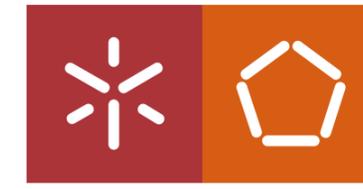


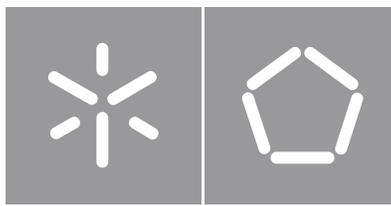


Mauro Daniel Monteiro Queirós

**Célula de trabalho para montagem de
pequenos componentes em PCB**

Universidade do Minho
Escola de Engenharia





Universidade do Minho

Escola de Engenharia

Mauro Daniel Monteiro Queirós

**Célula de trabalho para montagem de
pequenos componentes em PCB**

Dissertação de Mestrado

Mestrado Integrado em Engenharia Eletrónica Industrial e
Computadores

Controlo, Automação e Robótica

Trabalho efetuado sob a orientação do(a)

Professor Doutor Jaime Francisco Cruz Fonseca

Direitos de autor e condições de utilização do trabalho por terceiros

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos. Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada. Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição-NãoComercial
CC BY-NC

<https://creativecommons.org/licenses/by-nc/4.0/>

Agradecimentos

Tendo chegado ao fim esta caminhada académica, desejo exprimir os meus sinceros agradecimentos a todos aqueles que de alguma forma me acompanharam nesta jornada e contribuíram para a conclusão deste importante capítulo da minha vida.

Inicialmente, gostaria de agradecer ao meu orientador, Doutor Jaime Fonseca, por me ter guiado e incentivado nesta caminhada. Além disso, demonstrou um acompanhamento muito próximo ao longo de todo o trabalho realizado, e mostrou-se sempre disponível para me ajudar e guiar pelos diferentes obstáculos, garantindo o sucesso do trabalho. Agradeço ainda a forma como sempre abordou as minhas dificuldades, tentando sempre da melhor forma fornecer os ingredientes para que eu fosse capaz de avançar. Um sincero obrigado, não só pelo acompanhamento próximo e por todo o conhecimento e sabedoria transmitidos, mas também pela relação criada ao longo deste projeto.

Gostaria também de agradecer a todos os colegas que trabalharam no laboratório MotionLab direta ou indiretamente comigo, que também me apoiaram e souberam guiar em diferentes fases do trabalho. Um agradecimento também aos meus colegas e amigos que fiz ao longo deste percurso académico que de alguma forma tornaram este percurso memorável, em especial ao meu amigo João Lobato Pereira.

Aos meus pais, Henrique Queirós e Paula Queirós, que tornaram possível a minha caminhada nesta universidade, que se sacrificaram para que eu me tornasse mestre em Engenharia Eletrónica Industrial e de Computadores, mostrando sempre apoio, incentivando em todos os momentos menos felizes e obstáculos sentidos, e pelo estímulo dado para que tivesse a força necessária para continuar esta jornada. Ao meu irmão, Sandro Queirós, que sempre esteve presente, para me aconselhar, para me guiar, servindo sempre o seu propósito de irmão mais velho e de exemplo a seguir. À minha irmã, Tatiana Queirós, que também como irmã mais velha, demonstrou presença mesmo não estando perto, me aconselhou em alturas menos boas, e me mostrou sempre uma solução para todos os problemas sentidos. À minha companheira, Diana Oliveira, que ao longo desta jornada transmitiu confiança, me deu energia para avançar, demonstrou paciência e suportou fases menos felizes. Um agradecimento sincero a toda a minha família, que longe ou perto, demonstraram sempre carinho e força para que eu conseguisse atingir este meu objetivo.

A estes pilares, a estes alicerces, nada mais a dizer que um muito obrigado por tudo.

Este trabalho é co-financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER) através do Programa Operacional Competitividade e Internacionalização (COMPETE 2020) do Programa Portugal 2020 [Projecto nº 45070, "FlexASComp"; Referência de Financiamento: POCI-01-0247-FEDER-045070].

Declaração de integridade

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Resumo

Na montagem de uma placa de circuito impresso (PCB), quando todos os componentes são dispositivos montados à superfície (SMD), o processo de montagem é automático. No entanto, se a placa incluir componentes de pino através de furos (PTH), estes têm de ser inseridos manualmente, o que leva a atrasos na linha de produção.

Neste contexto, o presente trabalho propõe e valida uma célula de trabalho para a inserção automática de diferentes conectores e componentes eletrônicos, como condensadores, tiristores, entre outros (do tipo PTH), em placas de circuito impresso, baseada num robô colaborativo, num sistema de visão e num controlador lógico programável (PLC, do inglês *Programmable Logic Controller*). Especificamente, foi utilizado o robô colaborativo TM5-700, capaz de realizar movimentos rápidos e seguros, garantindo uma tarefa de *pick and place* precisa e eficiente, o sistema de visão externo FH1050 da OMRON, capacitado de funcionalidades de melhoramento de imagem e inspeção, e ainda o PLC NX102-9020, capaz de monitorizar e controlar a célula de trabalho. Para além da tarefa de *pick and place*, o robô está ainda encarregue de, através do seu sistema de visão interno, fazer a leitura do *part number* da PCB e a referenciação da mesma, garantindo a correta inserção de todos os componentes nas placas. Por sua vez, o sistema de visão externo valida cada componente antes da sua inserção, dividindo para tal o componente em secções e contabilizando o número de pinos presentes face ao esperado. Por fim, o PLC controla o fluxo da célula de trabalho desenvolvida, recebendo os diferentes sinais provenientes dos restantes sistemas e sensores, e posteriormente tomando decisões e dando ordens para que os sistemas realizem as suas respetivas tarefas de acordo com o fluxo de trabalho. Os diferentes módulos foram validados, tendo sido realizados testes à rapidez e segurança do sistema robô, à fiabilidade e precisão do sistema de visão externo, e ainda à capacidade de monitorização do PLC, bem como de comunicação à base de dados. Os resultados obtidos foram satisfatórios, tendo sido verificados todos os requisitos propostos inicialmente para a célula de trabalho.

