, é aplicada apenas uma única esponja para as peças do lado direito e outra esponja para as peças do lado esquerdo. Esta distinção deve ser tida em conta no momento em que a ferramenta realiza o *picking* de cada esponja. A ferramenta é, assim, composta por três zonas funcionais: uma zona de soldadura e aplicação de tabs, outra para aplicação de esponjas no lado direito e uma última para aplicação de esponjas no lado esquerdo.

Programação

Nesta fase, será apresentado o código que foi abordado apenas para simulação, pelo que serão descartados os sinais provenientes do PLC. Nesta fase, foi solicitado que se detetassem possíveis colisões com a estrutura, que serão corrigidas mais tarde. Todo o código foi idealizado para permitir a programação posterior dos robôs.

A utilização de variáveis para definir velocidades e posições torna o código mais dinâmico e simples, facilitando a alteração dos valores para realizar ajustes e depurar o código, se necessário. As velocidades das juntas dos robôs ABB são definidas por meio de variáveis. A utilização de variáveis para armazenar dados permite que, caso seja necessário alterar uma velocidade, tal alteração seja efetuada apenas numa parte do código, e não em todo o código, poupando tempo na fase de testes. As velocidades são guardadas por meio de três variáveis: uma para a velocidade de aproximação das peças ou da estrutura, outra para as velocidades intermédias e uma última para a

velocidade da peça durante o processo de aplicação de calor, por via de plasma.

Para além das posições marcadas, são também definidas variáveis temporárias de posição, denominadas por "POS_PlasmaX_Y".

```
1 !Variaveis de velocidade
2 PERS speeddata velAproximacao;
3 PERS speeddata velIntermedia;
4 PERS speeddata velPlasma;
```

Como existem pelo menos duas referências de peças, a utilização de variáveis permite simplificar o código, como vai ser abordado mais à frente.

```
1 !Variaveis de posicao
2 !Posicao do plasma
3 !Peca 1
4 PERS robtarget POS_Plasma1_1;
5 PERS robtarget POS_Plasma1_2;
6 PERS robtarget POS_Plasma1_3;
7 !Peca 2
8 PERS robtarget POS_Plasma2_1;
9 PERS robtarget POS_Plasma2_2;
10 PERS robtarget POS_Plasma2_3;
11 !Peca 3
12 PERS robtarget POS_Plasma3_1;
13 PERS robtarget POS_Plasma3_2;
14 PERS robtarget POS_Plasma3_3;
15 !Peca 4
16 PERS robtarget POS_Plasma4_1;
17 PERS robtarget POS_Plasma4_2;
18 PERS robtarget POS_Plasma4_3;
```

Para simplificar o código, o *main* é reduzido, invocando apenas os subprogramas, que na linguagem "RAPID" tem a nomenclatura de "PROC". No exemplo seguinte, é possível verificar que o programa executa uma série de subprogramas associados a funções que o robô deve realizar.

```
1 PROC main()
2 inicializacao;
3 carregamentoPosicoes 2;
4 Path_Home;
5 Path_PickTool;
6 Path_Home;
7 Path_PickPecaTapete;
8 Path_Plasma;
9 Path_Home;
10 Path_PlacePeca;
11 Path_Home;
12 ENDPROC
```

É na inicialização que todas as variáveis voltam aos valores predefinidos, aplicável às variáveis de velocidade, aos contadores e a algumas variáveis temporárias. O subprograma "carregamentoPosicoes" permite selecionar quais são as posições que devem ser usadas nas variáveis de posição. A seleção é passada por argumento como é mostrado no exemplo acima. O valor do argumento dois refere-se às variáveis de posição das peças da marca Volkswagen. No excerto de código do programa representado de seguida, podemos encontrar a atribuição das variáveis "POS_PlamaX_Y" pelos valores dos pontos previamente marcados.

```

1 PROC carregamentoPosicoes(num versao)
2     !versao=1 – Skoda
3     !versao=2 – VW
4     IF versao=1 THEN
5         !Posicao do plasma
6         !Peca 1
7         POS_Plasma1_1:=POS_Plasma1_1_Skoda ;
8         POS_Plasma1_2:=POS_Plasma1_2_Skoda ;
9         POS_Plasma1_3:=POS_Plasma1_3_Skoda ;
10        !Peca 2
11        ...
12        !Peca 3
13        ...
14        !Peca 4
15        ...
16    ENDIF
17    IF versao=2 THEN
18        !Posicao do plasma
19        !Peca 1
20        POS_Plasma1_1:=POS_Plasma1_1_VW ;
21        POS_Plasma1_2:=POS_Plasma1_2_VW ;
22        POS_Plasma1_3:=POS_Plasma1_3_VW ;
23        !Peca 2
24        ...
25        !Peca 3
26        ...
27        !Peca 4
28        ...
29    ENDIF
30 ENDPROC

```

As variáveis anteriores serão utilizadas no programa "Path_Plasma", que tem como função aplicar plasma numa zona específica das peças em questão.

```

1 PROC Path_Plasma()
2     ...
3     MoveJ
4         RelTool(POS_Plasma1_1,0,0,-50),velIntermedia,fine,Tool_1\WObj:=wobj0;
5     MoveL POS_Plasma1_1,velAproximacao,fine,Tool_1\WObj:=wobj0;

```

```
5 MoveC POS_Plasma1_2 , POS_Plasma1_3 , velPlasma , fine , Tool_1 \WObj:=wobj0 ;  
6 MoveL  
    RelTool(POS_Plasma1_3,0,0,-50) , velAproximacao , fine , Tool_1 \WObj:=wobj0 ;  
7 ...  
8  
9 ENDPROC
```

Em cada subprograma de código escrito, existe uma parte inicial cuja principal funcionalidade é a calibração manual das posições, conforme já foi referido anteriormente. Na fase de testes, os pontos terão de ser verificados e reajustados, sendo que é nessa fase que entra essa parte de código. Na fase de ajustes e calibrações, o utilizador força a entrada nessas linhas de código e "corre" cada linha de código verificando a posição, se necessário faz o devido ajuste e logo de seguida faz a gravação das novas coordenadas do ponto, porém, o subprograma "carregamentoPosicoes" tem de ser corrido novamente para atualizar as variáveis de posição.

3.2.3 Robô número 2

O segundo robô será responsável pela aplicação de fita nas quatro peças, embora o fabricante possa também selecionar quais as peças às quais pretende aplicar fita. Esta flexibilidade é especialmente vantajosa na produção de peças com diferentes especificações *aftermarket* para modelos de veículos em diferentes fases do respetivo ciclo de vida. Na fase normal de produção, será necessário fabricar todas as peças. No entanto, numa fase posterior — quando a produção dos veículos for descontinuada —, a célula poderá produzir apenas as peças solicitadas pelo cliente final. Por exemplo, no caso de um acidente em que apenas a lateral frontal esquerda do veículo ficou danificada, a célula terá capacidade para fabricar exclusivamente essa peça. Esta flexibilidade está a ser cuidadosamente considerada durante as fases de conceção e programação das diferentes máquinas da célula.

Este robô pode também operar com duas ferramentas, que são alimentadas por um sistema designado por "caixa mágica", representado na figura 3.40.

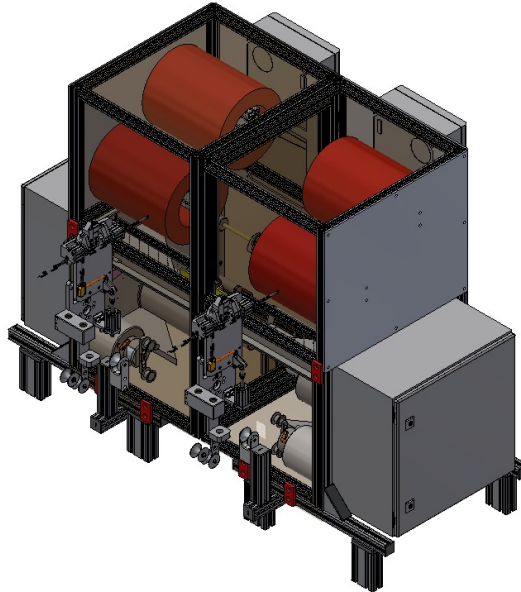


Figura 3.40: Caixa mágica

A "caixa mágica" permite que a fita se mantenha esticada, independentemente da posição do robô. Além disso, possui duas ferramentas, sendo utilizada apenas uma de cada vez. Esta configuração confere flexibilidade à célula, visto que, quando um rolo de fita termina, o processo não é interrompido, permitindo ao operador substituir o rolo na caixa correspondente sem causar paragens na produção.

O acoplamento representado na figura 3.41 liga a ferramenta ao robô e assegura a interligação dos sinais digitais provenientes da ferramenta, por exemplo, dos sensores de curso dos cilindros pneumáticos e a codificação da própria ferramenta.

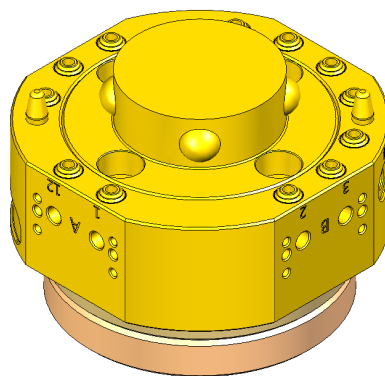


Figura 3.41: Acoplamento de ferramentas

Calibração de ferramentas

O segundo robô pode ter duas ferramentas de dispensação de fita, porém, foram registadas três ferramentas. Na figura 3.42 estão representadas as posições dos TCPs, no ambiente de simulação RobotStudio.

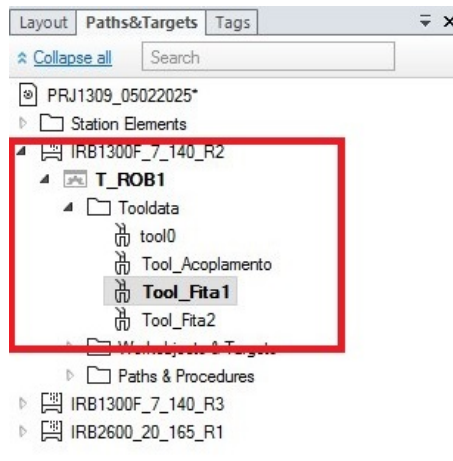


Figura 3.42: Registo de TCPs do segundo robô da célula 1309

Na figura 3.43, estão assinalados a laranja os dois TCPs da ferramenta de aplicação de fita: a "Tool_Fita1" e a "Tool_Fita2". Isto deve-se ao facto de, em teoria, as duas ferramentas serem idênticas. O ponto assinalado corresponde à posição de aplicação da fita, ou seja, ao local onde esta sai da ferramenta e é aplicada na peça.

Na zona assinalada a azul, encontra-se outro TCP, denominado "Tool_Acoplamento", que é utilizado quando não está acoplada qualquer ferramenta. Este ponto é útil para ajustar as posições de captura da ferramenta e também funciona como posição de *home*.

A figura seguinte ilustra os TCP das ferramentas, sendo que o "tool0" representa o ponto do flange do robô, definido por padrão pela ABB.

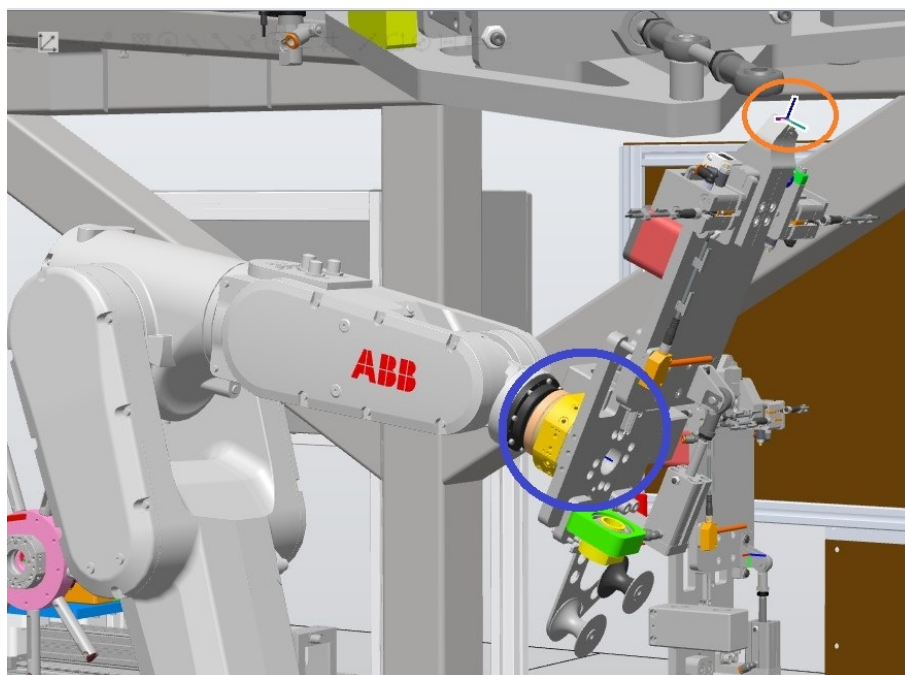


Figura 3.43: Localização dos TPCs do segundo robô da célula 1309

3.2.4 Robô número 3

Após a aplicação da fita, um manipulador cartesiano retira as peças da primeira base e coloca-as na segunda, para que mais tarde o robô número três possa inserir os clips, com a ferramenta designada por "estrela de nove dedos" representada na figura 3.44.

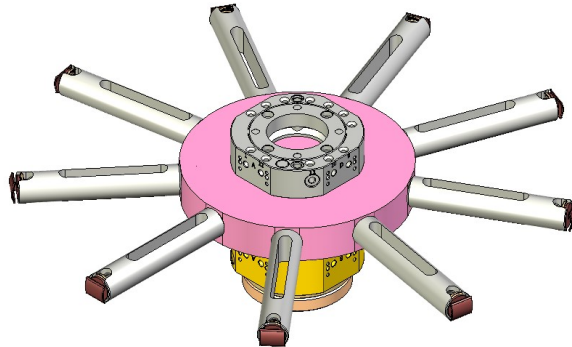


Figura 3.44: Ferramenta Estrela de 9 dedos

Calibração de ferramentas

A calibração das ferramentas deste robô específico é muito semelhante à descrita para os robôs da secção anterior, sendo a localização dos pontos dos TCP representada nas figuras 3.45, 3.46 e 3.47.

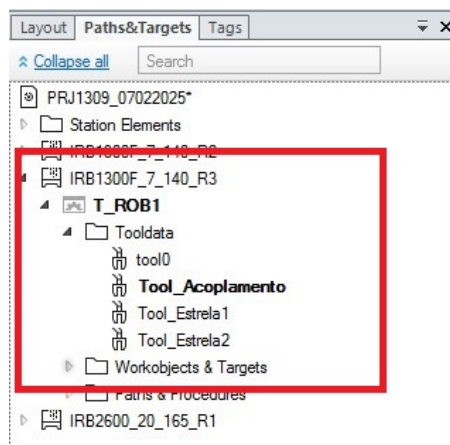


Figura 3.45: Registo de TCPs do terceiro robô da célula 1309

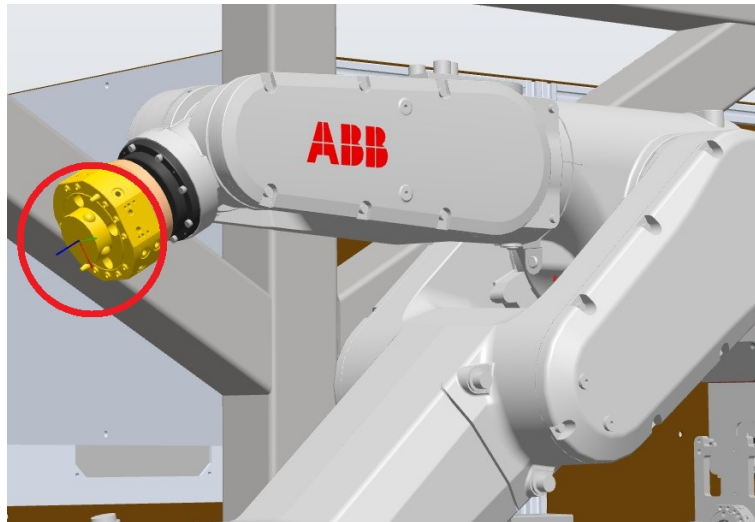


Figura 3.46: Localização do TCP do acoplamento do terceiro robô

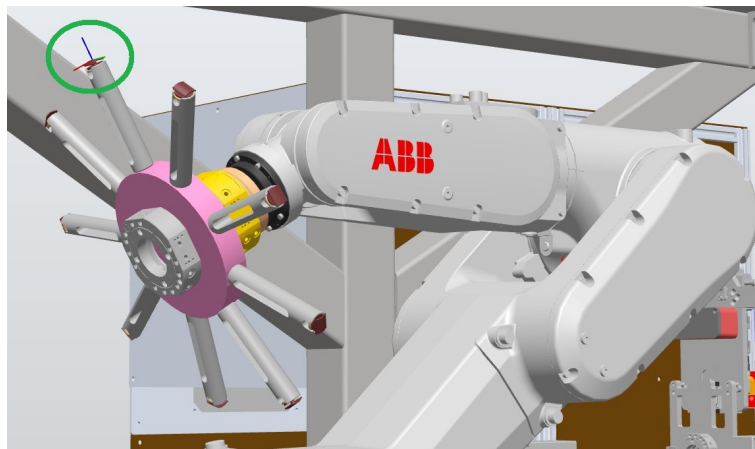


Figura 3.47: Localização do TCP da ferramenta-estrela 9 dedos do terceiro robô

A cinemática da inserção dos clips é descrita por cinco pontos para cada clip, no entanto, apenas dois deles são registados na memória do robô, sendo os três restantes calculados a partir destes. A figura 3.48 mostra a cinemática da inserção de clips, sendo que cada dedo tem de realizar estas três etapas, começando no ponto um seguindo para o segundo e assim sucessivamente, conforme representado na figura.

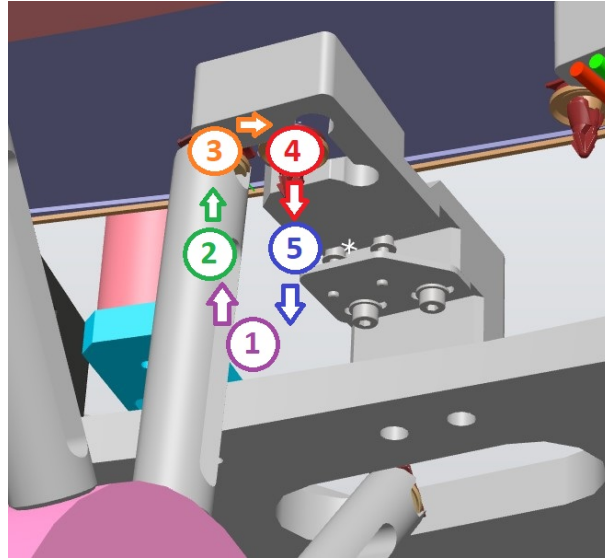


Figura 3.48: Cinemática de inserção de clips

No que toca a posições da figura 3.48 temos:

1. Posição intermédia de inserção;
2. Posição de aproximação de entrada;
3. Posição de entrada do clip;
4. Posição de inserção do clip;
5. Posição de saída do clip.

As posições três e quatro são marcadas manualmente, ao passo que todas as outras são posições calculadas. No excerto de código seguinte, é representada a dinâmica de utilização da função "RelTool" que calcula internamente um dado deslocamento em x, y e z em relação ao sistema de orientação da ferramenta, o que obriga à configuração da ferramenta. No entanto, uma má configuração da ferramenta pode resultar num deslocamento incorreto.

```

1 !Clip 2
2 MoveJ RelTool(POS_InsertClip1_2,0,-10,-100)
3 ,velIntermedia,fine,Tool_Estrela1\WObj:=wobj0;
4 MoveL RelTool(POS_EntradaClip1_2,0,0,-50),
5 velIntermedia,fine,Tool_Estrela1\WObj:=wobj0;
6 MoveL POS_EntradaClip1_2,velAproximacao,
7 fine,Tool_Estrela1\WObj:=wobj0;
8 MoveL POS_InsertClip1_2,velAproximacao,
9 fine,Tool_Estrela1\WObj:=wobj0;
10 MoveL RelTool(POS_InsertClip1_2,0,0,-50),velAproximacao,
11 fine,Tool_Estrela1\WObj:=wobj0;
12 MoveJ RelTool(POS_InsertClip1_2,0,-10,-100)
13 ,velIntermedia,fine,Tool_Estrela1\WObj:=wobj0;
    
```

3.2.5 Robô número 4

O robô número quatro é responsável pela aplicação de tabs e de esponjas nas peças. Este robô possui duas ferramentas idênticas, cada uma com duas funções distintas: fornecer tabs e aplicar esponjas.

Cada peça recebe um tab, que é soldado à fita previamente posicionada pelo robô número dois. Esta aplicação é realizada através de uma sequência de movimentos executados pelo robô, com a intervenção de um cilindro com um elemento aquecido, o qual permite a fusão dos dois materiais: o tab e a fita. É de salientar que os tabs são provenientes de um rolo, que é cortado pela própria ferramenta.

Quando o rolo termina, o robô procede à substituição automática da ferramenta por outra que já contenha um novo rolo carregado. A existência de duas ferramentas justifica-se precisamente por esta razão: permitir a substituição do rolo de tabs sem a necessidade de interromper o ciclo de produção da máquina.

No que se refere ao processo de aplicação de esponjas, é utilizada a mesma ferramenta, que possui um sistema de vácuo na sua parte inferior. Este sistema permite descolar a fita do dispensador de esponjas e proceder à sua aplicação na peça. A ferramenta em questão está representada na figura 3.49

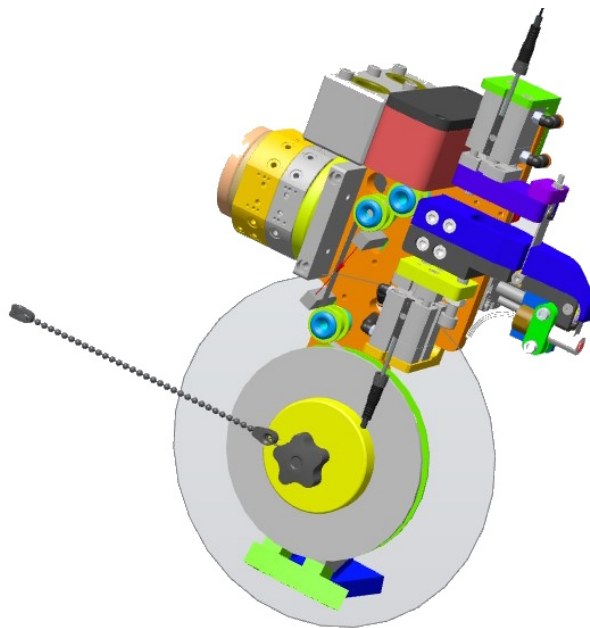


Figura 3.49: Ferramenta de aplicação de tabs e de esponjas

Calibração de ferramentas

Este robô tem uma calibração de ferramentas ligeiramente diferente da utilizada nos robôs anteriores. Possui quatro calibrações distintas, conforme ilustrado na figura 3.50.

