

isec
Engenharia

MESTRADO EM ENGENHARIA
ELETROTÉCNICA

Sistemas de Visão na Área da Cerâmica

Autor

André Daniel Martins Santos

Orientadores

Professor Coordenador com Agregação

Doutor Nuno Miguel Fonseca Ferreira

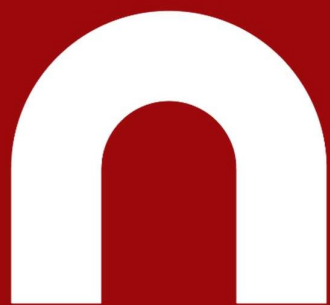
Professor Coordenador

Doutor Victor Daniel Neto dos Santos

INSTITUTO POLITÉCNICO
DE COIMBRA

Coimbra, janeiro de 2024

INSTITUTO SUPERIOR
DE ENGENHARIA



isec

Engenharia

DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA

Sistemas de Visão na Área da Cerâmica

Relatório de Trabalho de Projeto para a obtenção do grau de
Mestre em Engenharia Eletrotécnica

Especialização em Automação e Comunicações em Sistemas
Industriais

Autor

André Daniel Martins Santos

Orientadores

Professor Coordenador com Agregação

Doutor Nuno Miguel Fonseca Ferreira

Professor Coordenador

Doutor Victor Daniel Neto dos Santos

Supervisor na empresa Centro Tecnológico da Cerâmica e Vidro

Engenheiro Victor Francisco

INSTITUTO POLITÉCNICO
DE COIMBRA

INSTITUTO SUPERIOR
DE ENGENHARIA

Coimbra, janeiro de 2024

AGRADECIMENTOS

Após o término deste percurso académico, cabe expressar o meu agradecimento:

Aos meus orientadores do Instituto Superior de Engenharia de Coimbra (ISEC), o Professor Doutor Nuno Miguel Fonseca Ferreira e o Professor Doutor Victor Daniel Neto dos Santos, que desde o primeiro dia me incentivaram e orientaram de forma extraordinária e contínua.

Aos meus amigos e colegas, Rogério Torres e João Mendes, que desempenharam um papel fundamental no que toca à realização deste projeto.

Agradeço à empresa Centro Tecnológico da Cerâmica e do Vidro (CTCV) e ao meu supervisor, o Eng.º Victor Francisco e Rui Gouveia, que viram o meu potencial e me acolheram de forma exemplar.

À minha família que foi o pilar nesta reta final do meu percurso académico, pela coragem e incentivo, por nunca desistirem de acreditar que tudo isto era possível. Obrigado pela vossa compreensão. A ti, mãe, obrigado por todas as orientações que me deste, quando tudo parecia desmoronar. A ti, pai, agradeço por seres o representante da descontração.

À minha namorada pelo apoio, pela paciência e por me incentivar todos os dias incansavelmente, para ser uma pessoa trabalhadora, empenhada e focada naquilo que me faz feliz.

Por último, mas não menos importante, um agradecimento com carinho:

Ao ISEC que foi indispensável como instituição académica formadora, uma vez que me proporcionou boas experiências, oportunidades de aprendizagem e me deu ferramentas essenciais para ser um profissional de excelência.

RESUMO

Este relatório tem como objetivo apresentar as atividades realizadas ao longo de um projeto curricular, desenvolvido no âmbito da componente Projeto do plano de estudos do Mestrado em Engenharia Eletrotécnica (MEE), área de especialização em “Automação e Comunicações em Sistemas Industriais”, ministrado pelo Instituto Superior de Engenharia de Coimbra (ISEC). Este projeto decorreu nas instalações do Centro Tecnológico da Cerâmica e Vidro (CTCV) localizadas em Antanhol – Coimbra.

O relatório apresenta um breve enquadramento teórico, de suporte às atividades desenvolvidas no projeto. De seguida, é apresentado um guia do KUKA Sim Pro que permite a simulação e programação de linhas de produção industrial automatizadas. Foi também desenvolvida uma solução tecnológica, a aplicar na indústria cerâmica, que utiliza uma câmara localizada no teto, para determinar o centro dos pratos cerâmicos, de modo a permitir a um robô manipulador agarrar nesses pratos, através de um ponto relativo calculado para o efeito, de forma a otimizar e a tornar mais seguro o processo de vidragem. Para além disso, foi desenvolvido um programa Matlab que através de imagens adquiridas por uma câmara, identificava os objetos cerâmicos que estavam num dado local, classificando-os num dos seguintes três tipos considerados: chávena, pires e caneca. Por último, são apresentadas aplicações de visão artificial para a área da indústria cerâmica baseadas em algoritmos de aprendizagem.

Palavras-Chave: Programação; Simulação; KUKA Sim Pro; Visão Artificial.

ABSTRACT

This report aims to present the activities carried out throughout a curricular project, developed within the scope of the Project unit of the Master in Electrical Engineering (MEE), specialization area in "Automation and Communications in Industrial Systems", taught at Instituto Superior de Engenharia de Coimbra (ISEC). The project activities took place in the facilities of the Centro Tecnológico da Cerâmica e Vidro (CTCV) located in Antanhol - Coimbra.

The report presents a brief theoretical framework to support the activities developed in the project. It then presents a guide to "KUKA Sim Pro" which allows the simulation and programming of automated industrial production lines. A technological solution was also developed, to be applied in the ceramics industry, using a camera located on the ceiling, to determine the center of the ceramic dishes, in order to allow a robotic arm to grab these dishes, through a relative point calculated for that purpose, in order to optimize and make safer the glazing process. In addition, a Matlab program was developed which, through images acquired by a camera, identified the ceramic objects that are in a given location, classifying them into one of the following three types: cup, saucer and mug. Finally, artificial vision applications for the ceramics industry based on learning algorithms are presented.

Keywords: Programming; Simulation; KUKA Sim Pro; Artificial Vision.

ÍNDICE

CAPÍTULO 1: INTRODUÇÃO	1
1.1 – Objetivos do projeto	1
1.2 – Estrutura do documento	2
CAPÍTULO 2: ENQUADRAMENTO TEÓRICO-CONCETUAL	3
2.1 – Automação Industrial	3
2.1.1 – Indústria 4.0.....	3
2.1.1.1 – Princípios da Indústria 4.0.....	4
2.1.1.2 – Pilares da Indústria 4.0.....	4
2.1.2 – Indústria 5.0.....	6
2.2 – A Robótica no Quotidiano.....	7
2.2.1 – Robótica Colaborativa	8
2.2.2 – Robótica Móvel	11
2.2.3 – Robótica Industrial	12
2.2.3.1 – Elementos do Robô.....	13
2.2.3.2 – Tipos de Robô	14
2.2.3.3 – Cinemática.....	17
2.2.3.4 – Tipo de movimentos	18
2.3 – Processamento de Imagem	20
2.3.1 – Relação da Câmara com o olho humano.....	20
2.3.2 – Câmaras.....	21
2.3.3 – Visão Computacional.....	22
2.4 – Inteligência Artificial.....	25
2.4.1 – Tipos de Inteligência Artificial.....	25
2.4.2 – Impactos da Inteligência Artificial	26
2.4.3 – Aprendizagem profunda.....	26

CAPÍTULO 3: GUIA RÁPIDO SIMULADOR KUKA.....	31
3.1 – Novo Programa	31
3.2 – Save Programa.....	36
3.3 – Componentes	37
3.3.1 – Save Components	37
3.3.2 – Criação de garra.....	38
3.3.3 – Criação Creator.....	45
3.4 – Programação do Simulador	49
3.4.1 – Sinais de Entrada e Saída.....	49
3.4.2 – Programação do robô.....	52
3.5 – Exemplos de simulações	54
CAPÍTULO 4: Visão Aplicada com Robótica	59
4.1 – Mako Allied Vision	59
4.2 – Matlab identificação Cerâmica	65
4.3 – Algoritmos de Aprendizagem	67
CONCLUSÃO.....	69
Referências	71

ÍNDICE DE FIGURAS

Figura 1 – Unimate. Fonte: (ELETROGATE, 2022).	8
Figura 2 – Robótica Colaborativa. Fonte: (Universal Robots, 2019).	9
Figura 3 – Tipos de relações com o robô.	10
Figura 4 – Exemplos de robôs móveis em diferentes ambientes (Fonte: (Defesanet, 2016), (Encyclopedia Britannica, 15 Jul. 2020) e (Eos, 2017)).	11
Figura 5 – AGV Robot Fonte: (EDS ROBOTICS).	11
Figura 6 – Robótica Industrial. Fonte: (blog.kalatec).	12
Figura 7 – Sequência de elos numa junta de um braço robótico. Fonte: (Groveer 1988).....	13
Figura 8 – Controlador de Robô. Fonte: (directindustry)	14
Figura 9 – Robô Cartesiano. Fonte: (campetella).	15
Figura 10 – Robô Esférico. Fonte: (professor.ufop).....	15
Figura 11 – Kuka Scara Robot. Fonte: (Kuka Robotics).....	16
Figura 12 – Robô delta Fanuc. Fonte: (Fanuc).....	16
Figura 13 – Movimento PTP.	19
Figura 14 – Movimento Linear.....	19
Figura 15 – Movimento Circular.	20
Figura 16 – Movimento Spline.....	20
Figura 17 – Olho Humano. Fonte: (A Física do Olho Humano, 2020).....	21
Figura 18 – Cognex Fonte: (eletronor).	22
Figura 19 – Relação entre diferentes disciplinas da Inteligência Artificial.	27
Figura 20 – Relação existente entre diferentes conceitos de IA. Fonte: (Kavlakoglu, 2020)....	28
Figura 21 – Ambiente Inicial do programa.	31
Figura 22 – Menu Modeling.....	32
Figura 23 – Eixos do componente.	32
Figura 24 – Menu para adicionar objetos.....	33
Figura 25 – Propriedades do bloco.	33
Figura 26 – Cubo Criado.....	34
Figura 27 – Referência descentralizada.	34
Figura 28 – Função Move.	35
Figura 29 – Processo de centralização da referência.	35

Figura 30 – Cubo com referência centralizada.....	36
Figura 31 – Menu Home.	36
Figura 32 – Menu File.....	37
Figura 33 – Guardar componente.	38
Figura 34 – Garra criada.	38
Figura 35 – Primeira forma de Criar um link.	39
Figura 36 – Segunda forma de Criar um link.	39
Figura 37 – Propriedades do Link.	40
Figura 38 – Configuração do Link.....	40
Figura 39 – Adicionar Servo.	41
Figura 40 – Configuração do Link 2.....	41
Figura 41 – Submenu Wizards.	42
Figura 42 – Menu End Effector.....	42
Figura 43 – Garra com os sensores.....	43
Figura 44 – Adicionar eixo de referência.....	44
Figura 45 – Garra com eixo de referência.....	44
Figura 46 – Propriedades do EOAT_MountInterface.....	45
Figura 47 – Menu Behaviors.	46
Figura 48 – Propriedades do ComponentCreator.	46
Figura 49 – Adicionar Container.	47
Figura 50 – Propriedades do Output do Container.	47
Figura 51 – Configuração do Eixo Referencial da Criação.	48
Figura 52 – Configuração da movimentação do tapete.....	48
Figura 53 – Propriedades do Input do One Way Path.	49
Figura 54 – Componentes com sinais.	49
Figura 55 – Submenu Connect.	50
Figura 56 – Menu Connect Signal.	50
Figura 57 – Configuração de sinais.	51
Figura 58 – Interligação de Componentes.....	51
Figura 59 – Menu Program.	52
Figura 60 – Consola Virtual do Robô.	52
Figura 61 – Painel de Programação.	53
Figura 62 – Robô em modo "JOG".....	54

Figura 63 – Ambiente Fabril em Simulação.	54
Figura 64 – Barra de controlo de simulação.....	55
Figura 65 – Azulejos a aguardar deslocação.	55
Figura 66 – Colocação dos azulejos no tapete.....	56
Figura 67 – Colocação dos azulejos em caixas.	56
Figura 68 – Colocação de cintas na caixa.	57
Figura 69 – Paletização dos azulejos.	57
Figura 70 – Imagem guardada.	59
Figura 71 – Setup Câmara, realidade vs simulador.	60
Figura 72 – a) Imagem Cartesiano; b) Detetar Centros.	61
Figura 73 – Detetar centros ROI.....	62
Figura 74 – Fluxograma do programa.....	63
Figura 75 – Imagens com luminosidade baixa e alta.....	64
Figura 76 – Objetos Cerâmicos.	65
Figura 77 – Código Matlab.	66
Figura 78 – Identificação de Objetos Cerâmicos.....	66
Figura 79 – Imagem processada com Yolo.....	68

SIMBOLOGIA E ABREVIATURAS

AGV	<i>Automated Guided Vehicle</i>
AUV	<i>Autonomous Underwater Vehicle</i>
CISCO	<i>Commercial and Industrial Security Corporation</i>
CNN	<i>Convolutional Neural Networks</i>
CTCV	Centro Tecnológico da Cerâmica e do Vidro
IA	Inteligência Artificial
IDT	Investigação e Desenvolvimento Tecnológico
IFR	<i>International Federation of Robotics</i>
IoS	<i>Internet of Services</i>
IoT	<i>Internet of Things</i>
IPAC	Instituto Português de Acreditação
ISEC	Instituto Superior de Engenharia de Coimbra
MEE	Mestrado em Engenharia Eletrotécnica
PoE	<i>Power over Ethernet</i>
PPP	<i>Prismatic Prismatic Prismatic</i>
RL	<i>Reinforcement Learning</i>
ROI	<i>Region Of Interest</i>
RPP	<i>Revolute Prismatic Prismatic</i>
RRP	<i>Revolute Revolute Prismatic</i>
SCARA	<i>Selective Compliance Assembly Robot Arm</i>
UAV	<i>Unmanned Aerial Vehicle</i>
VANT	Veículo Aéreo Não Tripulado

CAPÍTULO 1: INTRODUÇÃO

O presente relatório de projeto, inserido no Mestrado de Engenharia Eletrotécnica (MEE) na Área de Especialização em “Automação e Comunicações em Sistemas Industriais” pelo Instituto Superior de Engenharia de Coimbra (ISEC), intitula-se como Sistemas de Visão na Área da Cerâmica. Este pretende demonstrar as atividades realizadas ao longo do projeto, que decorreram na empresa Centro Tecnológico da Cerâmica e Vidro (CTCV).

A referida empresa é uma Entidade do Sistema Científico e Tecnológico que dispõe de laboratórios acreditados pelo Instituto Português de Acreditação (IPAC) vocacionados para a realização de diversos ensaios e análises. Para além disso, na sua qualidade de Organismo de Normalização Setorial participa ou acompanha trabalhos de normalização nacional, europeia e internacional com interesse para os sectores da Cerâmica e do Vidro. Para o efeito dispõe de uma equipa pluridisciplinar com valências científicas e competências específicas sólidas em diversas áreas de conhecimento. Este perfil de qualidade foi solidificado ao longo dos 30 anos de existência da organização resultado da formação especializada e experiência acumulada. Assim sendo, o CTCV providencia um conjunto de soluções tanto ao nível da consultoria como da auditoria, formação profissional, soluções de Investigação e Desenvolvimento Tecnológico (IDT), entre outros, orientados maioritariamente para os setores da Cerâmica e do Vidro.

1.1 – Objetivos do projeto

O documento corresponde a um relatório de projeto, no âmbito da componente Projeto/Dissertação/Estágio do plano de estudos do Mestrado em Engenharia Eletrotécnica, mais concretamente na Área de Especialização em “Automação e Comunicações em Sistemas Industriais”, acompanhado pelos orientadores e supervisor estabelecidos pelo ISEC. A duração deste projeto foi aproximadamente de 8 meses e realizou-se na empresa CTCV. A orientação do Projeto ficou ao cuidado dos professores Nuno Ferreira e Victor Santos do ISEC e do responsável da respetiva empresa CTCV, o Engenheiro Victor Francisco. Neste sentido, o Projeto teve como principal objetivo dar a conhecer as aplicações da robótica e da visão na área da cerâmica. Para isso, foram concretizadas diversas atividades, nomeadamente programação de robôs em aplicações para a cerâmica, trabalhos de simulação de robótica e aplicações de visão artificial, por exemplo *Pick and Place*.

1.2 – Estrutura do documento

O presente relatório encontra-se dividido em quatro capítulos, nos quais é feita uma exposição e descrição detalhada das atividades desenvolvidas ao longo do projeto.

No primeiro capítulo é apresentada uma breve introdução a este trabalho de projeto e aos seus objetivos, bem como um breve resumo acerca da empresa de acolhimento onde o projeto se realizou.

O segundo capítulo aborda o estado da arte, ou seja, os respetivos conceitos de automação industrial, visão artificial, robótica industrial e colaborativa.

O terceiro capítulo descreve duas atividades realizadas ao longo do projeto. A primeira atividade consistiu na utilização do *software* KUKA Sim Pro para realizar a programação de uma linha de produção para a empresa Aleluia Cerâmicas, S.A., a qual será implementada na prática num futuro próximo. Relativamente à segunda atividade apresentada, esta foi desenvolvida com o objetivo de identificar se um dado objeto cerâmico é uma chávena, pires ou caneca, por outro lado, detetar os centros destes objetos para que no futuro se possa aplicar essa informação, para transportar esses objetos de um lado para o outro, por meio de um robô, sem intervenção humana.

No último capítulo são apresentadas as considerações finais e conclusões do projeto desenvolvido, aqui descrito, mencionando os pontos positivos e negativos, as respetivas limitações e, por último, os objetivos atingidos com a realização deste Trabalho de Projeto.

CAPÍTULO 2: ENQUADRAMENTO TEÓRICO-CONCETUAL

2.1 – Automação Industrial

Antes da Primeira Revolução Industrial, os produtos eram produzidos de forma manual (Santos, Alberto, Lima, & Charrua-Santos, 2018), o que resultava numa baixa produção, enquanto a população crescia cada vez mais, levando assim a um lucro reduzido. Entre 1760 e 1860 surgiu a Primeira Revolução Industrial que permitiu a evolução dos setores produtivo e de transporte. O carvão começou a ser utilizado como fonte de energia desde a descoberta da máquina a vapor (Venturelli, 2021). Esta nova tecnologia foi aplicada pela primeira vez na indústria têxtil, tornando-se o símbolo da produção excedente, inspirando outros setores a renovar a sua forma de agir e pensar, bem como, automatizar os seus processos (Sakurai & Zuchi, 2018). Esta primeira revolução transformou completamente a indústria e a vida das pessoas, sendo ainda hoje possível observar essa mudança positiva (Cavalcante, 2011).

Desde a Primeira Revolução Industrial que a indústria tem vindo a sofrer um desenvolvimento ao longo do tempo, passando pela Segunda e a Terceira Revolução Industrial. A Indústria 2.0 ficou conhecida pela descoberta da eletricidade, a transformação do ferro em aço, o surgimento e modernização dos veículos de transporte e pelo avanço dos meios de comunicação (Sakurai & Zuchi, 2018). Em relação à Indústria 3.0, esta foi responsável por inovações tecnológicas como, por exemplo, avanços no campo da informática, da robótica, das telecomunicações, dos transportes e da biotecnologia (BOETTCHER, 2015). Através destes avanços resultou o surgimento da Indústria 4.0, também conhecida por Quarta Revolução Industrial.

2.1.1 – INDÚSTRIA 4.0

O termo indústria 4.0 foi pronunciado pela primeira vez na feira Hannover Messe 2011, em outubro de 2011. No ano seguinte, os autores deste projeto, supervisionado por Siegfried Dais e Kagermann, apresentaram um relatório de recomendação para o Governo Federal Alemão para projetar a implantação da Indústria 4.0 (RIBEIRO, 2018). O objetivo deste relatório é promover a informatização da manufatura e a integração dos dados, criando redes inteligentes que permitem controlar os módulos de produção de forma autónoma.

2.1.1.1 – PRINCÍPIOS DA INDÚSTRIA 4.0

Existem princípios que devem ser respeitados na implementação da indústria 4.0 e que definem os sistemas inteligentes (Silveira, 2016):

- Virtualização – Promove a existência de uma cópia virtual das fábricas inteligentes, permitindo rastrear e monitorizar remotamente todos os processos, através de sensores implementados ao longo destas;
- Capacidade de operação em tempo real – Promove a aquisição e tratamento de dados de forma direta e rápida, permitindo ter a opção de decisão em tempo real;
- Orientação a serviços – Existência de arquiteturas de aplicações, associadas ao conceito de *Internet of Services* (IoS);
- Modularidade – Produção de acordo com a procura, ligação de módulos na produção, permitindo alterar tarefas nas máquinas o que leva a uma produção mais flexível;
- Descentralização – O sistema cyber-físico poderá tomar as decisões de acordo com as necessidades da produção em tempo real. As máquinas não só receberão comandos, como também fornecerão informações sobre o seu ciclo de trabalho, permitindo que os módulos da fábrica trabalhem de forma descentralizada, melhorando vários processos de produção.