Em suma, desenvolveu-se uma célula de trabalho operacional capaz de substituir o processo manual na montagem de PCBs com componentes do tipo PTH, otimizando este processo e mantendo os altos níveis de desempenho.

Palavras-Chave

Robô colaborativo, Montagem de PCBs, Componentes PTH, Célula de trabalho.

Abstract

When assembling a printed circuit board (PCB), when all components are surface mount devices (SMD), the process can be automatic. However, if the board includes pin through-hole (PTH) components, these must be manually inserted, which leads to delays in the production line.

In this context, this work proposes and validates a work cell for the automatic insertion of different connectors and electronic components, such as capacitors, thyristors, among others (of the PTH type), in printed circuit boards, based on a collaborative robot, a vision system and a programmable logic controller (PLC). Specifically, it was used the TM5-700 collaborative robot, capable of fast and safe movements, ensuring an accurate and efficient pick and place task, the OMRON FH1050 external vision system, integrating image enhancement and inspection functionalities, and also the NX102-9020 PLC, capable of monitoring and controlling the work cell. Besides the pick and place task, the robot is also in charge of, through its internal vision system, reading the PCB part number and perform its referencing, ensuring the correct insertion of all components in the board. In turn, the external vision system validates each component before its insertion, dividing it into sections and counting the number of pins present compared to what is expected. Finally, the PLC controls the flow of the developed work cell, receiving the different signals from the other systems and sensors, and then making decisions and giving orders to the other systems to perform their respective tasks according to the workflow. The different modules were validated, and tests were made to the speed and safety of the robot system, the reliability and precision of the external vision system, and the PLC monitoring capacity, as well as the database communication features. The results obtained were satisfactory, having verified all the requirements initially proposed for the work cell.

In short, an operational work cell was developed capable of replacing the manual process in the assembly of PCBs with PTH components, optimizing this process and maintaining high levels of performance.

Keywords

Collaborative robot, PCB assembly, PTH components, Work cell.

Índice

1.	Introdução	1
1.1	Enquadramento e motivação	1
1.2	Objetivos	2
1.3	Estrutura da dissertação	3
2.	Estado de arte.....	5
2.1	Indústria 4.0.....	5
2.2	Fundamentos de Robótica	7
2.2.1	Constituição, movimentos, trajetórias e classificação	7
2.2.2	Cinemática	11
2.2.3	Robôs industriais vs. robôs colaborativos.....	13
2.3	Fundamentos de Visão por Computador.....	16
2.3.1	Conceitos básicos	17
2.3.2	Algoritmos de processamento de imagem	19
2.3.3	Principais aplicações dos sistemas de visão	22
2.3.4	Componentes do sistema de visão	23
2.4	Montagem de PCBs.....	24
3.	Requisitos e arquitetura do sistema	27
3.1	Robô TM5-700	27
3.2	Sistema de visão	31
3.3	Controlador Lógico Programável, PLC	37
4.	Métodos.....	38
4.1	Implementação do sistema robô	39
4.1.1	Movimentação	40
4.1.2	Visão interna do sistema robô	41
4.2	Implementação do sistema de visão	45
4.3	Implementação do PLC	47
4.3.1	Integração do sistema de visão externo	52

4.3.2	Integração da base de dados.....	54
5.	Testes e Discussão.....	57
5.1	Testes	57
5.1.1	Quanto tempo leva para o <i>cobot</i> preencher uma PCB.....	57
5.1.2	Fiabilidade do sistema de visão externo	60
5.1.3	Monitorização e aquisição de dados	63
5.1.4	Validação do fluxo de informação para a base de dados local	64
5.1.5	Implementação do protocolo OPC-UA para disponibilização de variáveis	65
5.2	Discussão.....	66
6.	Conclusão.....	69
	Referências Bibliográficas.....	70
	Anexo A - Work cell for assembling small components in pcb	72
	Anexo B - Human-robot collaboration (hrc) with vision inspection for pcb assembly.....	77

Índice de Figuras

Figura 1 - Conjunto de componentes: a) SMD [1]; e b) PTH.	1
Figura 2 - Ilustração geral da célula do trabalho a desenvolver.	3
Figura 3 - Indústria 4.0 [4].	6
Figura 7 - Os 4 tipos de juntas simples [12].	8
Figura 8 - Configuração cartesiana [14].	9
Figura 9 - Configuração cilíndrica [14].	9
Figura 10 - Configuração esférica [14].	9
Figura 11 - Configuração SCARA [14].	10
Figura 12 - Configuração articulada [14].	10
Figura 4 - Cinemática direta e inversa [10].	11
Figura 5 - Cinemática direta [11].	12
Figura 6 - Cinemática inversa [11].	12
Figura 13 - Robô Unimate [16].	13
Figura 14 - Robô colaborativo TM5-700 [20].	16
Figura 15 - Mudança de aparência de um objeto [21].	17
Figura 16 - Pinhole camera model [23].	18
Figura 17 - Detecção de imagem [23].	18
Figura 18 - Modelo RBG [23].	19
Figura 19 - Exemplo de aprimoramento de imagem [28].	20
Figura 20 - Exemplo de segmentação por <i>thresholding</i> [31].	21
Figura 21 - Partilha de informação entre os três subsistemas da célula de trabalho desenvolvida.	28
Figura 22 - Interface de programação do <i>software TMflow</i> [39].	29
Figura 23 - Sistema de visão interno do robô TM5-700 [20].	29
Figura 24 - Funcionalidades de melhoramento de imagem existentes no <i>TMvision</i> [39].	30
Figura 25 - Sistema de visão com controlador e sensor [40].	32
Figura 26 - Search function [41].	33
Figura 27 - Shape Search III [41].	33
Figura 28 - Edge Pitch [41].	34
Figura 29 - Defect [41].	34