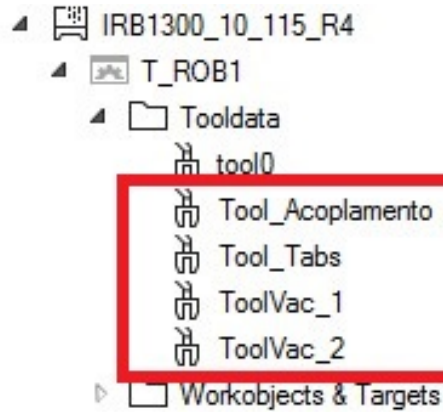


Figura 3.50: Registo de TCP's do quarto robô da célula 1309

Tal como os restantes robôs que requerem a troca de ferramentas, este está equipado com um TCP dedicado ao acoplamento. Dispõe também de um TCP específico para a manipulação dos tabs, visível na figura 3.51.

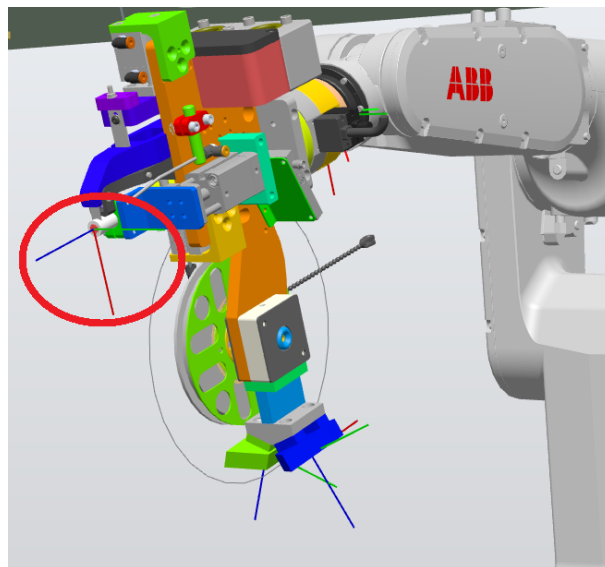


Figura 3.51: Localização dos TCPs "Tool_Tabs1" do quarto robô

Para além deste, foram configurados dois TCP adicionais, designados por "ToolVac_1" e "ToolVac_2", responsáveis pela colocação precisa das esponjas nas respetivas posições (um para o lado esquerdo e outro para o lado direito da peça), conforme representado na figura 3.52.

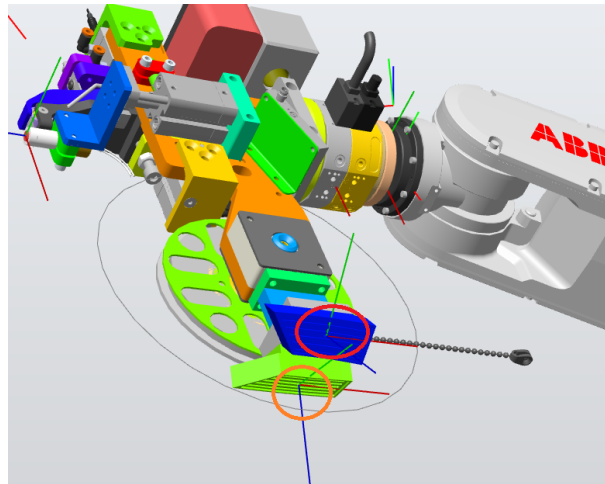


Figura 3.52: Localização dos TCPs "ToolVac_1" e "ToolVac_2" do quarto robô

3.2.6 Colisões

Durante as primeiras simulações realizadas, foi detetada a ocorrência de colisões, o que conduziu a algumas alterações na ferramenta. A figura 3.53 seguinte ilustra uma das colisões que ocorria entre a ferramenta e a base das peças. Deste modo, foi necessário alterar a ferramenta de colocação da fita, visto não ser possível proceder à alteração da base neste cenário.

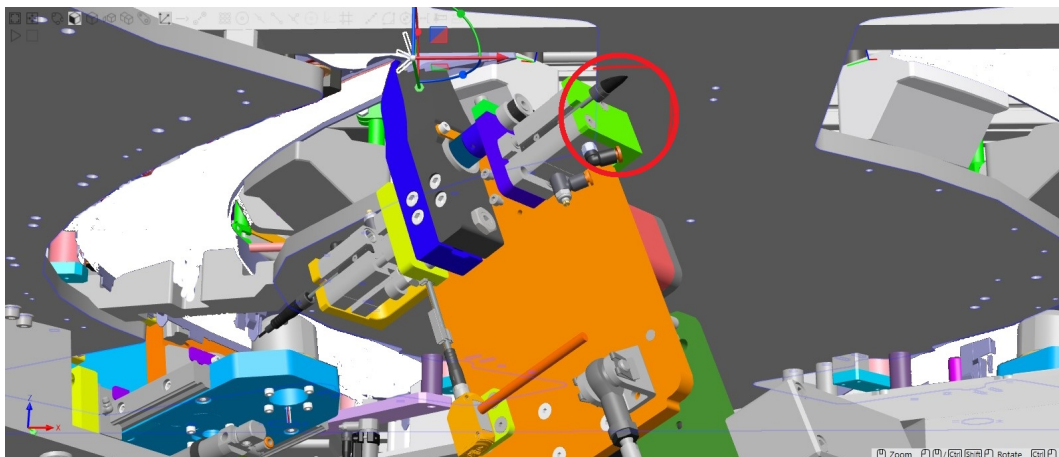


Figura 3.53: Colisão durante a inserção de fita na célula 1309

As sucessivas alterações à ferramenta, decorrentes das colisões mencionadas, motivaram a existência de várias versões até esta ser dada como finalizada. A ferramenta final é mais compacta na sua globalidade, com exceção do bico de papagaio (assinalado a vermelho na figura anterior), o que permite que esta desça evitando assim a ocorrência de colisões. A figura 3.54 representa a primeira versão e a última até ao momento da escrita do presente documento.

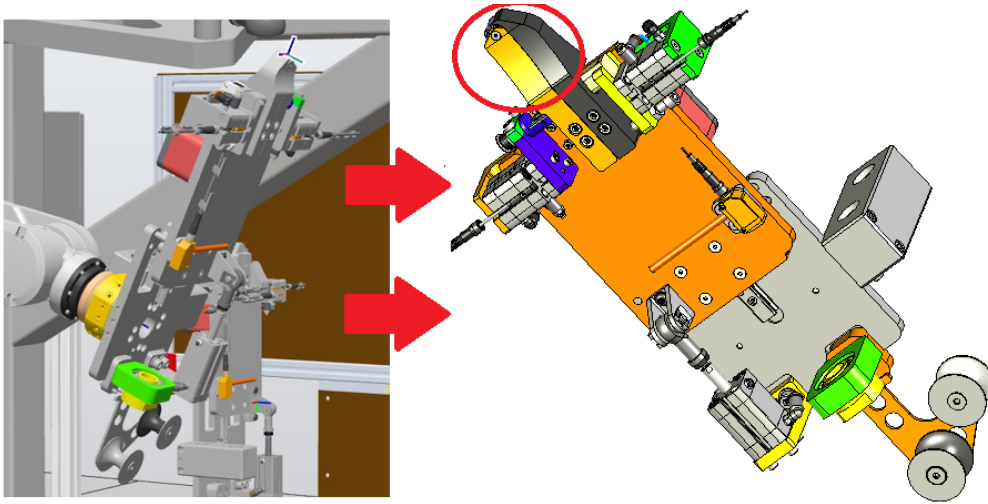


Figura 3.54: Representação da ferramenta da primeira versão e da última

É de salientar que as alterações mencionadas foram aplicadas a todas as ferramentas de aplicação de fita, tanto desta célula como de outras células descritas neste documento. Esta uniformização das ferramentas permite evitar colisões, dado que as bases das peças são semelhantes, mesmo para peças de diferentes fabricantes.

No que se refere à ferramenta de aplicação de tabs, foi também detetada uma colisão (figura 3.55) com a base, o que levou a várias iterações na edição da ferramenta.

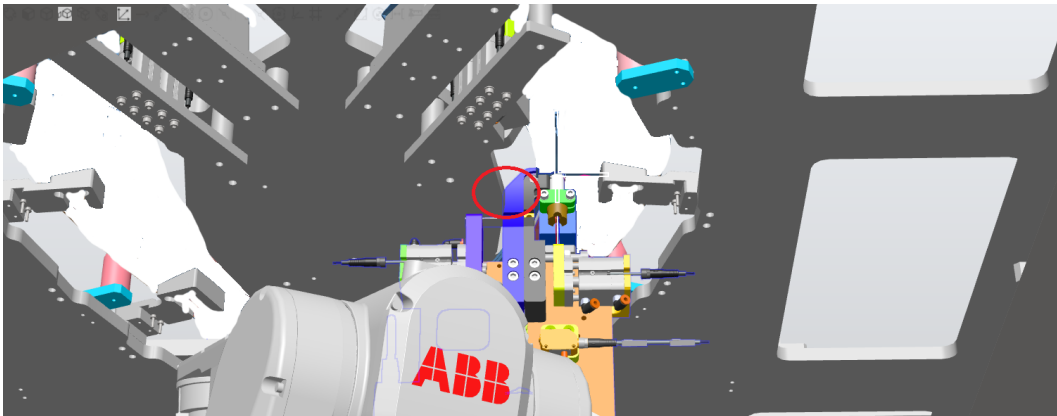


Figura 3.55: Colisão na aplicação de tabs na célula 1309

3.3 Célula PRJ 1311

Nesta célula, fui mais uma vez responsável pela criação da estrutura do código com os movimentos dos robôs e pela sua simulação, com o objetivo de identificar possíveis colisões.

3.3.1 Layout e descrição de funcionamento

A célula "PRJ 1311" está responsável por:

- Manipulação de peças;
- Inserção de clips.

Esta célula trabalha com duas referências de peças que estão divididas em peças do lado direito ou esquerdo do carro.

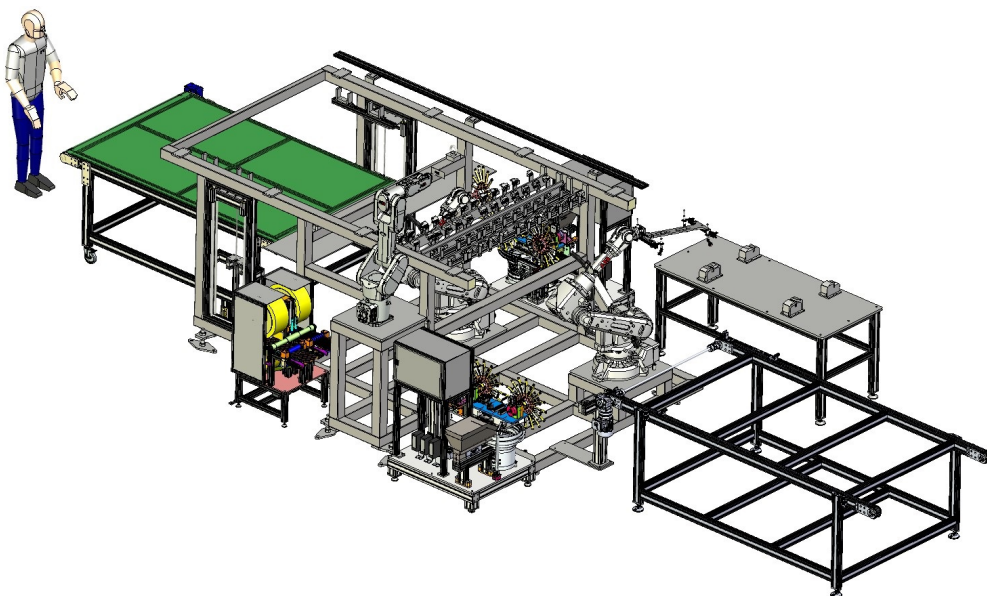


Figura 3.56: Layout da célula PRJ 1311

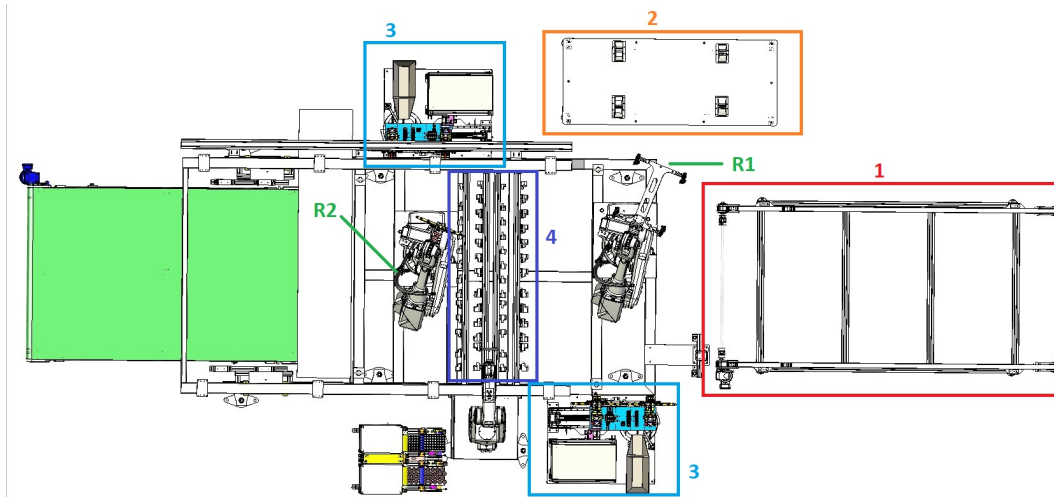


Figura 3.57: Representação da célula 1311 por áreas

Legenda da figura 3.57:

1. Tapete de entrada de peças;
2. Mesa de entrada manual de peças;
3. Máquina de carregamento automático de clips nas estrelas;
4. Zona de colocação de peças.

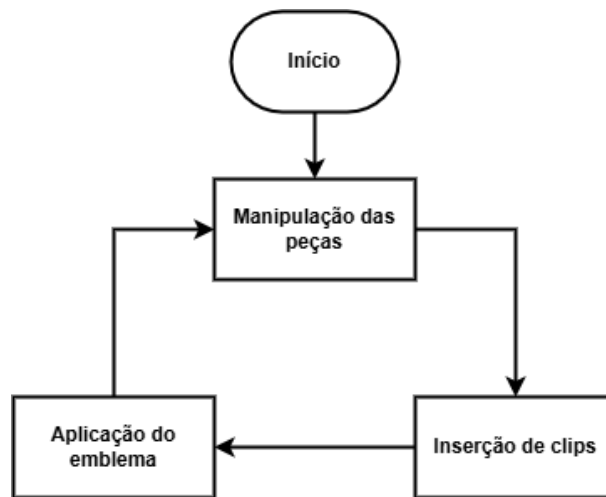


Figura 3.58: Fluxograma de funcionamento geral da célula 1311

3.3.2 Robô número 1

O robô número um é responsável por manipular as peças do tapete de entrada ou da mesa de entrada manual, para as bases. De seguida, este terá de substituir a ferramenta de manipulação das peças (a garra) por uma estrela de dez dedos, para realizar a inserção de clips nas peças. Por fim, coloca a ferramenta (estrela de dez dedos) na máquina de carregamento automático de clips.

Calibração de ferramentas

O presente robô está num cenário que necessita de duas ferramentas, uma garra e duas estrelas de dez dedos, logo foram configuradas três ferramentas. A ferramenta do acoplamento (na figura 3.59 está assinalado a localização do TCP) e as estrelas (na figura 3.60 está assinalado a sua localização), ficando a faltar a garra porque até ao momento ainda não tinha sido projetada.

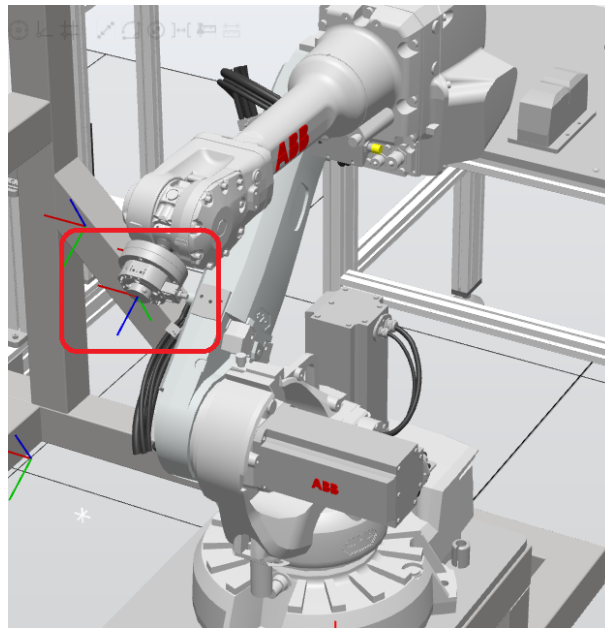


Figura 3.59: Localização do TCP do acoplamento do primeiro robô da célula 1311

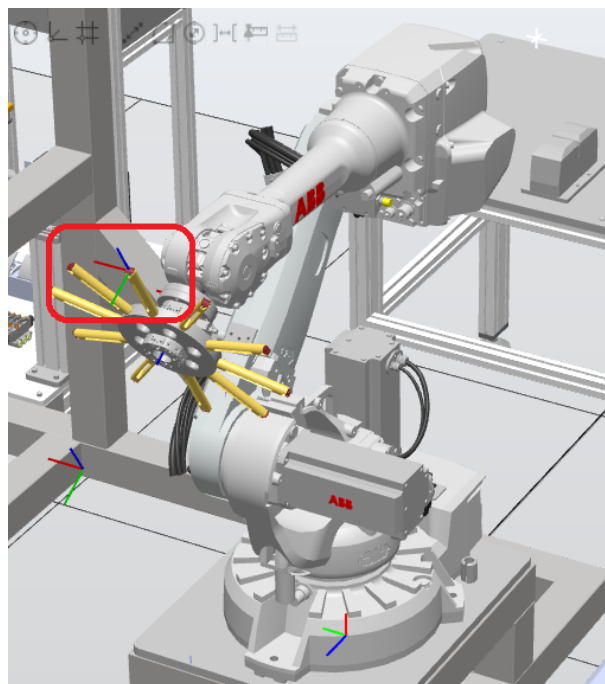


Figura 3.60: Localização do TCP da "Estrela de 10 dedos" do primeiro robô da célula 1311

Tal como descrito no subcapítulo anterior, foi necessário configurar um TCP para cada dedo, o que resultou na disposição de *layout* apresentada na figura 3.61.

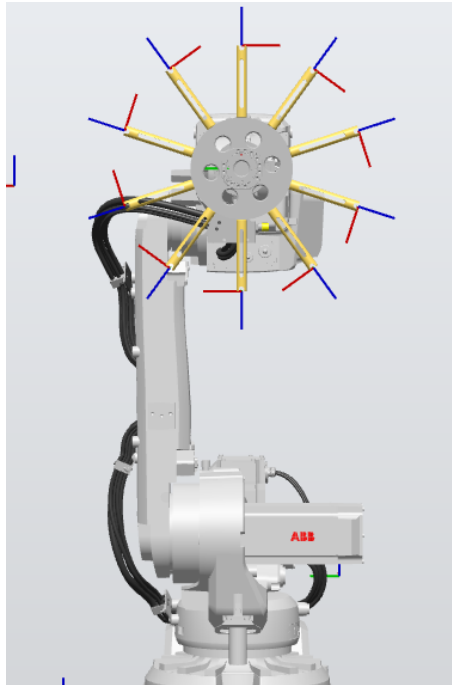


Figura 3.61: Localização dos novos TCPs da ferramenta "Estrela de 10 dedos" do robô nº 1 da célula 1311

3.3.3 Robô número 2

O robô número dois é responsável pela inserção de clips, pelo que terá de ter uma ferramenta do tipo "estrela" com acoplamento, visto que as estrelas serão carregadas por uma máquina de carregamento automático de clips.

Calibração de ferramentas

Mais uma vez procedeu-se com a calibração e a definição de ferramentas, com as localizações e sentidos muito semelhantes às calibrações anteriores, como demonstram as figuras 3.62 e 3.63

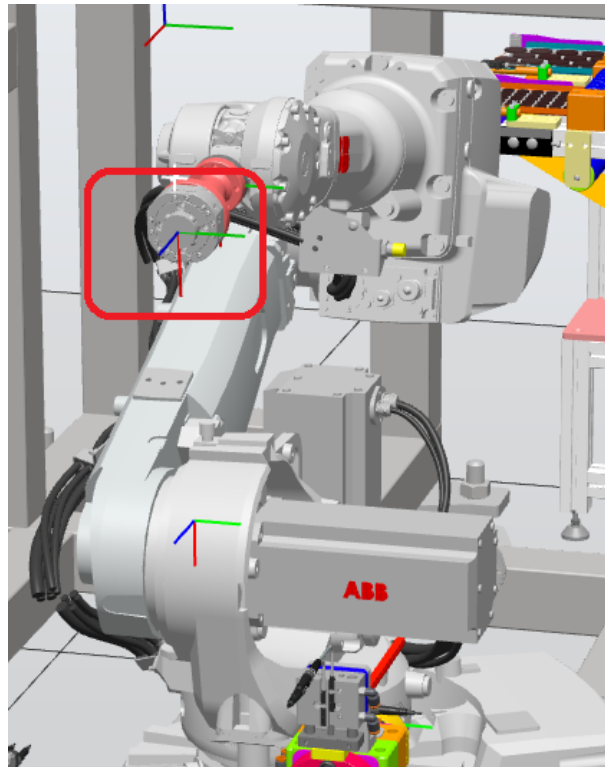


Figura 3.62: Localização do TCP do acoplamento do segundo robô da célula 1311

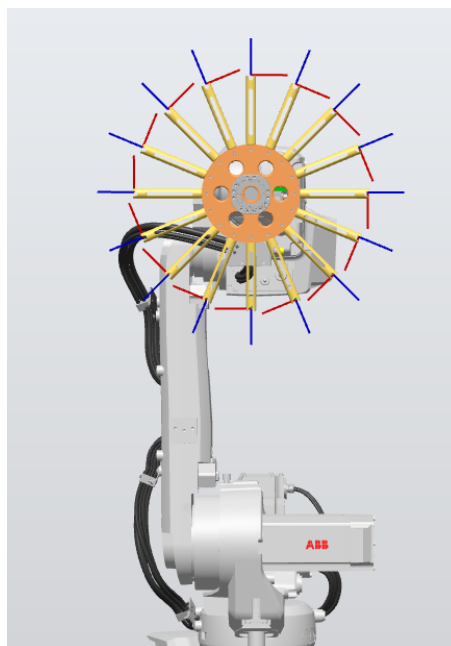


Figura 3.63: Localização dos novos TCPs da ferramenta "Estrela de 16 dedos" do robô nº 2 da célula 1311

3.3.4 Alterações das ferramentas

Nesta célula, foi identificada uma colisão com o robô número dois. Verificou-se que a última junta do robô colidia com a estrutura da célula. Para resolver o problema,

aumentou-se a distância entre o flange do robô e o acoplamento, como mostra na figura 3.64.