2.1.1.2 – PILARES DA INDÚSTRIA 4.0

Os avanços tecnológicos verificados nos últimos anos, nomeadamente na área da automação e robótica deram origem à Quarta Revolução Industrial. Muitas das tecnologias acima referidas podem integrar-se neste sistema, mas dentro delas existem algumas que acabam por se destacar mais, representando os pilares da Indústria 4.0 (Pasquini, 2018):

- Internet das Coisas (IoT – *Internet of Things*): Trata-se de um conceito no qual se pretendem ligar objetos e dispositivos físicos à Internet através de uma rede de comunicação, com ou sem fios, e protocolos que possibilitam a troca de dados digitais entre os referidos objetos e a Internet. O funcionamento destes sistemas é a base de Internet das Coisas constituídos por vários sensores e atuadores, que permitem a recolha de dados/sistemas cyber-físicos;

- *Big Data Analytics in the fog*: São sistemas de dados extensos e complexos que utilizam novas abordagens para a aquisição, verificação e manipulação das respectivas informações. O *Big Data Analytics* é composto por 6 C's para tratar a informação, Conexão (ligações dos equipamentos através da rede industrial inteligente), Cloud (nuvem de dados), Cyber (modelo e memória), Conteúdo (informação a enviar), Comunidade (compartilhamento das informações) e Customização (personalização e valores). Já o *fog* é uma rede de computadores entre a fonte de dados e a *cloud*, ou seja, uma *cloud* local. Assim o objetivo passa por uma manipulação da informação localmente, diminuindo a quantidade de dados transmitidos na rede e a sua complexidade computacional necessária na *cloud*. De acordo com a CISCO, o *fog* consiste em expandir a *cloud* por meio da colocação de nós estrategicamente numa infraestrutura de computação distribuída, levando vantagens a todos os dispositivos já ligados à IoT;
- Cibersegurança: Atualmente, com os avanços das tecnologias, a segurança tem vindo a ser uma das maiores preocupações a nível mundial, uma vez que antigamente as redes entre os equipamentos eram fechadas e, por isso, havia um maior controlo por parte das empresas. Deste modo, é necessário que existam mais “meios de comunicação cada vez mais confiáveis e sofisticados” (RUBMANN, et al., 2015). A cibersegurança passou a ser uma área de extrema importância com a entrada do *big data*, IoT e outros protocolos de redes abertas, pois com as ameaças que têm vindo a surgir é possível que os trabalhos desenvolvidos possam ser facilmente comprometidos;
- Inteligência Artificial ou *Deep Learning*: A Inteligência Artificial (IA) representa a habilidade da máquina executar tarefas normalmente associadas a seres inteligentes. Desde a criação do computador é possível que os mesmos consigam realizar tarefas complexas através da utilização de sistemas de redes neuronais. Na visão artificial é usada cada vez mais IA de forma que seja possível optar por decisões mais benéficas, permitindo que os dados possam ter diagnósticos, previsões e prescrições;
- Simulação: É possível realizar simulações em 3D nos produtos e materiais dos processos de produção, o que permite uma melhor qualidade do produto e uma maior eficiência na fabricação. Deste modo, permite a simulação e teste de uma ideia para que se tenha uma noção do tempo e dos recursos exigidos (Longo, 2018).

2.1.2 – INDÚSTRIA 5.0

Com os avanços tecnológicos verificados na Indústria 4.0, já há quem fale no surgimento de uma quinta revolução industrial, também conhecida por Indústria 5.0. Este termo foi mencionado pela primeira vez em 2015, e retrata a personalização dos produtos e sistemas em que o homem interage com máquinas inteligentes (HALEEM & JAVAID, 2019), combinando a criatividade humana com a precisão do robô. Estes sistemas inteligentes também têm a vantagem de operarem em ambientes hostis assegurando a saúde humana (PASCHEK, MOCAN, & DRAGHICI, 2019).

A necessidade da Indústria 5.0 deve-se à exigência de produtos mais personalizados, isto é, foca-se em pequenos lotes de produção com produtos individualizados e consequentemente com maior valor agregado, enquanto a Indústria 4.0 utiliza um modelo de produção em massa composto por linhas de produção autónomas, o que leva a grandes quantidades de produção com o intuito de uma elevada procura de produtos (Maestri, 2021).

Tabela 1 – Comparações entre Indústria 4.0 e 5.0. Fonte: Adaptado de (Maestri, 2021)

Indústria 4.0	Indústria 5.0
Indústria inteligente utilizando tecnologias como IoT e sistemas cyber-físicos	Combinação entre a criatividade humana com a rapidez e a execução de trabalhos repetitivos e/ou árduos de robôs
Customização em massa	Personalização dos produtos
O sistema, que é integrado, é capaz de prever a possibilidade de ocorrer defeitos nos produtos, e informar o tipo de falha, dependendo do banco de dados inseridos no sistema	A robotização auxilia na redução de defeitos; controle de produtos personalizados em tempo real
Auto configurável e otimizável	Dispositivos digitais com a supervisão de seres humanos
Robótica Industrial	Robótica Colaborativa

As tecnologias que compõem a Indústria 5.0 (ASLAM, AIMIN, LI, & REHMAN, 2020) são as seguintes:

- Manufatura aditiva;
- Robôs colaborativos;
- IoT;
- Produção inteligente;
- Ecossistema digital;
- Inteligência Artificial emergente;
- Sistemas adaptativos complexos;
- Tecnologias e multiagentes.

Apesar da Indústria 5.0 não estar implementada já existem indícios da sua utilização na Indústria da Cerâmica na vidragem e manipulação de peças cerâmicas sendo que o funcionário fabril vai complementar as tarefas que a máquina ainda não consegue realizar de forma autónoma.

2.2 – A Robótica no Quotidiano

Atualmente, a tecnologia é uma parte fundamental da vida das pessoas, presente no seu dia a dia, encontrando-se em qualquer lugar, seja no supermercado, na indústria, nas suas próprias casas.

A robótica representa o estudo dos robôs, sendo esta palavra “robô” proveniente da palavra checa *robota* (MOLLE & ADAMS, 2019), que significa “trabalho forçado, servidão”. Esta expressão submete para a ideia de o robô trabalhar por nós, dando a perceção de um certo facilitismo no que toca à execução das tarefas. O robô é composto maioritariamente por sensores, manipuladores, sistemas de controlo, *software* e uma fonte de energia.

Na indústria, o facto de haver diversas tarefas repetitivas que exigem movimentos de peças/materiais muito pesados levou à necessidade de criar um robô para uso na área da indústria. Em 1954 concretizou-se essa necessidade, quando George Devol criou um robô chamado “*Unimate*” (Lousada, 2022) que integrou a linha de produção da *General Motors* (empresa de fabrico de carros), com o objetivo de transportar/manipular, agarrar as peças aquecidas de metal e colocar nos chassis dos carros. Este robô obedecia a comandos gravados em fitas magnéticas.

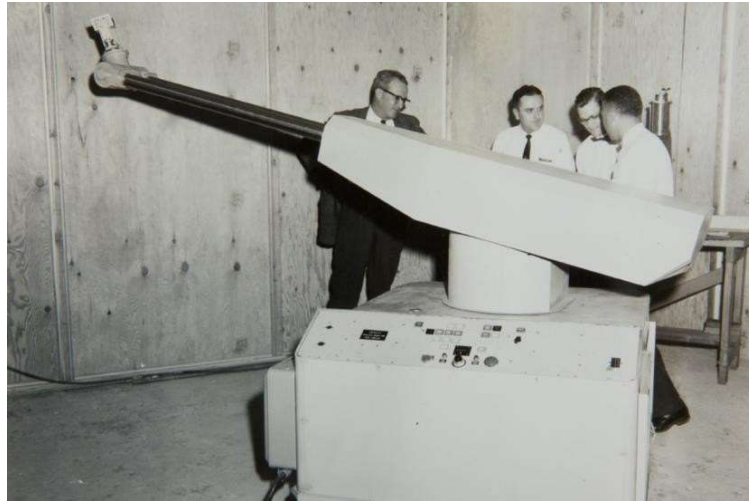


Figura 1 – Unimate. Fonte: (ELETROGATE, 2022).

Considerando a robótica no geral podemos distinguir três categorias: robótica colaborativa, robótica móvel e robótica industrial (GALIN & MESHCHERYAKOV, 2019).

2.2.1 – ROBÓTICA COLABORATIVA

Com o passar do tempo, a investigação na área da robótica mostrou ser uma mais-valia na área da indústria, permitindo rentabilizar diversos processos, contudo a solução mais eficiente em termos de produção industrial flexível não é apenas ter um robô. “A solução perfeita passa pela junção de uma linha de produção com humanos e robôs, permitindo aproveitar as qualidades de cada um como, por exemplo, a capacidade cognitiva e coordenação dos humanos e a precisão e repetibilidade dos robôs” (GALIN & MESHCHERYAKOV, 2019). Deste modo, começaram a aparecer os robôs colaborativos, estes permitem a interação com os humanos de uma maneira segura, como representado na Figura 2.



Figura 2 – Robótica Colaborativa. Fonte: (Universal Robots, 2019).

Estes robôs estão equipados com vários sensores e sistemas de visão, no entanto, há robôs que não apresentam qualquer garra sensorial e, por isso, é preciso acrescentar outras funcionalidades que permitam uma maior segurança para quem o rodeia. Esses sensores permitem que, quando o indivíduo entra na zona de trabalho do robô, é necessário que este esteja em segurança, uma vez que o robô é capaz de alterar o seu comportamento dependendo do algoritmo de comportamento. Caso o robô esteja em movimento e um humano se coloque na sua trajetória, ele deverá parar ou alterar a trajetória de movimento, de modo a garantir a segurança. Quando se fala de um robô colaborativo pretende-se descrever um ambiente em que tanto o humano como o robô possam operar, ou seja, a pessoa é vista como um colega do ponto de vista do robô e vice-versa, pois a criação deste tipo de robôs é, efetivamente, objetivos e trabalhar a par com o ser humano.

Este tipo de conceito do desenvolvimento da interação humana com o robô encontra-se presente na indústria 4.0, cujo objetivo é tornar o robô uma ferramenta auxiliar para o humano enquanto este está a produzir, levando à compreensão do conceito de “*Smart production*” (GALIN & MESHCHERYAKOV, 2019).

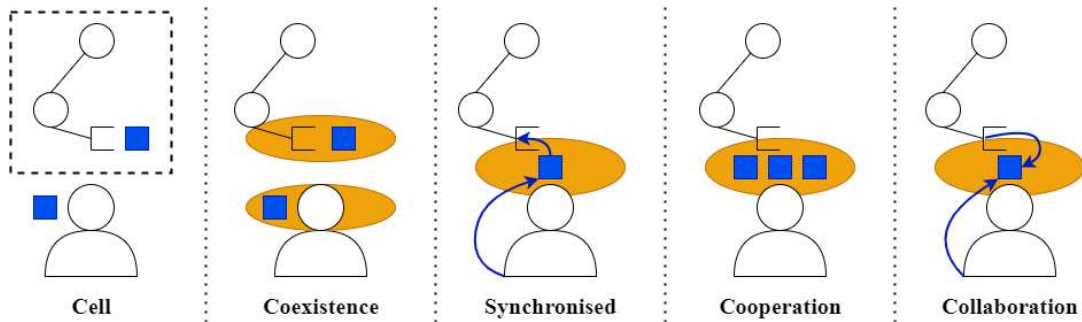


Figura 3 – Tipos de relações com o robô.

Olhando para a imagem anterior (Figura 3) pode-se perceber os diferentes conceitos de relação de um humano e de um robô (Matheson, Minto, Zampieri, Faccio, & Rosati, 2019).

- Cell: retrata a independência das tarefas realizadas por cada um, não havendo necessidade de interação já que também possui uma célula com esse objetivo;
- Coexistence: operador humano e robô estão no mesmo ambiente, mas normalmente não interagem;
- Synchronised: operador humano e robô trabalham no mesmo ambiente, mas em tempos diferentes;
- Cooperation: operador humano e robô trabalham no mesmo espaço ao mesmo tempo, embora desempenhem tarefas diferentes um do outro;
- Collaboration: operador humano e robô executam a mesma tarefa juntos, a ação de um tem efeitos no outro, empregando aos sensores e sistemas de visão.

Os robôs colaborativos são cada vez mais utilizados na indústria e apresentam vantagens bastante promissoras, tais como a segurança através dos mecanismos criados para esse fim, pois são bastante flexíveis devido à sua estrutura leve e ao espaço que ocupam. Tudo isto permitirá o acesso a diversas aplicações e a uma configuração e programação mais acessível. Apesar das vantagens descritas também apresentam algumas desvantagens, nomeadamente: custo elevado, baixa capacidade de carga e reduzida velocidade de trabalho relativamente aos restantes robôs.

2.2.2 – ROBÓTICA MÓVEL

Embora os robôs manipuladores sejam uma grande ajuda na área industrial, possuem limitações ao nível da mobilidade. Através dos robôs móveis é possível criar uma maior liberdade, permitindo uma adaptação ao ambiente e conseqüentemente uma outra perspectiva no que concerne às novas ideias de futuros projetos. Por outro lado, estes podem ser caracterizados consoante o ambiente em que se movem, por exemplo, robôs terrestres que geralmente são compostos por rodas, mas também se podem encontrar no espaço, aéreos designadamente Veículos Aéreos Não Tripulados (VANTs) ou *Unmanned Aerial Vehicles* (UAVs) e em ambientes aquáticos *Autonomous Underwater Vehicles* (AUVs).



Figura 4 – Exemplos de robôs móveis em diferentes ambientes (Fonte: (Defesanet, 2016), (Encyclopedia Britannica, 15 Jul. 2020) e (Eos, 2017).

Apesar da robótica móvel se encontrar ainda em desenvolvimento verifica-se que cada vez mais uma maior procura para robôs desse tipo. Durante a Segunda Guerra Mundial, com os avanços na área da ciência da computação, surgiu o robô móvel intitulado como “bombas voadoras”, isto é, mísseis. A partir daí notou-se uma evolução significativa e uma maior procura, por parte das pessoas, tanto para fins industriais como para uso doméstico.



Figura 5 – AGV Robot Fonte: (EDS ROBOTICS).

Os *Automated Guided Vehicle* (AGV) são usados para o transporte de matérias-primas ou produtos finais sendo um exemplo de robótica móvel (Figura 5). Os AGVs podem também ser utilizados como uma alternativa às passadeiras industriais que transportam os produtos inacabados por diversas etapas de produção de uma forma sequencial. (Lynch, et al., 2019). O primeiro AGV seguia o campo magnético, gerado por fios de cobre embutidos no chão de uma fábrica, o qual era detetado pelo AGV indicando dessa forma o caminho a seguir. Atualmente os AGV empregam outros métodos e sistemas designadamente de navegação laser, de navegação por meio de fitas magnéticas, de navegação inercial ou giroscópica e sistemas de visão.

2.2.3 – ROBÓTICA INDUSTRIAL

A robótica industrial, como já foi referido anteriormente, é uma área em desenvolvimento, com aplicação em tarefas de soldadura, pintura, montagem, paletização, movimentação de cargas pesadas e inspeção de produtos fabricados, sendo todas essas tarefas realizadas com elevada exatidão e rapidez. Cada processo possui um robô específico que se enquadra nos requisitos associados a cada uma das tarefas enunciadas. Existem vários exemplos de robôs (Figura 6), de acordo com a sua composição (elementos) e movimentos que consegue executar.

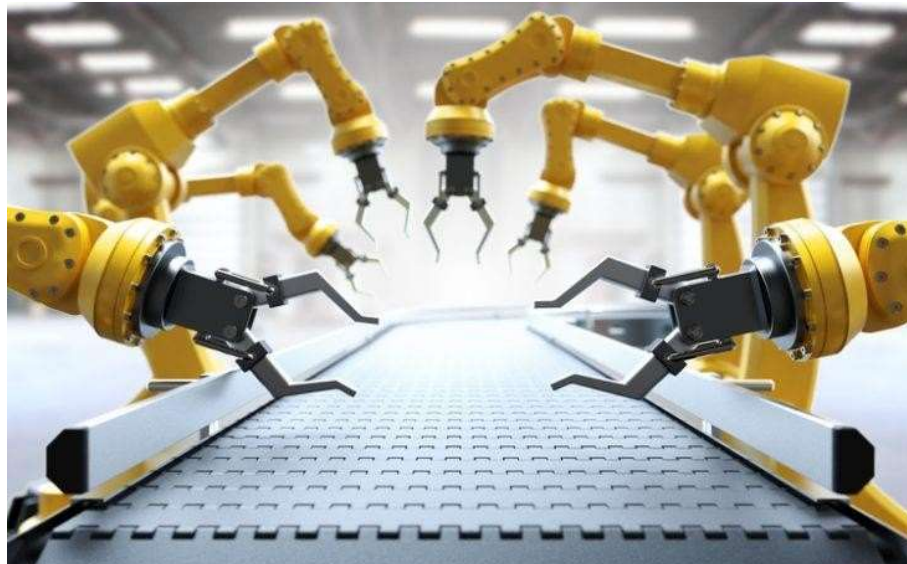


Figura 6 – Robótica Industrial. Fonte: (blog.kalatec).

2.2.3.1 – ELEMENTOS DO ROBÔ

O robô industrial é formado por elos e juntas, sensores, atuadores e um controlador, sendo possível a comparação a um braço do ser humano.

Os elos e juntas são partes mecânicas rígidas dos robôs (Figura 7) que permitem a definição da posição cartesiana em que se encontra.

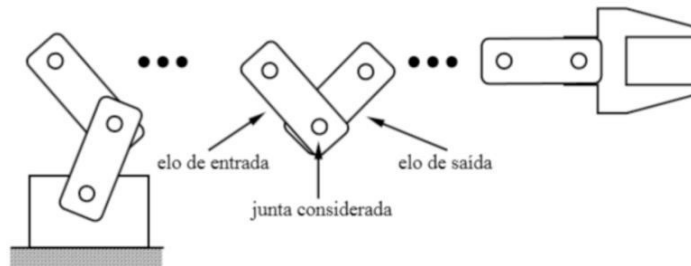


Figura 7 – Sequência de elos numa junta de um braço robótico. Fonte: (Groveer 1988)

Os sensores são responsáveis por obter informações do ambiente interno e externo em que se encontra o robô transforma-as em valores que possam ser compreendidos pelo controlador. Existem vários tipos de sensores, mas alguns deles designados por padrão, permitem medir o ângulo e a posição em que se encontra o robô e até mesmo a velocidade de deslocamento, sendo possível através destes dados calcular a cinemática.

Os atuadores são responsáveis por transformar a informação digital do controlador para valores no mundo real, através disso dão força aos movimentos.

O controlador é como o cérebro do robô, vai processar em tempo real, toda a informação proveniente dos sensores, de modo que depois envie informação através de atuadores. Este componente, representado na Figura 8, acaba por ser um computador que possui um *software* que controla o robô.



Figura 8 – Controlador de Robô. Fonte: (directindustry)

2.2.3.2 – TIPOS DE ROBÔ

Existem vários tipos de robôs industriais que podem ser relacionados a tarefas específicas, de modo que estas sejam desempenhadas com maior precisão (Simplício, 2016). Em 2004, a *International Federation of Robotics* (IFR) classificou os robôs com base na combinação das suas juntas e elos (International Federation of Robotics, 2021), resultando a seguinte classificação:

- Robôs Cilíndricos;
- Robôs Esféricos;
- Robôs Articulados;
- Robôs Delta.

Os robôs de coordenadas cartesianas possuem três articulações deslizantes conhecidas como *Prismatic-Prismatic-Prismatic* (PPP), como se pode observar na Figura 9. Estes robôs são caracterizados pela sua pequena área de trabalho e por não conseguirem movimentar múltiplos eixos em simultâneo. Normalmente são usados em tarefas de *pick and place*, dada à sua alta precisão.

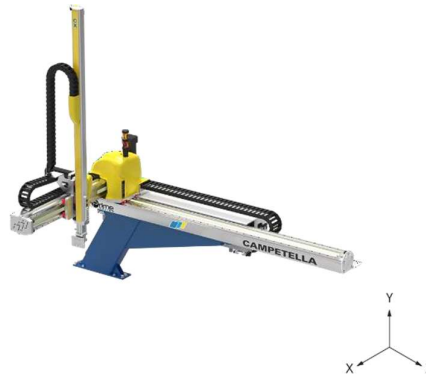


Figura 9 – Robô Cartesiano. Fonte: (campetella).

Os robôs cilíndricos são compostos por três juntas: uma rotacional e duas prismáticas *Revolute-Prismatic-Prismatic* (RPP). Através dessas juntas é possível combinar movimentos rotacionais e lineares, permitindo uma área de trabalho um pouco maior que o robô de coordenadas cartesianas.

Os robôs esféricos *Revolute-Revolute-Prismatic* (RRP) possuem uma área de trabalho ainda maior que os robôs cilíndricos, estes são dotados de dois movimentos rotacionais tornando-os mais complexos que o dos robôs cilíndricos (Figura 10).



Figura 10 – Robô Esférico. Fonte: (professor.ufop)

Os robôs articulados podem ser diferenciados em horizontais (RRP) e verticais (RRR). Quanto aos articulados horizontais também podem ser conhecidos como robôs *Selective Compliance Assembly Robot Arm* (SCARA), os quais são compostos por duas juntas rotacionais e uma linear, sendo a sua principal utilização em montagens de equipamentos. São conhecidos pela sua elevada velocidade durante os movimentos e a sua precisão. Já os articulados verticais possuem uma grande flexibilidade e podem ocupar uma grande área de trabalho, assim como

os horizontais. Estes robôs funcionam normalmente com atuadores elétricos, a sua grande desvantagem advém de possuírem uma cinemática complexa e um controlo difícil dos movimentos lineares. Contudo é visto como o modelo mais utilizado na indústria, realizando tarefas como manipulação de componentes, solda, pintura e automação de linhas de produção (Figura 11) (Bayer, 2011).



Figura 11 – Kuka Scara Robot. Fonte: (Kuka Robotics)

Os robôs delta (Figura 12), também chamados de robôs paralelos são geralmente usados em *pick and place*, tipicamente utilizados em sistemas que necessitem de alta velocidade e elevada precisão. Estes robôs podem ser usados quando se quer rapidez em espaços mais fechados, por norma são constituídos por três ou mais juntas do tipo lineares ou rotacionais.



Figura 12 – Robô delta Fanuc. Fonte: (Fanuc).

A escolha de um robô é feita sobretudo tendo em conta as suas dimensões, capacidade de carga e às vezes até dos movimentos. Algumas das características de um robô são:

- Número de eixos: Refere-se aos graus de liberdade que o robô tem, para alcançar um ponto no plano são necessários dois eixos, quando nos referimos a três eixos já é possível alcançar um ponto no espaço;
- Cinemática: Está relacionado com a distribuição dos elos e das juntas do robô, que vai originar os seus movimentos, podem ser articulados, cartesianos, paralelos e SCARA como foram acima referidos;
- Alcance: Trata-se da região onde o robô poderá trabalhar, logo depende essencialmente do comprimento dos seus elos;
- Capacidade de carga: Representa o valor que o robô consegue levantar, a carga que conseguirá suportar. Contudo temos de ter em atenção que o peso da garra também está presente neste valor, pelo que o tipo de garra e o peso da mesma, deve ser levado em conta para que seja possível ao robô aguentar o peso;
- Velocidade: A rapidez que um robô possui também é uma característica importante, porque é fundamental se quisermos ter movimentos rápidos que nos poupem tempo;
- Precisão: a precisão representa a posição mais correta em que o robô deve estar, ou seja, quanto maior for o seu rigor no que diz respeito à sua tarefa, melhor serão os seus movimentos. Contudo, também é possível calibrar sempre que necessário, para que o valor esteja o mais próximo do objetivo a cumprir.