Figura 30 - Fine Matching [41].	35
Figura 31 - Filtering [41].	35
Figura 32 - Calculation [41].	36
Figura 33 - Parallel Judgement Output [41].	36
Figura 34 - Célula de trabalho proposta.	38
Figura 35 - Fluxograma de operação do subsistema robô.	39
Figura 36 - Movimentação em dois passos, <i>waypoint motion</i> .	40
Figura 37 - Definições de segurança homem-máquina.	41
Figura 38 - Funcionalidades disponíveis no sistema de visão interno: a) funcionalidades de identificação; b) funcionalidades de procura.	41
Figura 39 - Exemplo de um código de barras na PCB.	42
Figura 40 - Funcionalidade <i>código de barras</i> implementada no <i>TMvision</i> .	42
Figura 41 - Exemplo das <i>marcas fiduciais</i> numa PCB.	43
Figura 42 - Funcionalidade <i>Fiducial Marks</i> implementada no <i>TMvision</i> .	44
Figura 43 - Identificação das marcas fiduciais e da área de procura.	44
Figura 44 - Detecção das marcas fiduciais na placa de montagem.	45
Figura 45 - Fluxograma de operação do subsistema visão.	46
Figura 46 - Componente dividido em secções e número de pinos por secção.	46
Figura 47 - <i>Vision block</i> .	47
Figura 48 - Fluxograma de operação da célula de trabalho, e informação partilhada entre subsistemas.	48
Figura 49 - <i>TMmodbus function</i> , bloco de comunicação direta com o robo TM5-700.	49
Figura 50 - Interpretação do código de barras, e retorno da informação.	50
Figura 51 - Preenchimento e interpretação de dados da PCB em estudo.	51
Figura 52 - Variáveis Modbus.	51
Figura 53 - <i>User area</i> : a) Input; b) Output.	52
Figura 54 - Duas estruturas de dados para acesso aos comandos e à área de resposta.	52
Figura 55 - Registo das <i>tags</i> de entrada e saída.	53
Figura 56 - Estabelecer as conexões das respetivas <i>tags</i> .	53
Figura 57 - Comandos específicos para o sistema de visão externo.	53
Figura 58 - Bloco <i>testpins4</i> .	54
Figura 59 - Tabela com as variáveis a guardar do sistema.	54

Figura 60 - Estrutura com as variáveis da base de dados e variável do tipo <i>SQLData</i>	55
Figura 61 - Conexão à base de dados no <i>software Sysmac Studio</i>	55
Figura 62 - Blocos de funções para implementação da base de dados.	55
Figura 63 - Ambiente da célula de trabalho.....	57
Figura 64 - Fluxograma da tarefa principal da PCB tipo 4.....	58
Figura 65 - Fluxograma da tarefa comparativa ao ser humano.	59
Figura 66 - Exemplos dos componentes utilizados.	60
Figura 67 - Componente em bom estado, apresentando uma contagem de 44 pinos.	61
Figura 68 - Componente em mau estado, sendo possível observar qual o pino que se encontra danificado.	62
Figura 69 - Interface construída no <i>software CX-Designer</i>	63
Figura 70 - Interface durante iterações sequenciadas.....	64
Figura 71 - Observação das variáveis da tabela criada na base de dados.	65
Figura 72 - Visualização em tempo real das variáveis.....	66

Índice de Tabelas

Tabela 1 - Vantagens e desvantagens dos robôs colaborativos	15
Tabela 2 - Troca de informação entre o PLC e os restantes subsistemas.	49
Tabela 3 - Tempo necessário para preencher a PCB a diferentes percentagens de velocidade. ...	58

Lista de Abreviaturas, Siglas e Acrónimos

DH	Denavit-Hartenberg
DOF	<i>Degrees of Freedom</i>
I/O	<i>Input/Output</i>
IoT	<i>Internet of Things</i>
PCB	<i>Printed Circuit Board</i>
PLC	<i>Programmable Logic Controller</i>
PTH	<i>Pin Through-Hole</i>
PTP	<i>Point-to-Point</i>
ROI	<i>Region of Interest</i>
SMD	<i>Surface-Mounted Device</i>
SMT	<i>Surface-Mounted Technology</i>
THT	<i>Through-Hole Technology</i>

1. INTRODUÇÃO

Este primeiro capítulo descreve o enquadramento e a motivação para a presente dissertação, os objetivos propostos para a sua realização e uma breve descrição da estrutura do documento da dissertação.

1.1 Enquadramento e motivação

A montagem de placas de circuito impresso (PCB, do inglês *printed circuit board*) contempla duas tecnologias: a tecnologia de montagem de superfície SMT (do inglês *surface-mounted technology*), onde são utilizados componentes de montagem em superfície, SMD (do inglês *surface-mounted device*; Figura 1a), os quais são soldados diretamente na superfície de um dos lados da placa de circuito impresso, permitindo aproveitar ambas as faces da PCB e, assim, economizar espaço e reduzir o tamanho da placa; e a THT (do inglês *through-hole technology*), que utiliza componentes PTH (do inglês *pin through-hole*; Figura 1b), e na qual se inserem os pinos do componente em furos efetuados na placa de circuito impresso e soldam-se na face oposta.

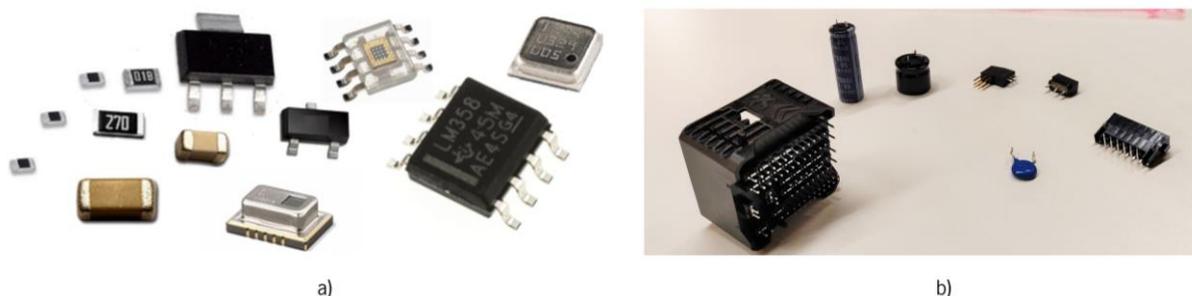


Figura 1 - Conjunto de componentes: a) SMD [1]; e b) PTH.