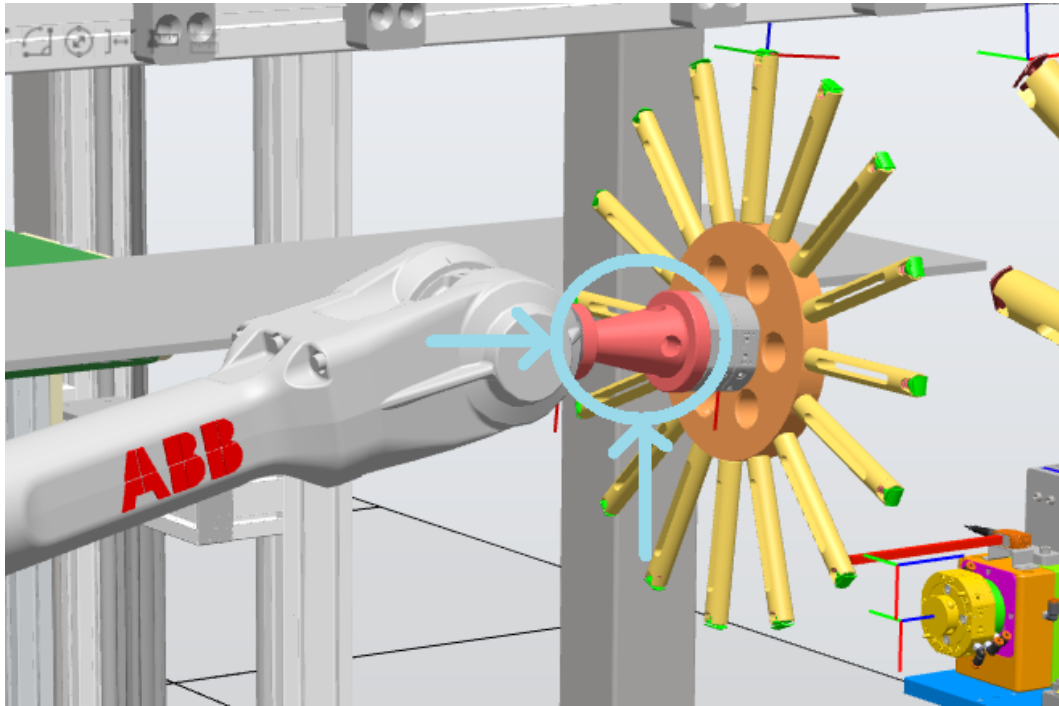


Figura 3.64: Alteração da ferramenta do robô número 2 da célula 1311

Mais uma vez, a utilização de configurações de ferramenta permitiu evitar a calibração de todos os pontos previamente assinalados. Com a substituição do TCP antigo pelo atual, todas as posições do robô foram atualizadas automaticamente.

4 PROJETOS REALIZADOS COM RECURSO A ROBÔS COLABORATIVOS

Neste capítulo, são apresentados quatro projetos desenvolvidos durante o estágio, com o objetivo de proporcionar uma introdução à robótica colaborativa. Cada projeto será descrito de forma a destacar as metodologias utilizadas, os principais resultados obtidos, oferecendo uma visão geral do percurso formativo no contexto da robótica colaborativa.

4.1 Soldadura a laser usando robô colaborativo

Neste subcapítulo, será abordado um projeto que utiliza um robô colaborativo para soldar duas peças com recurso a um laser. Refira-se que o robô colaborativo utilizado nas soldaduras a laser era propriedade do cliente. O principal desafio consistia em programar o robô para soldar peças cilíndricas, de acordo com alguns requisitos.

4.1.1 Descrição

Este projeto teve diferentes fases de implementação, durante as quais os requisitos foram modificados. Inicialmente, o robô teria de soldar duas peças cilíndricas com cerca de um centímetro de raio. As peças cilíndricas a soldar eram inicialmente sobrepostas manualmente. Posteriormente, o robô deveria incidir o laser na zona a soldar com uma inclinação específica descrevendo uma trajetória circular. Esse ângulo em relação à perpendicular da peça é programável, de modo a garantir que a soldadura é realizada em conformidade com as especificações. Numa fase posterior, solicitou-se que o robô efetuasse uma rota circular com diferentes valores de raio. Na versão final do programa, será possível definir dinamicamente o valor do raio e o ângulo de inclinação pretendidos.

A figura 4.1 ilustra a direção de incidência do laser nas peças, sendo que a cor verde indica o ângulo que o laser deve formar com a peça.

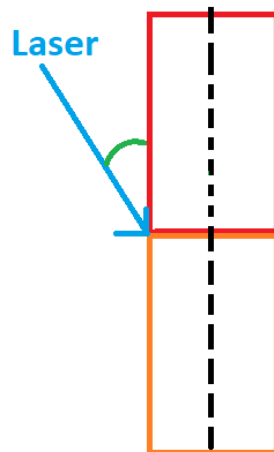


Figura 4.1: Exemplo do ângulo incidência do laser na peça

4.1.2 Robô utilizado

Para este projeto, foi utilizado um robô colaborativo, o JAKA ZU-12, modelo apresentado na figura 4.2, visto que o cliente já possuía este modelo e pretendia utilizá-lo para o efeito.



Figura 4.2: Robô colaborativo JAKA ZU-12

Este tipo de robôs utiliza uma linguagem de programação por blocos semelhante à programação "Scratch". Na figura 4.3 mostra uma parte do programa elaborado no software da JAKA.

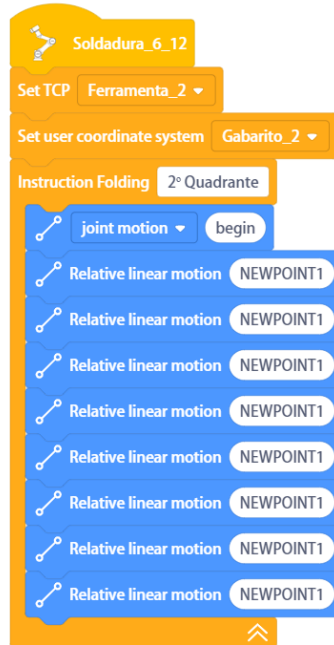


Figura 4.3: Exemplo de programação dos robôs JAKA

4.1.3 Primeira Proposta

Inicialmente, uma vez que o cliente não necessitava de um programa dinâmico e apenas queria confirmar se o robô seria capaz de realizar uma soldadura, foi necessário garantir que este não pudesse rodar sobre o seu eixo longitudinal, utilizando o robô disponível. Para esse efeito, procedeu-se à marcação manual de pontos numa circunferência, dividindo-a em vários pontos (24) e marcando-os um a um.

Antes de proceder à marcação dos pontos, foi necessário configurar a ferramenta e o *framework*. Este tipo de configuração permite efetuar ajustes finos nos pontos. Os *frameworks* permitem que o utilizador configure várias áreas de trabalho e marque pontos dentro dessas áreas. Se a área de trabalho sofrer alterações nas suas coordenadas, todos os pontos previamente marcados nessa área são "arrastados" para a nova posição, facilitando os ajustes e poupando tempo no processo de *setup*. Já no caso da configuração da ferramenta, esta permite definir um ponto virtual e um sistema de orientação novo em relação ao flange do robô. Para configurar determinadas posições, esta abordagem pode ser mais rápida do que utilizar o sistema de orientação "mundo". Por exemplo, num trabalho que implique a rotação da ferramenta, em X, Y, ou Z, é necessário configurar a ferramenta, para que esta efetue a rotação pretendida. Esta nova abordagem na configuração permite simplificar o código.

Como esta abordagem não era muito prática, recorreu-se a uma função do robô que permite trabalhar por incrementos.

4.1.4 Segunda Proposta

Visto que o robô consegue percorrer a trajetória pretendida, o cliente solicitou que o programa fosse dinâmico. Agora, é necessário inserir o raio do cilindro no início do programa, para que o robô ajuste todos os pontos de acordo com esse valor. Inicialmente, é definido o sistema de coordenadas, designadamente o *user frame* e o TCP da ferramenta, e são inicializadas as variáveis do raio da peça a soldar e do ângulo de incidência do feixe laser.

O funcionamento está descrito no fluxograma do Anexo D.

A figura 4.4 apresenta o código de inicialização do programa.

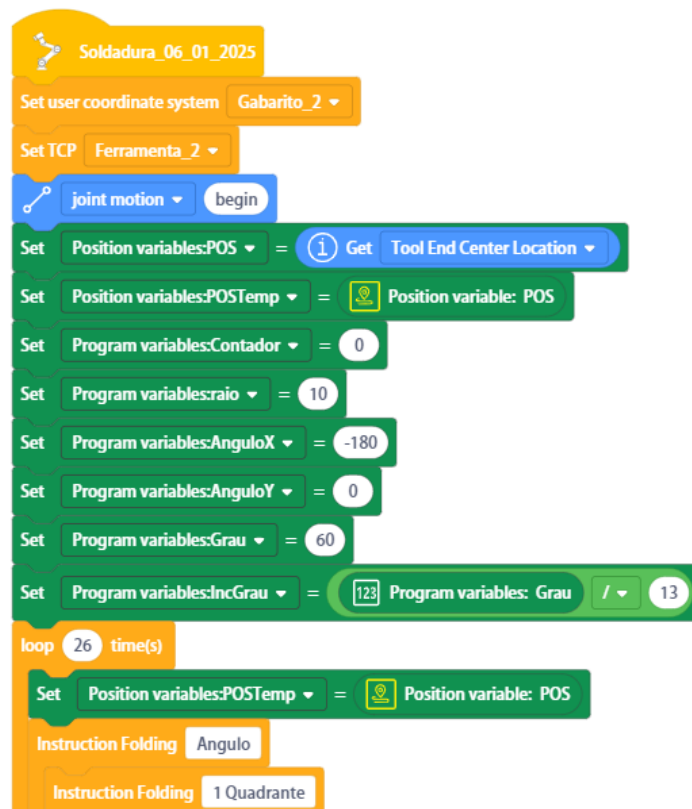


Figura 4.4: Inicializações do programa da segunda proposta

O código seguinte está subdividido em quadrantes. Cada quadrante tem um algoritmo distinto, adequado à posição do manipulador para realizar a soldadura nesse quadrante específico. O programa atual trabalha apenas com um ponto físico marcado, todos os outros pontos são calculados. O algoritmo trabalha com um *array* de seis posições, em que cada posição do *array* corresponde às posições X, Y, Z e às respectivas rotações de cada eixo.

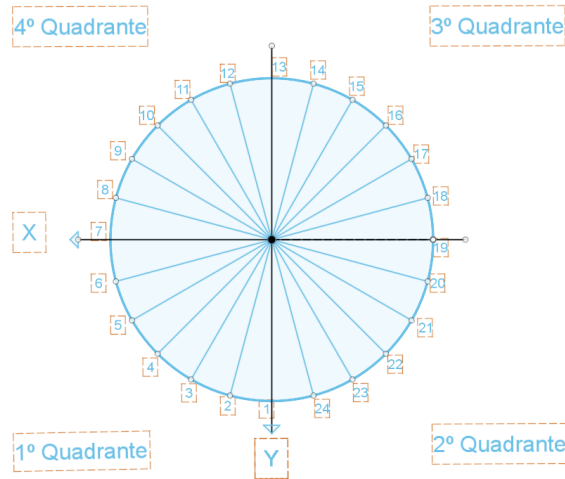


Figura 4.5: Representação 2D do círculo

No anexo C está presente o código desenvolvido.

4.2 Alimentador de CNC usando Robô colaborativo

No âmbito da produção em massa, foi necessário modificar um processo industrial que envolvia uma máquina CNC e um robô colaborativo responsável por alimentar a máquina. Inicialmente, o robô manipulador fornecia à CNC peças brutas de latão, que esta transformava em dois mosquetões, utilizados na produção de malas de senhora de elevada qualidade. O bloco de latão bruto era desbastado pela CNC, resultando numa peça que era posteriormente, cortada ao meio, obtendo-se, assim os dois mosquetões. Estes são, por fim, finalizados com a aplicação do revestimento solicitado. O processo de maquinação das peças implica a modificação da programação do robô, de modo a que este possa realizar novas tarefas.



Figura 4.6: Ilustração das mudanças na peça por fases

O novo processo consiste em três etapas distintas. Inicialmente, no primeiro ciclo de funcionamento, um operador humano coloca uma peça bruta em cada um dos gabaritos da mesa um da CNC. A mesa vira, a CNC maquina as peças brutas resultando na "peça fase intermédia 1", apresentada na figura seguinte. Durante a maquinação das peças pela CNC, o operador coloca as peças das fases intermédias um e dois na mesa dois da CNC. Posteriormente, o processo de fabrico passa a ser automático, sendo o operador humano substituído por um robô colaborativo. A referida máquina está representada na figura 4.11

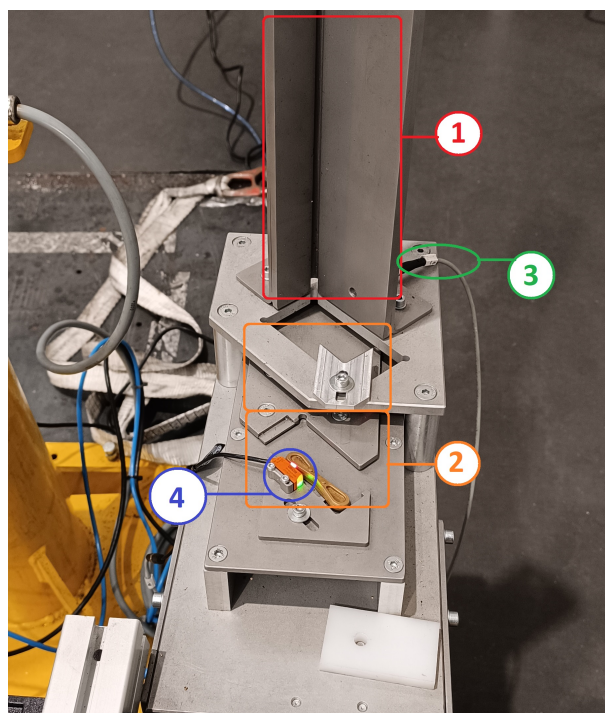


Figura 4.7: Alimentador de peças

Legenda:

1. Armazém de peças brutas;
2. Zona de captura de peças e dispensação de peças;
3. Sensor de presença de peças brutas;
4. Sensor de presença de peças na zona de captura.

O robô é alimentado por peças brutas por um alimentador automático, colocando cada uma delas num gabarito da mesa um. A CNC desbasta as peças brutas presentes nesses gabaritos e corta as peças da fase intermédia um resultando em oito mosquetões. De seguida é necessário fazer uma rosca na face do topo da peça. A figura 4.6 representa as peças em diferentes fases do processo de fabrico.

Além de alimentar a CNC, o robô terá de colocar os mosquetões previamente concluídos em *blister*. Para programar esta funcionalidade, é utilizada uma função de pa-

letização do robô JAKA. Cada *blisters* tem uma capacidade de armazenamento de 48 peças. Assim, o tabuleiro, que contém quatro *blisters*, tem uma capacidade agregada de armazenamento de 192 peças.

As figuras seguintes 4.8 e 4.9 representam os gabaritos do lado da CNC, identificados por cada fase de produção.

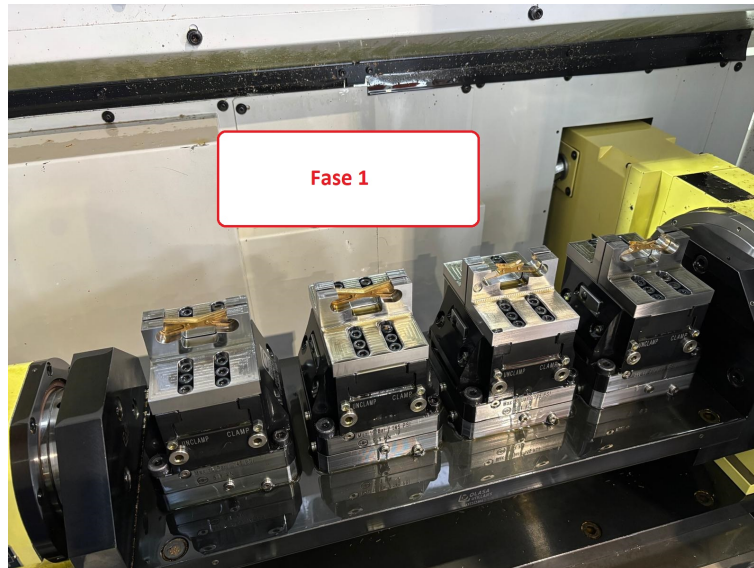


Figura 4.8: Gabarito número 1

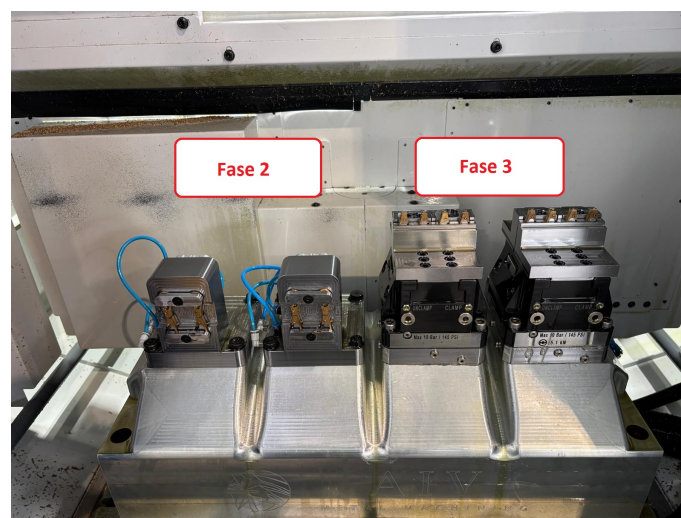


Figura 4.9: Gabarito número 2

Refira-se que enquanto a CNC maquina as peças dispostas num dos gabaritos, o robô retira as peças concluídas e coloca novas peças brutas no outro gabarito. Para facilitar o processo, e por motivos de segurança, existe uma separação física entre a área de maquinação da CNC e a área de manipulação das peças nos gabaritos. Quando o robô termina a sua função, ele envia um sinal à CNC para indicar que existe um gabarito

com peças novas prontas a serem maquinadas. Quando a CNC termina a maquinação das peças, os gabaritos são reposicionados por meio de uma mesa rotativa. A figura 4.10 mostra o fluxograma do funcionamento geral da máquina.

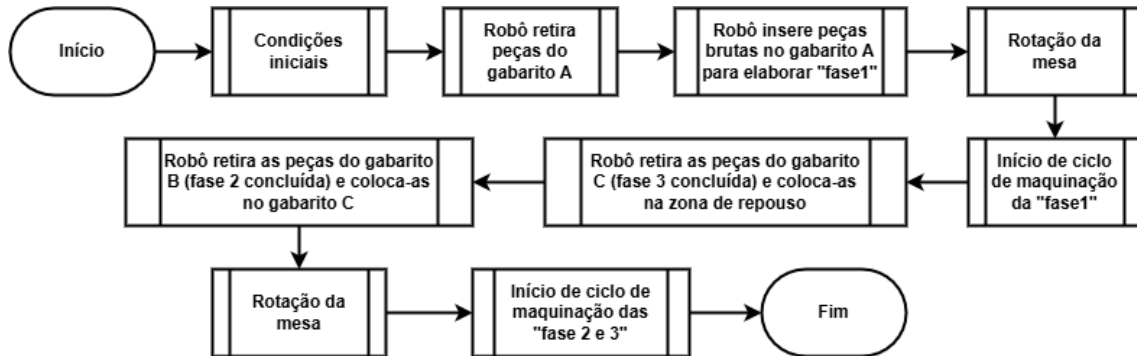


Figura 4.10: Fluxograma do funcionamento geral da máquina em um ciclo de operação

A máquina onde o robô está inserido é constituída por várias partes, nomeadamente o alimentador de peças brutas (figura 4.7), o tabuleiro dos *blisters* e a zona de repouso assinalados na figura 4.12.

O alimentador, representado na figura 4.7, (sujeito a alterações) é responsável por manter uma quantidade considerável de peças brutas no seu armazém, por detetar se alguma peça ficou encravada e por fornecer as peças para a zona de captura.

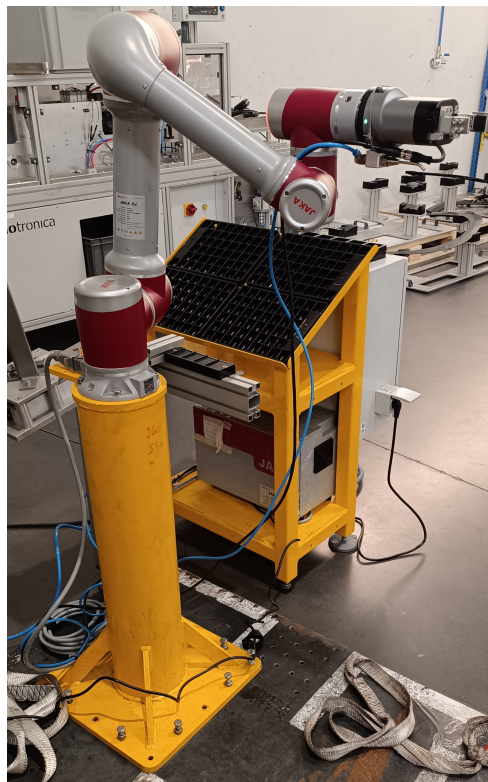


Figura 4.11: Estrutura do alimentador da CNC

O tabuleiro é responsável pelo armazenamento de peças concluídas e tem espaço para quatro *blisters*. Na parte inferior da máquina (zona 1), existe uma área de repouso destinada a armazenar peças provenientes da primeira fase, para que, estas possam ser colocadas no gabarito correspondente à segunda fase. A figura 4.12 representa todas as zonas.