2.2.3.3 – CINEMÁTICA

Cinemática significa movimento, ou seja, representa o estudo do movimento de um objeto juntamente com as forças que o originam. No âmbito da área da robótica, relaciona a ligação do espaço e do tempo entre os elos, juntas e componentes presentes no robô.

Cinemática Direta:

Trata-se de uma modelação do robô, em que se estabelece uma relação entre as juntas a partir da origem do robô (eixo inicial). Para isso utiliza-se por exemplo o método de Denavit-Hartenberg que permite para cada junta criar uma matriz, possuindo quatro parâmetros (distâncias e ângulos).

Através destas matrizes, se houver uma multiplicação entre todas as juntas, obtém-se uma matriz, uma relação entre o referencial, entre a junta inicial e final do robô. Através desta análise também é possível prevenir impactos entre os componentes do robô.

$$\mathbf{A}_0^1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ e } \mathbf{A}_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

$$\mathbf{A}_0^2 = \mathbf{A}_0^1 \mathbf{A}_1^2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_1 + a_2 C_{12} \\ S_{12} & C_{12} & 0 & a_1 S_1 + a_2 S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2)$$

Cinemática Inversa:

A Cinemática Inversa tem como objetivo movimentar o robô para uma posição, sendo necessário descobrir a posição em que as juntas têm de estar para a atingirem. Existe uma matriz inversa e vai-se substituindo os ângulos do robô até se chegar a uma dada coordenada.

Trata-se de um tipo de movimento inverso da cinemática direta, contudo existe o problema de haver várias posições da junta e algumas são impossíveis. Uma das possíveis soluções é entrar também com a velocidade do robô, ou seja, através da matriz jacobiana.

2.2.3.4 – TIPO DE MOVIMENTOS

- **PTP**

O PTP é um posicionamento aproximado, ou seja, este posicionamento não vai parar exatamente na posição configurada. O robô sai do ponto inicial e desloca-se para o ponto final e a sua trajetória irá executar o caminho mais rápido. Por esse motivo e pela rapidez do deslocamento nas juntas do robô faz com que este se desvie do seu caminho original.

É um movimento imprevisível para o ser humano já que não se conhece o caminho a tomar, por ser um movimento linear nas diferentes juntas do robô. Além disso pode ser perigoso em algumas situações, o desconhecimento da trajetória pode fazer com que o robô possa colidir contra objetos. Quanto mais pontos forem utilizados no caminho, mais este irá ser parecido com a trajetória original (Figura 13).



Figura 13 – Movimento PTP.

- **LIN**

O LIN é um movimento linear no espaço operacional. O robô ao executar este movimento irá efetuar o movimento mais curto, tipicamente é quando se efetua uma deslocação de um ponto para o outro e só se move num eixo. Desta maneira o caminho apresenta semelhanças ao original. Se colocarmos mais que dois pontos é possível executar um ângulo de 90 graus, com se pode observar na Figura 14.

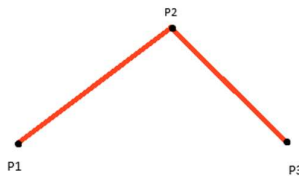


Figura 14 – Movimento Linear.

- **CIRC**

O CIRC é um movimento circular, como o próprio nome do comando indica, sendo necessário criar um ponto auxiliar para permitir que se consiga executar um arco e, para tal, é necessário ter três pontos. Apesar de ser um movimento bastante simples, este permite que se consiga executar um círculo através de quatro arcos (Figura 15).

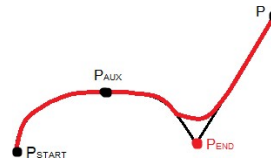


Figura 15 – Movimento Circular.

- **SPLINE**

O SPLINE é um tipo de movimento particularmente adequado para caminhos complexos e curvos, também é possível usar o LIN ou CIRC, no entanto, apresentam algumas desvantagens. O movimento SPLINE é mais versátil, usa-se para agrupar vários movimentos num único movimento. Desta forma permite a criação de vários tipos de movimento, o que irá permitir que o robô execute um caminho sempre mais preciso e mais suave (Figura 16).

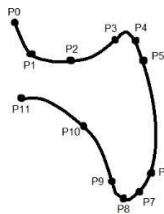


Figura 16 – Movimento Spline.

2.3 – Processamento de Imagem

2.3.1 – RELAÇÃO DA CÂMARA COM O OLHO HUMANO

Quando se pretende fazer uma análise de informação mais complexa, esta é de maior qualidade, mas menos eficiente no humano do que num sistema automatizado, uma vez que este está dotado de vários “sensores” que se encontram em constante comunicação. Um desses sensores, que vamos abordar de forma mais significativa, é o que se relaciona com a visão.

Os olhos estabelecem comunicação com outras partes do corpo, fazendo chegar a informação ao cérebro que, depois de ser processada, permite por exemplo saber a posição e a cor de um determinado objeto, ou qualquer outra informação. A Figura 17 apresenta as componentes do olho humano e, com base nestas, descreve o seu funcionamento. A entrada da luz na córnea, sendo esta uma lente fixa, vai direcionar a luz para o cristalino, de forma a focar o que se está a observar. A pupila encontra-se ao pé do cristalino e adapta-se consoante a intensidade da luz que está a receber. Esta permite controlar a quantidade de luz que entra no olho, ou seja, quando se está em locais muito escuros a pupila dilata-se, por outro lado, em sítios com muita luz, a pupila contrai-se e fecha, permitindo que em todas as situações seja possível captar a quantidade de luz certa para formar a imagem. A luz chega finalmente à retina que funciona como um sensor formado por bastonetes e cones, identificando a luz monocromática e as cores, respetivamente (Da MOTA, 2020).

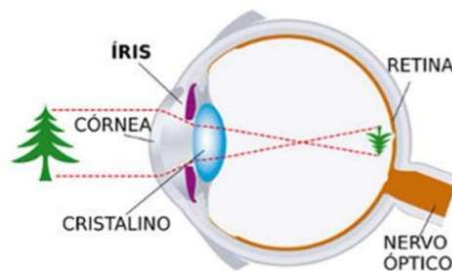


Figura 17 – Olho Humano. Fonte: (A Física do Olho Humano, 2020)

2.3.2 – CÂMARAS

Existem vários tipos de câmaras, mas as que têm uma maior utilização são as digitais. A primeira câmara foi concebida pela *Texas Instruments* em 1972 e, três anos depois, a *Cromemco Cyclops* criou a primeira câmara digital que passou a ser comercializada no mercado, permitindo a comunicação com um computador (FALCÃO, 2019). A principal diferença entre as câmaras digitais e analógicas é o facto de ser possível armazenar a informação ou imagem na memória ou em cartões de armazenamento, além de também permitir estabelecer comunicações, o que foi uma mais-valia para a sua utilização juntamente com sistemas na indústria (Moreno, 2016).

Atualmente as câmaras digitais são muito avançadas existindo empresas direcionadas para o desenvolvimento tecnológico de sistemas de visão e soluções de supervisão através de câmaras.

Através do processamento de imagem, juntamente com a câmara, foi possível expandir para outras aplicações, nomeadamente nas áreas da indústria (controlo de qualidade), dos transportes (SILVA, 2020) (veículos autónomos), da segurança (GONZAGA, 2017) (detecção de pessoas), da agricultura (SANTOS, 2020) (detecção de doenças nas plantas, pragas, etc.) e da saúde (SAMPAIO, 2018) (análise de tumores) entre outras.

No mercado já estão disponíveis vários tipos de soluções e sistemas de visão, existindo até empresas que oferecem *kits*, já com os objetivos pretendidos, como a *SICK Inc*, a *Allied Vision* e a *Cognex* que já possuem câmaras 2D e 3D com elevada capacidade de analisar detalhes a grandes velocidades de processamento (Figura 18).



Figura 18 – Cognex Fonte: (eletronor).

2.3.3 – VISÃO COMPUTACIONAL

A visão computacional é um tema recente, que está em desenvolvimento, tratando-se de uma ferramenta que permite aos robôs visualizar, por exemplo objetos, proporcionando um maior conhecimento do que existe ao seu redor (NEVES, 2022).

Este tipo de visão só foi possível graças ao trabalho de inúmeros investigadores, que estudaram de forma intensiva a aplicação de Redes Neurais Convulsionais (RNC) em tarefas reais. Estas redes recebem imagens, sendo caracterizadas no computador como matrizes de pixéis que se referem às cores da respetiva imagem que irá ser filtrada, possibilitando assim a extração das suas características. Após esta extração, utiliza-se a camada de rede neural para classificar a imagem obtida (Kuric, et al., 2022).

Desta forma, quando o sistema capta uma imagem, esta é examinada em cada pixel pela qual é constituída, pois correspondem aos menores elementos que constituem uma imagem e cada cor possui uma informação em forma de código diferente. O algoritmo, por sua vez, analisa

esses fatores e apreende, passando a reconhecer imagens. Sendo possível identificar os padrões nas imagens por meio da sua análise, possibilita assim, a comparação de imagens semelhantes (DA SILVA, VIDAL, & DA SILVA, 2020).

As *Convolutional Neural Network* (CNN) que correspondem a algoritmos de *deep learning* e que extraem as características dos *datasets* das imagens por meio da aplicação de filtros. Posteriormente utilizam as camadas de redes neurais para reconhecer e classificar (Kuric, et al., 2022).

São diversas camadas sequenciais, que analisam uma imagem. A primeira camada é responsável por detetar características mais básicas, como pixels, linhas verticais e horizontais. Já as camadas mais profundas possuem funções mais complexas, sendo capazes de identificar elementos como formatos e bordas. Assim, as camadas associadas geram uma rede neural que permite às máquinas identificar faces, edifícios, locais, algarismos, entre outros (Voulodimos, Doulamis, Doulamis, & Protopapadakis, 2018).

A função da visão computacional possui algumas características específicas:

- Aquisição da imagem: por meio de câmaras e sensores são captadas imagens, que podem ser bidimensionais ou tridimensionais. A máquina captura os pixels para identificar a intensidade da luz e as cores e também aspetos como profundidade;
- Pré-processamento: o sistema processa a imagem para assegurar a qualidade da análise. Isso envolve etapas como a redução de ruídos, ajuste de contraste, entre outros;
- Extração de características: reconhecimento das características matemáticas da imagem, como texturas, formatos e movimentos. Possibilitando assim, que as máquinas identifiquem bordas e cantos;
- Deteção e segmentação: trata-se da priorização da relevância de cada região da imagem para o processamento final;
- Processamento: é a etapa onde é realizada a classificação do objeto detetado.

Assim, a visão computacional, além de reproduzir determinados aspetos da visão humana também se torna mais precisa e exata, uma vez que identifica os detalhes que possam passar despercebidos por um observador.

Este tipo de visão tem diversas aplicações sendo uma delas o reconhecimento facial. Este processo para estar apto a funcionar basta que num primeiro momento seja fornecido com um número suficiente de imagens de rostos humanos, de seguida os algoritmos das redes neuronais conseguem aprender e detetar faces de forma autónoma (FERREIRA, 2021).

Existem diversas aplicações da visão computacional, sendo algumas delas:

- Na medicina, nomeadamente nas tarefas de diagnóstico por meio de extração de informações de radiografias, ultrassonografias, ressonâncias, entre outras (SOUZA, 2020);
- Veículos autónomos que são capazes de reconhecer a sinalização nas vias por meio do processamento das imagens das placas e sinais de trânsito em tempo real, obtendo assim maior assertividade que o olho/cérebro humano conseguiria. Estas soluções são usadas inclusivamente na exploração espacial como na *Mars Exploration Rover*, da NASA (Hammes, 2020);
- Na agricultura também é benéfico o uso desta tecnologia a qual auxilia na robotização da colheita (diferença entre frutos maduros e verdes), gestão de fertilizantes e na identificação de ervas daninhas.

Em conclusão, a visão computacional é uma tecnologia altamente disruptiva e tem desempenhado um importante papel na transformação digital, sobretudo no que se refere à indústria 4.0 (DE ALMEIDA, 2021). Seja no ramo alimentar, têxtil, automóvel, na saúde, a indústria tem vindo a beneficiar cada vez mais de sistemas inteligentes e com alta capacidade de processamento, ampliando as opções de utilização desta visão. Os sistemas de supervisão são úteis para reconhecer padrões e identificar falhas, sendo vistos como complemento ao trabalho humano, fornecendo assim, um grau de precisão e eficiência elevados para captar detalhes que poderiam passar despercebidos e causar impactos negativos na cadeia de produção. Desta forma, a visão computacional torna-se crucial para o futuro e para as atividades quotidianas (RECALCATI & MINGORI, 2020).

2.4 – Inteligência Artificial

A Inteligência Artificial (IA) teve início após a Segunda Guerra Mundial e atualmente já se encontra presente em várias áreas (GOMES, 2010). Apesar de ser um conceito complexo, é possível ter uma ideia sobre a Inteligência Artificial/Aumentada admitindo:

- 1ª – Sistemas que pensam como seres humanos: “O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal” (HAUGELAND, 1985);
- 2ª – Sistemas que atuam como seres humanos: “A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (KURZWEIL, 1990);
- 3ª – Sistemas que pensam racionalmente: “O estudo das faculdades mentais pela utilização de modelos computacionais.” (CHARNIAK & MCDERMOTT, 1985);
- 4ª – Sistemas que atuam racionalmente: “A Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (POOLE, MACKWORTH, & GOEBEL, 1998).

A primeira e terceira linhas de pensamento referem-se ao processo de raciocínio e pensamento, em contrapartida a segunda e quarta referem-se ao comportamento.

2.4.1 – TIPOS DE INTELIGÊNCIA ARTIFICIAL

Segundo (Ludermir, 2021), existem três tipos de IA: a IA Focada, a IA Generalizada e a IA Superinteligente.

A IA Focada, denominada também de IA fraca, foi criada com o objetivo de resolver problemas numa área ou num problema específico. Os sistemas de armazenamento possuem uma capacidade enorme para guardar informação e os algoritmos conseguem efetuar tarefas complexas, mas sempre focadas no objetivo para o qual foi criada.

A IA Generalizada, ou IA forte, foi criada para que consiga ter as mesmas capacidades dos humanos em várias tarefas, como por exemplo visão computacional.

A IA Superinteligente, refere-se a uma IA que seja capaz de ultrapassar as capacidades dos humanos em qualquer tipo de tarefas, mas atualmente ainda não existe este tipo de IA.

2.4.2 – IMPACTOS DA INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial apresenta vantagens e desvantagens para o ser humano, ela pode ajudar nas tarefas tornando-as automáticas, permitindo assim que os humanos lidem com outro tipo de tarefas, levando assim a uma melhor gestão de recursos humanos (Ludermir, 2021).

A utilização de IA proporcionou vários benefícios, como por exemplo melhorias no serviço de saúde, deteção de fraudes, entretenimento, casas inteligentes, veículos autónomos, permitindo que as pessoas possam ter uma vida mais simples. Contudo também apresenta aspetos negativos, por exemplo a invasão da privacidade das pessoas, a utilização de informações das suas vidas sem o seu consentimento, para que a IA atinja os seus propósitos.

2.4.3 – APRENDIZAGEM PROFUNDA

Antes de entrarmos propriamente no conceito de Aprendizagem Profunda, é importante saber e perceber a origem desta e a ligação que a IA tem com a Aprendizagem de Máquina.

Embora, as recentes notícias e opiniões por parte dos meios de comunicação social sobre a valorização da indústria no que toca à IA, este conceito já é conhecido desde os anos 50, aquando da intervenção dos pioneiros do campo da ciência da computação que vieram pôr em cima da mesa, a possibilidade de os computadores pensarem como os seres humanos. Atualmente, o conceito de IA apresenta variadas definições, no entanto, a que demonstra ser a mais adequada é “o esforço para automatizar tarefas intelectuais normalmente realizadas por humanos” (Chollet, 2018).

Neste sentido, é importante interiorizar que a IA é um conceito geral que visa inúmeras abordagens, sendo que a Aprendizagem de Máquina é uma das demais que acaba por englobar a Aprendizagem Profunda, como podemos observar na Figura 19.

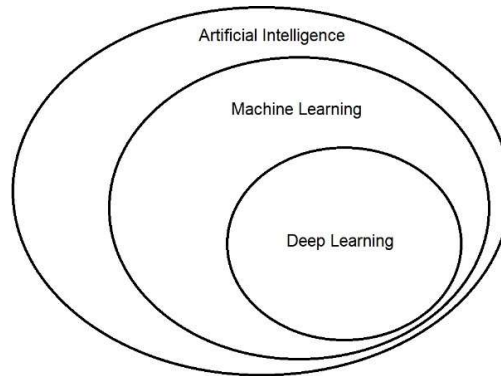


Figura 19 – Relação entre diferentes disciplinas da Inteligência Artificial.

Machine Learning

Na década de 1990, a Aprendizagem de Máquina começou a ganhar visibilidade, tornando-se assim num subcampo mais popular no que toca ao conceito de IA, devido ao aumento da disponibilidade do conjunto de determinados dados e da rapidez do *hardware* (Chollet, 2018).

Assim sendo, este conceito de Aprendizagem de Máquina veio beneficiar a adoção de um novo paradigma que difere da IA original, que era utilizado maioritariamente na década de 1980 (Chollet, 2018).

Desta forma, enquanto nesta os dados e as regras eram introduzidas manualmente de maneira a produzir respostas, a Aprendizagem de Máquina já tinha estipulado a introdução destes pelos computadores, de maneira que se obtivesse as regras que seriam disponibilizados pela máquina (COPELAND, 2016).

Por tudo isto, os algoritmos de Aprendizagem de Máquina desenvolvem o processo intitulado como “treino”. Este visa a apresentação de exemplares relevantes para uma dada tarefa, de modo a encontrar uma estrutura estatística nos exemplos que, talvez, permitam que o algoritmo venha a criar regras para automatizar a tarefa (Dickson, 2019).

Deep Learning

Este conceito de Aprendizagem Profunda é conhecido por ser um subcampo da Aprendizagem de Máquina, mas, de uma forma mais precisa, é importante perceber e interiorizar que a Aprendizagem Profunda usa como meio de trabalho as Redes Neurais que fazem parte da Aprendizagem de Máquina, como se pode observar na Figura 20.

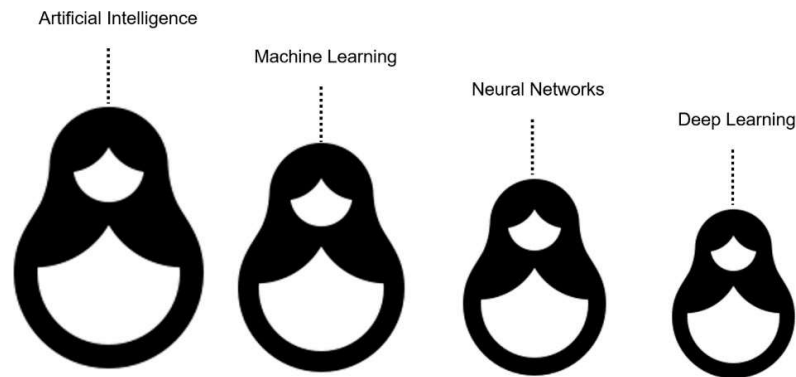


Figura 20 – Relação existente entre diferentes conceitos de IA. Fonte: (Kavlakoglu, 2020).

Perante este segmento do que é a Aprendizagem Profunda e do que dela se retrai surge a seguinte questão “Em que consistem as Redes Neurais?”. De forma a responder a esta pergunta é imprescindível recuar aos tempos em que Frank Rosenblatt dá a conhecer o conceito de Perceptron (Rosenblatt, 1958). Neste sentido, este conceito apresenta-se como sendo um modelo computacional relativamente ao funcionamento de um neurónio biológico. Contudo, a ideia original do autor era ensinar uma máquina a reconhecer imagens. Assim sendo, este veio a recorrer ao Perceptron, de modo a atingir os seus objetivos iniciais, utilizando determinadas características de uma dada imagem, isto é, utilizou os valores de intensidade de cada pixel associando-os a cada uma dessas características um “peso”.

A Aprendizagem Profunda distingue-se das Redes Neurais pela quantidade de camadas de neurónios ou profundidade utilizadas, caso existam mais de três camadas de nós/neurónios então pode-se considerar como sendo um algoritmo de Aprendizagem Profunda (Kavlakoglu, 2020). Por outro lado, a Aprendizagem Profunda pode-se distinguir ainda por outros fatores da Aprendizagem de Máquina, através da extração de características de forma automática no que diz respeito às primeiras camadas dos modelos, já a Aprendizagem de Máquina precisa de uma extração manual dessas respetivas características (Coursera, 2022).

Por outras palavras, a Aprendizagem Profunda faz uma aprendizagem de representações dos respetivos dados através de múltiplas etapas, porém é importante ter em linha de conta como se processa toda essa aprendizagem. Além disto, esta aprendizagem associa os pesos a cada uma das camadas da rede o que vem a traduzir-se em valores aleatórios. Estes são utilizados nas respetivas camadas do modelo traduzindo-se em resultados recebidos como *input*. Assim sendo, é realizada uma previsão dos dados transformados, permitindo fazer uma comparação da mesma com a anotação que foi feita previamente sobre os dados. Posto isto, é feito um cálculo que nos dá um valor de perda (*loss score*) associada à comparação, tendo em vista o objetivo do modelo. Este valor de perda alude à distância a que a previsão está do objetivo. E, quanto mais longe estivermos do objetivo maior será a distância, ou seja, o treino tem como objetivo tentar minimizar este valor de perda. Tendo em conta este valor, os pesos atribuídos às camadas são atualizados com recurso ao algoritmo de *Backpropagation*, para que na próxima previsão os resultados estejam de acordo com as anotações previamente estabelecidas. Este processo é repetido com um novo conjunto de dados com os pesos atualizados, para que o valor de perda venha a atingir o valor mínimo possível (Silva, 2022).

Embora este conceito de Aprendizagem Profunda já tenha sido concetualizado antes do ano de 2010, só agora é que se veio a intensificar a adesão deste conceito. O crescimento da utilização dos algoritmos de Aprendizagem Profunda deveu-se basicamente à evolução do *hardware*, utilizando os processadores gráficos de alto desempenho que foram desenvolvidos para jogos e à disponibilização de grandes quantidades de dados, através da evolução do uso da *Internet* por parte não só das massas, como também das empresas (Chollet, 2018).