Atualmente, os componentes SMD são inseridos de forma automática por robôs *pick and place*. Estes são robôs simples, de dois ou três eixos, de alta precisão e velocidade, capazes de inserir um elevado número de componentes num curto intervalo de tempo. Na sua maioria, estes encontram-se destinados a pequenos grupos de componentes, mas alguns possuem uma maior flexibilidade, podendo ser utilizados para um conjunto alargado de componentes SMD distintos.

Em contrapartida, componentes do tipo PTH não são possíveis de serem inseridos de forma automática em PCBs por este tipo de robôs, sendo atualmente este processo realizado de forma manual, isto é, com o recurso a um operador humano (através da técnica de soldadura acima descrita). A inserção

por este método possui algumas vantagens, como a facilidade na prototipagem e realização de testes, mas apresenta desvantagens significativas em contexto industrial. De facto, por necessitar de um operador humano está associada a uma menor eficiência e à necessidade de pausas ocasionais no processo de montagem, seja devido a erros, falhas ou simplesmente pela dificuldade em inserir um componente com um vasto número de pinos, provocando deste modo aumentos nos custos de produção.

Nas últimas décadas, o uso de robôs industriais tem vindo a crescer em inúmeras aplicações, visando a substituição dos operadores encarregues por trabalhos repetitivos, não saudáveis, ou então, tarefas com uma necessidade elevada de precisão. Estes robôs encontram-se, no entanto, limitados à sua programação, bem como à sua essência de trabalharem em espaços isolados. Mais recentemente, o surgimento de robôs colaborativos permitiu quebrar estas barreiras, possibilitando a interação humano-robô e trazendo uma maior flexibilidade no meio industrial, pois permite aliar a inteligência e adaptabilidade do operador humano com a precisão e a repetibilidade do robô.

Com vista a tirar partido do surgimento dos robôs colaborativos, a Bosch, em parceria com a Universidade do Minho e a DIB4T, encontra-se atualmente a trabalhar no desenvolvimento de um novo processo de industrialização: uma célula de trabalho para a inserção automática de diferentes conectores e componentes eletrónicos, como condensadores, tiristores, entre outros (do tipo PTH), em placas de circuito impresso.

1.2 Objetivos

O presente trabalho está inserido no projeto acima mencionado e visa, especificamente, o projeto, implementação e validação da referida célula de trabalho, utilizando entre outras tecnologias, um robô colaborativo, capaz de trabalhar ora em conjunto com o operador humano ora de forma isolada, e um sistema de visão artificial, capaz de realizar testes e validar os componentes e a sua inserção na PCB. Esta célula deverá ser automatizada e autossuficiente, sendo capaz de se adaptar a diferentes situações, como a alteração de componentes a inserir ou o ajuste na abordagem à placa de circuito impresso.

A implementação desta célula de trabalho encontra-se dividida em três módulos principais (Figura 2): o primeiro valida a qualidade dos pinos do conector a inserir, o segundo efetua a inserção do conector na PCB, e o terceiro verifica o resultado da inserção para garantir a adequabilidade da placa.

Especificamente, no primeiro módulo, o robô deverá pegar no conector e levá-lo até à zona onde se encontra o sistema de visão artificial. Aqui, com o recurso a funções de processamento digital de imagem, o sistema de visão deverá validar se o conector está apto para ser inserido, verificando se existem pinos dobrados ou quebrados. Enquanto esta validação ocorre, o robô deverá aguardar pelo resultado da

inspeção do componente. Se o resultado for positivo, o robô deverá passar ao estágio seguinte. Caso contrário, deverá descartar o componente defeituoso e repetir o processo para outro componente.

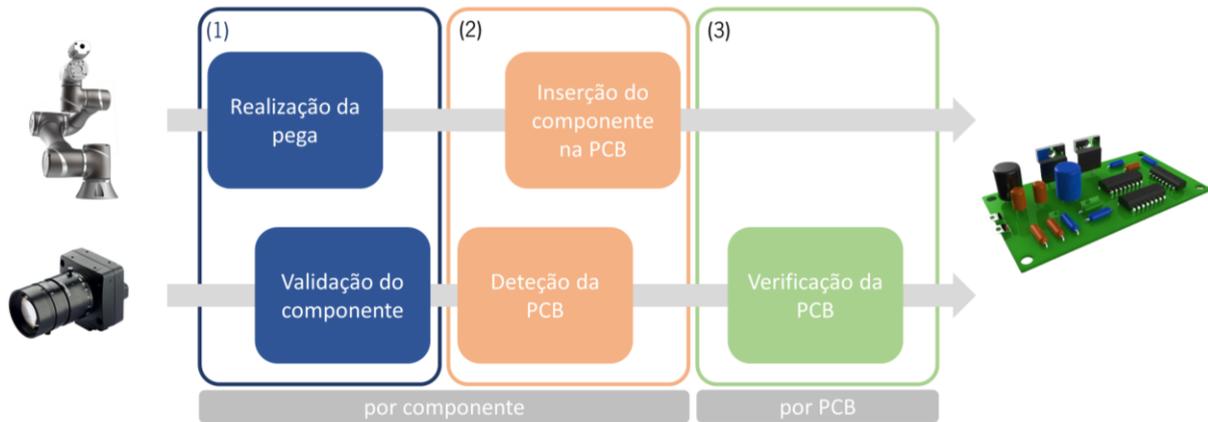


Figura 2 - Ilustração geral da célula do trabalho a desenvolver.