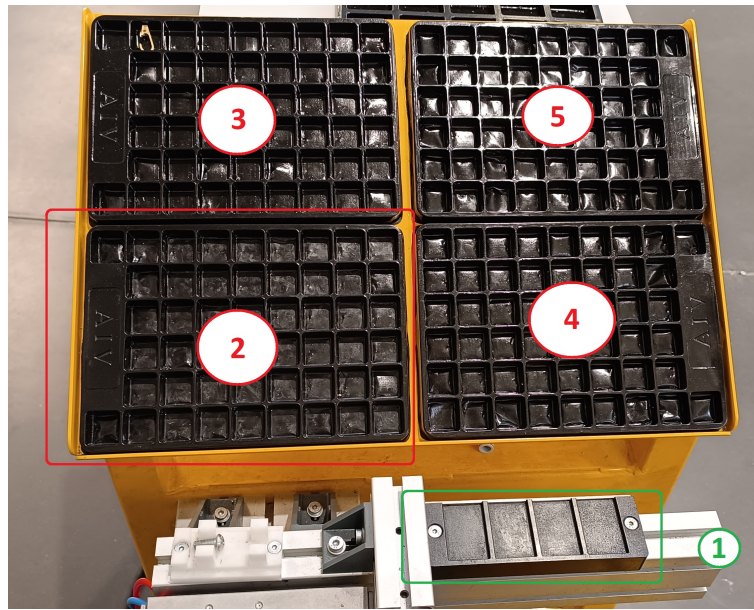


Figura 4.12: Tabuleiro com os quatro *blisters*

Legenda da figura anterior

1. Zona de repouso;
2. *Blister* 1;
3. *Blister* 2;
4. *Blister* 3;
5. *Blister* 4.

4.2.1 Configurações do robô

Inicialmente, foram configurados três *user frames* na ferramenta de software do robô JAKA, as quais estão assinaladas a vermelho na figura 4.13.

Name	X(mm)	Y(mm)	Z(mm)	RX°	RY°	RZ°	Edit
Maquina	-593.193	-270.866	180.000	0.000	0.567	90.000	
tabuleiro	461.181	436.325	108.305	29.888	0.000	-90.000	
mat prima	139.643	308.445	123.405	174.398	-6.064	43.916	
tabuleiro 2	550.223	-455.171	5.600	30.331	-0.266	177.323	
mat prima 2	452.192	-202.473	86.166	2.292	-1.962	-177.812	
ZR Daniel	403.167	286.898	268.365	34.924	1.970	-93.267	
USRFRM7	0.000	0.000	0.000	0.000	0.000	0.000	
USRFRM8	0.000	0.000	0.000	0.000	0.000	0.000	
USRFRM9	0.000	0.000	0.000	0.000	0.000	0.000	
firstzero	391.136	154.906	400.432	0.000	0.000	-90.000	

Figura 4.13: Definição de *user frames*

Cada *user frame* corresponde a uma área de trabalho distinta: o "tabuleiro 2" corresponde à zona de partida das peças prontas, a "mat prima" corresponde à zona de captura de peças brutas e a área "Maquina" corresponde à zona da máquina CNC.

O peso e o centro de massa da ferramenta foram igualmente configurados, conforme é possível verificar na figura 4.14.

Figura 4.14: Configurações da carga da ferramenta

A configuração do peso e do centro de massa da ferramenta permite que o robô ajuste as suas rotas. Sabendo que o flange tem um peso adicional de dois quilogramas, é

necessário configurar o centro de massa numa nova posição, seis centímetros mais à frente no eixo X. Após a conclusão das configurações, foi necessário definir o programa de arranque do robô. Isto permite que o programa tenha início com as ordens dadas pelo comando externo. A figura 4.15, mostra a configuração do programa de arranque.

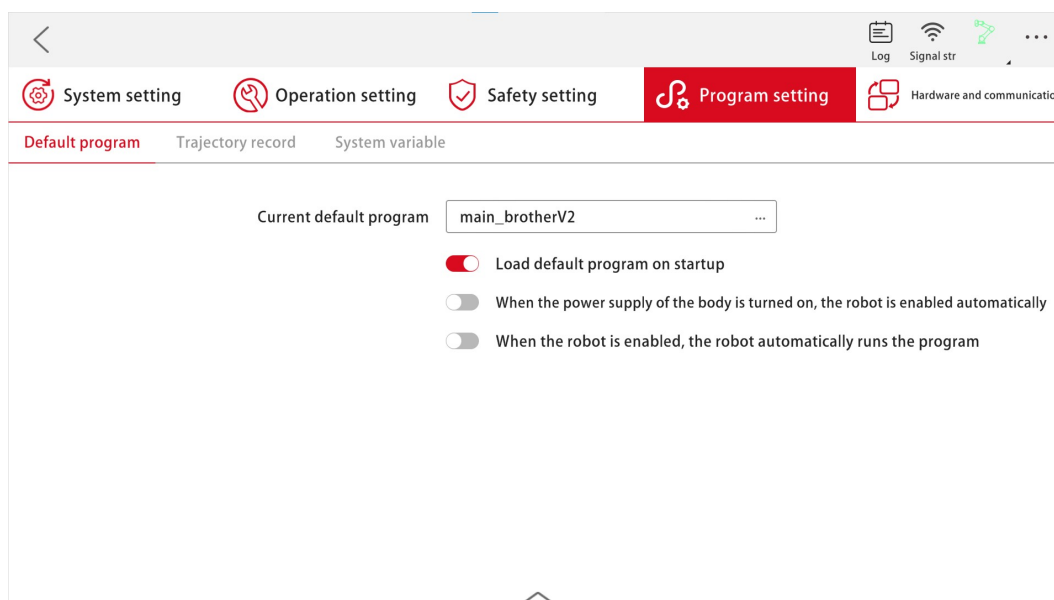


Figura 4.15: Configuração do programa de arranque

Estes robôs possuem um comando externo que permite executar algumas funções básicas sem ser necessário ligar um computador.

4.2.2 Fluxogramas de funcionamento

O programa do robô está dividido em vários programas e subprogramas, sendo o programa principal descrito pelo fluxograma apresentado no Anexo E.

O Anexo E ilustra o fluxograma relativo ao funcionamento do robô no processo de preparação do gabarito número um. O Anexo G apresenta o fluxograma do robô relativo à preparação do gabarito número dois.

4.3 Parametrização de um sistema de visão para um sistema robotizado de detecção de peças

A detecção de peças é uma etapa fundamental nos processos industriais modernos, sobretudo nas linhas de produção automatizadas. Este processo consiste na identificação, localização e classificação de componentes ou produtos ao longo da cadeia de produção. Tradicionalmente, esta tarefa era realizada por sensores simples ou operadores humanos, limitando a precisão, a velocidade e a consistência do processo.

Com o avanço das tecnologias de visão computacional, a utilização de câmaras de visão tornou-se uma solução eficaz e amplamente adotada para a deteção de peças. Em conjunto com algoritmos de processamento de imagem, estas câmaras permitem uma análise visual detalhada do ambiente de produção em tempo real.

Para demonstrar dois produtos comercializados pela Fluidotronica em feiras e exposições, foi montada uma pequena célula de demonstração. Este projeto utiliza uma câmara de reconhecimento 2D para detetar peças com o mesmo formato, as quais são posteriormente manipuladas por um robô colaborativo. A câmara fornece as coordenadas para a captura das peças ao robô, utilizando a rede de comunicação Ethernet. A minha participação neste projeto compreendeu a configuração do sistema de visão. A figura 4.16 representa o sistema em questão.

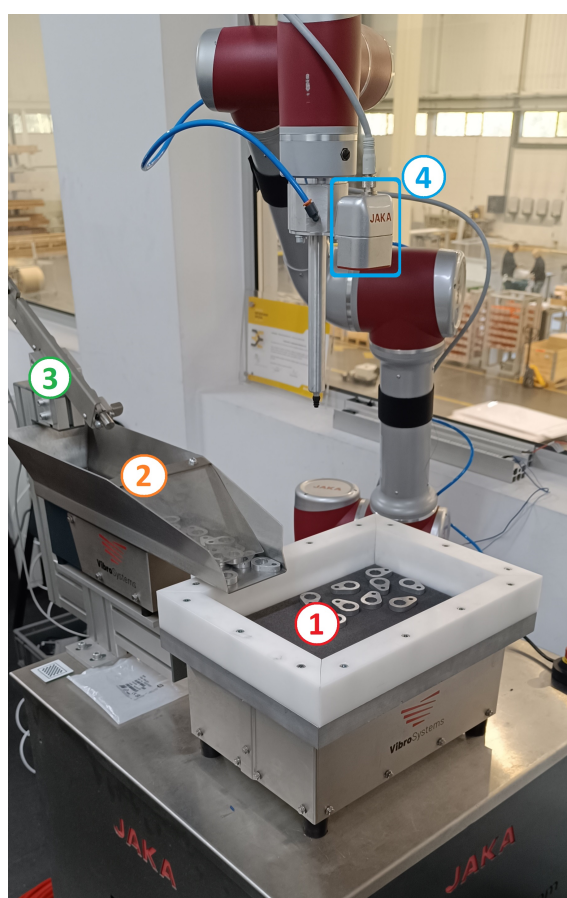


Figura 4.16: Layout do projeto de visão

Legenda da figura anterior:

1. Base de receção de peças;
2. Silo;
3. Calha de armazenamento de peças;
4. Câmara de visão.

De seguida, irei explicar o funcionamento do sistema desenvolvido. As peças são inicialmente introduzidas manualmente no silo por uma pessoa e, de seguida, é iniciado o funcionamento do robô. Nesse momento, o robô desloca-se até a uma posição perpendicular e ligeiramente superior à posição da base de receção de peças, onde procede ao reconhecimento das peças. Inicialmente, caso a base não tenha peças, o sistema vibratório do silo e da base é ativado por um curto período de tempo, até que a base tenha peças e estas sejam detetadas pela câmara. Após a geração das coordenadas absolutas da peça, estas são transmitidas da câmara para o robô por meio de comunicação *Transmission Control Protocol / Internet Protocol* (TCP/IP). O robô processa as coordenadas recebidas para proceder à manipulação das peças provenientes da base para a calha de armazenamento, conforme representado na figura 4.17.

Refira-se que, é necessário que o sistema de vácuo fique centrado no meio dos dois orifícios da peça para que esta possa ser manipulada.

Assim que a calha estiver cheia de peças provenientes da base, o robô abre a comporta e as peças caem novamente no silo, num processo repetitivo e demonstrativo dos sistemas de visão, processamento e manipulação utilizados.

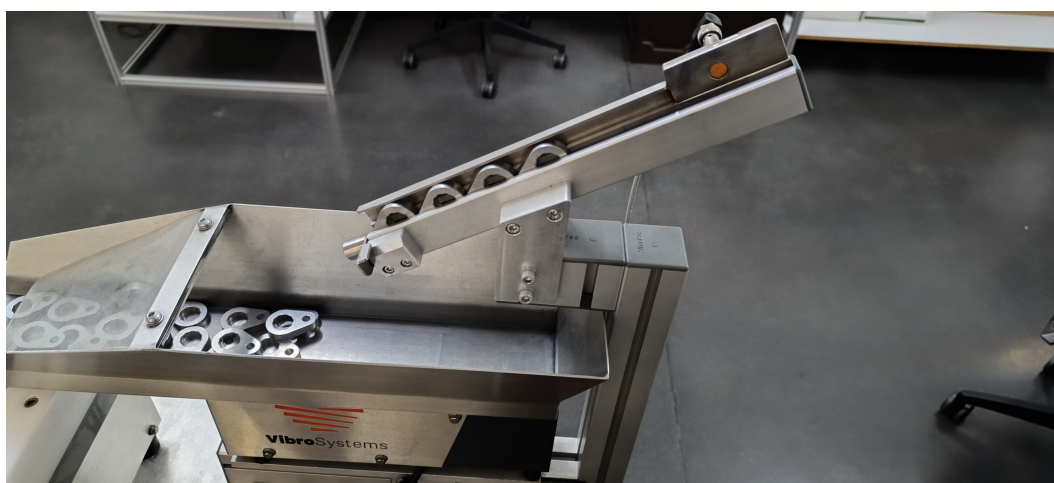


Figura 4.17: Calha de armazenamento de peças

A movimentação das peças do silo para a base é feita recorrendo a um sistema de vibração incorporado em ambos os componentes: base e silo. A vibração que ocorre no silo faz com que as peças se desloquem em direção à base. A vibração da base tem como objetivo evitar que as peças fiquem sobrepostas, facilitando o processo de reconhecimento por parte do sistema de visão. A Figura 4.18 mostra os dois sistemas vibratórios, visualizados a partir de outro plano de visão.

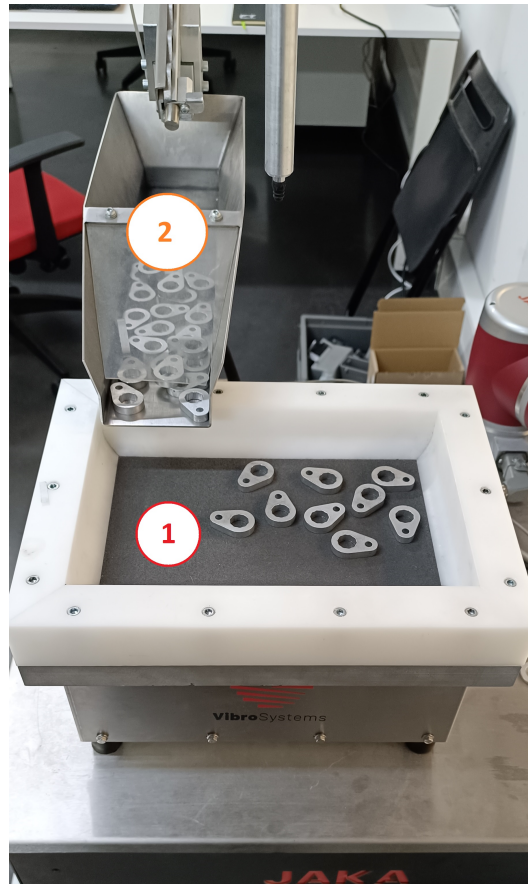


Figura 4.18: Silo e Base

4.3.1 Parametrização da câmara

Os passos estão subdivididos nas seguintes etapas:

- Calibração visual da câmara;
- Configuração da conexão;
- Criar o projeto de reconhecimento.

Inicialmente, a câmara apresentava uma configuração de comunicação desconhecida. O primeiro passo consistiu, portanto, em identificar o seu endereço IP, dado que a câmara comunicava através do protocolo TCP/IP. Para esse efeito, foi utilizado o programa GigeCameraConfig x64, que indicou que a câmara estava configurada com o endereço IP 192.168.120.15, conforme representado na figura 4.19. Com esta informação, concluiu-se que a gama de endereços IP à qual o dispositivo deveria pertencer era 192.168.120.X.

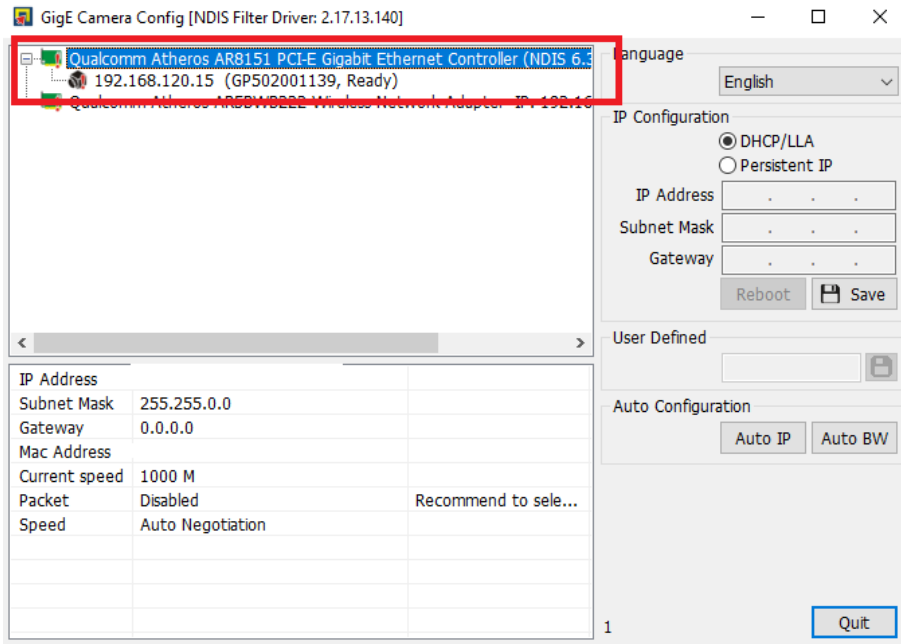


Figura 4.19: Endereço IP da câmara

De seguida, foram refeitas as ligações físicas de rede, a câmara que estava ligada diretamente ao PC para efetuar a sua configuração, é neste ponto ligada a uma porta do controlador do robô. Existe, contudo, outra ligação de Ethernet entre o PC e o controlador para efeitos de programação. As ligações estão representadas na figura 4.20.

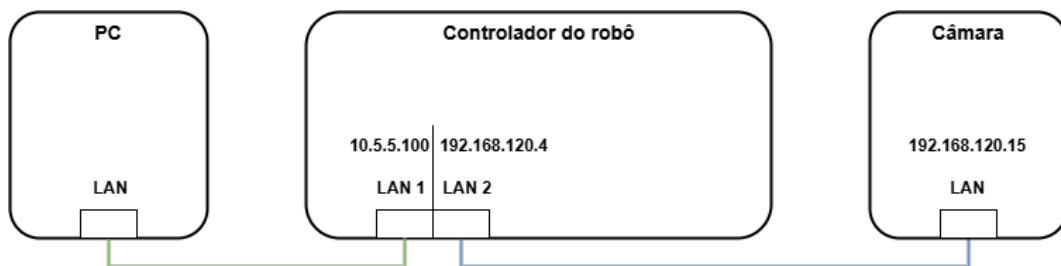


Figura 4.20: Configuração das ligações da rede

Após a conclusão das ligações e da configuração do endereço IP da câmara, foi necessário reconfigurar a porta 'LAN 2' do controlador, representada na figura 4.21. Inicialmente, esta porta encontrava-se com a definição de IP pré-configurada para o modo DHCP. No entanto, para este caso específico, foi necessário atribuir-lhe um IP estático dentro da gama de IPs previamente identificada.

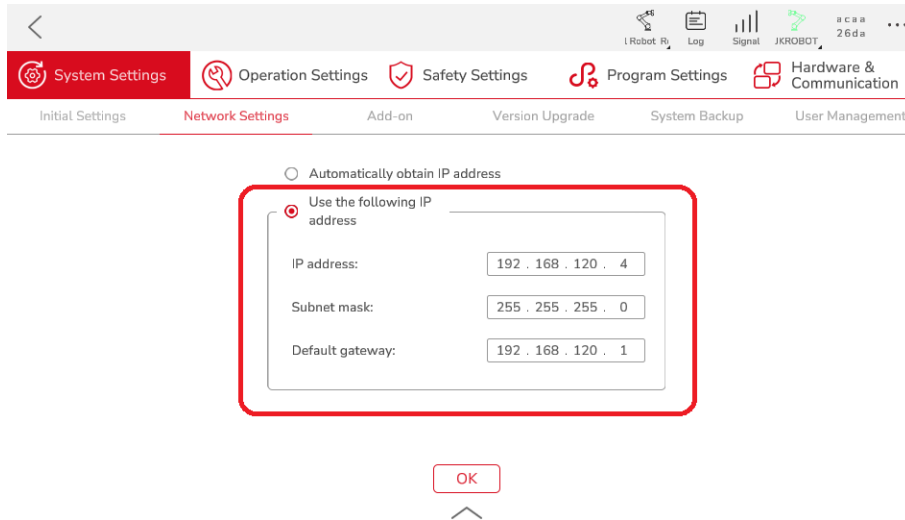


Figura 4.21: Configuração da LAN 2 do controlador do robô

Somente após a conclusão de todos os passos anteriores foi possível iniciar a configuração do software de visão. A câmara utilizada para reconhecimento de peças foi a "JAKA Lens 2D", da mesma marca do robô, o que facilitou a troca de comandos entre ambos.

Relativamente ao ambiente de parametrização da câmara, o controlador permite que um computador ligado à mesma rede da 'LAN 1' aceda ao software de configuração. O acesso ao software é efetuado através de uma interface *web*, utilizando o endereço IP da *gateway* predefinida, conforme ilustrado na figura 4.22.

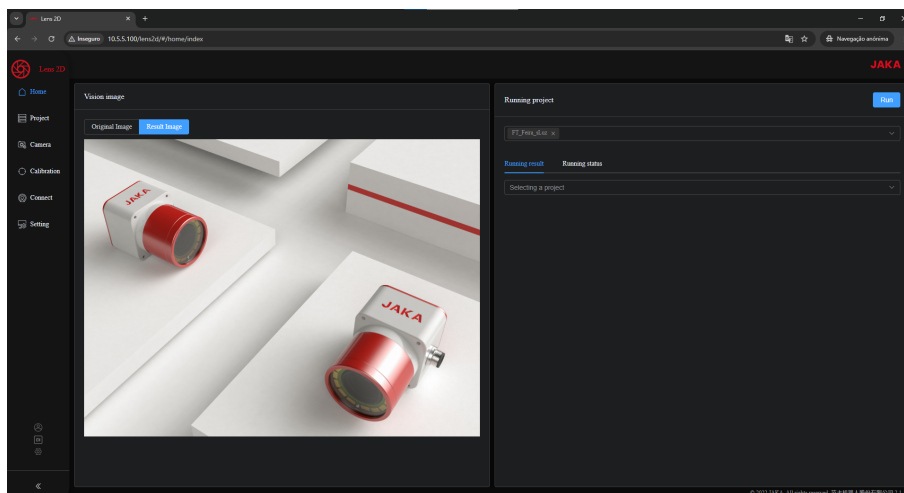


Figura 4.22: Ambiente de parametrização da câmara

Um dos primeiros passos consistiu em definir uma posição específica para a captura de imagens no robô, a qual é armazenada na memória. De seguida, a lente focal foi ajustada para garantir a obtenção de uma imagem nítida. Concluída esta etapa, a câmara foi calibrada, o que envolveu a captação de várias imagens em diferentes posições, mas

sempre no mesmo plano do ponto de captura, em torno de uma matriz de calibração. Com o *setup* atual — ou seja, a câmara montada no flange do robô — os parâmetros de calibração obtidos estão representados na figura 4.23."