CAPÍTULO 3: GUIA RÁPIDO SIMULADOR KUKA

O programa KUKA Sim Pro apresenta-se como um programa de simulação que foi desenvolvido para programação *offline* dos robôs KUKA. Este programa permite que sejam adicionados robôs num ambiente fabril com o objetivo de serem programados e desenvolvidas linhas de produção, garantindo numa fase de planeamento não apenas uma previsão do novo contexto da área industrial, mas também por exemplo avaliar economicamente o custo/benefício do esforço de engenharia.

Assim, é importante salientar o bom funcionamento do KUKA Sim Pro, sendo este mencionado ao longo do relatório, devido ao facto de ter sido uma ferramenta bastante utilizada durante o período do projeto. Estes tipos de simuladores são bastante importantes, pois permitem ter uma ideia de como ficará o aspeto de uma linha de produção numa fábrica.

Neste sentido, este capítulo tem como principal objetivo demonstrar as vantagens do seu uso, tratando-se de um guia que permite a sua utilização e, por outro lado, dar a conhecer as suas principais funcionalidades.

3.1 – Novo Programa

Na criação de um novo programa inicia-se o KUKA Sim Pro para começar um programa de início, conforme se observa na Figura 21. Para a execução da aplicação é necessário possuir uma licença de utilização para o KUKA Sim Pro.

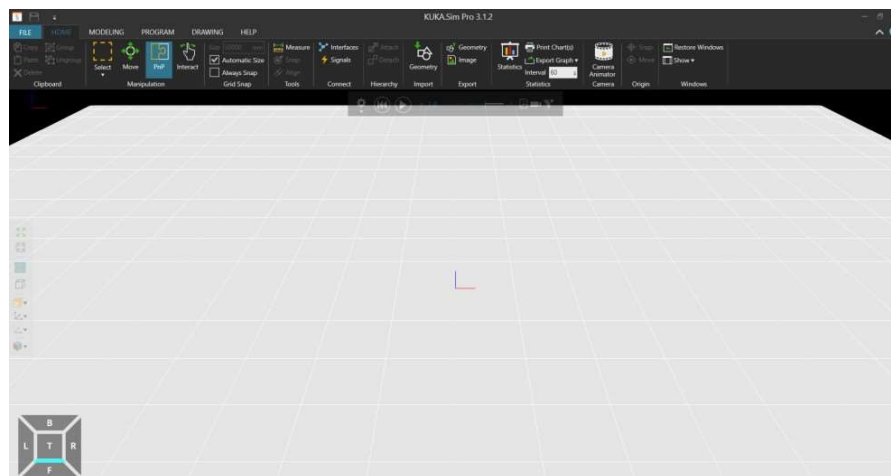


Figura 21 – Ambiente Inicial do programa.

A criação de um objeto no simulador é uma tarefa complexa, contudo acessível caso se disponha de conhecimento e experiência prévia no referido simulador. Para o efeito, seleciona-se a opção “Modeling” (Figura 22) e escolhe-se a opção “New” para criar um componente que pode ser um cubo, uma esfera, etc.

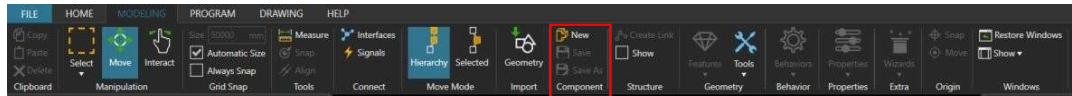


Figura 22 – Menu Modeling.

Após ser criado um componente aparece um ponto com vários eixos que será a origem do nosso componente (Figura 23).

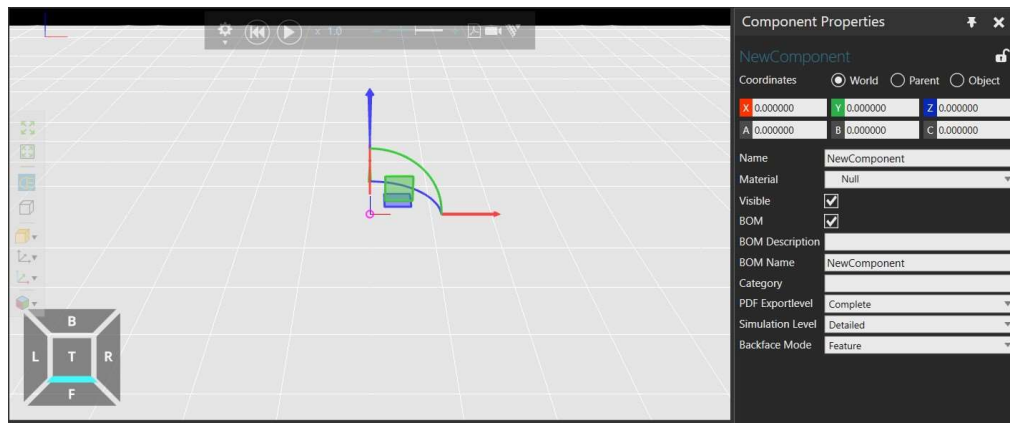


Figura 23 – Eixos do componente.

Depois de ter uma origem que irá ser o ponto de referência do objeto, é necessário adicionar algumas “features”, que neste exemplo será um cubo. Para adicionar um cubo é necessário ir a “features”, posteriormente abre-se um submenu e de seguida seleciona-se a opção “box”, como está demonstrado na Figura 24.

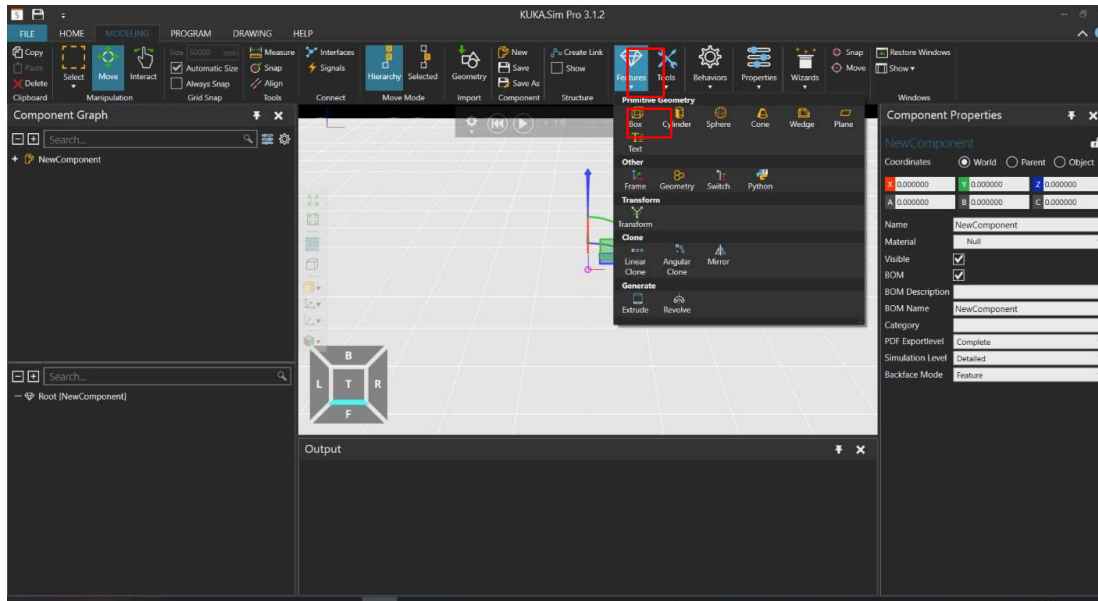


Figura 24 – Menu para adicionar objetos.

Após selecionar é criado um cubo laranja com determinadas dimensões. Podemos alterar as medidas apresentadas numa janela do lado direito.

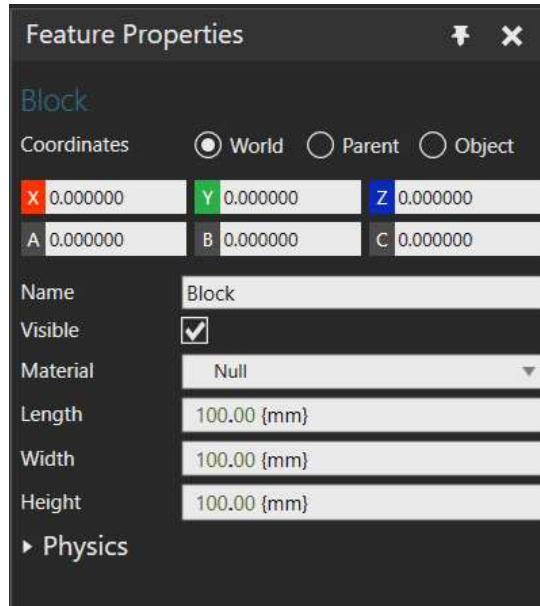


Figura 25 – Propriedades do bloco.

Logo após se colocarem as medidas do cubo basta fechar essa janela, sendo neste caso aconselhado usar os valores em “mm” (Figura 25). Caso se pretenda usar outras unidades também é possível fazê-lo. Após a criação do cubo se formos ao menu “Home” irá aparecer um cubo como o que se encontra na Figura 26.

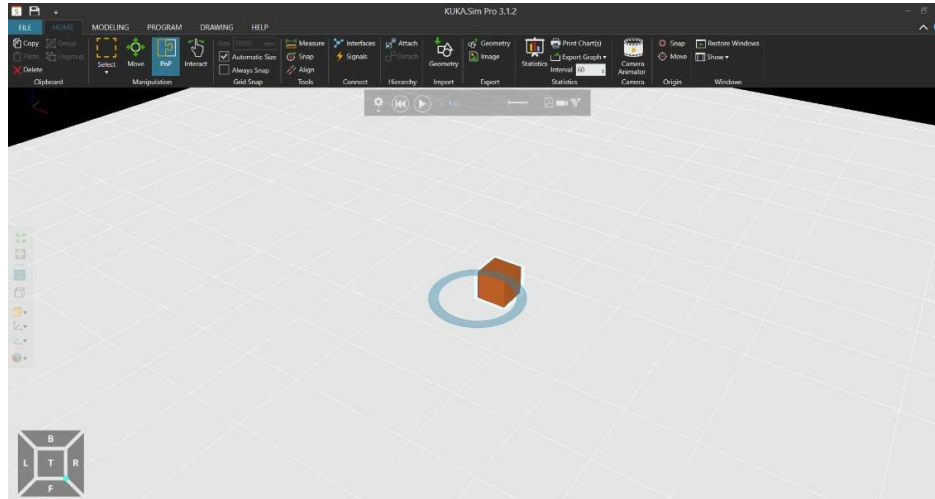


Figura 26 – Cubo Criado.

Na movimentação do objeto é necessário haver alguma atenção, pois pode-se estar a movimentar o cubo mantendo-se o ponto de origem na mesma posição. Na movimentação é necessário pressionar duas vezes no objeto para selecionar todo o componente, de modo a não acontecer o que se vê na Figura 27.

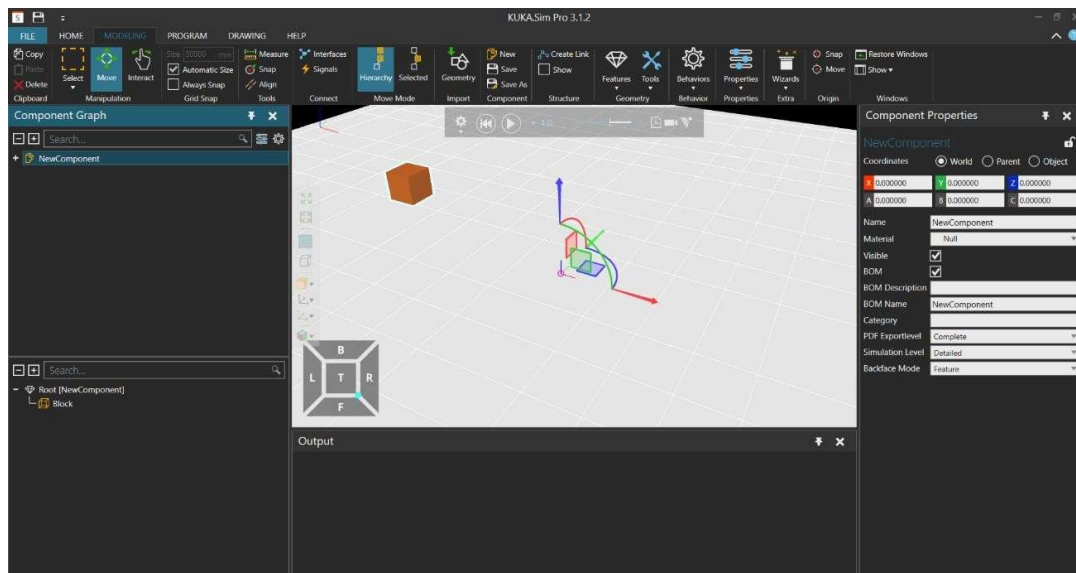


Figura 27 – Referência descentralizada.

Como se pode observar na Figura 28, o centro do cubo, isto é, a origem do objeto encontra-se noutra ponto diferente, isto pode dar problemas no futuro com as movimentações do objeto, portanto é necessário que esta origem coincida sempre com o cubo.

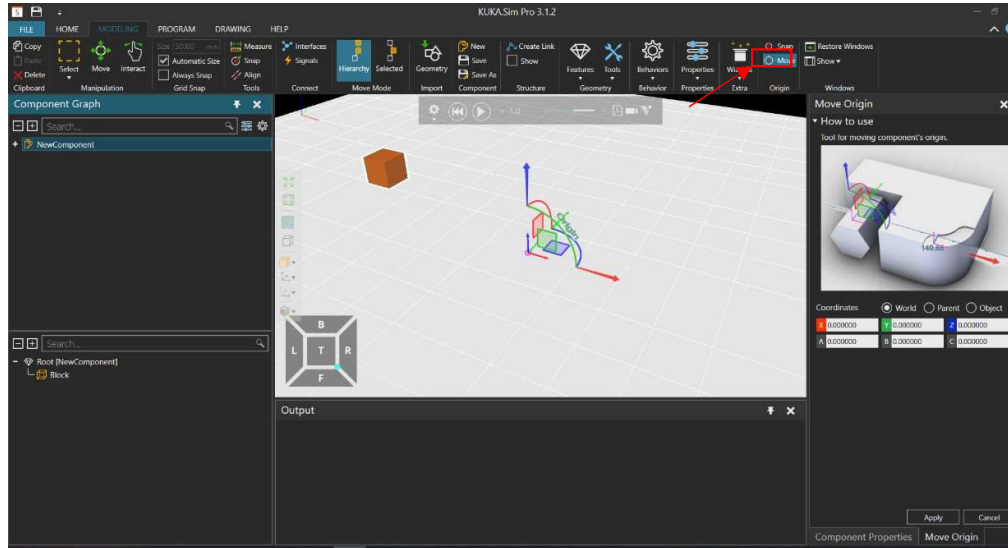


Figura 28 – Função Move.

Para resolver o problema descrito sem apagar o objeto, basta ir a “modeling” e selecionar “move”. Vai aparecer uma janela do lado direito (Figura 29), seleciona-se a origem do referencial e move-se para um vértice do cubo, como se apresenta na imagem seguinte.

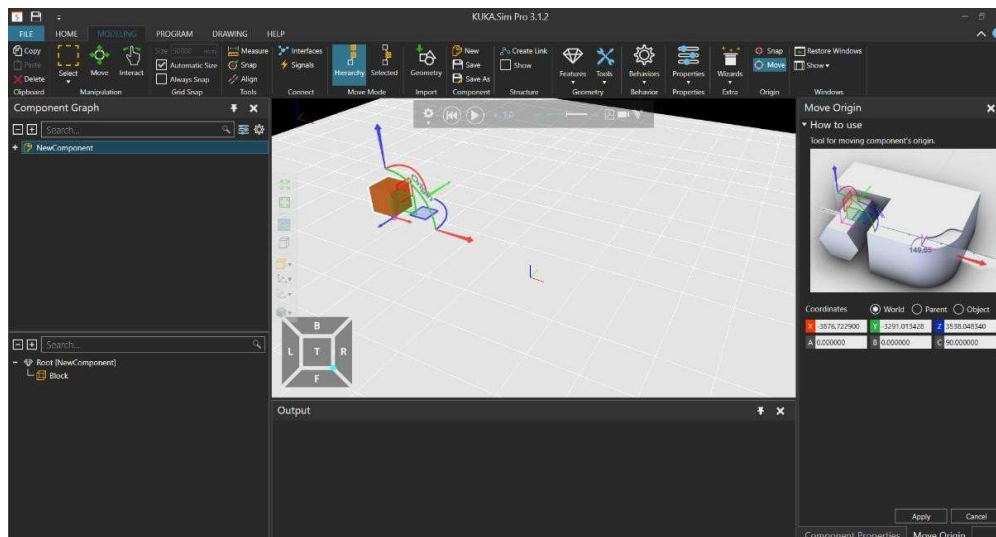


Figura 29 – Processo de centralização da referência.

Depois de estar situada na posição que queremos basta selecionar “*apply*” que se encontra no lado direito, ficando o cubo a coincidir com a origem do referencial (Figura 30) e, no futuro, já não vão existir problemas com a movimentação do cubo.

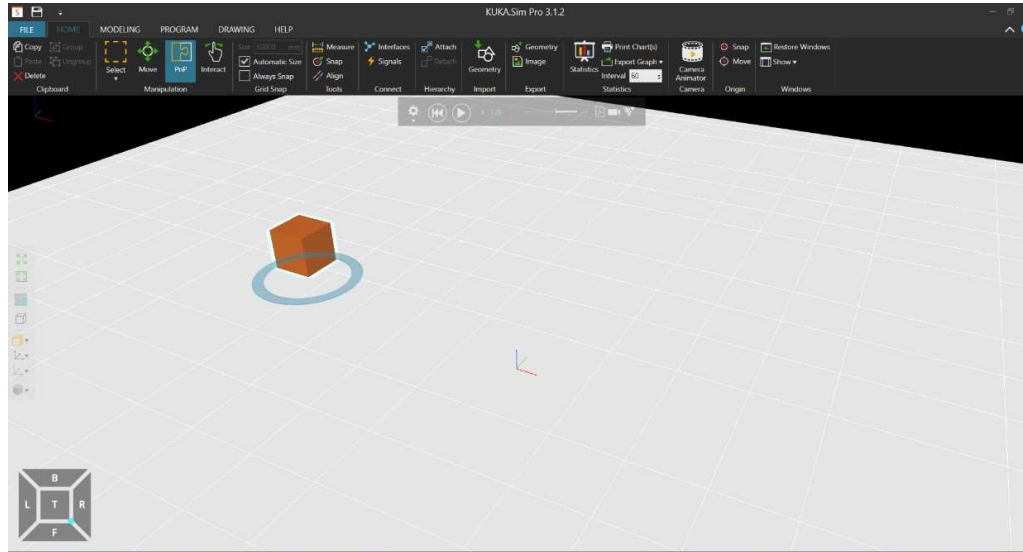


Figura 30 – Cubo com referência centralizada.

Se voltar para o menu “*Home*” e selecionar o “*PnP*”, aparece o que se visualiza na imagem acima, isto significa que o objeto tem o seu ponto de origem no objeto.

3.2 – Save Programa

Após se ter criado o cubo é importante guardar o programa, para isso basta ir a “*File*” (Figura 31).

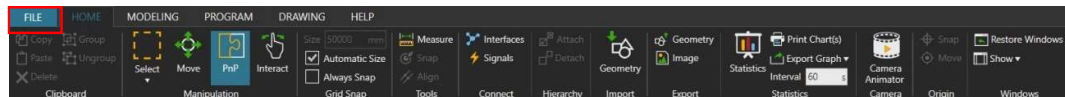


Figura 31 – Menu Home.

De seguida, será aberto um submenu (Figura 32) com várias opções, sendo uma delas “*Save*”. Ao escolher essa opção o programa irá ser guardado e é importante ter várias gravações separadas, para o caso de ocorrer algum problema existir sempre um *backup*.

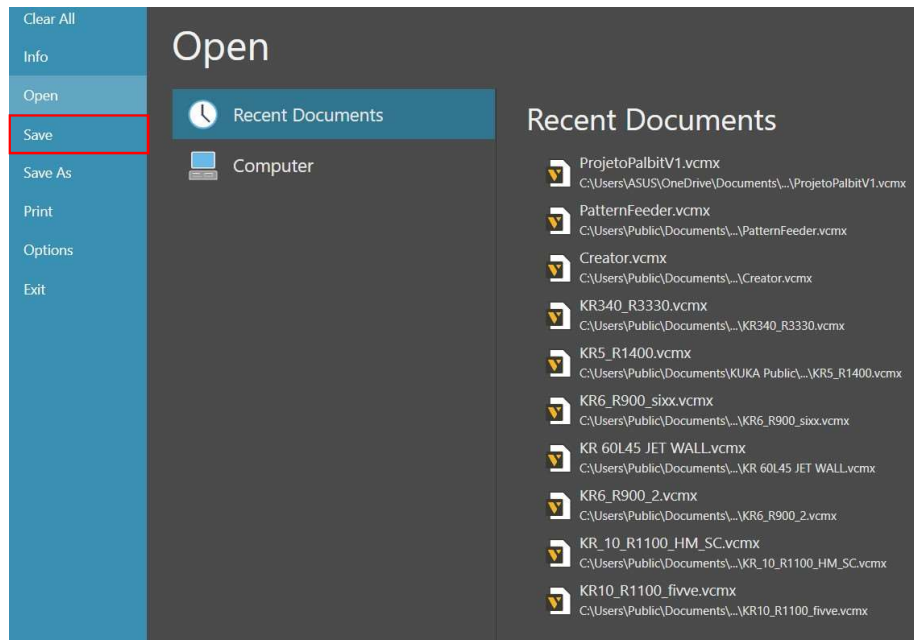


Figura 32 – Menu File.