No segundo módulo, e no caso de o componente estar em perfeito estado, o robô deverá realizar a inserção automática deste na placa de circuito impresso. De notar que, apesar da posição relativa do componente na PCB ser conhecida, a posição desta última poderá variar uns milímetros/graus de placa para placa. Como tal, utilizando novamente o sistema de visão artificial, o robô deverá ajustar-se perante a posição da placa, para assim detetar a posição correta do componente na PCB.

No último módulo, após a inserção dos respetivos componentes numa determinada PCB, o robô, através do sistema de visão artificial, deverá verificar se estes de facto se encontram bem inseridos. Posteriormente aguardará pela chegada de uma nova placa de circuito impresso, vazia, para iniciar novamente o processo.

O desenvolvimento com sucesso desta nova célula de trabalho permitirá tornar a inserção deste tipo de componentes PTH em PCBs mais rápida, mais precisa, e com a mesma qualidade de quando os componentes eram inseridos manualmente. Este avanço permitiria retirar quase na totalidade a intervenção humana existente previamente e, deste modo, contribuiria para um dos maiores desafios atuais do setor eletrónico que passa pela redução de custos e aceleração de processos, sem nunca abrir mão da qualidade destes.

1.3 Estrutura da dissertação

Esta dissertação está dividida em seis capítulos, organizados sequencialmente de modo a permitir ao leitor acompanhar o processo realizado e compreender o trabalho desenvolvido.

No primeiro capítulo, Introdução, foi efetuado o enquadramento do projeto, apresentando os objetivos deste e a respetiva estrutura da dissertação.

No segundo capítulo, Estado de Arte, são exploradas as temáticas essenciais para a compreensão do projeto. Inicialmente é efetuada uma introdução ao tema subjacente deste projeto, a indústria 4.0, apresentando algumas das vantagens e novidades que advém deste novo conceito. De seguida, são abordados alguns conceitos fundamentais dos dois principais tópicos deste projeto, a robótica e a visão por computador. Por fim, é apresentada e sumariada a literatura existente sobre os diferentes trabalhos já realizados nesta área, mencionando alguns aspetos relevantes para o objetivo deste projeto.

No terceiro capítulo, Requisitos e arquitetura do sistema, é feita uma apreciação dos requisitos a cumprir na execução desta célula de trabalho, demonstrando quais são os componentes principais da mesma: robô colaborativo, sistema de visão externo e um controlador lógico programável, PLC (do inglês *programmable logic controller*). Aqui é explicado as diferentes características de cada sistema, apresentando as suas vantagens e principais capacidades.

No quarto capítulo, Métodos, são explorados os três sistemas mencionados anteriormente. Assim, é descrita detalhadamente a metodologia empregue em cada sistema, de forma a ser possível compreender a implementação da referida célula de trabalho.

No quinto capítulo, Testes e Discussão, são apresentados os testes efetuados para a validação da célula de trabalho, bem como verificação do cumprimento dos requisitos estabelecidos inicialmente. No final, é feita uma análise aos resultados obtidos.

No sexto e último capítulo, Conclusão, são efetuadas conclusões gerais do trabalho e discussão de potencial trabalho futuro.

2. ESTADO DE ARTE

Este capítulo apresenta conceitos teóricos relevantes no contexto do presente projeto, bem como uma revisão da literatura que serviu de apoio para os desenvolvimentos efetuados.

Tendo por base o tema subjacente ao presente projeto, inicialmente introduz-se o conceito da indústria 4.0, bem como as principais características trazidas por esta no contexto das fábricas inteligentes (*smart factories*).

Posteriormente, e visto serem duas áreas de conhecimento fundamentais no âmbito da célula de trabalho que se pretende desenvolver, é efetuada uma introdução teórica à robótica, incluindo a temática dos robôs colaborativos, e à visão por computador. Por um lado, na robótica, são abordados alguns conceitos fundamentais como a cinemática direta e inversa, tipos de movimentos e trajetórias possíveis de serem executadas pelo robô, configurações existentes e classificação dos robôs. Para além disso, também é apresentada uma comparação entre robôs industriais e colaborativos. Por outro lado, são mencionados alguns tópicos de visão artificial e processamento de imagem, incluindo técnicas de melhoria, segmentação e análise de imagem, e ainda de deteção de objetos. Mais ainda, são abordadas as principais áreas onde estes sistemas são integrados, bem como os seus principais constituintes e a importância destes na correta implementação de um sistema de visão artificial.

Por último, integrando os tópicos acima mencionados e visto que o sistema a desenvolver tem como objetivo a inserção de componentes em placas de circuito impresso, são abordados alguns tópicos relevantes no que toca a este processo e são apresentadas as tecnologias utilizadas na área. Além disto, é apresentada a literatura existente no tópico, mencionado trabalhos relevantes para o objetivo desta dissertação.

2.1 Indústria 4.0

A Indústria 4.0 veio com o objetivo de, não só melhorar a comunicação entre o homem e a máquina, possibilitando uma maior informação sobre os dados e operações realizadas, mas também aprimorar a comunicação máquina-máquina. Para tal ser possível, esta revolução industrial engloba essencialmente algumas tecnologias para automação e troca de dados utilizando conceitos recentes como *Big Data*, *Cloud Computing*, *Cyber-physical systems* e também *Internet of Things (IoT)*. Estes avanços permitem a implementação das *smart factories* [2], isto é, fábricas que possuem processos de produção inteligentes e autossuficientes, devido à interligação digital existente entre as diferentes máquinas e dispositivos, com

o único objetivo de caracterizar estes processos de uma forma digital, onde o fluxo de informação entre dispositivos se dará num meio com a mínima intervenção humana. Para além disto, a ideia foca-se na existência de uma arquitetura flexível e adaptável a diferentes protocolos e dispositivos, com o intuito de fornecer a estas indústrias a capacidade de criarem redes inteligentes ao longo da cadeia de valor, para poderem controlar e comandar estes processos de produção [3] (Figura 3).



Figura 3 - Indústria 4.0 [4].