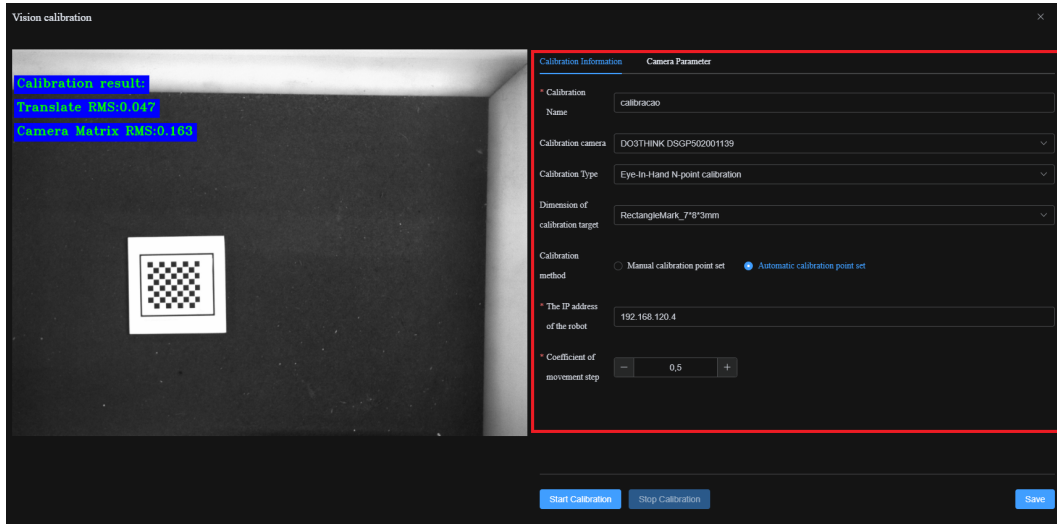


Figura 4.23: Parâmetros de calibração

Uma vez concluída a calibração, os valores "*Translate RMS*" e "*Camera matrix RMS*" são representados no ambiente de parametrização. Estes parâmetros permitem corrigir a distorção da lente, para uma reconstrução correta da figura 3D a partir de várias fotos 2D.

- O "*Translate RMS*" é o valor do erro médio da posição dos pontos do mundo real (quadrado) e as projeções da câmara após a calibração.
- O "*Camera matrix RMS*" contém os parâmetros intrínsecos da câmara, que incluem a distância focal, o ponto principal (centro da figura), e os coeficientes de distorção. O "*Camera Matrix RMS*" representa o erro médio associado à estimativa desses parâmetros intrínsecos[28].

Após concluída a calibração do ponto focal, procedeu-se à configuração do algoritmo de deteção de peças, por meio de dois projetos que traduziam diferentes condições de iluminação. As diferenças entre os projetos residem apenas nos parâmetros de captura da imagem, que serão apresentados mais adiante neste documento. Com recurso à página *web* "Lens 2D", que tem o mesmo endereço de IP do robô, foi possível construir o algoritmo. Esta página contém várias ferramentas incorporadas, nomeadamente de deteção de linhas, círculos, formas, códigos QR, entre outras. Contudo, para este efeito, optou-se por utilizar apenas duas ferramentas: a de captura de imagens e a de deteção de formas, recorrendo a imagens em tons de cinza. A figura 4.24 representa o *flowchart* do algoritmo.

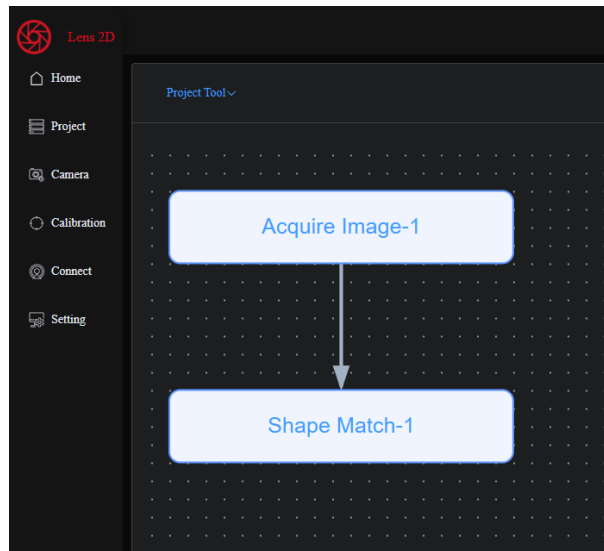


Figura 4.24: Algoritmo de detecção de peças

Dentro de cada bloco estão presentes vários parâmetros a configurar, começando pelo bloco *Acquire Image*, presente na figura 4.25.

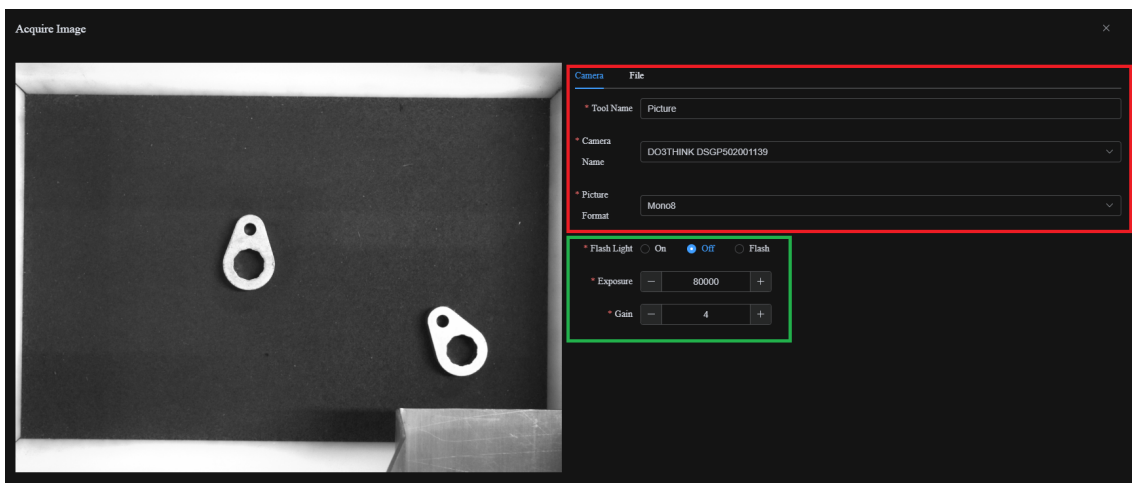


Figura 4.25: Parâmetros do bloco de captura de figura

Na figura anterior, a câmara selecionada e o formato da figura estão assinalados a vermelho. Neste caso, foi selecionado o formato "Mono8" que corresponde a uma imagem a tons de cinzento (oito bits). Os elementos assinalados a verde correspondem a parâmetros fulcrais para a obtenção de uma imagem de qualidade, nomeadamente os parâmetros de ganho da imagem e de exposição da imagem. O ganho corresponde ao fator de amplificação da intensidade da imagem, ou seja, a um aumento dos valores de cor (neste caso, o branco). Por outro lado, o parâmetro de exposição refere-se ao tempo durante o qual o sensor da câmara permanece exposto à iluminação. Uma calibração adequada destes parâmetros melhora a capacidade de reconhecimento de padrões. A

única alteração necessária na configuração da câmara com iluminação própria foi precisamente a modificação dos parâmetros de ganho e de exposição.

O segundo bloco da figura 4.24 tem como função principal a deteção de formas. Uma vez que o modelo se baseia num algoritmo de treino, este foi previamente treinado. Para tal, é necessário assinalar a área de treino, o ponto de referência e a área de pesquisa. A peça no centro da área delimitada por um quadrado azul foi definida como a peça de treino do algoritmo. A vermelho está assinalado o ponto de referência e de captura da peça. Para além da seleção da área de treino, foram também definidas duas zonas adicionais, responsáveis por identificar formas idênticas à forma de treino, assinaladas a laranja.

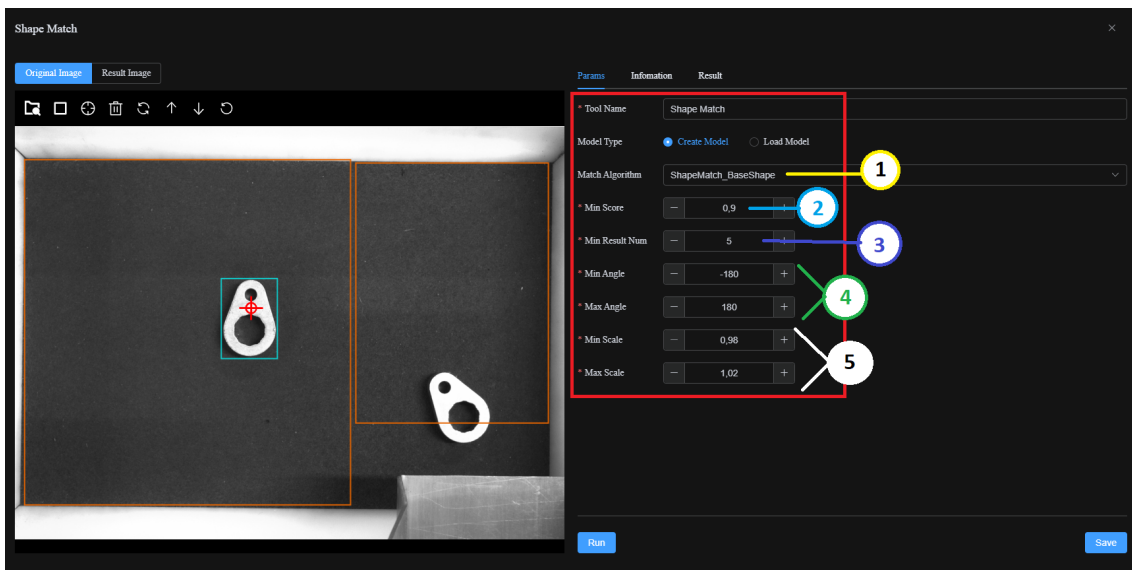


Figura 4.26: Parâmetros do bloco de reconhecimento de formas

Para a obtenção de uma peça conforme, foram escolhidos os valores anteriores, que se baseiam no seguinte:

1. Algoritmo de deteção de peça;
2. Valor mínimo de percentagem de correspondência com a peça de treino;
3. Teto máximo de peças a detetar;
4. Intervalo de rotação que a peça pode ter, para ser detetada em comparação com a peça de treino;
5. Intervalo de discrepância de tamanho da peça em comparação com a original.

A existência de duas áreas de deteção de peças visa maximizar o espaço útil para a identificação de objetos. Não foi definida uma única área de deteção que abrangesse toda a imagem, pois algumas zonas apresentam uma elevação significativa no eixo Z (no silo), encontrando-se em patamares diferentes. Não se pretende que o robô capture peças nessas regiões, motivo pelo qual essas áreas foram excluídas da deteção.

O passo seguinte compreendeu a configuração da comunicação, definindo: o tipo de comunicação, o endereço IP, a porta e os comandos de recepção, ilustrados na figura 4.27.

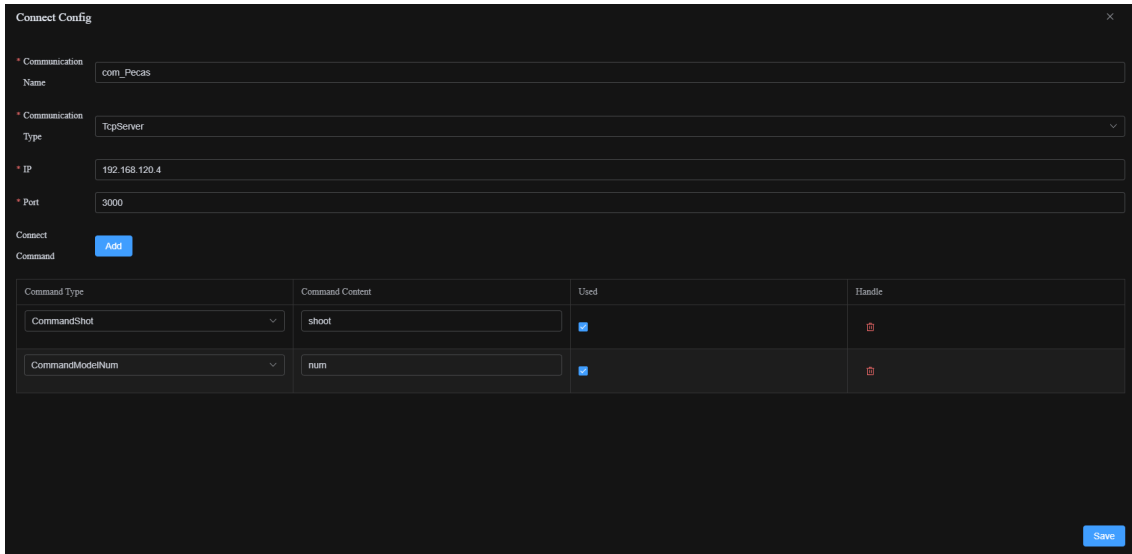


Figura 4.27: Parâmetros de comunicação

Consoante o comando recebido, a câmara executará uma tarefa específica. Esses comandos estão apresentados na figura anterior. O passo seguinte consiste na configuração das coordenadas absolutas, que correspondem ao centro da peça definida na figura 4.28.

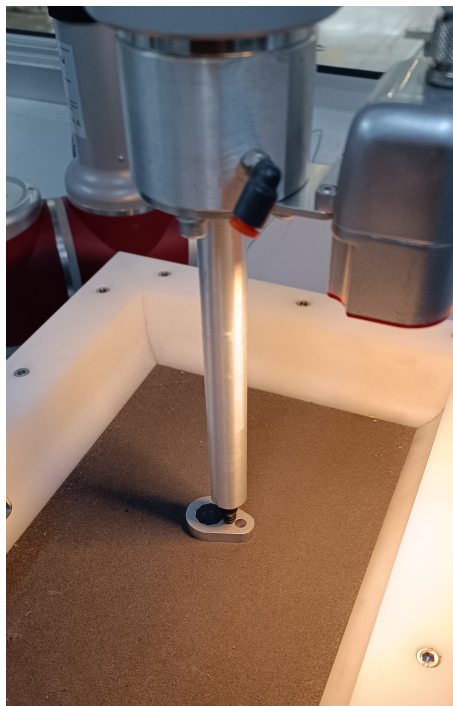


Figura 4.28: Ponto de calibração da peça

A seleção da opção "Enable Vision Absolute coordinate" na figura 4.30, permite que as coordenadas enviadas ao robô quando uma peça for detetada, correspondam a um incremento relativo à posição de calibração e, por conseguinte, às coordenadas de captura da peça. Isto significa que, se uma peça estiver nas coordenadas 1000 em X, 0 em Y e 200 em Z, e estiver deslocada 20 milímetros em X, a câmara indicará ao robô que este deve deslocar-se 20 milímetros em X relativamente à sua posição atual.

A introdução de valores absolutos incorretos resulta em posições erradas para a captura da peça. Estas coordenadas podem ser obtidas ao posicionar o robô na posição de calibração da peça, que corresponde às coordenadas da captura da imagem. No entanto, é necessário efetuar um deslocamento no eixo Z, até ao ponto em que a ventosa entra em contacto com a peça. Sem esta opção ativada, as coordenadas enviadas ao robô correspondem ao ponto de captura do pixel da câmara, e não ao valor correto de deslocamento para a captura. Em seguida, o robô deve ser elevado à posição de captura de foto.

Na figura 4.29 mostra a simulação da troca de comandos entre a câmara de visão e o software Hercules.

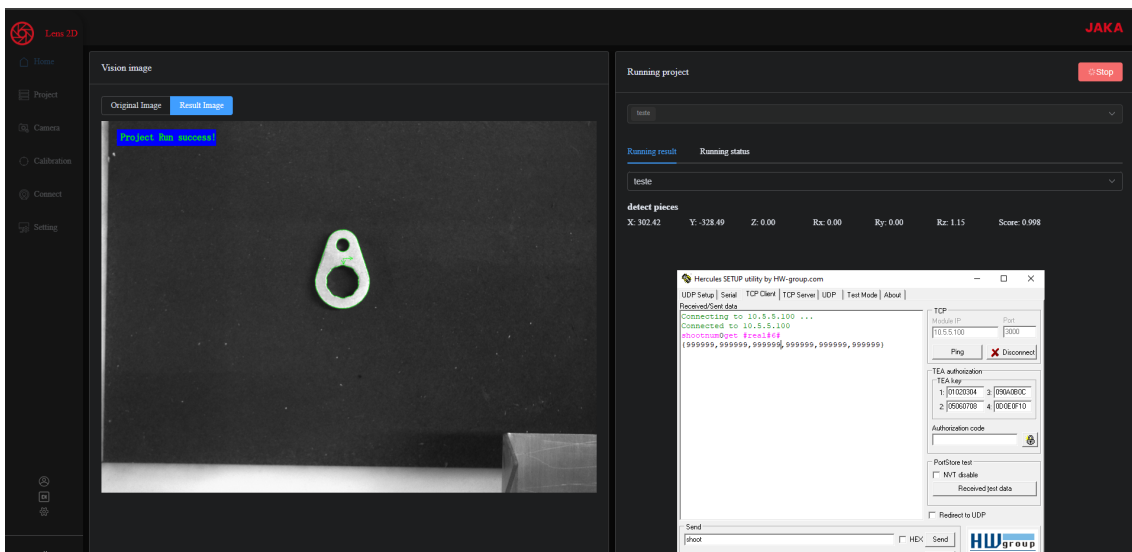


Figura 4.29: Detecção de uma peça

Para validar a correta configuração do sistema, foram simulados os comandos de envio via TCP/IP com recurso ao software Hercules. Esta simulação permitiu confirmar o bom funcionamento da comunicação e garantir que todos os parâmetros estavam corretamente configurados antes da implementação final. Durante este processo, os valores da posição absoluta do projeto foram ajustados de modo a garantir que o robô executa os movimentos com a precisão necessária para manusear as peças. Estes ajustes são essenciais para garantir a fiabilidade e a repetibilidade das operações num ambiente real.

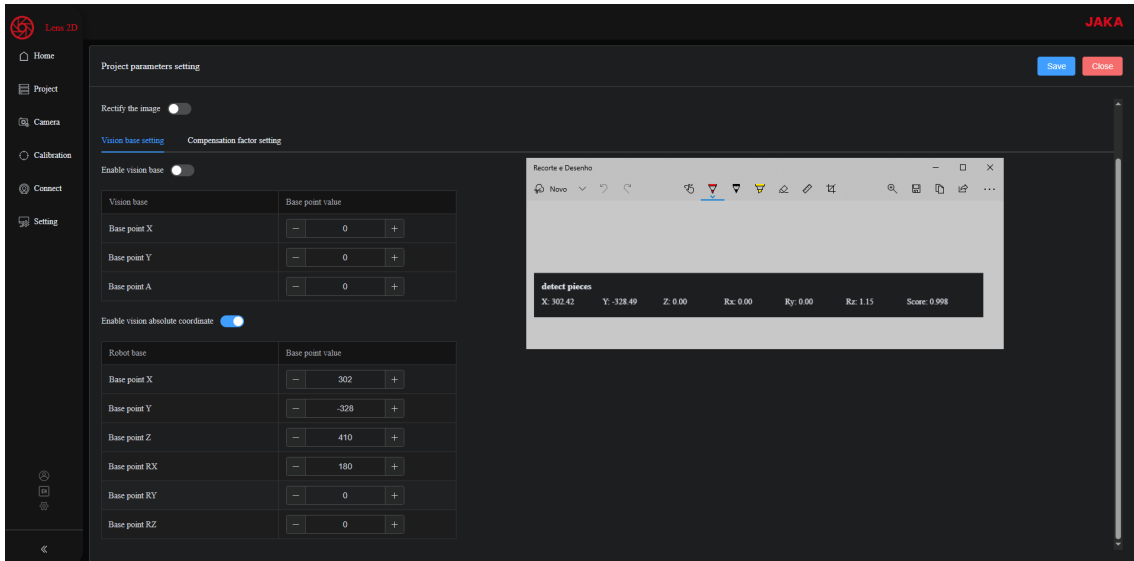


Figura 4.30: Coordenadas da peça de calibração

4.3.2 Resultados

A figura 4.31 ilustra a interligação com a aplicação Hercules, utilizada para enviar comandos de captura de imagem, obter o número de peças e determinar as respetivas coordenadas, com vista à realização de testes.

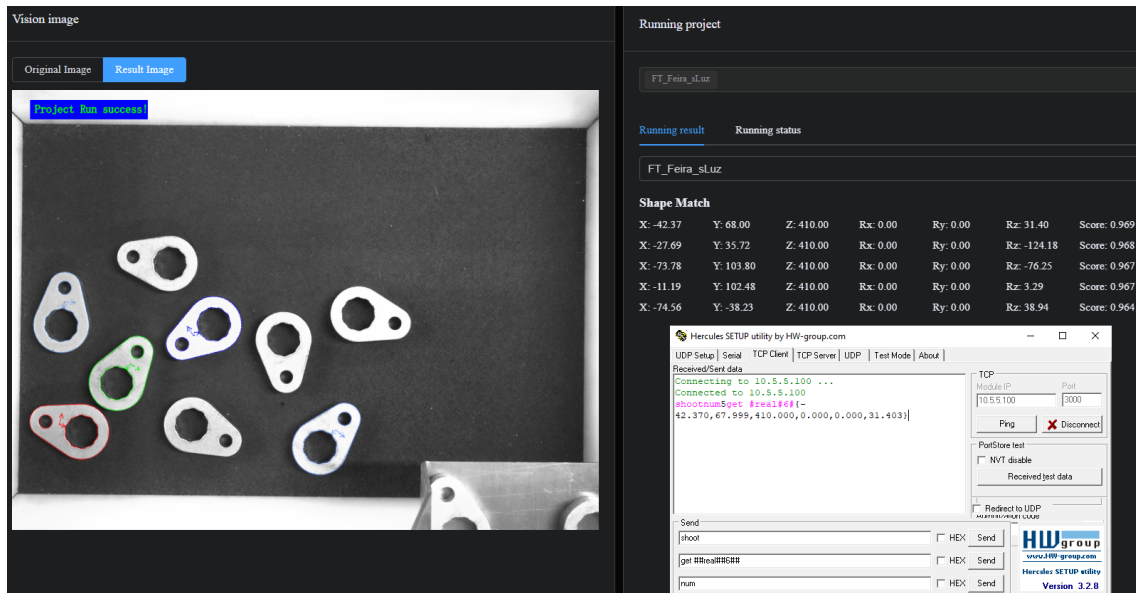


Figura 4.31: Resultado da deteção de peças

Como é possível observar na figura anterior, a deteção de peças não excede as cinco, mesmo que haja um número superior de peças presentes. Esta limitação resulta de um requisito do código do robô, que estabelece que o número de peças detetadas deve ser inferior a seis.