3.3 – Componentes

3.3.1 – SAVE COMPONENTS

Nesta secção pretende-se demonstrar como se podem guardar componentes criados e adicioná-los à base de dados do programa, para serem importados posteriormente para o projeto. Esta ferramenta é uma mais-valia para criar componentes que não existam na base de dados que já vem com o simulador, pois caso seja necessário usá-los outra vez é só importar. Para guardar um componente é necessário seleccionar duas vezes o objeto e de seguida ir ao “*Modeling*” e escolher a opção “*Save*” ou “*Save As*” (Figura 33).

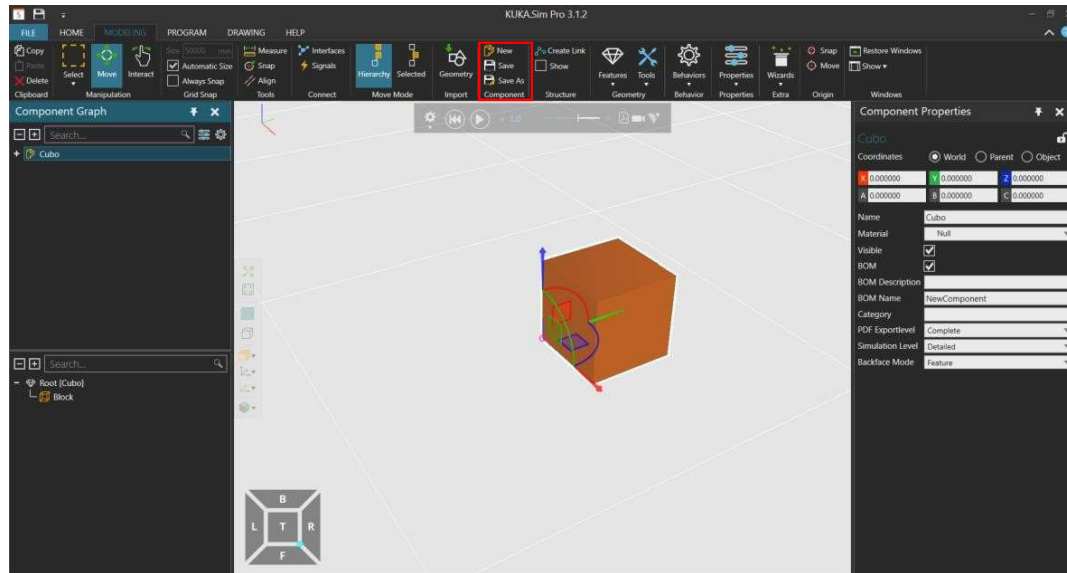


Figura 33 – Guardar componente.

3.3.2 – CRIAÇÃO DE GARRA

Sendo o simulador uma ferramenta que permite criar componentes é sempre importante idealizar uma garra para diferentes situações. Nesta secção irá ser descrita a criação da garra de um robô, com um aspeto semelhante à da Figura 34.

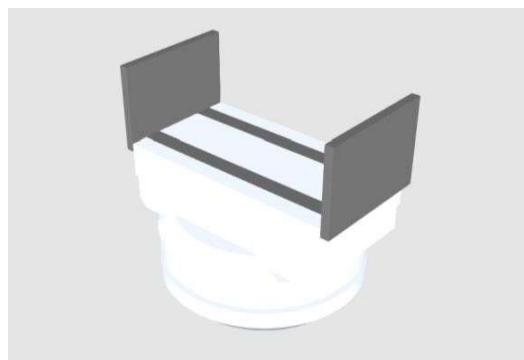


Figura 34 – Garra criada.

O primeiro passo para criar uma garra, não precisa de ser específica, mas só o facto de abrir e fechar é suficiente para essa criação. Após isso seleciona-se uma das partes móveis da garra e cria-se um “Link”. Este processo pode ser efetuado de duas formas, uma delas é selecionar a parte móvel e ir ao menu “Modeling”, clicar em “Tools” e de seguida em “Link” (Figura 35).

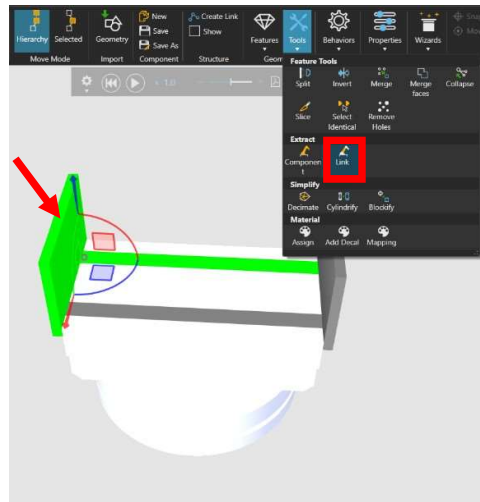


Figura 35 – Primeira forma de Criar um link.

Outra forma é clicar no botão direito do rato, seleccionar a parte que interessa e depois seleccionar “*Extract*” e depois “*Link*”, como é demonstrado na Figura 36.

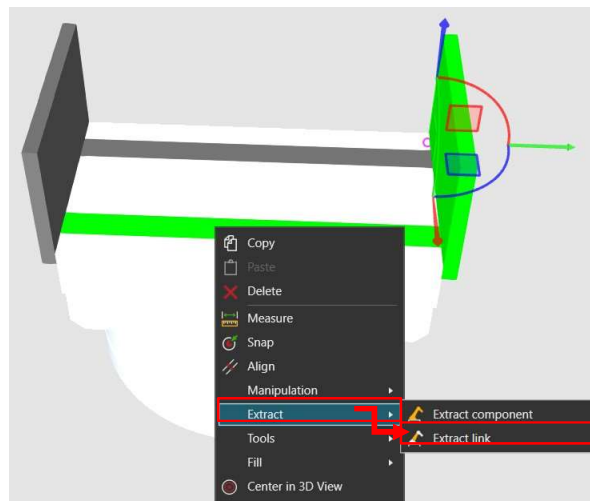


Figura 36 – Segunda forma de Criar um link.

Depois de criado o “*Link*” repete-se o processo para a outra parte da garra, desta forma irão ser criados dois “*Links*” que podem agora ser amovíveis após a sua configuração. Na configuração deles é necessário seleccionar algumas opções para permitir a sua interação.

Como se pode observar na Figura 37, a junta encontra-se “Fixed”, ou seja, não executa qualquer tipo de movimento, para se movimentar é necessário escolher:

- Rotational: para executar uma rotação;
- Translational: para executar uma translação;
- Rotational Follower: para executar uma rotação acompanhando outra junta;
- Translational Follower: para executar uma translação acompanhando outra junta.

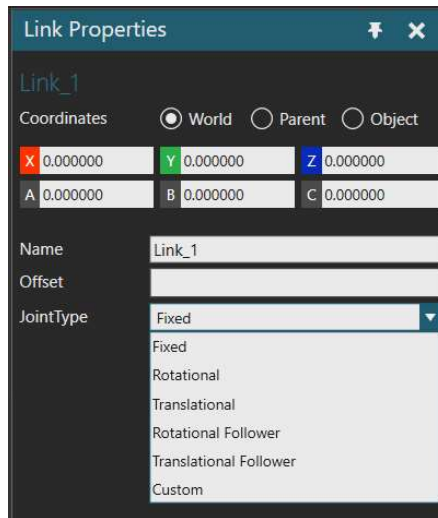


Figura 37 – Propriedades do Link.

Nesta situação em que se pretende uma garra que permita agarrar algum objeto (abrir e a fechar) vai-se optar pela “Translational”, após a sua seleção será pedido para selecionar o seu sentido. O sentido que se deve escolher é “+Y” para se movimentar para a direita, tal como o movimento da Figura 38, devendo ter uma configuração igual à que se optou no exemplo.

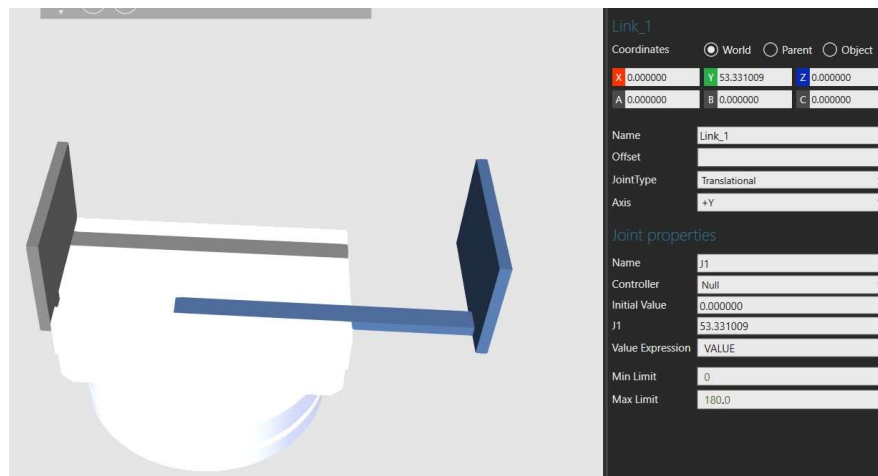


Figura 38 – Configuração do Link.

O valor do “J1” refere-se ao valor da posição em que se encontra, também se observa que o valor mínimo é 0 para impedir que a parte da garra se mova demais para a esquerda. Após este passo estar concluído vai-se adicionar um controlador, para isso basta ir a “*Controller*” e adicionar um novo servo como na Figura 39.

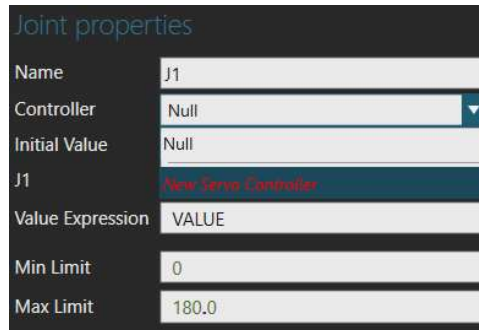


Figura 39 – Adicionar Servo.

Depois de se criar o “*Servo Controller*”, o próximo passo é configurar o “*Link 2*” (Figura 40), seleciona-se e coloca-se a configuração como aparece em baixo.

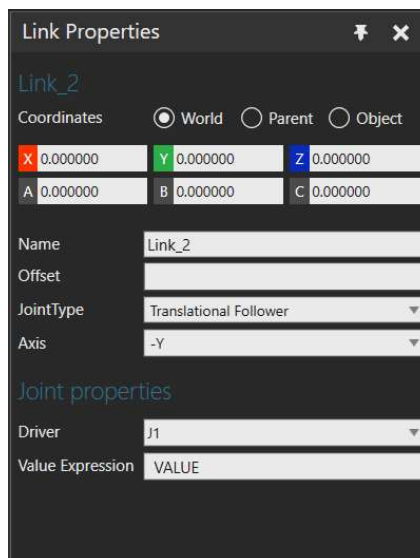


Figura 40 – Configuração do Link 2.

Neste caso já se insere uma junta do tipo “*Translational Follower*”, para que se mova quando a outra alterar também a sua posição. Em “*Driver*” seleciona-se “J1” pelo facto de ter um “*servo controller*” para ser controlado por outra junta, desta forma garantimos simetria nos movimentos.

Neste ponto a garra já se encontra operacional e é possível interagir com as juntas para observar o comportamento delas, mas o objetivo é que esta seja usada num robô, para isso tem que se identificar como ferramenta e adicionar sensores para que se feche a garra com saídas do robô. Começa-se por selecionar a garra e de seguida vai-se a a “Wizards” (Figura 41) e adiciona-se “End Effector”, que permite adicionar sensores e fazer com que a garra seja identificada como tal pelo robô.

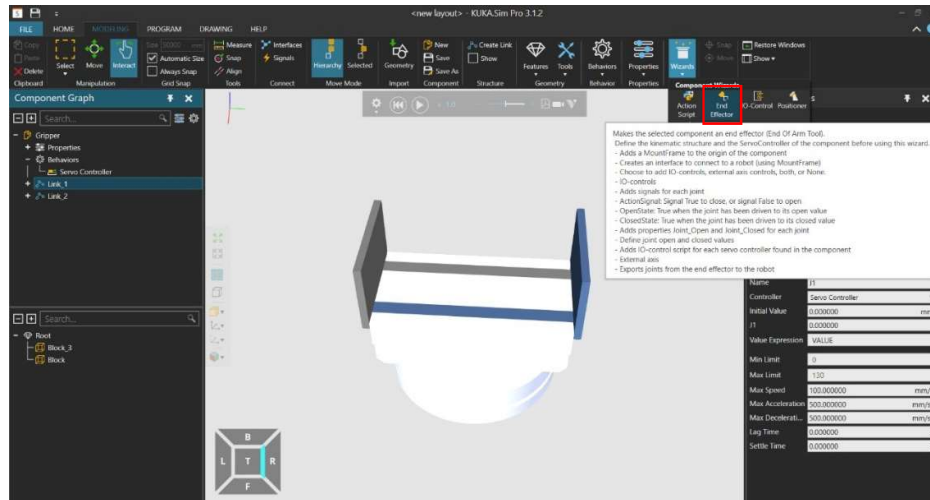


Figura 41 – Submenu Wizards.

Após executar o passo referido anteriormente, vai ser aberto um menu igual ao da Figura 42 que tem de ser configurado.

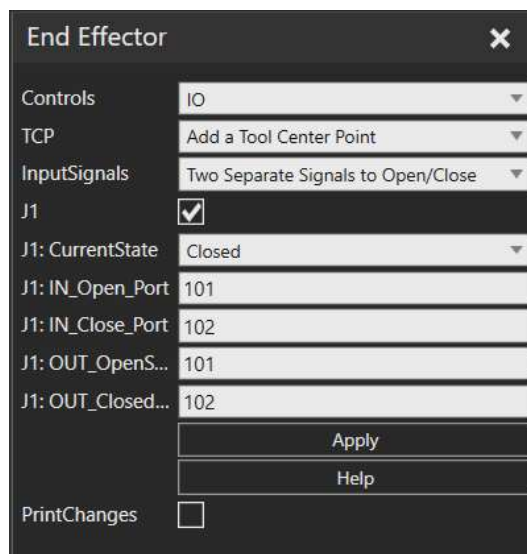


Figura 42 – Menu End Effector.

Neste caso é importante analisar o estado da ferramenta, se esta se encontra aberta ou fechada na sua forma inicial, porque se não se tiver isto em atenção a ferramenta pode estar aberta e dar sinal que está fechada.

Depois de concluir a configuração, se abrir o menu dos “signals” e seleccionar o “gripper” irá ver os quatro sensores que possui. Analisando a Figura 43 é possível verificar que dois deles servem para dar a ordem de abrir e fechar e os outros dois para dar o sinal de que a garra se encontra aberta ou fechada.

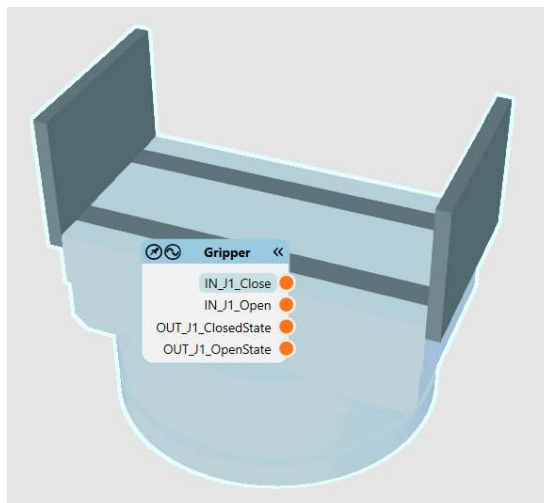


Figura 43 – Garra com os sensores.

Agora que os sensores foram criados, o próximo passo é adicionar o eixo referencial onde a peça vai ficar na garra. Para isso é necessário seleccioná-la e no “ToolContainer” adicionar uma *frame*, como está explícito na Figura 44.

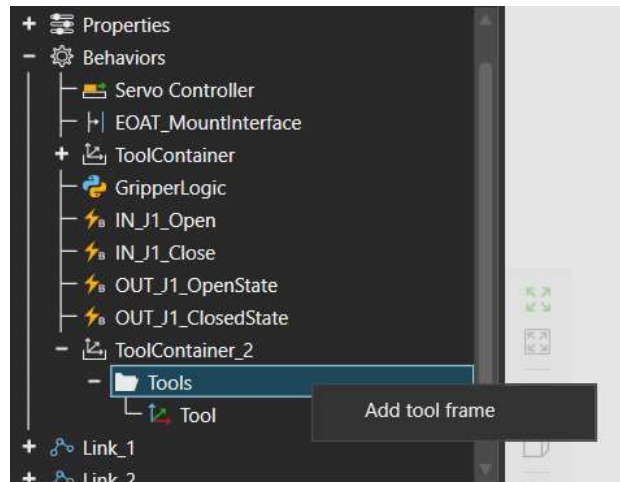


Figura 44 – Adicionar eixo de referência.

Depois de criada, a *frame* terá um aspeto idêntico ao da Figura 45, mas não basta adicioná-la, é necessário movê-la para o local onde se quer que a peça seja agarrada.

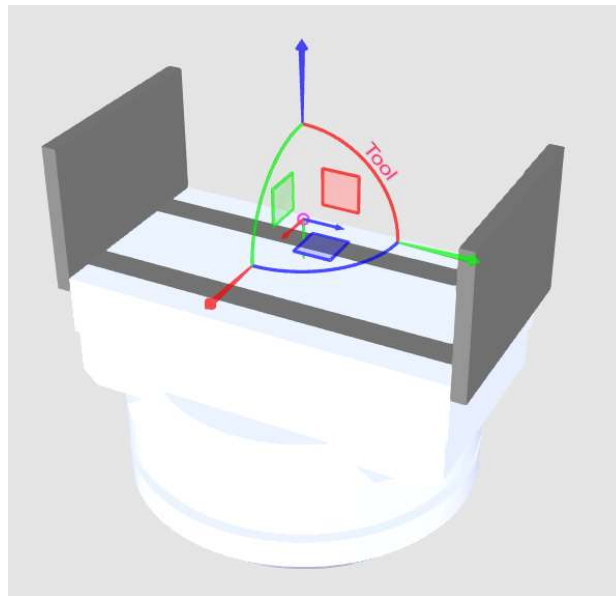


Figura 45 – Garra com eixo de referência.

Após este passo estar concluído é preciso ir a “EOAT_MountInterface” e adicionar o referencial à garra para que este seja usado nos movimentos do robô. Em “Selection and Fields” e, onde diz “Add new field”, adiciona-se um “ToolExport”, depois é só adicionar o nosso “ToolContainer” no “ToolList” (Figura 46).

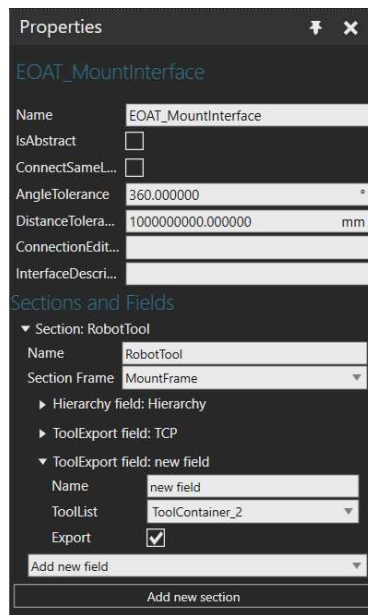


Figura 46 – Propriedades do EOAT_MountInterface.

Agora que o último passo foi configurado basta acoplar a garra ao robô e está pronta a ser usada. Neste caso criou-se este tipo de garra, porém é possível optar por outros tipos, bastando criar e depois configurar os movimentos e sensores que estão presentes.

3.3.3 – CRIAÇÃO CREATOR

Um *creator* tem como objetivo criar um objeto através de outro, isto é, se houver por exemplo uma mesa é possível fazer com que apareça em cima da mesa um objeto em determinado período. Este objeto não se encontra invisível apenas está a ser chamado de uma base de dados para aparecer naquela mesa, neste caso vai ser usado um retângulo para criar um objeto em cima dele, por exemplo uma caixa. O primeiro passo é seleccionar o retângulo onde se pretenda que apareça o objeto e clicar em “*behaviors*” no menu “*Modeling*”, isto vai abrir um submenu em que se tem de ir à secção “*Material Flow*” e clicar em “*Component Creator*” (Figura 47), para adicionar o comportamento de criação.

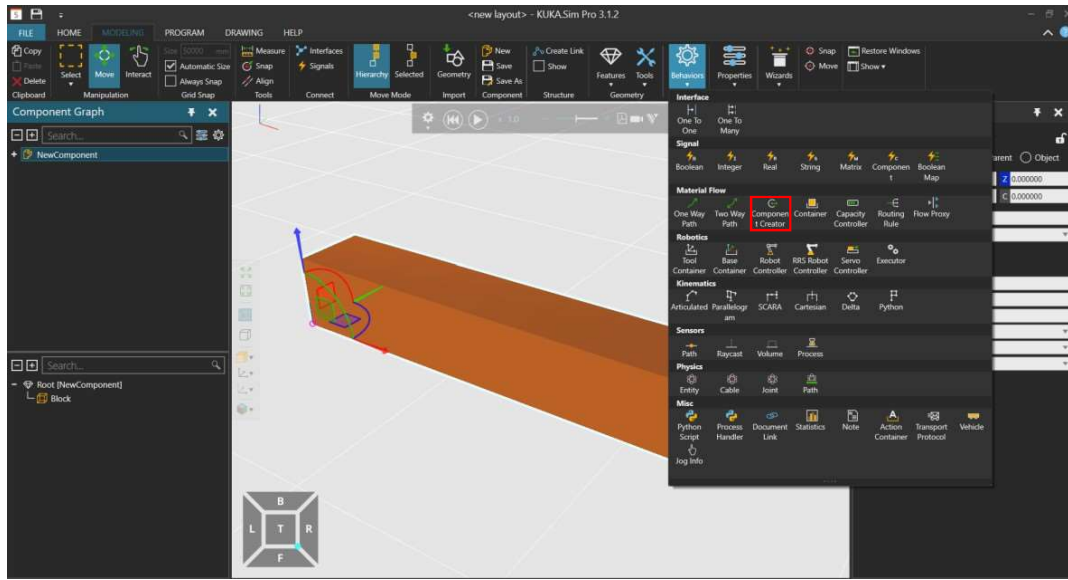


Figura 47 – Menu Behaviors.

Depois desta ação, vai aparecer uma janela do lado direito, igual à Figura 48, para que seja possível colocar as configurações a usar.