Como todas as evoluções e revoluções, existem sempre impactos positivos e negativos. O principal impacto negativo do surgimento das *smart factories* está ligado à diminuição do número de postos de trabalho. De facto, diversas empresas têm vindo a substituir grande parte da mão de obra por robôs e sistemas automatizados, mantendo apenas os trabalhadores necessários para o controlo, calibração e/ou manutenção destes sistemas, ou necessários nos poucos processos que ainda possam requerer um operador humano. No entanto, a Indústria 4.0 proporcionou bastantes benefícios [5] [6]:

- **Produção modular** → veio facilitar a troca e atualização de programas, possibilitando uma maior rapidez na troca de produtos e aumentando, por isso, a sua flexibilidade;
- **Operações em tempo real**
- **Operações integradas** → permitindo o contacto entre toda a cadeia de valor, desde o cliente ao fornecedor;
- **Otimização** → aumento da produtividade e qualidade, e diminuição dos custos, uma vez que as máquinas são mais precisas e não necessitam de descanso;

- **Interoperabilidade** → fornece uma maior transparência nas operações e na troca de dados, possibilitando a conexão entre os diferentes órgãos do processo. É dentro deste contexto que o desenvolvimento desta célula de trabalho se insere.

2.2 Fundamentos de Robótica

A palavra *robot* deriva da língua checa (Robota) e significa “trabalhador forçado” ou “escravo/servo”. Foi pela primeira vez utilizada em 1921, por Karel Capek, um escritor checo, que descreveu os robôs como sendo máquinas que se assemelham a pessoas na sua estrutura, mas que acabam por desempenhar o dobro do trabalho e nunca apresentavam sentimentos humanos. Com o passar do tempo, este termo veio ocupar a mente do homem até que, em 1950, com a contribuição de Isaac Asimov, foram definidas as três Leis da Robótica [7]:

- **1ª Lei** → Um robô não pode maltratar um ser humano, ou pela sua passividade deixar que um ser humano seja maltratado;
- **2ª Lei** → Um robô deve obedecer às ordens dadas por um ser humano, exceto se entrarem em conflito com a 1ª lei;
- **3ª Lei** → Um robô deve proteger a sua própria existência desde que essa proteção não entre em conflito com as duas leis anteriores.

Com a evolução da robótica e de tudo o que de si depende, os avanços tecnológicos permitiram estabelecer salas de trabalho, e secções de operação totalmente automatizadas, isto é, sem qualquer tipo de presença humana durante todo o processo, levando a uma maior presença deste mundo robótico no atual mundo industrial.

Posto isto, são mencionados de seguida alguns fundamentos essenciais no que toca à robótica industrial, desde a cinemática dos robôs, a sua constituição, tipo de movimento e trajetórias, entre outros aspetos. Por fim, é feita uma comparação entre os primeiros robôs a aparecerem no mundo industrial, e os robôs colaborativos, recentemente criados e inseridos no mundo fabril.

2.2.1 Constituição, movimentos, trajetórias e classificação

Mecanicamente um manipulador robótico é um equipamento formado por uma cadeia de elos rígidos conectados entre si por diferentes tipos de juntas, sendo estas as responsáveis pela movimentação do robô. Em robôs industriais, dois tipos de juntas são geralmente utilizadas, as lineares (ou prismáticas, definidas por um movimento linear) e as rotativas (movimento em torno de um eixo de

rotação), podendo estas últimas ser divididas em 3 tipos de acordo com as direções dos elos de entrada e de saída em relação ao eixo de rotação. Assim, no total, existem 4 tipos de juntas simples (Figura 4):

- **Junta prismática** → Movimento linear do elo de saída em relação ao elo de entrada (Figura 4);
- **Junta rotacional** → O elo de saída realiza uma rotação em torno da junta, sendo o eixo de rotação perpendicular a ambos os elos (Figura 4);
- **Junta de torção ou torsional** → O eixo de rotação é paralelo aos elos de entrada e saída (Figura 4);
- **Junta revolvente** → O eixo de rotação é paralelo ao elo de entrada, mas o elo de saída é perpendicular a este (Figura 4).

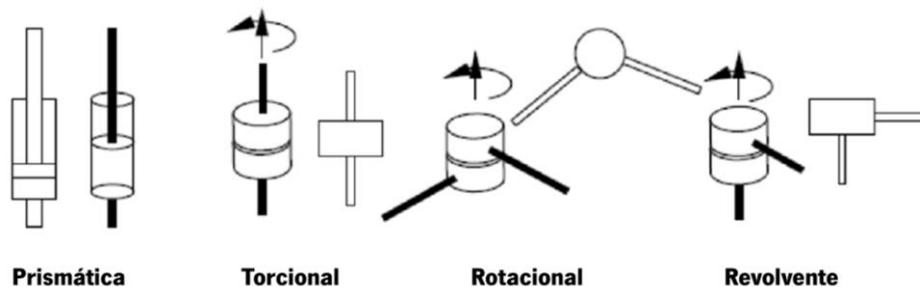


Figura 4 - Os 4 tipos de juntas simples [12].

A localização da junta do robô é definida em relação à sua junta vizinha. Neste caso a relação entre juntas sucessivas é descrita por matrizes de transformação homogêneas 4×4 , onde o número de matrizes necessárias para descrever a constituição do robô indica o número de graus de liberdade que o robô possui. Os graus de liberdade que um robô possui, DOF (do inglês, *degrees of freedom*), é o número total de movimentos independentes que este pode efetuar.

Consoante o comprimento dos elos e os movimentos permitidos pelas juntas, cada robô possui um espaço de trabalho único, isto é, cada robô possui uma região dentro da qual o manipulador poderá posicionar a sua extremidade. Dependendo da aplicação que se pretende atribuir ao manipulador e as juntas que o constituem, o processo de seleção é mediante a sua dimensão e capacidade de carga, precisão e tipo de movimento pretendido. De acordo com a ISO 8373, estes encontram-se divididos nas seguintes categorias [13]:

- **Cartesiano** → Possui apenas juntas prismáticas, permitindo unicamente movimentos lineares e sendo normalmente utilizado em aplicações de manipulação e movimentação (Figura 5);

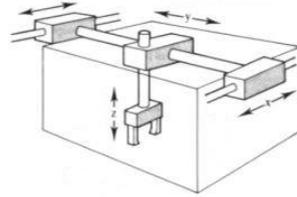


Figura 5 - Configuração cartesiana [14].