4.4 Conformadora de caixas usando um robô colaborativo

Esta secção aborda um projeto desenvolvido que recorre um robô colaborativo para realizar a montagem automática de caixas de cartão canelado. A referida solução técnica foi apresentada como demonstração na feira EMAF, que decorreu na EXPONOR em Matosinhos. Esta feira é o maior evento industrial em Portugal e abrange uma ampla gama de soluções industriais, incluindo automação, digitalização, robótica e inovação. A figura 4.32 mostra a máquina devidamente legendada.

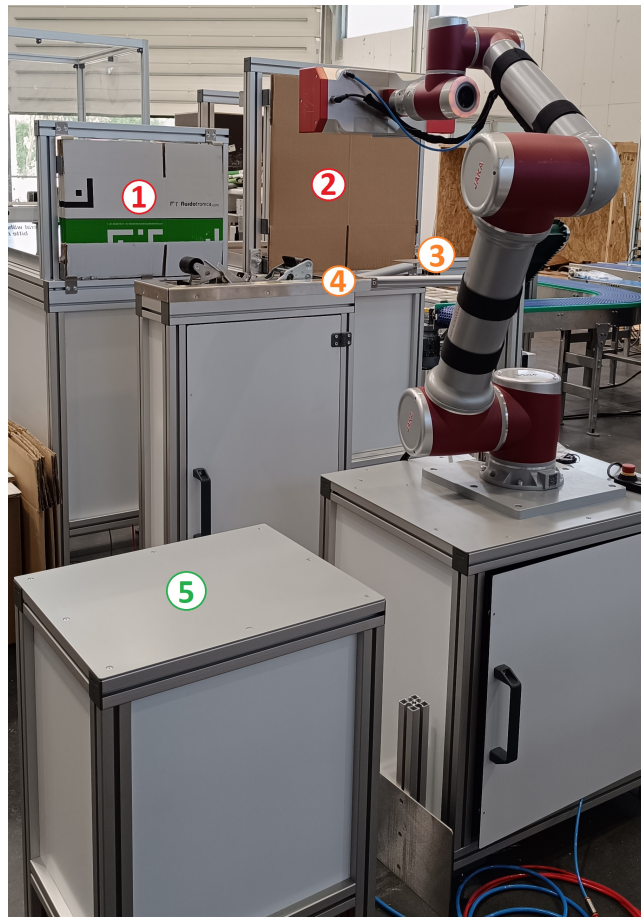


Figura 4.32: Estrutura da conformadora

A máquina anterior é constituída pelas seguintes zonas:

1. Armazém de caixas de cartão planas pequenas;
2. Armazém de caixas de cartão planas grandes;
3. Zona de dobragem da caixa de cartão;
4. Zona de aplicação de fita cola;
5. Zona de colocação de caixas conformadas.

Este robô tem dois modos de funcionamento. No primeiro, o robô efetua sequencialmente várias operações, nomeadamente a captura de uma caixa de cartão plana, a conformação da caixa, que compreende a execução de dobras nas abas mais pequenas para dentro, seguidas pelas abas grandes, e, por fim, a colocação de fita adesiva para selar a parte inferior da referida caixa.

O segundo modo de funcionamento é meramente demonstrativo, no qual o robô "simula" a conformação das caixas e executa todos os tipos de movimentos anteriormente descritos, porém sem atuar as garras de vácuo que são responsáveis por agarrar a caixa. O ciclo de funcionamento consiste na captura de um dos dois tipos de caixas (grandes ou pequenas), em que o critério de seleção do tipo de caixa é dado por um seletor (disponível na figura 4.33).

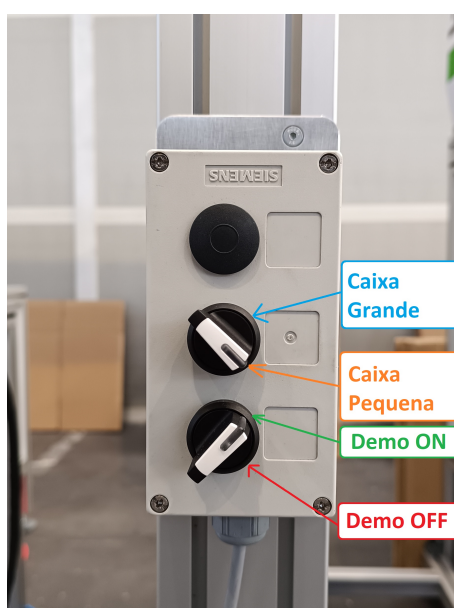


Figura 4.33: Botoneira de seleção de modos de funcionamento

No que diz respeito ao funcionamento do sistema de manuseamento de caixas, o robô desloca-se até ao armazém correspondente, de acordo com o tipo de caixa selecionado, escolha essa efetuada através de um botão seletor. A ativação do vácuo depende do modo de funcionamento escolhido, que é determinado pela ativação ou não do modo de demonstração.

Após a seleção, o robô procede à captura da caixa e, em seguida, desloca-se até à zona três, onde é realizada a preparação para a dobragem. De seguida, são executados todos os movimentos necessários para aplicar a fita-cola, na zona quatro, conforme ilustrado na figura anterior.

No final do ciclo, a caixa é libertada na zona cinco, onde será posteriormente recolhida pelo operador. Dependendo do tipo de caixa, o robô executa movimentos muito semelhantes. Como as caixas têm dimensões diferentes, os pontos de dobragem e de

aplicação da fita variam ligeiramente, embora sigam uma lógica semelhante. O Anexo H apresenta o fluxograma que ilustra o funcionamento geral do ciclo operacional do robô. Cada armazém de caixas planas está equipado com um cilindro pneumático que posiciona corretamente as caixas no local de captura, mediante o seu avanço. A ausência de caixas no armazém é detetada por sensores indutivos que monitorizam o movimento dos cilindros para a frente e para trás. Quando é enviado um sinal digital ao robô, este inicia uma rotina que identifica a falta de caixas e executa uma sequência lógica. Dependendo do estado dos sensores, são ativadas as eletroválvulas responsáveis pelo movimento dos cilindros. O Anexo I representa a lógica de deteção de um armazém vazio (sem caixas de cartão disponíveis). Neste programa não foram utilizados *user frames*, visto que todos os postos estão fixos à mesma estrutura. Apenas o TCP da garra e o respetivo centro de massa foram configurados. Adicionalmente, foram também configuradas as entradas e saídas digitais.

Esta configuração assegura o funcionamento eficiente do sistema robótico, permitindo a execução precisa das tarefas de manuseamento e conformação. A ausência de *user frames* simplifica a programação, ao passo que a correta definição do TCP e das entradas e saídas digitais garante a fiabilidade e a repetibilidade do processo.

5 CONCLUSÃO

Durante o estágio, participei no desenvolvimento e na simulação de diversas soluções robóticas destinadas à indústria automóvel. O sistema automatizado a implementar tinha como objetivo colocar clips, brackets, tabs e fita adesiva num componente de plástico do veículo.

A primeira fase do projeto consistiu na simulação das diferentes células robotizadas responsáveis por essas operações. Nesta fase de desenvolvimento, a simulação revelou-se uma ferramenta essencial, na medida em que permitiu identificar potenciais erros de conceção antes da fase de produção. Desta forma, foi possível reduzir custos, minimizar desperdícios e aumentar a segurança do equipamento e dos operadores.

Constatou-se também que a configuração correta da ferramenta do robô é determinante para otimizar o processo de marcação de pontos e tornar as operações mais ágeis e precisas.

Esta configuração permite automatizar e padronizar etapas que, de outro modo, seriam realizadas manualmente aumentando, por conseguinte, o risco de erros. Como resultado, obtém-se um fluxo de trabalho mais eficiente e consistente. A flexibilidade introduzida por esta abordagem facilita também a adaptação a diferentes cenários e requisitos operacionais, contribuindo para a melhoria contínua do desempenho do sistema.

Uma das melhorias mais relevantes foi a transição de um único ponto central de ferramenta TCP para múltiplos TCPs, localizados em cada um dos dedos da “estrela” da ferramenta. Esta alteração representou um avanço significativo na afinação dos pontos de aplicação dos clips, permitindo um controlo mais preciso e específico em cada dedo. Como resultado, a precisão do ajuste dos pontos de contacto aumentou consideravelmente, tornando o processo de calibração mais eficiente e garantindo um desempenho mais fiável na manipulação de objetos pelo robô.

No que diz respeito aos robôs colaborativos, participei em quatro projetos distintos. Antes de mais, é importante referir que este tipo de robótica tem estado cada vez mais presente em ambientes fabris que exigem a execução de tarefas repetitivas e fisicamente exigentes por parte dos operadores. A robótica colaborativa visa melhorar as condições de trabalho e aumentar a longevidade funcional dos operadores, promovendo uma maior flexibilidade na interação entre humanos e robôs. Estes sistemas permitem que os robôs operem lado a lado com os operadores, sem ser necessário recorrer a barreiras físicas, mantendo a segurança e a eficiência do processo produtivo.

O primeiro projeto em que estive envolvido, que empregava robôs colaborativos, consistia no desenvolvimento de um sistema de soldadura a laser para unir dois objetos cilíndricos.

Um dos principais desafios prendia-se com o elevado número de pontos de soldadura necessários, que tornavam o processo moroso e suscetível a erros, caso esses pontos tivessem de ser marcados manualmente pelo operador. Para contornar esta limitação, foram aplicados cálculos automáticos que agilizaram e tornaram mais dinâmica a definição dos pontos de soldadura.

Sabe-se que, quanto maior for o número de pontos definidos, maior será a qualidade da soldadura, uma vez que o erro de execução tende a diminuir e as semirretas a efetuar são mais curtas. Assim sendo, foi necessário dotar o robô da capacidade de realizar esses cálculos de forma autónoma e precisa.

A simplificação do procedimento de marcação de pontos requer, contudo, algoritmos mais robustos e estruturas de código mais complexas do ponto de vista computacional. Esta abordagem permite otimizar significativamente o tempo despendido na marcação dos pontos.

O segundo projeto em que estive envolvido compreendia o desenvolvimento de um sistema alimentador de CNC usando um robô colaborativo. Este projeto revelou-se particularmente desafiante, dado que o espaço disponível era bastante limitado e a arquitetura do robô era pouco convencional, o que exigiu a definição de trajetórias complexas para evitar colisões com a máquina CNC.

O objetivo deste projeto era efetuar de uma forma automática a manipulação de peças num processo industrial, que incluía a colocação de peças em bruto em gabaritos. Após a sua maquinação, por uma máquina CNC, o robô colaborativo colocava as peças resultantes em blisters. Ambas as operações eram automáticas, requerendo apenas a intervenção do operador no início do ciclo. Posteriormente, participei num terceiro projeto que consistia no desenvolvimento de uma solução de demonstração para a deteção de objetos. Esta solução visava detetar objetos provenientes de um sistema de alimentação, recorrendo a um braço robotizado equipado com uma câmara.

Ao longo deste projeto, verificaram-se algumas dificuldades resultantes de fatores externos, como a presença de iluminação ambiente e a acumulação de detritos ou resíduos na base de observação, que interferiram significativamente no processo de deteção das peças, provocando leituras incorretas ou dificultando a sua identificação.

A presença de iluminação externa constituiu um desafio considerável para a deteção precisa das peças, tendo afetado a qualidade das imagens capturadas e, conseqüentemente, a eficiência do algoritmo. Verificou-se também que a seleção dos parâmetros de exposição e de ganho da configuração matinal não era adequada para o período da tarde, o que comprometia a nitidez e o contraste das imagens adquiridas e dificultava

a correta identificação dos objetos.

Em suma, esta solução mostrou ser viável em ambientes com iluminação controlada. No entanto, em caso de mudança de local, poderá ser necessário ajustar os parâmetros de ganho e de contraste, para adaptar a solução às condições de iluminação globais do novo espaço.

Por fim, participei num quarto projeto cujo objetivo era a conformação de caixas de dois tamanhos distintos. Um dos principais desafios deste processo foi a precisão exigida na marcação dos pontos, de forma a garantir a conformação correta das caixas. Inicialmente, a marcação dos *user frames* foi desconsiderada, uma vez que toda a estrutura estava fixa ao robô, e foi garantido que esta não seria desmontada. No entanto, mais tarde, verificou-se que seria inevitável desmontar a estrutura para a deslocar, o que implicou a necessidade de reajustar manualmente todos os pontos previamente definidos.

Como solução, foi sugerida a utilização de *user frames*. Contudo, esta abordagem mostrou-se inviável, dado que o método de definição exigia a marcação de três pontos de referência, e o robô não dispunha de espaço suficiente para tal operação. Por conseguinte, o projeto ficou condicionado por esta limitação, decorrente da falta de um planeamento adequado na fase inicial.

Outra dificuldade encontrada foi a definição da posição ideal do robô. Inicialmente, o robô não conseguia executar todas as trajetórias devido à falta de espaço. Esta situação foi resolvida com sucesso através de uma proposta da minha autoria, que consistiu na alteração do *layout* da estrutura, permitindo assim a execução completa dos movimentos necessários.

REFERÊNCIAS BIBLIOGRÁFICAS

BIBLIOGRAFIA

- [1] Eletrogate. (2020, Jul.) O que é robótica: Conceito, história e evolução. <https://blog.eletrogate.com/o-que-e-robotica-conceito-historia-e-evolucao/> (acedido em 04/12/2025).
- [2] O futuro do trabalho: a robótica. <https://oshwiki.osha.europa.eu/pt/themes/review-future-work-robotics> (acedido em 04/12/2025).
- [3] Apresentação - FLUIDOTRONICA. <https://www.fluidotronica.com/pt/empresa/apresentacao/> (acedido em 04/12/2025).
- [4] U. R. Brasil. Tipos de robôs industriais e suas aplicações. <https://www.universal-robots.com/br/blog/tipos-de-rob%C3%B4s-industriais-e-suas-aplicac%C3%B5es/> (acedido em 04/12/2025).
- [5] (2025, Jan.) Robôs articulados: o que são e como funcionam? <https://pahcautomacao.com.br/robos-articulados-o-que-sao-e-como-funcionam/> (acedido em 04/12/2025).
- [6] (2018, Jun.) Que robô industrial escolher? <https://guide.directindustry.com/pt/que-robo-industrial-escolher/> (acedido em 04/12/2025).
- [7] IRB 1300. <https://new.abb.com/products/robotics/robots/articulated-robots/irb-1300> (acedido em 04/12/2025).
- [8] (2024, Jul.) Como funciona um robô cartesiano. <https://www.pensenova.com/robo-cartesiano/> (acedido em 04/12/2025).
- [9] CO1-CO2 X-Series. <https://www.campetella.com/pt/rob%C3%B4s-cartesianos/co1-co2-x-series/> (acedido em 04/12/2025).
- [10] C. B. Silveira. (2019, Aug.) Os 6 principais tipos de robôs industriais. <https://www.citisystems.com.br/tipos-de-robos/> (acedido em 04/12/2025).
- [11] (2020, Dec.) Os 7 principais robôs industriais que você deve conhecer. <https://www.evsint.com/pt/top-7-industrial-robots/> (acedido em 04/12/2025).
- [12] U. R. Brasil. O que é um robô delta e como ele funciona? <https://www.universal-robots.com/br/blog/o-que-%C3%A9-um-rob%C3%B4-delta-e-como-ele-funciona/> (acedido em 04/12/2025).
- [13] ABB lança robô delta de cinco eixos mais rápido para coleta, embalagem e reorientação de produtos leves. <https://new.abb.com/news/pt-br/detail/93173/prsrll>

-ABB-launches-five-axis-Delta-robot-fastest-for-lightweight-product-picking-packing-and-re-orientation (acedido em 04/12/2025).

- [14] jaka. https://www.jaka.com/en/ProductList/JAKA_Zu_Series (acedido em 04/12/2025).
- [15] D. Gonçalves, C. Augusto, C. Bissochi Junior, J. Morais, A. Morais, I. Tavares, F. Adriano, and F. Adriano da Silva Gonçalves. (2012, 07) Reformulação do hardware do braço robótico ma2000 com emprego da arquitetura arm para controle.
- [16] https://www.researchgate.net/figure/Junta-rotacional-unindo-dois-elos-consecutivos_fig1_305318224 (acedido em 04/12/2025).
- [17] (2022, Dec.) Tipos de juntas de robô: um guia detalhado - robô EVS. <https://www.evsint.com/pt/types-of-robot-joints/> (acedido em 04/12/2025).
- [18] E. Por Rvertulo. (2016, Mar.) Robôs de classe. <https://labdeeletronica.com.br/robos-de-classe/> (acedido em 04/12/2025).
- [19] J. (jaf). ROBÔS MANIPULADORES INDUSTRIAIS (conceitos teóricos). <https://automacaoerobotica.blogspot.com/2012/07/11-robos-manipuladores-industriais.html> (acedido em 04/12/2025).
- [20] ——. Classificação dos robôs manipuladores de acordo com a sua estrutura cinemática. <https://automacaoerobotica.blogspot.com/2012/07/robo-manipulador-cartesiano-um-robo.html> (acedido em 04/12/2025).
- [21] (2022, Jan.) Quais são as partes dum robô? <https://www.esneca.lat/pt/blog/principais-partes-dum-robo/> (acedido em 18/11/2025).
- [22] (2022, Jan.) Quais são as partes dum robô? <https://www.esneca.lat/pt/blog/principais-partes-dum-robo/> (acedido em 04/12/2025).
- [23] IRC5 controller. <https://new.abb.com/products/pt/3HAC020536-014/irc5-controller> (acedido em 04/12/2025).
- [24] (2023, Feb.) O que é um manipulador robótico? um guia - robô EVS. <https://www.evsint.com/pt/what-is-a-robotic-manipulator-a-guide/> (acedido em 04/12/2025).
- [25] <https://knowledge.motoman.com/hc/en-us/articles/21306478868375-Allowable-Load-and-Payload> (acedido em 04/12/2025).
- [26] S. Automação. (2024, Jul.) Especificação e programação de robôs para sistemas de automação. <https://sameautomacao.com.br/especificacao-e-programacao-d-e-robos-para-sistemas-de-automacao/> (acedido em 04/12/2025).
- [27] (2013, Feb.) UserFrame and ObjectFrame. <https://www.robot-forum.com/robotforum/thread/11189-userframe-and-objectframe/> (acedido em 04/12/2025).

- [28] https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html (acedido em 30/09/2025).