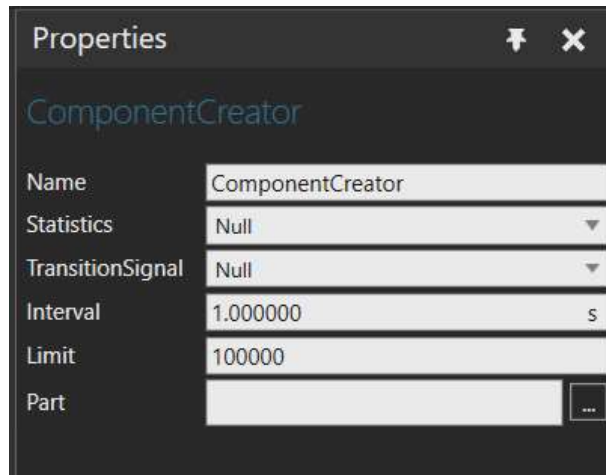


Figura 48 – Propriedades do ComponentCreator.

Nestas configurações o “Interval” é o tempo entre as peças criadas, ou seja, neste caso vai criar peças de 1 em 1 segundo, sendo o “Limit” o número máximo de peças que irá gerar, que neste caso são 100000. Nesta configuração também é necessário definir o “Part”, isto é, o objeto que irá aparecer no tapete. Após adicionar o objeto a ser criado, ele ainda não vai aparecer porque é necessário um “Container” (Figura 49), para que a peça fique visível.



Figura 49 – Adicionar Container.

Quando o “Container” já estiver adicionado é necessário relacioná-lo com o “ComponentCreator”, no “Output” tal como aparece na Figura 50. Depois de realizada esta operação, o objeto escolhido já vai aparecer, mas a sua referência de origem pode não estar bem configurada, é possível que aconteça a mesma situação da figura 50.

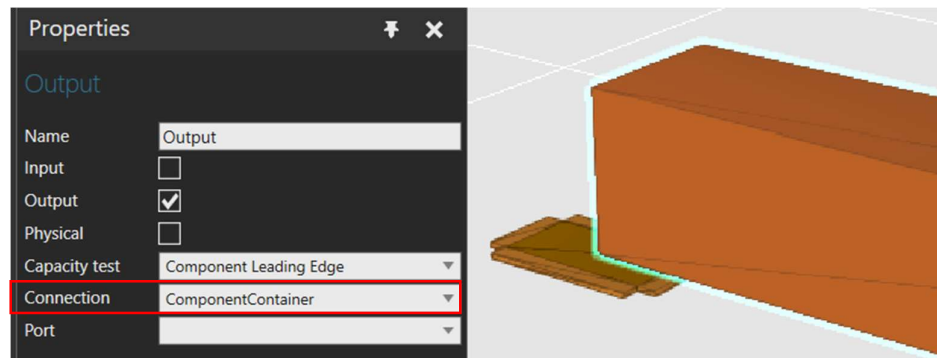


Figura 50 – Propriedades do Output do Container.

Uma solução neste caso é adicionar um eixo de referência e movimentá-lo para a posição desejada onde se pretende iniciar a criação do objeto (ver eixo da Figura 51). Depois basta ir às propriedades do “ComponentContainer” e na localização relacionar com o eixo de referencial.

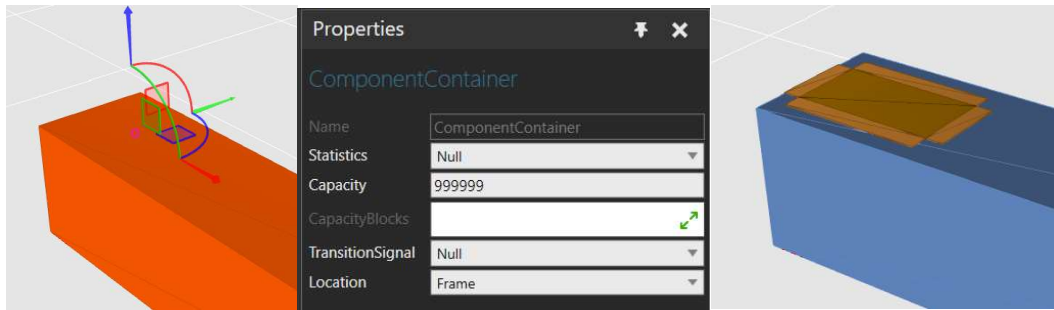


Figura 51 – Configuração do Eixo Referencial da Criação.

Em seguida, o próximo passo é colocar a peça a ser movida no tapete, para isso vai adicionar-se o “One Way Path” (Figura 52) e basta posicionar os eixos de acordo com o movimento pretendido. O eixo presente na imagem é um eixo criado que corresponde à posição final do tapete, sendo que a posição de início já foi configurada anteriormente. Nas propriedades de “OneWay Path” em “Path” adicionam-se os eixos de referências criados.

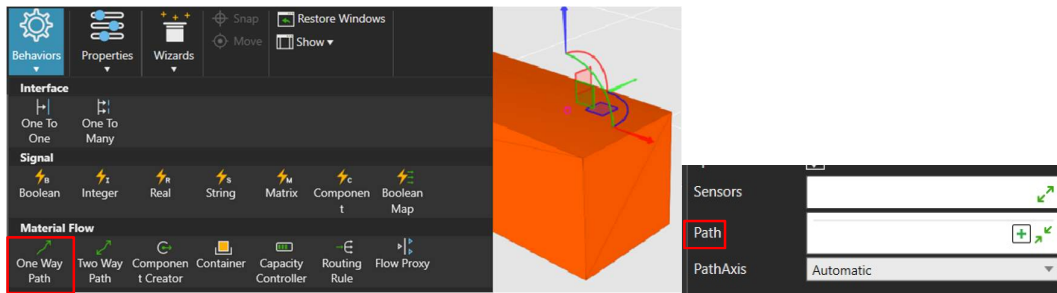


Figura 52 – Configuração da movimentação do tapete.

Depois é só necessário relacionar o “Input” do “One Way Path” com o “Output” do “ComponentContainer”, ficando a configuração igual à da Figura 53 e desta forma o *Creator* fica pronto a ser utilizado.

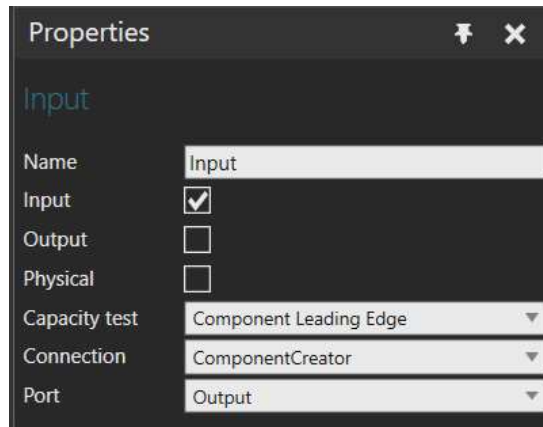


Figura 53 – Propriedades do Input do One Way Path.

3.4 – Programação do Simulador

3.4.1 – SINAIS DE ENTRADA E SAÍDA

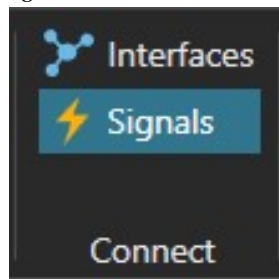
Para que o sistema funcione é necessário fazer as ligações de entradas e saídas, por exemplo se existir um tapete que esteja a transportar produtos e se pretenda configurar o robô como objetivo de os apanhar num determinado local, esse processo só será possível com um sensor. Neste caso seria necessário configurar, para que quando o sensor fica ativo, o robô receba essa informação e agarre no objeto, sendo este o processo mais benéfico. Outra forma seria, quando o sensor detetasse, o tapete parasse e o robô se deslocasse para o apanhar, mas neste caso o processo já não traria tantos benefícios, visto que atrasava o transporte das peças.



Figura 54 – Componentes com sinais.

No exemplo apresentado na Figura 54, é necessário interligar três componentes, um robô, um tapete e um sensor. Imagine-se a seguinte situação: ao colocar uma peça no tapete através do robô, este só coloca outra quando a primeira peça passar o sensor. Uma vez que o programa não possui nenhum autómato é necessário usar o robô como ligação comum, isto é, o sensor tem de se ligar à entrada do robô, de modo que este aguarde o sinal para colocar outra peça. Logo que o robô consiga colocar a primeira peça dá sinal ao tapete para iniciar o processo.

Figura 55 – Submenu Connect.



Para visualizar os sinais é necessário ir ao submenu “Connect” e selecionar “Signals”, este vai ter o símbolo de um raio como mostra a Figura 55, que permite o aparecimento das janelas de cada componente com a respetiva ligação.

Assim que se esteja nesse modo deverá aparecer uma janela do lado direito igual à da Figura 56, onde é possível relacionar os componentes como as figuras abaixo representadas.

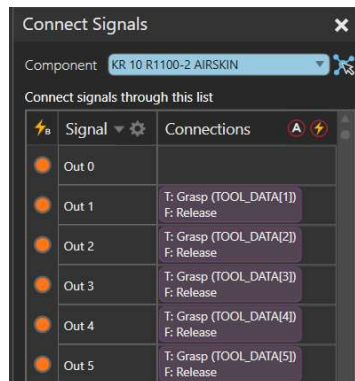


Figura 56 – Menu Connect Signal.

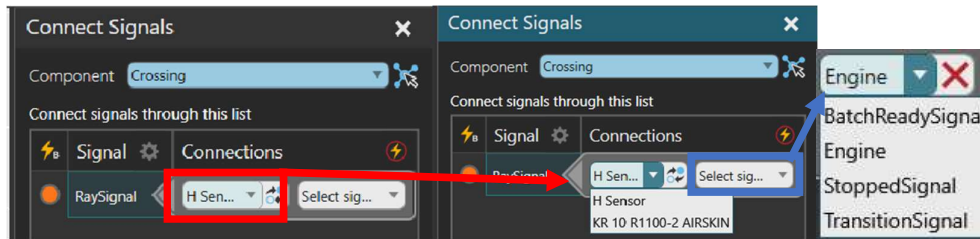


Figura 57 – Configuração de sinais.

No menu que aparece na direita, após escolher o modo “*Signal*”, é possível escolher qual o componente para o qual se pretende configurar as ligações, na esquerda encontra-se selecionado o robô e na direita o tapete rolante (Figura 57). Aconselha-se a que a interligação entre os componentes seja feita no robô porque este possui já algumas das saídas usadas.

Na região realçada a vermelho é onde se seleciona o componente que se quer relacionar, ou seja, neste caso é no tapete que é possível selecionar o robô ou o sensor, o lado direito dessa região permite que selecione o comportamento do sinal em questão. O sinal pode ser “*Engine*” permitindo dar ordens ao tapete para começar a trabalhar: sempre que estiver a “1” este encontra-se em andamento.

Após as ligações estarem concluídas, o resultado é o que está representado na Figura 58. Como se pode observar, o sensor encontra-se ligado a um “*Input*” do robô (IN1) enquanto o tapete se encontra ligado a uma saída do robô (OUT82). Assim, na parte da programação, uma vez que a ligação se encontra efetuada é possível aguardar pelo sinal do sensor ou ativar a saída do robô para colocar o tapete em movimento.

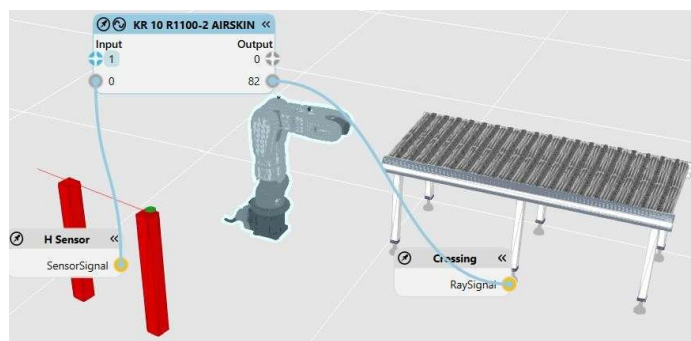


Figura 58 – Interligação de Componentes.

3.4.2 – PROGRAMAÇÃO DO ROBÔ

Para se programar um robô é necessário selecionar o menu “*Program*” (Figura 59) que vai abrir no lado esquerdo um separador designado por “*Job map*”.



Figura 59 – Menu Program.

No “*Job map*” representado na Figura 60 é possível programar as saídas, as posições, ou seja, basicamente contém todo o programa elaborado que irá ser seguido pelo robô.

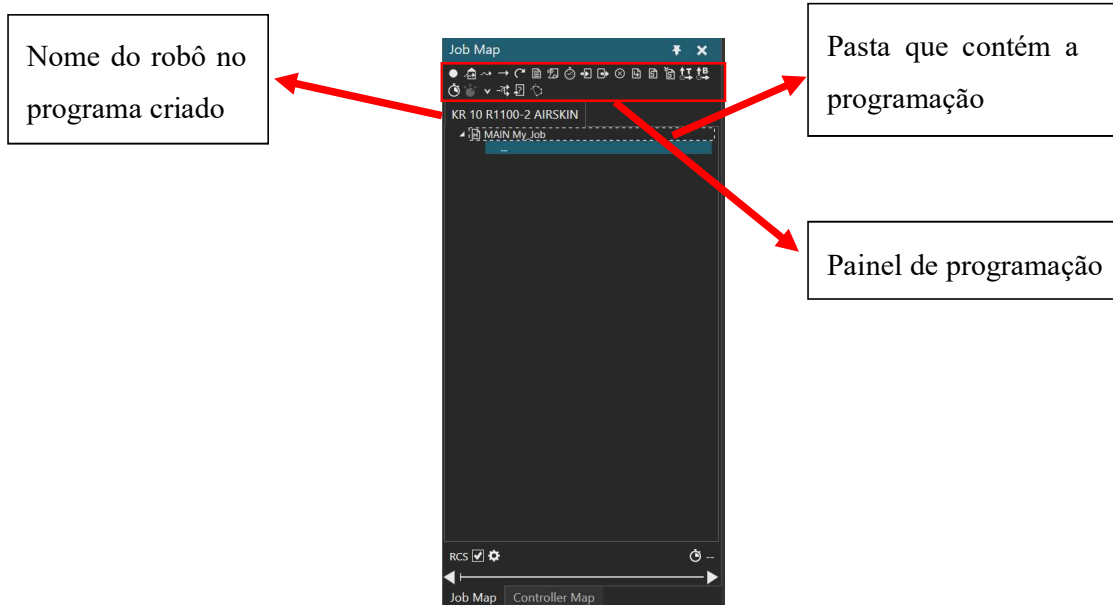


Figura 60 – Consola Virtual do Robô.

A Figura 61 apresenta as ferramentas ou os tipos de movimentos que poderão ser usados na programação do robô.



Figura 61 – Painel de Programação.

1. PTPHome: permite que seja guardada a posição onde se encontra como “home” do robô, ou seja, a posição inicial deste;
2. PTP: adiciona um movimento do tipo de juntas;
3. LIN: adiciona um movimento do tipo linear;
4. CIRC: adiciona um movimento do tipo circular;
5. COMMENT: adiciona um comentário para descrever algumas linhas da programação se for necessário;
6. WAIT: aguarda até que um tempo determinado tempo chegue ao fim;
7. WAIT FOR IN: aguarda que um determinado sinal de entrada esteja a 1, enquanto estiver a 0 fica preso nesta condição;
8. OUT: ativa uma saída do robô;
9. HALT: permite que o programa pare, ou seja, depois se quiser retomar basta clicar no play;
10. FOLDER: cria uma pasta onde se pode colocar a programação/ linhas de código;
11. SUBROUTINE: cria uma mini tarefa que pode ter uma determinada programação;
12. CALL-SUBROUTINE: permite que seja chamada uma parte do programa com uma tarefa específica;
13. SET TOOL: permite mudar o eixo de referência da ferramenta do robô;
14. SET BASE: permite mudar o eixo de referência da base do robô;
15. TIMER: permite contar o tempo;
16. IF: permite adicionar um IF que permite criar condições;
17. WHILE: permite adicionar um WHILE que permite criar ciclos dentro do programa;
18. PATH: permite que seja criado um caminho percorrido pelo robô, isto é, uma trajetória.

Para adicionar os movimentos do robô é necessário que ele esteja em posições diferentes, para isso basta clicar no “JOG” que se encontra presente em cima da janela no lado esquerdo e de seguida só é necessário ir ao eixo da ferramenta do robô e deslocá-la para movimentar. É possível também selecionar só um eixo se quisermos por exemplo efetuar um movimento linear.

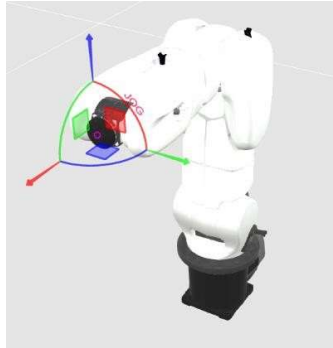


Figura 62 – Robô em modo "JOG".

3.5 – Exemplos de simulações

O exemplo demonstrado na Figura 63 veio no seguimento de uma proposta, da Aleluia Cerâmica, para simular uma linha de produção composta por três robôs. Através desta simulação é possível verificar o potencial deste programa, tendo em conta que permite simular um ambiente industrial em que já se encontre tudo sincronizado. É possível verificar também como fica o programa final através da utilização das funções em cima apresentadas.

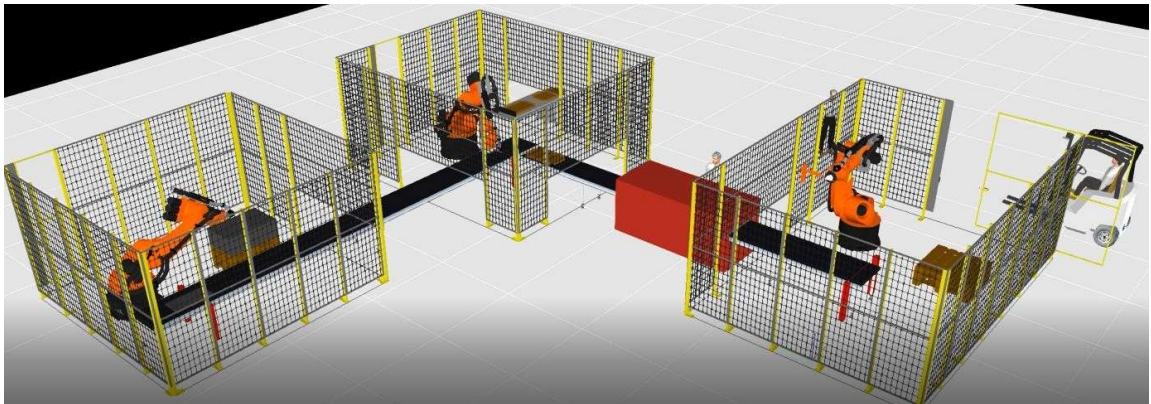


Figura 63 – Ambiente Fabril em Simulação.

Para iniciar o programa é necessário clicar no botão “Play” da Figura 64 que permite que as linhas de código sejam executadas até ao fim.



Figura 64 – Barra de controlo de simulação.

Os exemplos que se seguem permitem constatar que o Kuka Simulator é uma mais-valia, dando uma ideia de como será o processo e como estão organizadas as máquinas na realidade, pois possibilitará a visualização do aspeto do produto final.

De seguida apresentam-se algumas imagens que permitem comparar os processos atualmente existentes na empresa Aleluia Cerâmicas com os obtidos da ferramenta Kuka Simulator de acordo com os cenários de simulação estabelecidos.

A Figura 65 a) representa a passadeira de transporte de azulejos presente nas instalações fabris da empresa, bem como uma paleta de azulejos a aguardar deslocação. A Figura 65 b), a qual foi obtida por simulação, representa a passadeira de transporte dos azulejos e uma paleta, na qual serão colocados os azulejos, localizadas num espaço delimitado por grades de proteção industriais e um processo de paletização através de um robô.

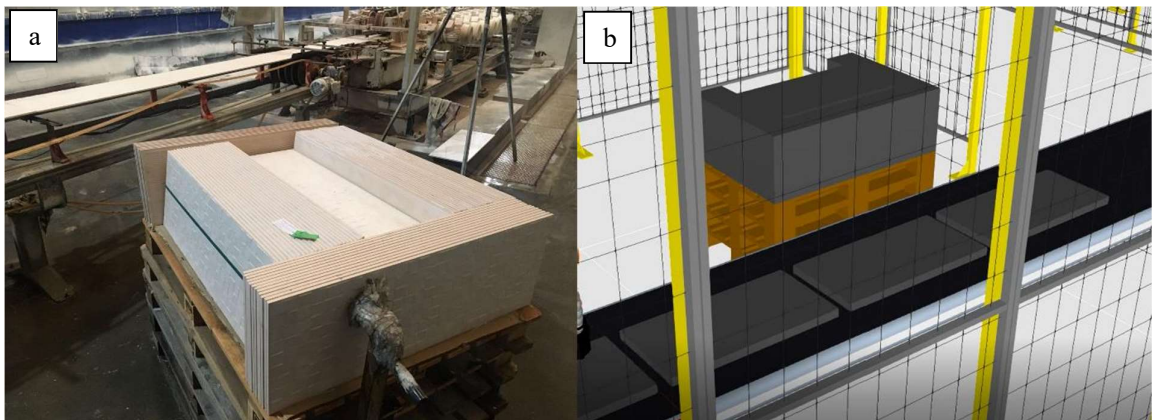


Figura 65 – Azulejos a aguardar deslocação.

A Figura 66 a) representa a colocação dos azulejos no tapete por um operador, em oposição com o processo automatizado obtido por simulação, representado na Figura 66 b).