- **Cilíndrico** → Possui duas juntas prismáticas e uma junta rotacional, sendo caracterizado por alta velocidade e precisão (Figura 6);

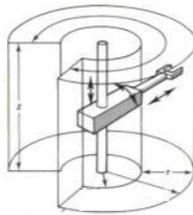


Figura 6 - Configuração cilíndrica [14].

- **Esférico** → Possui duas juntas rotacionais e uma junta prismática, apresentando uma menor resolução de posicionamento ao longo do eixo radial (Figura 7);

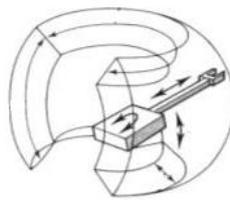


Figura 7 - Configuração esférica [14].

- **SCARA (do inglês *Selective Compliance Assembly Robot Arm*)** → Idêntico à configuração esférica, possuindo apenas uma junta prismática na sua extremidade e permitindo a sua utilização na manipulação de pequenos objetos. São caracterizados pela alta velocidade e precisão (Figura 8);

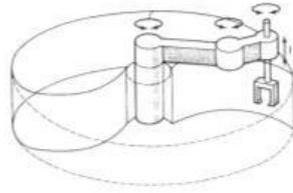


Figura 8 - Configuração SCARA [14].

- **Articulados** → Possui diversas juntas rotacionais, sendo a configuração mais flexível (Figura 9).

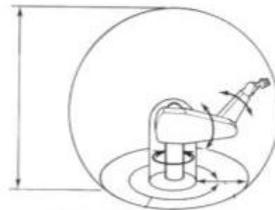


Figura 9 - Configuração articulada [14].

Após relacionar o espaço cartesiano com o tipo de juntas existentes, é necessário planejar a trajetória que o robô irá percorrer para atingir um determinado ponto no espaço. Este estudo engloba um conjunto de métodos que permitem definir as diferentes ações do robô, com a finalidade de atingir o objetivo planejado. Existem dois métodos para realizar o controle de trajetória:

- **Ponto-a-ponto (PTP)**

Neste tipo de trajetória o robô tenta mover a sua extremidade para o ponto destino, utilizando o caminho mais curto possível. Recorre-se à cinemática inversa, e tem a desvantagem de o utilizador não poder prever qual movimento que o robô irá realizar.

- **Trajecória Contínua**

Embora mais lento que o PTP, neste caso gera-se diversos pontos intermédios que serão percorridos sequencialmente. Este tipo de trajetória oferece-nos a possibilidade de definir caminhos mais complexos, e, inclusive, realizar outras operações durante o movimento. Certos controladores mais recentes permitem ao operador interpolar curvas sobre esses mesmos pontos intermédios, ajudando a tornar a trajetória mais fluida e rápida.

Por fim, é necessário ainda referir a classificação de robôs no que toca à sua inteligência. Por vezes, dependendo do tipo de aplicação, pode ser necessário possuir robôs simples, que se limitam a executar

uma determinada tarefa repetitiva, e noutros casos, é necessário que o robô seja capaz de aprender com os erros e se adaptar às características envolventes da área de trabalho. Deste modo, os robôs podem ser classificados em [15]:

- **Manual** → Operado diretamente pelos funcionários;
- **Sequência fixa** → Pode realizar várias funções consoante as informações predefinidas;
- **Sequência variável** → Possui a vantagem das informações dadas para a execução de determinada tarefa poderem facilmente ser alteradas;
- **Playback** → Aprende os movimentos ao observar o operador e executa a tarefa de seguida;
- **Controlo numérico** → O operador apenas escreve um programa usando fitas perfuradas e o robô desempenha a tarefa respeitando a ordem de operações;
- **Inteligentes** → Com base nas informações recebidas por meio de sensores, o robô adapta-se ao meio e decide que tipo de ação deve/pode realizar.

2.2.2 Cinemática

A cinemática é o estudo do movimento dos corpos sem levar em consideração as forças que causam o movimento [8]. Para ser possível analisar o comportamento dos manipuladores industriais robóticos é necessário formular os modelos cinemáticos adequados ao mecanismo do robô em questão.

Em 1955, Denavit e Hartenberg [9] demonstraram que uma transformação entre duas articulações de um robô se descreve a partir de quatro parâmetros apenas. Estes parâmetros são atualmente conhecidos como parâmetros de Denavit-Hartenberg (DH) e tornaram-se essenciais na descrição da cinemática de um robô. A cinemática de um manipulador pode ser dividida em cinemática direta e cinemática inversa, sendo apresentada na Figura 10 a relação existente entre as duas.

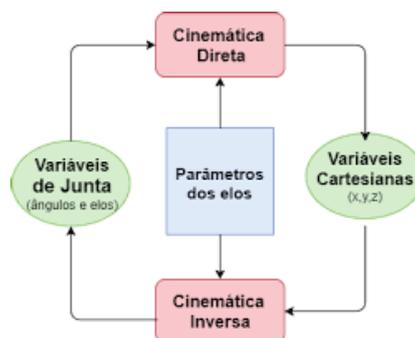


Figura 10 - Cinemática direta e inversa [10].

Cinemática direta

Tal como ilustrado na Figura 11, a cinemática direta diz respeito à determinação da posição (e orientação) do manipulador (juntas, elos ou órgão final) no espaço cartesiano, a partir do conhecimento dos valores dos ângulos das suas juntas.