ANEXOS

Anexo A - História da Fluidotronics

2004

- Fundada
- Parceria com a MiniTec, HepcoMotion, Fipa e Stoger Automation
- Presença na EMAF

2005

- Início da parceria com a empresa Bahr Modultechnik [Janeiro]
- Mudança para novas instalações, em Vilar, Oliveira de Azeméis
- Fluidotronica anuncia início da parceria com a E2 Systems [Abril]

2006

- Presença na EMAF

2007

- Parceria com Weiss

2008

- Presença na EMAF e MoldPlas

2010

- Atribuição do estatuto PME Líder
- Início da parceria com a empresa Omnitrack
- Fluidotronica na EMAF

2011

- Presença na MOLDPLÁS / TECNA 2011
- Renovação do estatuto PME Líder

2012

- Renovação do estatuto PME Líder
- Participação na Feira "Promoção PME"
- Missão cumprida e balanço positivo na EMAF 2012

2013

- Fluidotronica distinguida pela primeira vez como PME Excelência
- Participação na Feira "Open House"
- Atribuição do título "Top Exporta"

2014

- Fluidotronica completa 10 anos de história
- Fluidotronica distinguida como PME Excelência 2013
- Fluidotronica marcou novamente presença na MOLDPLAS/TECNA
- Automação num só lugar: este foi o mote da Fluidotronica na EMAF 2014

2015

- Participação na Feira "Salon de la sous-traitance automobile", em Marrocos
- Renovação do estatuto de "Top Exporta"
- Renovação do estatuto PME Líder
- Fluidotronica recebe o certificado do Sistema de Gestão de Qualidade
- Participação na Noite Europeia dos Investigadores

2016

- Fluidotronica comemora 12 anos de história

2017

- Mudança para novas instalações, na Zona Industrial de Ouriçosa, em UI - Oliveira de Azeméis
- Atribuição do título "Top Exporta"
- Fluidotronica mostrou as novidades na MOLDPLAS 2017
- A Fluidotronica tem para si um novo serviço: um centro de ensaios de moldes

2018

- Fluidotronica recebe a certificação ISO 14001:2015
- SMARTPICKING - A nova gama de produtos da Fluidotronica
- Impressão 3D - o novo serviço disponibilizado pela Fluidotronica
- Renovação de Estatuto PME Líder
- Fluidotronica voltou a marcar presença na EMAF

2019

- Fluidotronica celebra 15 anos
- Fluidotronica obteve a certificação no âmbito do Sistema de Gestão da Investigação Desenvolvimento e Inovação (IDI)
- Fluidotronica marcou presença em mais uma edição da MOLDPLAS

2020

- Elaboração plano estratégico 5 anos
- Chat bot na página web

2021

- Extinção UN Mould Test Center
- Parceria com Lions Club entre outubro e dezembro
- A Fluidotronica é membro da MOBINOVA, Cluster Automóvel de Portugal

2022

- Parceria com a JAKA
- Moldplás 9-12 novembro Batalha
- Visita Câmara comércio luso-mexicana
- Criação das newsletters internas
- Team Building Porto

2023

- 3M Technical days 19/20 abril
- EMAF 31 Maio-3 Junho
- Convívio Interno - Arraial Fluidotronica
- Empack 19/20 Abril
- Open Day

2024

- 20 anos FT - 22 janeiro

Anexo B - Programa do robô número 1 da célula 1309

```
1 MODULE Module1
2   !Posicao de home
3   CONST robtarget POS_Home:=[[267.278795794,0,1740.346900978],0,1,-0.000000001,0],
4   [-1,0,2,4],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
5   !Posicao de pick ferramenta 1
6   CONST robtarget POS_PickTool:=[[1254.711970625,613.355536973,1399.675936182],
7   [-0.000000055,0.923879577,0.382683325,-0.000000087],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
8   !Posicoes de pick pecas do tapete
9   CONST robtarget POS_PickTapeteEntrada_Skoda:=[[1583.410981065,-230.27438819,1021.157967098],
10  [-0.000000074,0.92387959,0.382683294,-0.000000113],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
11  CONST robtarget POS_PickTapeteEntrada_VW:=[[1583.410977197,-230.274386846,1021.158023077],
12  [-0.000000094,0.92387959,0.382683294,-0.000000091],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
13  CONST robtarget POS_PickTapeteIntermedia:=[[573.441344349,-573.441361346,1326.072098094],
14  [-0.000000013,0.923879529,-0.382683441,-0.000000002],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
15  !Posicao de pick de pecas da mesa
16  CONST robtarget POS_PickMesa_VW:=[[ -785.397219337,-1326.289960749,1021.158002951],
17  [0.000000064,-0.382683289,0.923879592,-0.000000005],[-2,2,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
18  CONST robtarget POS_PickMesa_Skoda:=[[ -785.397213456,-1326.289967159,1021.158005037],
19  [0.000000035,-0.382683306,0.923879585,-0.000000013],[-2,2,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
20  !Posicoes de place pecas
21  CONST robtarget POS_PlacePecaIntermedio:=[[ -91.414866694,251.160020517,1865.346994837],
22  [-0.000000108,0.573576323,0.819152124,-0.000000095],[1,0,2,4],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
23  CONST robtarget POS_Place_Skoda:=[[ -644.669188484,875.518243944,1519.675977662],
24  [0.00000003,0.382683473,0.923879516,-0.000000064],[1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
25  CONST robtarget POS_Place_VW:=[[ -644.669197488,875.518270992,1519.67599873],
26  [0.000000048,0.382683496,0.923879506,0.000000012],[1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
27  !Posicoes Plasma VW
28  CONST robtarget POS_Plasma1_1_VW:=[[808.555663648,-922.979304646,1221.15810203],
29  [-0.000000039,0.991444843,-0.130526328,-0.000000193],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
30  CONST robtarget POS_Plasma1_2_VW:=[[902.990821077,-729.725745029,1220.195214139],
31  [0.000000198,0.991444843,-0.130526328,0.000000051],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```

32  CONST robtarget POS_Plasma1_3_VW:=[[988.368948094,-562.509683975,1221.158203657],
33  [0.000000063,0.98480773,-0.17364831,-0.000000187],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
34  CONST robtarget POS_Plasma2_1_VW:=[[776.237951052,-428.574060096,1221.158098876],
35  [0.000000048,0.923879496,-0.38268352,-0.000000158],[0,0,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
36  CONST robtarget POS_Plasma2_2_VW:=[[593.293769072,-573.404186036,1221.157978459],
37  [-0.000000096,0.923879502,-0.382683506,-0.000000159],[-1,1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
38  CONST robtarget POS_Plasma2_3_VW:=[[504.905343319,-732.503200152,1221.15809561],
39  [-0.000000101,0.923879497,-0.382683517,-0.000000181],[-1,1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
40  CONST robtarget POS_Plasma3_1_VW:=[[399.284561656,-842.849510883,1221.158069702],
41  [-0.000000054,0.843391397,-0.537299685,-0.000000116],[-1,3,-2,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
42  CONST robtarget POS_Plasma3_2_VW:=[[253.400529182,-743.351945102,1221.158068139],
43  [0.000000042,0.793353294,-0.608761489,0.000000009],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
44  CONST robtarget POS_Plasma3_3_VW:=[[104.654127887,-598.489540074,1242.474080281],
45  [-0.000000003,0.737277286,-0.675590263,-0.000000005],[-2,3,-2,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
46  CONST robtarget POS_Plasma4_1_VW:=[[296.165175111,-1422.83993317,1242.474223418],
47  [-0.000000149,0.793353287,-0.608761498,-0.000000076],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
48  CONST robtarget POS_Plasma4_2_VW:=[[436.696279988,-1312.350923914,1242.47418944],
49  [0.000000033,0.793353271,-0.608761519,-0.000000156],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
50  CONST robtarget POS_Plasma4_3_VW:=[[558.807817166,-1227.120965063,1242.474087286],
51  [-0.000000092,0.866025357,-0.500000008,-0.000000165],[-1,3,-2,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
52  !Posicao Intermedia Plasma
53  CONST robtarget POS_PlasmaIntermedio:=[[493.395216202,-782.127511481,1411.158011045],
54  [-0.000000039,0.843391408,-0.537299667,-0.00000018],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
55  !Posicoes Plasma Skoda
56  CONST robtarget POS_Plasma1_1_Skoda:=[[702.303902794,-894.508972405,1231.158022186],
57  [0.000000028,0.991444849,-0.130526288,-0.000000231],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
58  CONST robtarget POS_Plasma1_2_Skoda:=[[786.386188888,-739.89267432,1220.195076527],
59  [0.000000242,0.991444848,-0.130526291,0.000000047],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
60  CONST robtarget POS_Plasma1_3_Skoda:=[[841.403701185,-615.436366728,1221.158048324],
61  [0.000000059,0.984807729,-0.173648316,-0.000000187],[-1,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
62  CONST robtarget POS_Plasma2_1_Skoda:=[[800.986690069,-559.388802484,1221.158025182],

```

```

63 [-0.000000047,0.9238795,-0.38268351,-0.000000155],[-1,1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
64 CONST robtarget POS_Plasma2_2_Skoda:=[[678.146495179,-658.256933739,1221.158015689],
65 [0.000000734,0.923879515,-0.382683474,0.000000085],[-1,1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
66 CONST robtarget POS_Plasma2_3_Skoda:=[[582.686951775,-810.284929049,1221.158026518],
67 [-0.00000009,0.923879485,-0.382683548,-0.000000168],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
68 CONST robtarget POS_Plasma3_1_Skoda:=[[437.320173988,-924.417176707,1221.158038302],
69 [0.000000022,0.843391395,-0.537299688,-0.000000175],[-1,3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
70 CONST robtarget POS_Plasma3_2_Skoda:=[[364.667881962,-868.829040338,1221.158054804],
71 [-0.000000062,0.793353333,-0.608761439,-0.000000027],[-1,1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
72 CONST robtarget POS_Plasma3_3_Skoda:=[[244.243233048,-817.155448938,1242.473967481],
73 [-0.000000118,0.73727729,-0.675590259,-0.000000051],[-1,3,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
74 CONST robtarget POS_Plasma4_1_Skoda:=[[315.297736123,-1262.421959818,1242.473978114],
75 [-0.000000177,0.793353286,-0.6087615,-0.000000178],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
76 CONST robtarget POS_Plasma4_2_Skoda:=[[444.275080594,-1186.086998631,1242.473979504],
77 [-0.000000051,0.793353328,-0.608761445,-0.000000156],[-1,1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
78 CONST robtarget POS_Plasma4_3_Skoda:=[[560.76936706,-1110.517928944,1242.473980732],
79 [0.000000169,0.866025346,-0.5000001,-0.000000082],[-1,2,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
80 !*****
81 !
82 ! Module: Module1
83 !
84 ! Description:
85 ! <Insert description here>
86 !
87 ! Author: est.automacao
88 !
89 ! Version: 1.0
90 !
91 !*****
92 !Variaveis de velocidade
93 VAR speeddata velAproximacao;

```

```
94  VAR speeddata velIntermedia;  
95  VAR speeddata velPlasma;  
96  
97  !-----  
98  !Variaveis de posicao  
99  !Posicao do plasma  
100 !Peca 1  
101  VAR robtarget POS_Plasma1_1;  
102  VAR robtarget POS_Plasma1_2;  
103  VAR robtarget POS_Plasma1_3;  
104  !Peca 2  
105  VAR robtarget POS_Plasma2_1;  
106  VAR robtarget POS_Plasma2_2;  
107  VAR robtarget POS_Plasma2_3;  
108  !Peca 3  
109  VAR robtarget POS_Plasma3_1;  
110  VAR robtarget POS_Plasma3_2;  
111  VAR robtarget POS_Plasma3_3;  
112  !Peca 4  
113  VAR robtarget POS_Plasma4_1;  
114  VAR robtarget POS_Plasma4_2;  
115  VAR robtarget POS_Plasma4_3;  
116  
117  !Posicoes pick tapete entrada  
118  VAR robtarget POS_PickTapeteEntrada;  
119  
120  !Posicoes place na base  
121  VAR robtarget POS_Place;  
122  VAR robtarget POS_AproxPlace;  
123  
124  !Posicoes de pick da pesa
```

```
125  VAR robtarget POS_PickMesa;
126
127  !*****
128  !
129  ! Procedure main
130  !
131  !   This is the entry point of your program
132  !
133  !*****
134  PROC main()
135      inicializacao;
136      carregamentoPosicoes 2;
137      !Path_Home;
138      !Path_PickTool;
139      !Path_Home;
140      !Path_PickPecaTapete;
141      !Path_Plasma;
142      Path_Home;
143      Path_PlacePeca;
144      Path_Home;
145  ENDPROC
146
147  PROC Path_PickTool()
148      GOTO inicio;
149      !Posicoes de calibracao
150      MoveJ POS_PickTool ,v50 ,fine , Tool_1\WObj:=wobj0;
151      inicio:
152      MoveJ RelTool(POS_PickTool , -50,0,0) ,velIntermedia ,fine , Tool_1\WObj:=wobj0;
153      MoveL POS_PickTool ,velAproximacao ,fine , Tool_1\WObj:=wobj0;
154      WaitTime 1;
155      MoveL RelTool(POS_PickTool,0,0, -50) ,velAproximacao ,fine , Tool_1\WObj:=wobj0;
```

```
156  ENDPROC
157
158  PROC Path_Home()
159      MoveJ POS_Home, v500, fine, Tool_1\WObj:=wobj0;
160  ENDPROC
161
162  PROC Path_PickPecaTapete()
163      GOTO inicio;
164      !Posicao de calibracao
165      MoveJ POS_PickTapeteEntrada_Skoda, v50, fine, Tool_1\WObj:=wobj0;
166      MoveJ POS_PickTapeteEntrada_VW, v100, fine, Tool_1\WObj:=wobj0;
167      MoveJ POS_PickTapeteIntermedia, v50, fine, Tool_1\WObj:=wobj0;
168      inicio:
169      MoveJ POS_PickTapeteIntermedia, velIntermedia, fine, Tool_1\WObj:=wobj0;
170      MoveL RelTool(POS_PickTapeteEntrada, 0, 0, -100), velIntermedia, fine, Tool_1\WObj:=wobj0;
171      MoveL POS_PickTapeteEntrada_Skoda, velAproximacao, fine, Tool_1\WObj:=wobj0;
172      WaitTime 1;
173      MoveL RelTool(POS_PickTapeteEntrada, 0, 0, -100), velIntermedia, fine, Tool_1\WObj:=wobj0;
174      MoveL POS_PickTapeteIntermedia, velIntermedia, fine, Tool_1\WObj:=wobj0;
175  ENDPROC
176
177  PROC Path_PlacePeca()
178      GOTO inicio;
179      !Posicao de calibracao
180      MoveJ POS_Place_Skoda, v50, fine, Tool_1\WObj:=wobj0;
181      MoveJ POS_Place_VW, v50, fine, Tool_1\WObj:=wobj0;
182      MoveJ POS_PlacePecaIntermedio, v100, fine, Tool_1\WObj:=wobj0;
183      inicio:
184      MoveJ POS_PlacePecaIntermedio, velIntermedia, fine, Tool_1\WObj:=wobj0;
185      MoveJ POS_AproxPlace, velIntermedia, fine, Tool_1\WObj:=wobj0;
186      MoveL POS_Place, velAproximacao, fine, Tool_1\WObj:=wobj0;
```

```

187     WaitTime 1;
188     MoveL POS_AproxPlace , velAproximacao , fine , Tool_1\WObj:=wobj0 ;
189     MoveJ POS_PlacePecaIntermedio , velIntermedia , fine , Tool_1\WObj:=wobj0 ;
190 ENDPROC
191
192 PROC Path_PickMesa ()
193     GOTO inicio ;
194     !Posicoes de calibracao
195     MoveJ POS_PickMesa_Skoda , v50 , fine , Tool_1\WObj:=wobj0 ;
196     MoveJ POS_PickMesa_VW , v100 , fine , Tool_1\WObj:=wobj0 ;
197     inicio:
198     MoveJ RelTool(POS_PickMesa,0,0,-100) , velIntermedia , fine , Tool_1\WObj:=wobj0 ;
199     MoveL POS_PickMesa , velAproximacao , fine , Tool_1\WObj:=wobj0 ;
200     WaitTime 1;
201     MoveL RelTool(POS_PickMesa,0,0,-100) , velIntermedia , fine , Tool_1\WObj:=wobj0 ;
202 ENDPROC
203
204 PROC Path_Plasma ()
205     GOTO inicio ;
206     !Posicoes de calibracao
207     !Skoda
208     MoveJ POS_Plasma1_1_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
209     MoveJ POS_Plasma1_2_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
210     MoveJ POS_Plasma1_3_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
211     MoveJ POS_Plasma2_1_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
212     MoveJ POS_Plasma2_2_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
213     MoveJ POS_Plasma2_3_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
214     MoveJ POS_Plasma3_1_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
215     MoveJ POS_Plasma3_2_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
216     MoveJ POS_Plasma3_3_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;
217     MoveJ POS_Plasma4_1_Skoda , v100 , fine , Tool_1\WObj:=wobj0 ;

```

218 MoveJ POS_Plasma4_2_Skoda ,v100 , fine , Tool_1\WObj:=wobj0 ;
219 MoveJ POS_Plasma4_3_Skoda ,v100 , fine , Tool_1\WObj:=wobj0 ;
220 !VW
221 MoveJ POS_Plasma1_1_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
222 MoveJ POS_Plasma1_2_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
223 MoveJ POS_Plasma1_3_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
224 MoveJ POS_Plasma2_1_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
225 MoveJ POS_Plasma2_2_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
226 MoveJ POS_Plasma2_3_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
227 MoveJ POS_Plasma3_1_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
228 MoveJ POS_Plasma3_2_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
229 MoveJ POS_Plasma3_3_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
230 MoveJ POS_Plasma4_1_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
231 MoveJ POS_Plasma4_2_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
232 MoveJ POS_Plasma4_3_VW ,v100 , fine , Tool_1\WObj:=wobj0 ;
233 !Posicao intermedia
234 MoveJ POS_PlasmaIntermedio ,v100 , fine , Tool_1\WObj:=wobj0 ;
235 inicio:
236 !Posicao intermedia
237 MoveJ POS_PlasmaIntermedio ,v100 , fine , Tool_1\WObj:=wobj0 ;
238 !Peca 1
239 MoveJ RelTool(POS_Plasma1_1,0,0,-50) ,velIntermedia , fine , Tool_1\WObj:=wobj0 ;
240 MoveL POS_Plasma1_1 ,velAproximacao , fine , Tool_1\WObj:=wobj0 ;
241 MoveC POS_Plasma1_2 ,POS_Plasma1_3 ,velPlasma , fine , Tool_1\WObj:=wobj0 ;
242 MoveL RelTool(POS_Plasma1_3,0,0,-50) ,velAproximacao , fine , Tool_1\WObj:=wobj0 ;
243 !Peca 2
244 MoveJ RelTool(POS_Plasma2_1,0,0,-50) ,velIntermedia , fine , Tool_1\WObj:=wobj0 ;
245 MoveL POS_Plasma2_1 ,velAproximacao , fine , Tool_1\WObj:=wobj0 ;
246 MoveC POS_Plasma2_2 ,POS_Plasma2_3 ,velPlasma , fine , Tool_1\WObj:=wobj0 ;
247 MoveL RelTool(POS_Plasma2_3,0,0,-50) ,velAproximacao , fine , Tool_1\WObj:=wobj0 ;
248 !Peca 3

```
249     MoveJ RelTool(POS_Plasma3_1,0,0,-50),velIntermedia,fine,Tool_1\WObj:=wobj0;
250     MoveL POS_Plasma3_1,velAproximacao,fine,Tool_1\WObj:=wobj0;
251     MoveC POS_Plasma3_2,POS_Plasma3_3,velPlasma,fine,Tool_1\WObj:=wobj0;
252     MoveL RelTool(POS_Plasma3_3,0,0,-50),velAproximacao,fine,Tool_1\WObj:=wobj0;
253     !Peca 4
254     MoveJ RelTool(POS_Plasma4_1,0,0,-50),velIntermedia,fine,Tool_1\WObj:=wobj0;
255     MoveL POS_Plasma4_1,velAproximacao,fine,Tool_1\WObj:=wobj0;
256     MoveC POS_Plasma4_2,POS_Plasma4_3,velPlasma,fine,Tool_1\WObj:=wobj0;
257     MoveL RelTool(POS_Plasma4_3,0,0,-50),velAproximacao,fine,Tool_1\WObj:=wobj0;
258
259     ENDPROC
260
261     !-----
262     PROC inicializacao()
263         velAproximacao:=v20;
264         velIntermedia:=v100;
265         velPlasma:=v60;
266     ENDPROC
267
268     PROC carregamentoPosicoes(num versao)
269         !versao=1 - Skoda
270         !versao=2 - VW
271         IF versao=1 THEN
272             !Posicao do plasma
273             !Peca 1
274             POS_Plasma1_1:=POS_Plasma1_1_Skoda;
275             POS_Plasma1_2:=POS_Plasma1_2_Skoda;
276             POS_Plasma1_3:=POS_Plasma1_3_Skoda;
277             !Peca 2
278             POS_Plasma2_1:=POS_Plasma2_1_Skoda;
279             POS_Plasma2_2:=POS_Plasma2_2_Skoda;
```

```
280     POS_Plasma2_3:=POS_Plasma2_3_Skoda ;
281     !Peca 3
282     POS_Plasma3_1:=POS_Plasma3_1_Skoda ;
283     POS_Plasma3_2:=POS_Plasma3_2_Skoda ;
284     POS_Plasma3_3:=POS_Plasma3_3_Skoda ;
285     !Peca 4
286     POS_Plasma4_1:=POS_Plasma4_1_Skoda ;
287     POS_Plasma4_2:=POS_Plasma4_2_Skoda ;
288     POS_Plasma4_3:=POS_Plasma4_3_Skoda ;
289     !Posicao de pick tapete de entrada
290     POS_PickTapeteEntrada:=POS_PickTapeteEntrada_Skoda ;
291     !Posicao de place na base
292     POS_Place:=POS_Place_Skoda ;
293     POS_AproxPlace:=RelTool ( POS_Place_Skoda ,0,0 , -150 ) ;
294     !Posicao de pick da mesa
295     POS_PickMesa:=POS_PickMesa_Skoda ;
296     ENDIF
297     IF versao=2 THEN
298         !Posicao do plasma
299         !Peca 1
300         POS_Plasma1_1:=POS_Plasma1_1_VW ;
301         POS_Plasma1_2:=POS_Plasma1_2_VW ;
302         POS_Plasma1_3:=POS_Plasma1_3_VW ;
303         !Peca 2
304         POS_Plasma2_1:=POS_Plasma2_1_VW ;
305         POS_Plasma2_2:=POS_Plasma2_2_VW ;
306         POS_Plasma2_3:=POS_Plasma2_3_VW ;
307         !Peca 3
308         POS_Plasma3_1:=POS_Plasma3_1_VW ;
309         POS_Plasma3_2:=POS_Plasma3_2_VW ;
310         POS_Plasma3_3:=POS_Plasma3_3_VW ;
```

```
311         !Peca 4
312         POS_Plasma4_1:=POS_Plasma4_1_VW;
313         POS_Plasma4_2:=POS_Plasma4_2_VW;
314         POS_Plasma4_3:=POS_Plasma4_3_VW;
315         !Posicao de pick tapete entrada
316         POS_PickTapeteEntrada:=POS_PickTapeteEntrada_VW;
317         !Posicao de place na base
318         POS_Place:=POS_Place_VW;
319         POS_AproxPlace:=RelTool(POS_Place_VW,0,0,-200);
320         !Posicao de pick da mesa
321         POS_PickMesa:=POS_PickMesa_VW;
322     ENDIF
323 ENDPROC
324
325 ENDMODULE
```

Anexo C - Programa do robô de soldadura laser



```
Instruction Folding Angulo
Instruction Folding 1 Quadrante
if [123] Program variables: Contador equal to 0 is true
  Set Program variables:AnguloX = -180 + [123] Program variables: Grau / 2
  Set Program variables:AnguloY = 0
  Set array elements name: Position variable: POSTemp subscript: 3 value: [123] Program variables: AnguloX
  Set array elements name: Position variable: POSTemp subscript: 4 value: [123] Program variables: AnguloY
if [123] Program variables: Contador greater than and equal to 1 and [123] Program variables: Contador less than and equal to 6 is true
  Set Program variables:AnguloX = [123] Program variables: AnguloX - [123] Program variables: IncGrau
  Set Program variables:AnguloY = [123] Program variables: AnguloY - [123] Program variables: IncGrau
  Set array elements name: Position variable: POSTemp subscript: 3 value: [123] Program variables: AnguloX
  Set array elements name: Position variable: POSTemp subscript: 4 value: [123] Program variables: AnguloY
Instruction Folding 2 Quadrante
if [123] Program variables: Contador equal to 7 is true
```

The image shows two sections of Scratch code blocks. The first section, titled "Instruction Folding 2 Quadrante", contains an "if" block where "Program variables: Contador" is equal to 7. Inside this "if" block, there are four "Set" blocks: "Program variables:AnguloX" is set to 180; "Program variables:AnguloY" is set to 0 minus "Program variables: Grau" divided by 2; "Set array elements name: Position variable: POSTemp" with subscript 3 is set to "Program variables: AnguloX"; and "Set array elements name: Position variable: POSTemp" with subscript 4 is set to "Program variables: AnguloY". Below this is another "if" block where "Program variables: Contador" is greater than or equal to 8 and less than or equal to 12. Inside this second "if" block, there are four "Set" blocks: "Program variables:AnguloX" is set to "Program variables: AnguloX" minus "Program variables: IncGrau"; "Program variables:AnguloY" is set to "Program variables: AnguloY" plus "Program variables: IncGrau"; "Set array elements name: Position variable: POSTemp" with subscript 3 is set to "Program variables: AnguloX"; and "Set array elements name: Position variable: POSTemp" with subscript 4 is set to "Program variables: AnguloY". The second section, titled "Instruction Folding 3 Quadrante", contains an "if" block where "Program variables: Contador" is equal to 13.