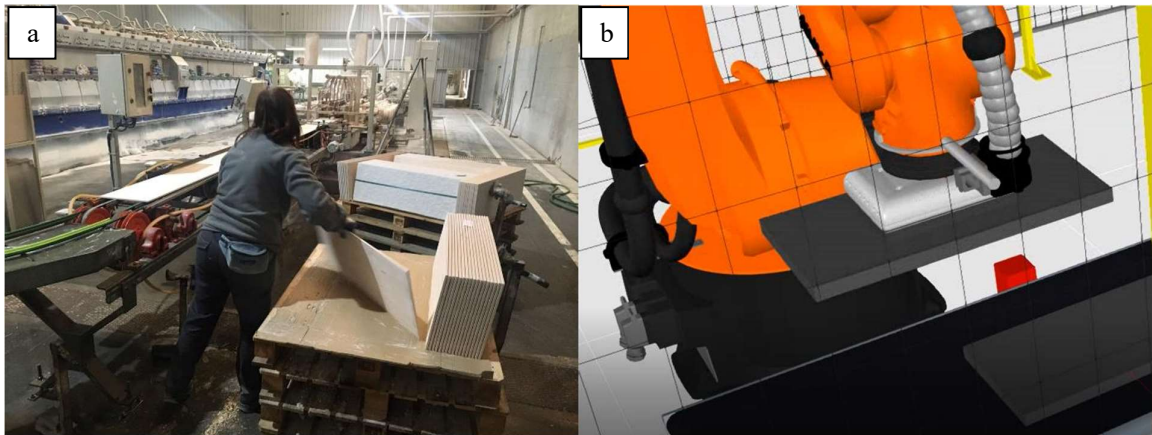


Figura 66 – Colocação dos azulejos no tapete.

A Figura 67 a) representa o processo de colocação manual dos azulejos nas caixas realizada por um operador em contraste não só com o processo automatizado obtido por simulação que compreende uma passadeira, mas também um manipulador robótico com uma garra apropriada ao peso dos referidos azulejos, representado na Figura 67 b).

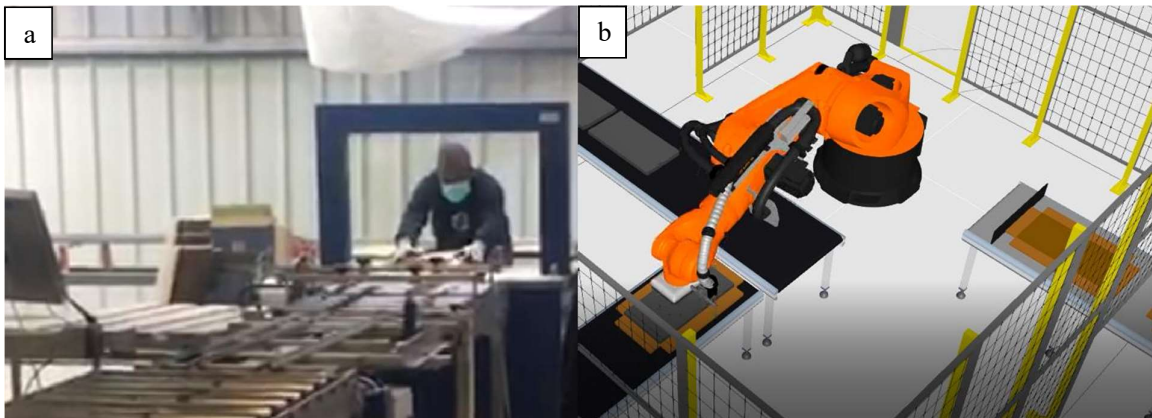


Figura 67 – Colocação dos azulejos em caixas.

A Figura 68 representa o processo manual e automático (simulado) no qual as caixas que contém os azulejos e que foram previamente fechadas. São seladas com recurso a uma fita de cinta de material apropriado.

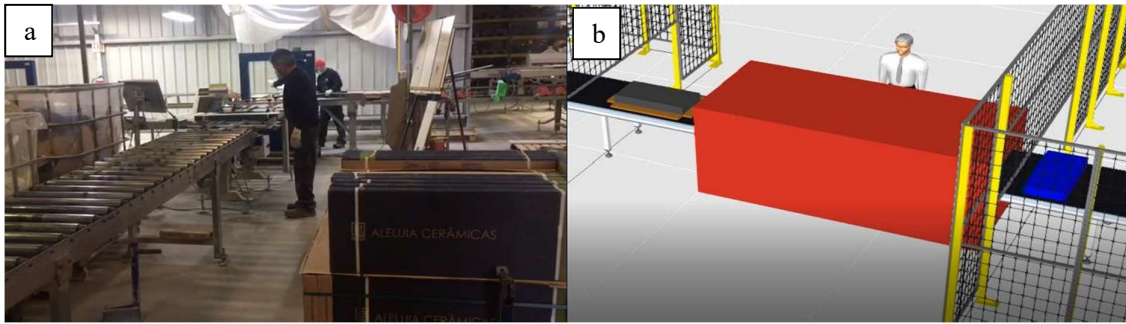


Figura 68 – Colocação de cintas na caixa.

Finalmente, a Figura 69 a) representa a paletização dos azulejos efetuada atualmente por um operador humano que os retira manualmente de uma passadeira e coloca-os numa palete em determinada posição pré-estabelecida. Por outro lado, na Figura 69 b), obtida do Kuka Simulator, observa-se um manipulador a efetuar a mesma tarefa de uma forma automatizada de acordo com a programação efetuada.

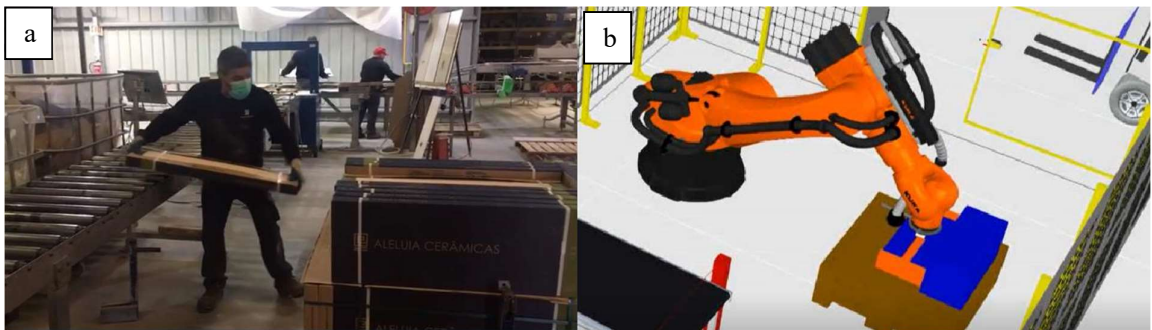


Figura 69 – Paletização dos azulejos.

CAPÍTULO 4 – Visão Aplicada à Robótica

4.1 – Mako Allied Vision

Relativamente ao campo da cerâmica foi desenvolvido um estudo em que se realizaram vários testes com uma câmara “Mako”. Esta câmara possui uma lente e um sensor monocromático com uma resolução de 2K (2560 x 1440 pixéis), sendo diretamente alimentada através de *Power Over Ethernet* (POE). A escolha da câmara foi feita tendo em conta a qualidade/preço, mas também a vantagem de ter exemplos de programa, o que é uma mais-valia. Em relação à comunicação, esta foi realizada através do sistema Ubuntu (Linux) usando a rede *Ethernet*, o que permitiu programar em OpenCV, devido ao facto de já existirem exemplos de programas. O desenvolvimento do programa foi feito através de um desses exemplos que permite à câmara visualizar os *frames* que captura com bastante rapidez, de modo a dar o efeito de um vídeo ao vivo. Normalmente este programa é usado para visualizar a zona de interesse, permitindo que se façam alterações e se verifique as suas influências na imagem.

Foi desenvolvido um algoritmo para guardar uma imagem retirada na câmara, sendo o princípio de funcionamento o mesmo referido anteriormente, modificando-se apenas a parte de mostrar a imagem, uma vez que só é necessário guardar a imagem e não o vídeo. Este programa pretende guardar a imagem, que posteriormente é importada mais tarde para outro programa. Para proceder à sua análise foi usada a função `cv2.imwrite` que permite que seja feito o *save* do *frame* sendo a imagem guardada em formato jpeg (Figura 70).

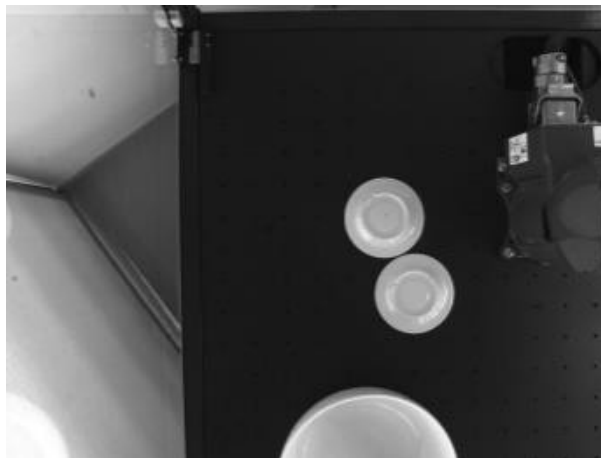


Figura 70 – Imagem guardada.

Depois de obtida a imagem, o próximo passo é processá-la, numa primeira fase é necessário convertê-la para tons de cinza tornando o prato mais visível, de seguida detetam-se os pratos e a localização do centro de cada um deles, de modo que essa informação seja enviada para um braço robótico à posteriori.

Como o sistema de medida se encontra fixo é necessário encontrar uma relação entre as coordenadas reais e as utilizadas no contexto computacional utilizado nomeadamente a quantificação das distâncias reais e os pixels das imagens. Para o efeito criou-se um eixo cartesiano com vários pontos, de modo a fazer comparações entre eles e descobrir uma medida constante que permita o cálculo dos valores reais.

Para proceder ao cálculo dos valores foi necessário usar uma régua que faz a medição entre os buracos (50 cm). Depois de obtido esse valor, através do referencial cartesiano, verificámos a distância na imagem entre esses mesmos pontos anteriormente medidos (135 pixels). De seguida basta usar apenas a regra de três simples, em que 50 cm são equivalentes a 135 pixels, ou seja, a distância que obtemos do ponto de referência até ao centro do prato é X .

Os valores referem-se unicamente a este caso específico, no qual a câmara encontra-se no teto (Figura 71), sempre à mesma distância garantindo estabilidade no cálculo. Caso a câmara estivesse na ferramenta do robô seria necessário calcular a distância do prato usando outra relação.

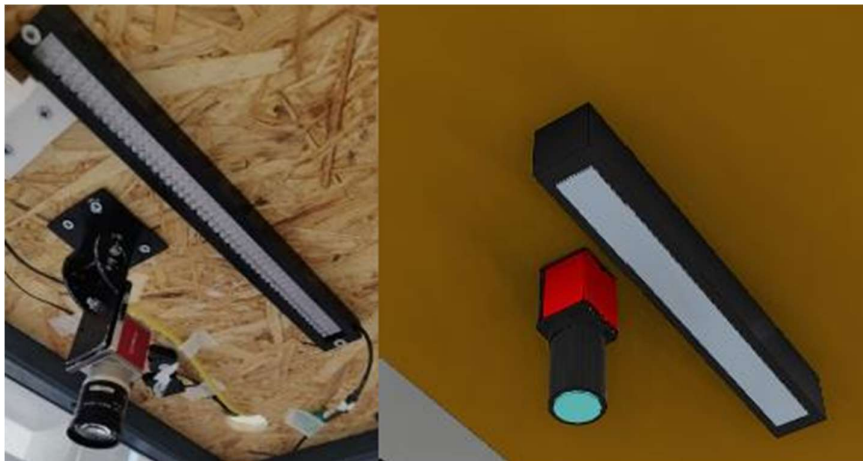


Figura 71 – Setup Câmara, realidade vs simulador.

Para representar os pontos usou-se a função `cv2.circle`, adicionando-se os valores constantes para que todos tivessem a mesma distância entre eles e mediu-se com uma régua o valor real (Figura 72 a)). Considerando os valores obtidos na experiência apresentada fez-se o cálculo da coordenada real com o centro do prato e identificou-se com um ponto vermelho, ao lado desse ponto apresentam-se as coordenadas na imagem e no mundo real.

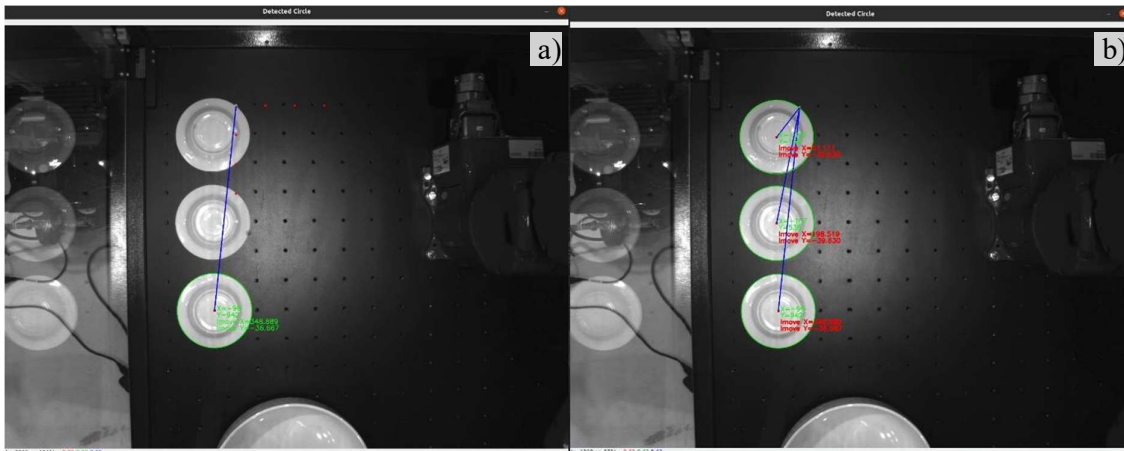


Figura 72 – a) Imagem Cartesiano; b) Detetar Centros.

No final, o programa fica com esta imagem (Figura 72 b)), realçando as coordenadas que o robô irá usar, estas distâncias representam a distância que o robô tem de fazer para apanhar o prato, sendo necessário ter um ponto de referência para fazer a diferença entre dois pontos e descobrir o espaço entre eles.

Para melhorar o programa, foi adicionada uma região de interesse, *Region of Interest* (ROI) de modo a processar apenas a zona onde estariam os pratos, tornando o programa mais preciso na identificação. A ROI da imagem está identificada, na Figura 73, por meio de um retângulo vermelho.

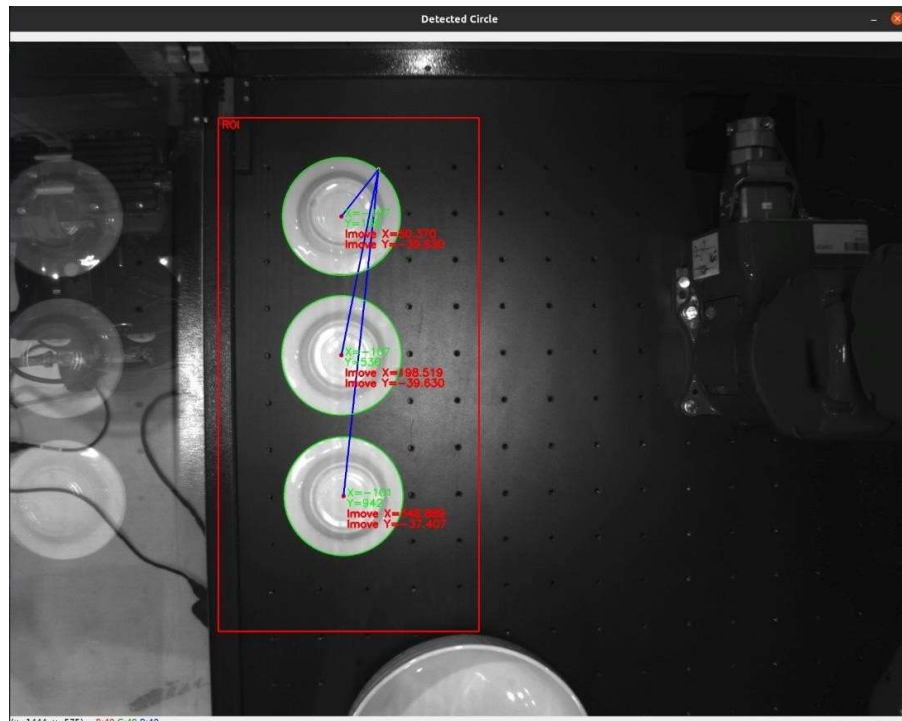


Figura 73 – Detetar centros ROI.

O processo final possui as etapas que estão presentes no fluxograma da Figura 74. Inicia-se com a leitura das configurações da câmara para assegurar que as imagens adquiridas possuem os parâmetros adequados. O próximo passo, após o registo da imagem, será convertê-la para tons de cinza, tornando o prato ainda mais visível. O passo seguinte compreende a deteção dos centros, caso não detete nenhum não vai abrir nenhuma imagem com a deteção, se detetar vai calcular o centro do prato e usar esse valor para chegar à distância em coordenadas do robô para que o prato seja apanhado. Para que este processo de pegar seja efetuado é necessária uma instrução humana, começando por colocar as respetivas coordenadas que foram calculadas anteriormente no robô, para que recolha o primeiro prato antes de ir para o próximo.

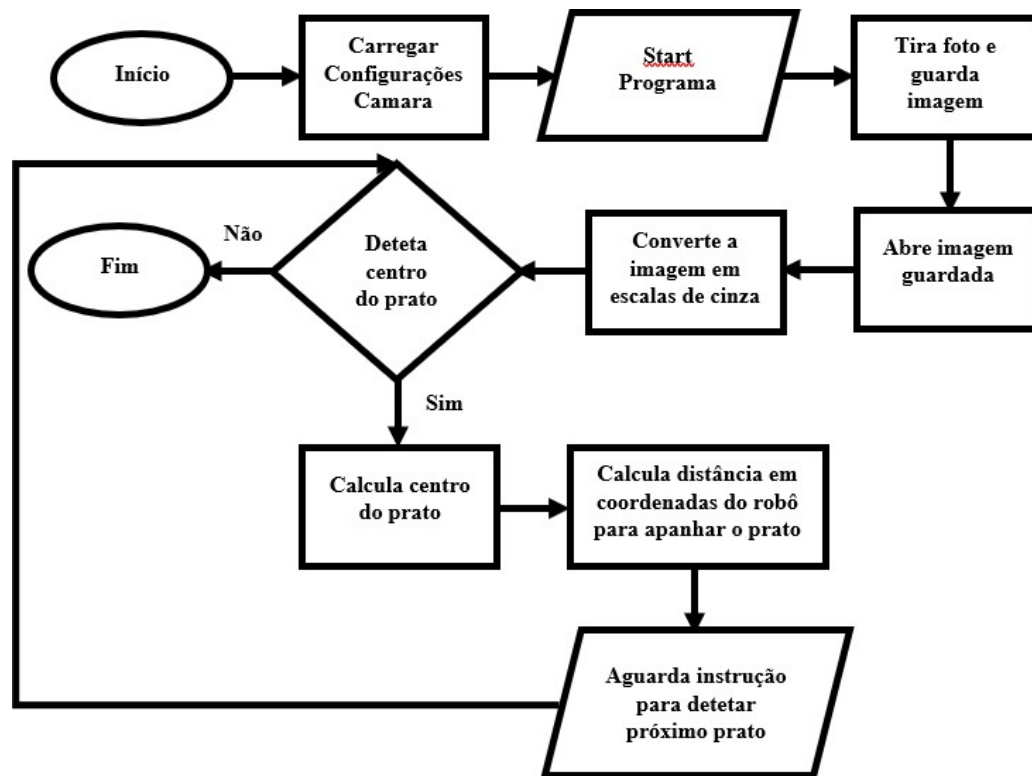


Figura 74 – Fluxograma do programa.

A iluminação representa uma condição importante para um processamento efetivo de uma imagem, podendo influenciar de uma forma positiva ou negativa. Uma boa iluminação permite realçar melhor e com mais facilidade regiões de interesse presentes na imagem permitindo um processamento mais eficiente. No entanto, a iluminação pode também apresentar desvantagens como, por exemplo, os reflexos. O tipo de luz que se usou na solução implementada é uma luminária LED direcional, a qual se encontra fixa no teto ao lado da câmara, usada para iluminar os objetos presentes nessa área.

Apesar do sistema já conseguir detetar as distâncias reais para apanhar o prato, ainda há aspetos a melhorar, nomeadamente a luminosidade (Figura 75). Devido ao facto de se tratar de uma câmara monocromática, as questões das luzes são muito importantes, o ideal seria fazer um ambiente fechado impedindo que a luz do sol influencie a imagem.



Figura 75 – Imagens com luminosidade baixa e alta.

Os resultados desta experiência foram positivos e promissores para o futuro, uma vez que o erro de posição do robô para agarrar o prato é muito reduzido. Contudo, ainda não se encontra totalmente otimizado, pois utiliza vários programas e ainda se trata de um sistema lento, mas para *pick and place* estáticos, pouco exigentes, acaba por funcionar muito bem. O programa ainda tem alguns aspetos a melhorar, sobretudo na parte da comunicação, visto que ainda não comunica com o robô, (é necessário comprar uma licença para o efeito), ou seja, a distância de movimento tem de ser inserida manualmente. Apesar de detetar o prato, apenas é possível em 2D, porque se este estivesse mais elevado podia não o detetar tão bem e o mesmo não teria a perceção que tinha de subir para o agarrar. Embora este programa funcione acaba por ter algumas limitações.

Considerando os resultados obtidos neste projeto pode-se constatar que os resultados foram bastantes positivos e, sem dúvida alguma, poderão ser uma mais-valia para várias aplicações, porém ainda é necessário melhorar alguns aspetos e otimizar, para que se torne um sistema mais eficaz e rápido. De salientar ainda, o facto de a visão ser uma mais-valia para a indústria da cerâmica, nomeadamente para a perceção de defeitos na louça identificando também a melhor posição para o robô agarrar as peças de louça.

4.2 – Matlab identificação Cerâmica

Nesta secção foi proposta a elaboração de um programa (Figura 77), de modo a identificar os objetos de cerâmica e apresentar as coordenadas do seu centro. Para esta experiência, o programa utilizado foi o Matlab, tendo-se para o efeito utilizado três imagens distintas que continham uma caneca, um pires e uma chávena (Figura 76).

As imagens foram tiradas com um telemóvel (16 MP) e manteve-se a altura para que os valores das experiências fossem precisos. A luminosidade foi de certa forma controlada e tentou-se usar um fundo mais escuro que os objetos.



Figura 76 – Objetos Cerâmicos.

Os objetos em questão foram identificados através do seu diâmetro, através de várias medições fez-se um levantamento das várias medidas possíveis destes. Após análise, o programa identifica cada um dos três objetos da imagem como sendo uma “chávena”, um “pires” e uma “caneca”.

```

clc;
clear all;
%-----
img = imread("3.jpg");
%-----
BW = im2bw(img);
[centers, radii, metric] = imfindcircles(BW,[80 1500], 'Sensitivity', 0.95);
circulos = size(radii);
for c = 1:circulos
    if radii(c)>90 && radii(c)<110
        text_str{1} = ['Chávena (' num2str(centers(c,1), '%0.2f') ', '
num2str(centers(c,2), '%0.2f') ')];
    end
    if radii(c)>160 && radii(c)<180
        text_str{2} = ['Caneca (' num2str(centers(c,1), '%0.2f') ', '
num2str(centers(c,2), '%0.2f') ')];
    end
end

```

```

    if radii(c)>180 && radii(c)<200
        text_str{3} = ['Pires (' num2str(centers(c,1),'%0.2f') ', '
num2str(centers(c,2),'%0.2f') ')];
    end
end
RGB = insertText(img,[centers(1,1)-40 centers(1,2)-40;centers(3,1)-40 centers(3,2)-
40;centers(2,1)-40 centers(2,2)-40],text_str,FontSize=40)
figure;
imshow(RGB);
viscircles(centers, radii,'EdgeColor','g');

```

Figura 77 – Código Matlab.

Através da função do Matlab “imfindcircles” é possível determinar o raio e o centro dos objetos cerâmicos, de seguida esse valor fica guardado numa variável que vai ser comparada dentro de intervalos, para que seja possível identificar que tipo de objeto representa. Foi necessário parametrizar a função para que a dimensão dos objetos fosse detetada nos intervalos de pesquisa.

Após o processamento das imagens presentes na Figura 78 é inserida uma legenda com a identificação do objeto e as coordenadas do centro de cada uma das peças cerâmicas.

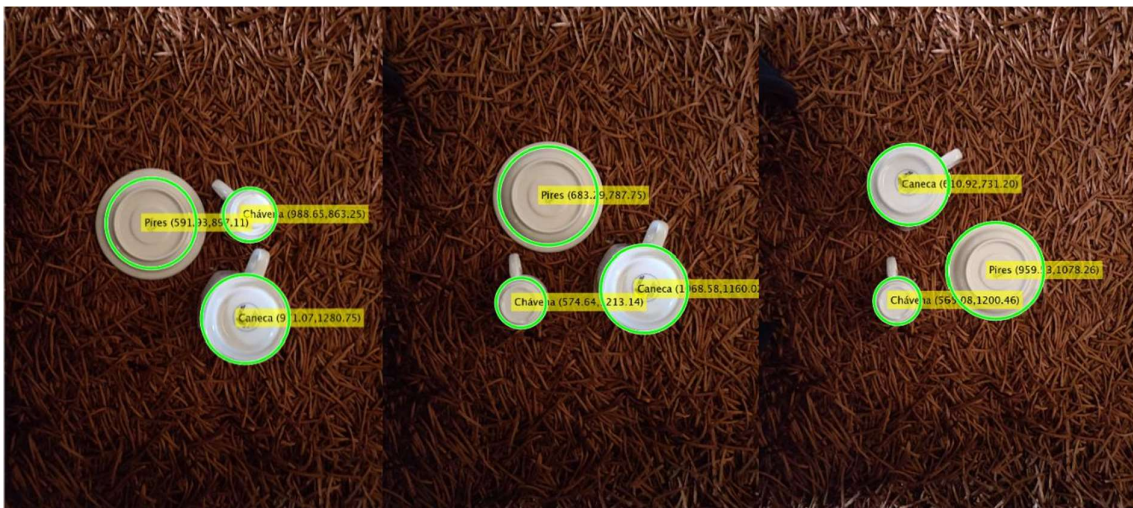


Figura 78 – Identificação de Objetos Cerâmicos.

Apesar desta experiência ter sido bastante positiva, o Matlab acaba por ter algumas limitações nomeadamente o tempo de processamento. Este código foi executado com apenas três imagens, contudo é bastante lento a processar cada uma delas. Embora tenha esta limitação, o acesso a funções do programa é uma mais-valia porque facilita este processo. Contudo se for necessário fazer uma deteção de defeitos num dos objetos cerâmicos, ou seja, um processo mais detalhado será necessário um sistema de processamento mais avançado visto que alguns defeitos nem a olho nu são perceptíveis.

4.3 – Algoritmos de Aprendizagem

O *Reinforcement Learning* (RL) é uma técnica de aprendizagem em que um agente aprende de acordo com a experiência, isto é, aprende a executar tarefas repetitivas em detrimento de tentativa de erro. Esta abordagem permite que o agente tome decisões, sem a intervenção humana e sem ser programado para executar a tarefa. Esta técnica acaba por aliviar a necessidade de adquirir dados, ou seja, não é necessária uma base de dados e evita o pré-processamento, outra vantagem é que com o RL é possível aprender um novo comportamento sem necessidade de supervisão humana. Para compor o *Reinforcement Learning* são necessários alguns passos:

- *Environment*: É necessário definir um ambiente onde o agente vai operar, incluindo a interface entre o agente e o ambiente. O ambiente pode ser de simulação ou realidade, mas no início os ambientes de simulação tendem a ser mais controlados o que permite uma melhor experiência;
- *Reward*: O agente necessita de um sinal para medir o seu desempenho em relação aos objetivos da tarefa;
- *Create the agent*: É necessário criar um agente, que consiste na política e no algoritmo;
- *Deploy the policy*: Implantar a representação de política treinada usando C/C++ ou CUDA.

Outro tipo de algoritmo de aprendizagem é o Yolo que é uma rede de detecção de objetos, usa uma rede de extração de recursos para fazer previsões dos objetos. Este usa uma CNN numa imagem, reúne e descodifica previsões para gerar caixas que limitam o objeto, identificando os objetos.

```
pretrainedURL =  
"https://www.mathworks.com/supportfiles/vision/data/yolov2IndoorObjectDetector  
.zip";  
pretrainedFolder = fullfile(tempdir, "pretrainedNetwork");  
pretrainedNetworkZip = fullfile(pretrainedFolder,  
"yolov2IndoorObjectDetector.zip");  
  
if ~exist(pretrainedNetworkZip, "file")  
    mkdir(pretrainedFolder);  
    disp("Downloading pretrained network (98 MB)...");  
    websave(pretrainedNetworkZip, pretrainedURL);  
end
```

```

unzip(pretrainedNetworkZip, pretrainedFolder)

pretrainedNetwork = fullfile(pretrainedFolder,
"yolov2IndoorObjectDetector.mat");
pretrained = load(pretrainedNetwork);
detector = pretrained.detector;

[bbox, score, label] = detect(detector, I);
    
```

Este exemplo permite a execução do Yolo v2 em que através de uma base de dados identifica os diferentes objetos da Figura 79 delimitando-os com linhas.



Figura 79 – Imagem processada com Yolo.

Este algoritmo é bastante apelativo tendo em conta que já se encontra disponível no Matlab e pode ser facilmente usado para identificar objetos. É possível que também identifique objetos cerâmicos, contudo para aplicações mais detalhadas, nomeadamente a deteção de defeitos na louça, poderá não ser a melhor opção. Em primeiro lugar seria necessária uma vasta recolha de dados numa fábrica, de modo a armazenar todos os tipos de defeitos existentes. Além disso há alguns defeitos que são difíceis de detetar a olho nu, pelo que seria necessário um computador com um maior poder de processamento, de modo a processar todo o detalhe da imagem e uma câmara que consiga capturar todas as informações presentes no prato. Apesar destas limitações, a capacidade deste programa continua a ser uma mais-valia para aplicações mais básicas como sejam as aplicações *pick and place*.

CONCLUSÃO

A Automação Industrial e Robótica são áreas complexas e extensas, com aplicação em diversos setores, nomeadamente na indústria, no comércio, na agricultura, na saúde, na educação, na habitação, entre outros. A diversidade de aplicações e a constante evolução das tecnologias associadas faz com que seja necessária uma atualização contínua de conhecimentos tendo em conta as especificidades de cada uma das tecnologias empregues, obrigando os profissionais a estarem sempre na vanguarda das inovações e a adotar uma postura proactiva na sua formação profissional. Assim, esta área é bastante desafiante pela necessidade de compreender os diferentes tipos de processos, sensores, atuadores, esquemas elétricos, materiais, dispositivos, linguagens de programação, protocolos de comunicação, tecnologias, entre outros.

Embora o tema central deste relatório de projeto se tenha focado mais na área de automação, robótica e visão artificial, para a realização do projeto foi necessário utilizar conhecimentos de diferentes áreas para além da Engenharia Eletrotécnica.

Durante o período de execução do projeto tive oportunidade de colocar em prática os conhecimentos adquiridos no curso de Licenciatura em Engenharia Eletrotécnica e no Mestrado em Engenharia Eletrotécnica com Especialização em Automação e Comunicações em Sistemas Industriais. No entanto, em algumas situações, existiu a necessidade de pesquisar e aprofundar novos conhecimentos, para dar resposta a algumas situações e dificuldades com que me fui deparando, durante a realização do projeto. Deste modo, em algumas situações tive de ser autodidata, uma vez que a empresa onde foi desenvolvido o projeto não tinha um profissional especializado em robótica. Contudo, foi uma experiência bastante gratificante, não só porque proporcionou a aquisição de novos conhecimentos, como também permitiu desenvolver competências e aplicar os conteúdos adquiridos ao longo dos últimos cinco anos.

Para a concretização dos trabalhos associados ao projeto foi necessário realizar e desenvolver diversas tarefas de entre eles refira-se: a simulação de soluções de robótica; a programação de robôs; o desenvolvimento de aplicações de visão artificial, o que evidenciou a importância e o papel fulcral da automação e robótica no setor da cerâmica. Deste modo considero que foram atingidos os objetivos propostos no início do trabalho.

Em suma, a Automação e a Robótica são áreas que estão em crescimento, pelo que é necessário a formação de mais profissionais especializados em Automação Industrial, Robótica e Visão Artificial, de forma a acompanhar a evolução e a dar resposta às necessidades e desafios de uma sociedade cada vez mais automatizada.

Considerando o trabalho realizado seria interessante, em trabalhos futuros, integrar o *Deep Learning* e a Inteligência Artificial no sistema de visão, de forma a torná-lo mais preciso e adaptável à execução das tarefas solicitadas. Do mesmo modo, seria uma mais valia programar a parte da comunicação do sistema *pick and place* com o robô, já que não foi possível desenvolvê-lo neste projeto, o que tornaria o sistema mais independente, sem que houvesse qualquer tipo de intervenção humana.

Outras sugestões de trabalhos a desenvolver seriam a utilização de robôs colaborativos, cada vez mais usados pelas empresas, de forma a criar um ambiente laboral que integrasse robôs e humanos, otimizando as tarefas, já que permitiria aproveitar as qualidades de cada um; a criação de um sistema de troca de ferramenta de trabalho do robô, em que, através da visão, se detetasse o tipo de peça e indexasse a ferramenta mais adequada a utilizar; e finalmente a integração de um braço robótico num robô móvel, de forma a permitir a execução de tarefas à medida que o robô móvel se desloca.

Referências

- Almeida, C. D. F. de, Lima, J. P. de O., Oliveira, J. V. M. de, Oliveira, R. C. de, & Oliveira, E. F. (2021). Arquitetura de Sistema em Nuvem para apoio à implantação de Visão computacional em Linhas de Produção na Indústria 4.0. Simpósio Brasileiro de Automação Inteligente - SBAI. https://www.sba.org.br/open_journal_systems/index.php/sbai/article/view/2862
- Aslam, F., Aimin, W., Li, M., & Ur Rehman, K. (2020). Innovation in the era of IOT and industry 5.0: Absolute Innovation Management (AIM) framework. MDPI. <https://www.mdpi.com/2078-2489/11/2/124>
- Bayer, F. (2011). Apostila Automação - Automação de Sistemas Fernando Mariano bayer Moacir Eckhardt Renato Machado. Studocu. <https://www.studocu.com/pt-br/document/ensino-medio-brasil/automacao-pneumatica-e-hidraulica/apostila-automacao/32567224>
- Boettcher, M. (2021, February 5). Revolução industrial - um pouco de história da indústria 1.0 até a Indústria 4.0. LinkedIn Portugal: <https://pt.linkedin.com/pulse/revolu%C3%A7%C3%A3o-industrial-um-pouco-de-hist%C3%B3ria-da-10-at%C3%A9-boettcher>
- Cavalcante, Z. (2011). A importância da revolução industrial no mundo da tecnologia. https://www.unicesumar.edu.br/epcc-2011/wp-content/uploads/sites/86/2016/07/zedequias_vieira_cavalcante2.pdf
- Charniak, E. (1985). (PDF) a Bayesian model of plan recognition - researchgate. https://www.researchgate.net/publication/222330914_A_Bayesian_model_of_plan_recognition
- Chollet, F. (2018). Deep learning. MITP-Verlags GmbH & Co. KG.
- COPELAND, M. (2016). Obtido de Nvidia: <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>
- Dickson, B. (9 de agosto de 2019). Obtido de Medium: <https://medium.com/pcmag-access/what-is-deep-learning-5785391fee39>
- Falcao, D. (2019, January 1). Timeline da história da fotografia. FALCÃO, David. https://www.academia.edu/41475716/Timeline_da_Hist%C3%B3ria_da_Fotografia
- Ferreira, M. H. S. (2021, December 10). Reconhecimento facial para detecção de Emoções Utilizando Redes neurais Convolucionais Com Tensorflow. Apache Tomcat/9.0.39. <https://repositorio.pucgoias.edu.br/jspui/handle/123456789/3704>
- Galin1, R., & Meshcheryakov1, R. (2019, May 1). IOPscience. IOP Conference Series: Materials Science and Engineering. <https://iopscience.iop.org/article/10.1088/1757-899X/537/3/032073>
- GOMES, D. D. S. (2010). Inteligência Artificial: Conceitos E Aplicações. https://www.professores.uff.br/screspo/wp-content/uploads/sites/127/2017/09/ia_intro.pdf

- Gonzaga, L. A. (2017, December 26). Aplicações da Visão computacional Utilizando Python. Repositório Institucional - Universidade Federal de Uberlândia: Home. <https://repositorio.ufu.br/handle/123456789/25519>
- Haleem, A., & Javaid, M. (2019). Industry 5.0 and its applications in orthopaedics. *Journal of Clinical Orthopaedics and Trauma*, 10(4), 807–808. <https://doi.org/10.1016/j.jcot.2018.12.010>
- Hammes, N. S. (17 de 12 de 2020). Algoritmo de visão computacional para recomendação de um manipulador robótico baseado na pose de objetos. TCC (graduação) - Universidade Federal de Santa Catarina. Campus Blumenau. Engenharia de Controle e Automação, pp. 1-49.
- Haugeland, J. (1986). Artificial Intelligence: The very idea. Google Books. https://books.google.com/books/about/Artificial_Intelligence.html?id=zLFSPdIuqKsC
- International Federation of Robotics. (2021). Executive Summary World Robotics 2021 Industrial Robots - Chapter 1. Obtido de International Federation of Robotics: https://ifr.org/img/worldrobotics/WR_Industrial_Robots_2021_Chapter_1.pdf
- Kavlakoglu. (2020). AI vs. Machine Learning vs. Deep Learning vs. neural networks: What's the difference? <https://www.ibm.com/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks/>
- Kuric, I., Klarák, J., Bulej, V., Sága, M., Kandra, M., Hajdučík, A., & Tucki, K. (2022). Approach to automated visual inspection of objects based on Artificial Intelligence. *Applied Sciences*, 12(2), 864. <https://doi.org/10.3390/app12020864>
- Kurzweil, R. (1990). *The Age of Spiritual Machines*. The MIT Press. Massachusetts.
- Longo, C. A. (9 de 1 de 2018). LinkedIn. Obtido de Pilar da Indústria 4.0 – Simulação: <https://pt.linkedin.com/pulse/3-pilar-da-ind%C3%BAstria-40-simula%C3%A7%C3%A3o-carlos-augusto-longo>
- Lousada, R. (16 de 5 de 2022). O que é Robótica: Conceito, história e evolução. (Blog Eletrogate) de <https://blog.eletrogate.com/o-que-e-robotica-conceito-historia-e-evolucao/> Obtido em 3 de 6 de 2022,
- Ludermir, T. B. (2021). Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. *Estudos Avançados*, 35, pp. 85-94.
- Lynch, L., Coleman, J., Newe, T., Walsh, J., Clifford, J., & Toal, D. (6 de 2019). Automated Ground Vehicle (AGV) and Sensor Technologies-A Review.
- Maestri, G. B. (2021). *Revista Eletrônica Perspectivas da Ciência e Tecnologia-ISSN*. Revoluções tecnológicas e a relação com o setor têxtil: perspectivas baseadas em Indústria 3.5, Indústria 4.0 e Indústria 5.0, pp. V.13 1984-5693.
- Matheson, E., Minto, R., Zampieri, E. G., Faccio, M., & Rosati, G. (2019). Human–robot collaboration in Manufacturing Applications: A Review. *Robotics*, 8(4), 100. <https://doi.org/10.3390/robotics8040100>

- Moreno, A. S. (2016). Revista Educação - ung-ser. Câmeras Analógicas e Digitais: Qual oferece melhor qualidade?, pp. 19-33; <http://revistas.ung.br/index.php/educacao/article/view/2331/1707> .
- Mota, E. M. da, Júnior, J. C. C. S., Martins, J. L., Carneiro, J. A., Almeida, J. G. L., Santana, V. M. de, & Pereira, J. da S. (2020). A física do Olho Humano. Cadernos Macambira. <https://revista.lapprudes.net/index.php/CM/article/view/402>
- Neves, L. A. (2022). Nexos entre visão computacional e computação afetiva para o desenvolvimento da robótica. Brazilian Journals, 8(No 4), 24991-25005.
- Paschek, D., Mocan, A., & Draghici, A. (2019). Industry 5.0—The Expected Impact of Next Industrial Revolution. In Thriving on Future Education, Industry, Business, and Society, Proceedings of the MakeLearn and TIIM International Conference (pp. 125-132). ToKnowPress.
- Pasquini, T. C. (2018). Proposta de ferramenta para relacionar os princípios da gestão da qualidade aos pilares da indústria 4.0: a influência da indústria 4.0 na área da qualidade. Ponta Grossa: Universidade Tecnológica Federal do Paraná.
- Poole, D., Mackworth, A. K., & Goebel, R. (1998). Computational Intelligence: A Logical Approach. Oxford University.
- Recalcati, F., & Mingori, F. R. (2020). Desenvolvimento de Sistema utilizando Visão Computacional para Interação Humana. Seminário de Iniciação Científica e Seminário Integrado de Ensino, Pesquisa e Extensão.
- Ribeiro, E. M. (2018). Inovação em Sistemas de Produção na Era da Indústria 4.0.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review.
- Rubmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., & Engel, P. (2015). The Boston Consulting Group. Industry 4.0: The Future of Productivity and Growth in Manufacturing.
- Sakurai, R., & Zuchi, J. D. (2018). Revista Interface Tecnológica. As revoluções industriais até a indústria 4.0, pp. 480-491.
- SAMPAIO, G. S. (8 de Fevereiro de 2018). Desenvolvimento de uma interface computacional natural para pessoas com deficiência motora baseada em visão computacional. Dissertação (Engenharia Elétrica) - Universidade Presbiteriana Mackenzie, São Paulo, pp. 1-107.
- Santos, B. P., Alberto, A., Lima, T. D., & Charrua-Santos, F. M. (15 de 01 de 2018). Revista Produção e Desenvolvimento. INDÚSTRIA 4.0: DESAFIOS E OPORTUNIDADES, p. 112.
- Santos, Beatrice & Alberto, Agostinho & Lima, Tânia & Santos, Beatrice. (2018). INDÚSTRIA 4.0: DESAFIOS E OPORTUNIDADES. Revista Produção e Desenvolvimento. 4.

- Santos, T. T. (2020). Visão computacional aplicada na agricultura. Agricultura digital: pesquisa, desenvolvimento e inovação nas cadeias produtivas, pp. cap. 6, p. 146-164.
- Silva, M. V., Costa, R. C., Lima, L. M., & Freitas, E. J. (2020). Sensor de Fadiga: Assistente Virtual Para Direção / fatigue sensor: Virtual assistant for steering. Brazilian Journal of Development, 6(10), 78540–78549. <https://doi.org/10.34117/bjdv6n10-330>
- Silva, R. M. (2022). Classificação de Termogramas do Pé Diabético Utilizando Machine Learning.
- Silveira, C. B. (11 de 02 de 2016). Indústria 4.0: O que é, e como ela vai impactar o mundo. Obtido de Citisystems: <https://www.citisystems.com.br/industria-4-0/>
- Simplício, P. V. (2016). Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT- SERGIPE. Manipuladores robóticos industriais, pp. 85-85.
- Souza, M. P. de. (2020, December). Explicabilidade de modelos de visão computacional aplicados à detecção de COVID-19 em radiografias: <https://hdl.handle.net/10438/30352>
- Staff , C. (2022, August 11). Deep Learning vs. machine learning: A beginner’s guide. Coursera. <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>
- Venturelli, M. (2021). Indústria 4.0: Uma Visão da Automação Industrial. Obtido de Automação Industrial: <https://www.automacaoindustrial.info/industria-4-0-uma-visao-da-automacao-industrial/>
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A brief review. Computational Intelligence and Neuroscience, 2018, 1–13. <https://doi.org/10.1155/2018/7068349>
- Wikipédia. (30 de 5 de 2022). Automated guided vehicle. Obtido em 13 de 6 de 2022, de https://en.wikipedia.org/wiki/Automated_guided_vehicle
- Zillmann da Silva, L. A., Ferreira Vidal, V., Santiago Cerqueira, A., Mello Honório, L. de, Ferreira da Silva, M., Ferreira dos Santos, M., & Lima de Carvalho, A. (2018). Reconhecimento automático de Componentes da Rede Elétrica de Ferrovias utilizando redes Neurais Convolucionais. Proceedings XXII Congresso Brasileiro de Automática. <https://doi.org/10.20906/cps/cba2018-0693>