Instruction Folding 3 Quadrante

if [123] Program variables: Contador equal to 13 is true

Set Program variables:AnguloX = 180 - [123] Program variables: Grau / 2

Set Program variables:AnguloY = 0

Set array elements name: Position variable: POSTemp subscript: 3 value: [123] Program variables: AnguloX

Set array elements name: Position variable: POSTemp subscript: 4 value: [123] Program variables: AnguloY

if [123] Program variables: Contador greater than and equal to 14 and [123] Program variables: Contador less than and equal to 18 is true

Set Program variables:AnguloX = [123] Program variables: AnguloX + [123] Program variables: IncGrau

Set Program variables:AnguloY = [123] Program variables: AnguloY + [123] Program variables: IncGrau

Set array elements name: Position variable: POSTemp subscript: 3 value: [123] Program variables: AnguloX

Set array elements name: Position variable: POSTemp subscript: 4 value: [123] Program variables: AnguloY

Instruction Folding 4 Quadrante

if [123] Program variables: Contador equal to 19 is true

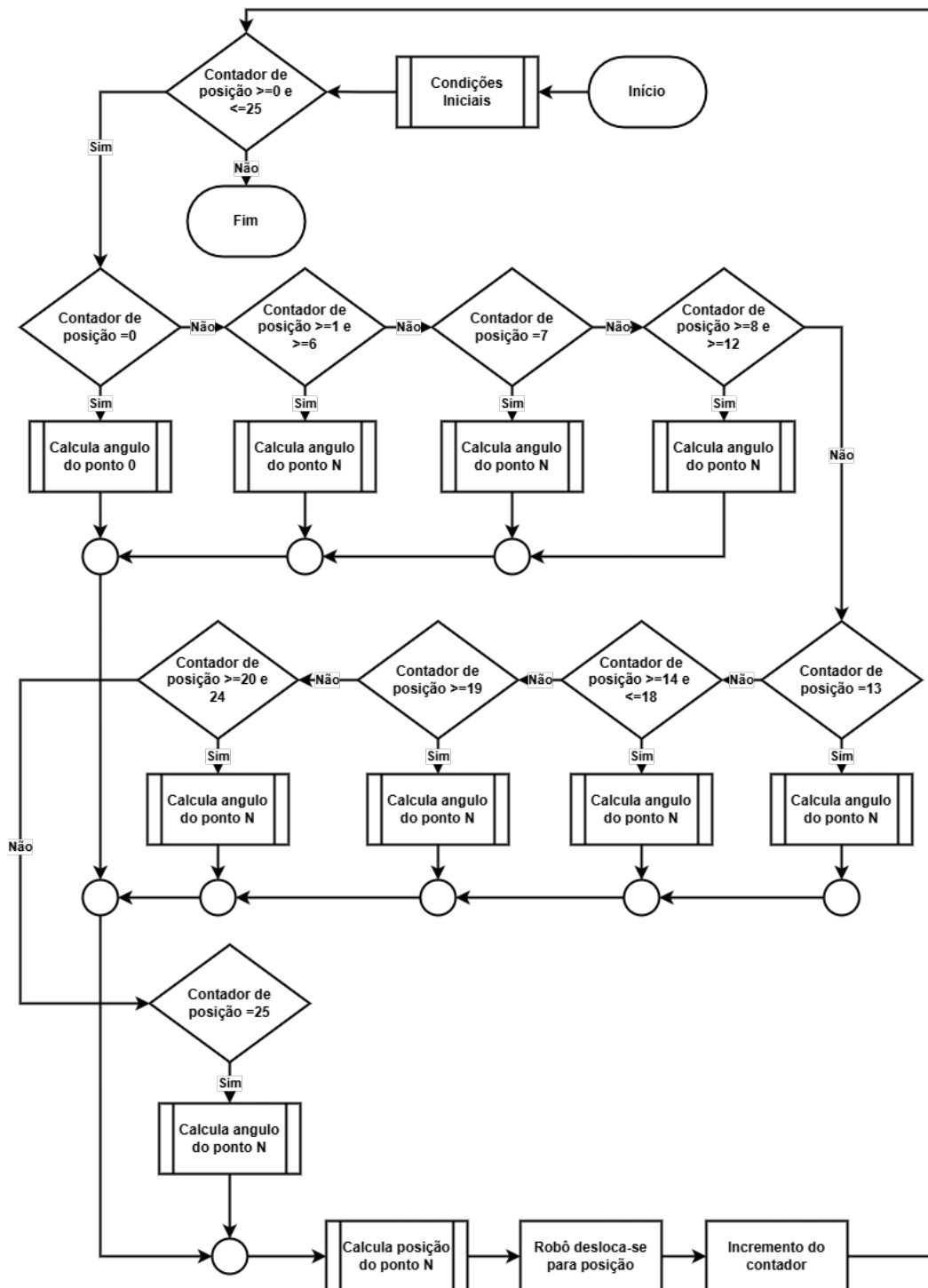
The image shows a Scratch script with two main sections: '4 Quadrante' and 'Posicao'. The '4 Quadrante' section contains three conditional blocks. The first 'if' block checks if 'Contador' is equal to 19, followed by 'Set AnguloX = -180', 'Set AnguloY = 0 + Grau / 2', and two 'Set array elements' blocks for 'POSTemp' at indices 3 and 4. The second 'if' block checks if 'Contador' is between 20 and 24, followed by 'Set AnguloX = AnguloX + IncGrau', 'Set AnguloY = AnguloY - IncGrau', and two 'Set array elements' blocks for 'POSTemp' at indices 3 and 4. The third 'if' block checks if 'Contador' is equal to 25, followed by 'Set AnguloX = -180 + Grau / 2', 'Set AnguloY = 0', and two 'Set array elements' blocks for 'POSTemp' at indices 3 and 4. The 'Posicao' section contains a single 'Set array elements' block for 'POSTemp' at index 0, with a value calculated as $\text{raio} \times \cos(\text{angle} - 90) \times 15 \times \text{Contador}$.

```
Instruction Folding 4 Quadrante  
if (123 Program variables: Contador equal to 19) is true  
  Set Program variables:AnguloX = -180  
  Set Program variables:AnguloY = 0 + (123 Program variables: Grau / 2)  
  Set array elements name: Position variable: POSTemp subscript: 3 value: (123 Program variables: AnguloX)  
  Set array elements name: Position variable: POSTemp subscript: 4 value: (123 Program variables: AnguloY)  
if (123 Program variables: Contador greater than and equal to 20 and (123 Program variables: Contador less than and equal to 24) is true  
  Set Program variables:AnguloX = (123 Program variables: AnguloX + (123 Program variables: IncGrau)  
  Set Program variables:AnguloY = (123 Program variables: AnguloY - (123 Program variables: IncGrau)  
  Set array elements name: Position variable: POSTemp subscript: 3 value: (123 Program variables: AnguloX)  
  Set array elements name: Position variable: POSTemp subscript: 4 value: (123 Program variables: AnguloY)  
if (123 Program variables: Contador equal to 25) is true  
  Set Program variables:AnguloX = -180 + (123 Program variables: Grau / 2)  
  Set Program variables:AnguloY = 0  
  Set array elements name: Position variable: POSTemp subscript: 3 value: (123 Program variables: AnguloX)  
  Set array elements name: Position variable: POSTemp subscript: 4 value: (123 Program variables: AnguloY)  
Instruction Folding Posicao  
Set array elements name: Position variable: POSTemp subscript: 0 value: (123 Program variables: raio * cosine (angle) - 90 * 15 * (123 Program variables: Contador)
```

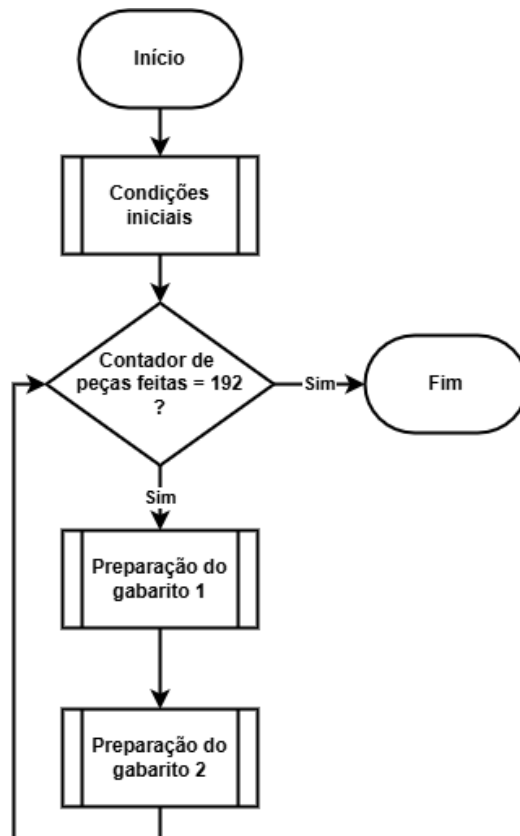
The image shows a Scratch script titled "Posicao" with the following blocks:

- Instruction Folding** (orange block)
- Set array elements** (green block): name: Position variable: POSTemp, subscript: 0, value: $123 \text{ Program variables: raio} * \cos(\text{angle}) 90 - 15 * 123 \text{ Program variables: Contador}$
- Set array elements** (green block): name: Position variable: POSTemp, subscript: 1, value: $123 \text{ Program variables: raio} * \sin(\text{angle}) 90 - 15 * 123 \text{ Program variables: Contador}$
- Set array elements** (green block): name: Position variable: POSTemp, subscript: 2, value: 10
- Set** (green block): Program variables:Contador = $123 \text{ Program variables: Contador} + 1$
- linear motion** (blue block): Position variable: POSTemp
- joint motion** (blue block): begin

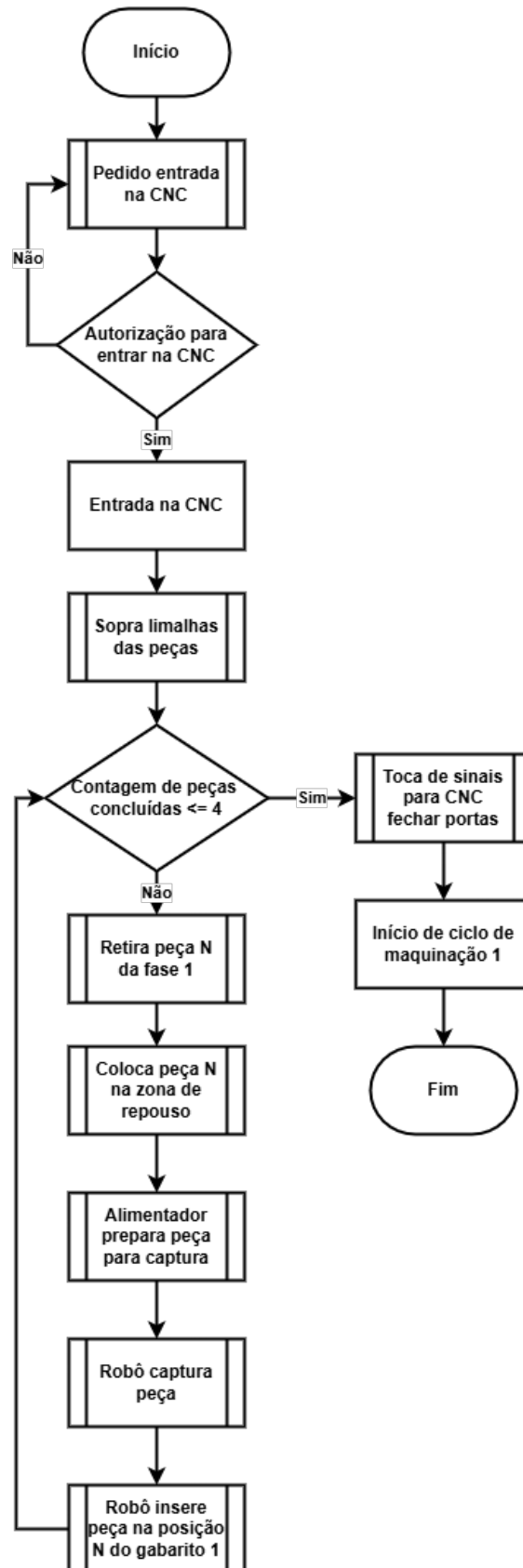
Anexo D - Fluxograma da segunda proposta do robô de soldadura laser



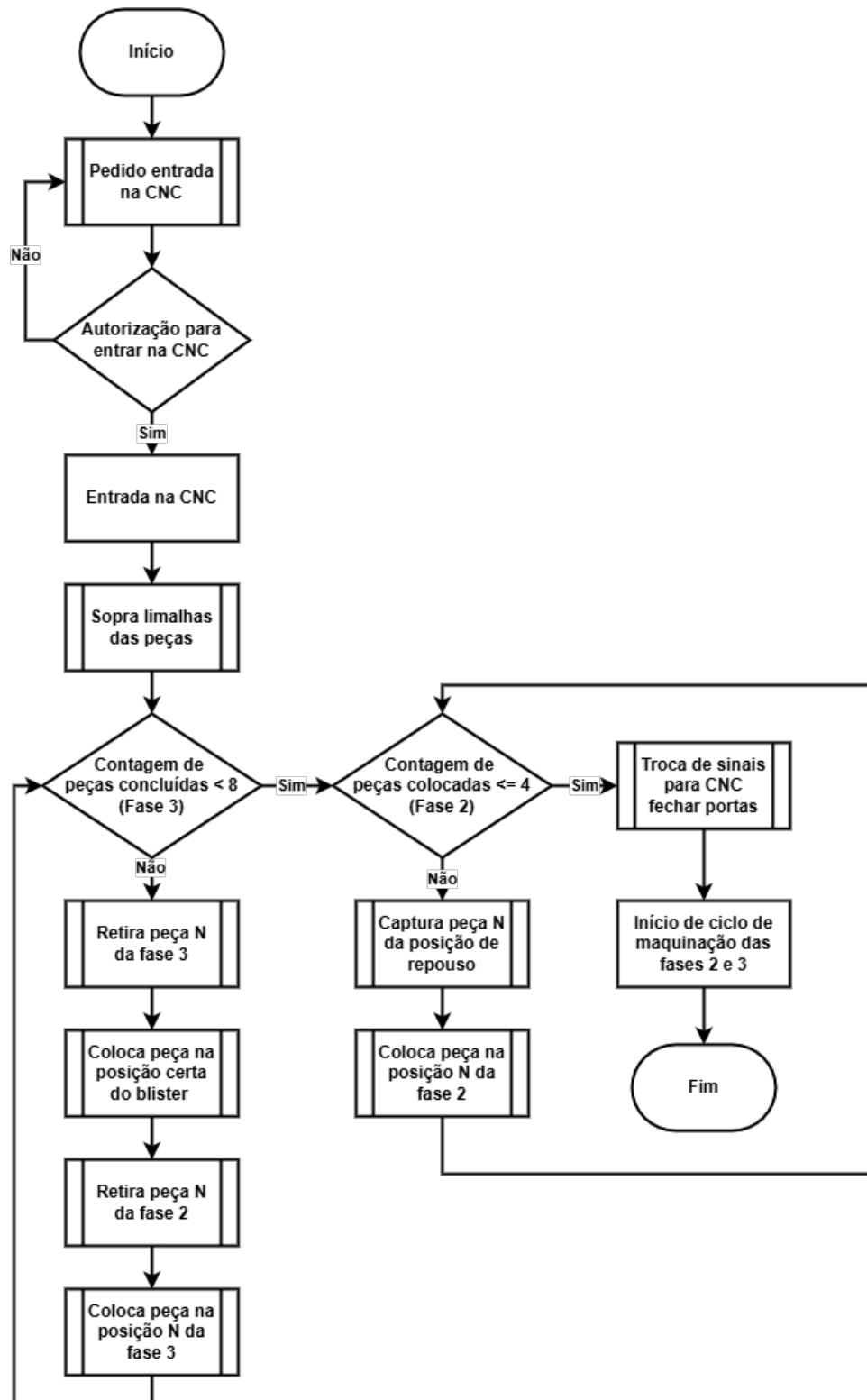
Anexo E - Fluxograma do Programa Principal do Alimentador de CNC



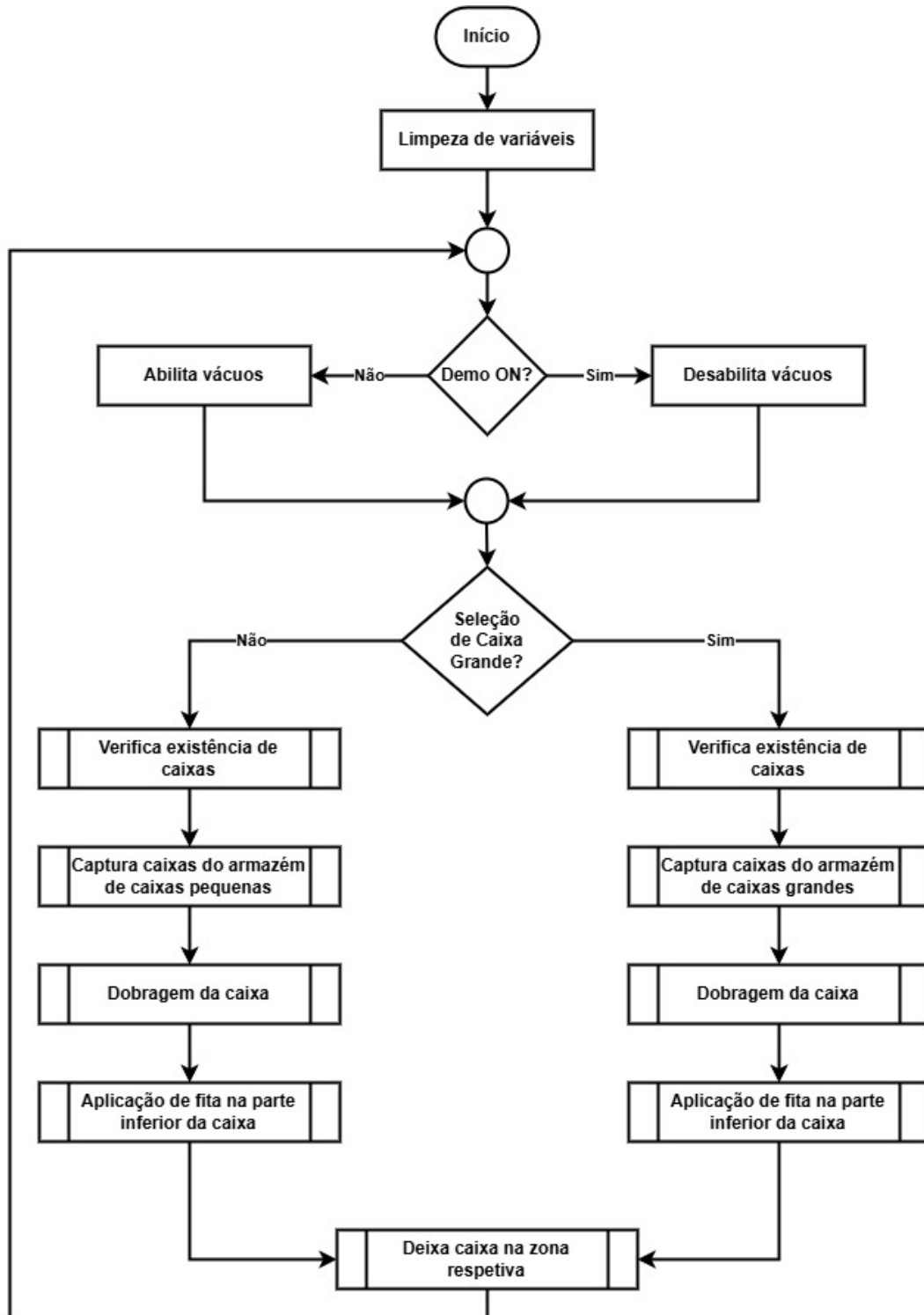
Anexo F - Fluxograma do Programa de Preparação do Gabarito 1 do Alimentador de CNC



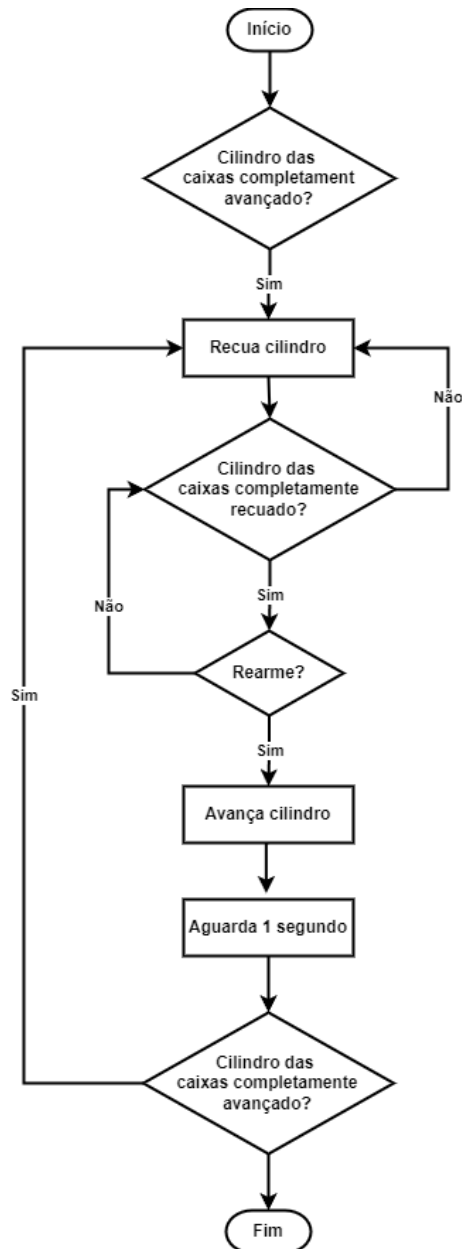
Anexo G - Fluxograma da preparação do gabarito 2



Anexo H - Fluxograma de funcionamento geral do robô empacotador



Anexo I - Fluxograma de funcionamento de deteção de armazém vazio



Anexo J - Diagrama de Gantt do estágio



**Instituto Superior
de Engenharia**

Politécnico de Coimbra