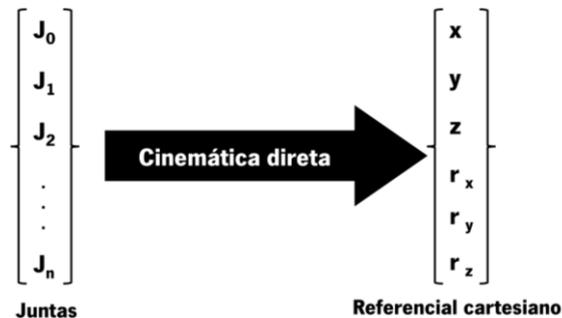


Figura 11 - Cinemática direta [11].

Para tal, é necessário recorrer ao método de Denavit-Hartenberg, mencionado anteriormente. Este método consiste numa série de regras e definições, onde inicialmente é necessário descrever as transformações que ocorrem entre as articulações do robô durante o movimento, sendo estas descritas através dos parâmetros DH. Estes fornecem as informações relativas a ângulos e distâncias entre os referenciais de duas juntas consecutivas, descrevendo respetivamente: a rotação no eixo do Z, “ θ ”, a translação no eixo do Z, “ d ”, a translação no eixo do X, “ l ”, e a rotação no eixo do X, “ α ”. Obtendo estes parâmetros, é então possível calcular as matrizes homogêneas que descrevem as transformações entre juntas. No fim, é possível obter a relação entre o referencial inicial (a base do robô) e o referencial relativo ao órgão final, através da multiplicação de todas as matrizes homogêneas, sendo então possível obter a posição final do manipulador robótico no referencial cartesiano.

Cinemática Inversa

Por sua vez, caso se pretenda colocar o robô numa posição (e orientação) específica, é necessário recorrer à cinemática inversa para se obter o valor dos ângulos das suas juntas (Figura 12).

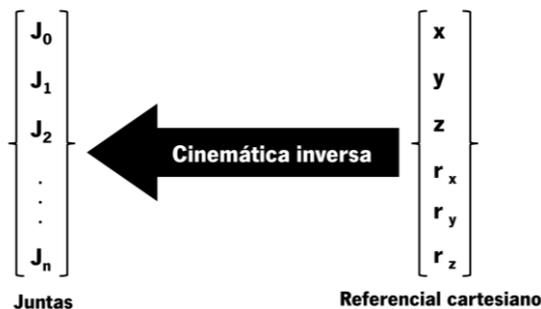


Figura 12 - Cinemática inversa [11].

Para isso, é necessário calcular a posição que cada junta do robô vai ter de estar para atingir a posição pretendida. Para este cálculo existem mais variantes, uma vez que, dependendo do tipo de robô, poderá existir múltiplas possibilidades de posicionar as juntas para a mesma posição do órgão final. De facto, quanto maior o número de juntas de um robô, maior os graus de liberdade deste, e maior é o número de possibilidades de posições diferentes para as juntas. Nesta cinemática existem 2 abordagens diferentes: a abordagem geométrica e a abordagem algébrica.

- A abordagem geométrica é utilizada para os manipuladores mais simples, uma vez que se torna bastante exaustiva e complexa com o aumento dos graus de liberdade do robô. Para manipuladores mais complexos, a abordagem algébrica é a mais utilizada na indústria.

2.2.3 Robôs industriais vs. robôs colaborativos

Os robôs industriais foram os primeiros a aparecer, sendo estes definidos pela ISO como “um manipulador controlado automaticamente, reprogramável, polivalente, programável em três ou mais eixos, que pode funcionar fixo ou em aplicações móveis”. Por serem pesados, grandes, robustos e com uma elevada força de movimentação e carga, tornam-se perigosos para a presença humana, sendo necessário cercas, sensores e sinalização à volta destes numa tentativa de alertar e tornar o ambiente de trabalho mais seguro.

O primeiro robô surgiu nos anos 60 criado por George Charles Devol (Figura 13). Foi denominado de Unimate e veio auxiliar o fabrico de automóveis da *General Motors* (USA), apoiando nas tarefas de movimentação e soldadura, tarefas essas que eram consideradas perigosas para o Homem. Nessa altura, era ainda algo muito limitado, possuía pouca memória para a realização de diferentes tarefas e era difícil de configurar. Com o objetivo de aumentar a flexibilidade deste tipo de robô industrial, ao longo dos anos diferentes empresas foram construindo os seus modelos. Todos estes modelos seriam diferentes, com capacidades diferentes, possuindo apenas uma semelhança: os movimentos que conseguiam executar.



Figura 13 - Robô Unimate [16].

No entanto, a evolução neste ramo industrial não ficou por aqui. Nos anos 90, mais precisamente em 1996, na tentativa de procurar algo mais evoluído, J. Edward Colgate e Michael Peshkin criaram o primeiro robô colaborativo. Este robô é destinado à interação direta com o operador humano dentro do espaço compartilhado para o trabalho. Isto é possível uma vez que o *cobot* (termo utilizado para a designação de robô colaborativo) é construído com o recurso a materiais mais leves, possui bordas arredondadas e permite o controlo da sua velocidade de movimento assim como a força que é exercida. Para além disto, possui uma vasta gama de sensores, e um *software* capaz de ser programável, possibilitando então a segurança necessária para este trabalho cooperativo [17]. Possuindo diferentes opções de utilização como apoio no transporte de materiais, fornecimento de informação em espaços públicos, automatismos em diferentes fases do processo de produção, entre outros, estes robôs possuem 4 tipos de colaboração com o ser humano [17]:

- **Coexistência** → Trabalham lado a lado, mas sem compartilhar o espaço de trabalho;
- **Colaboração sequencial** → Trabalham no mesmo espaço, mas em tempos sequenciais, isto é, não operam ao mesmo tempo sobre a mesma peça;
- **Cooperação** → Trabalham na mesma peça ao mesmo tempo, estando os dois em movimento;
- **Colaboração responsiva** → Trabalham sobre a mesma peça, mas o robô responde em tempo real conforme o movimento do operador humano.

Em suma, a principal distinção entre estes dois tipos de robôs, os robôs industriais tradicionais e os *cobots*, está na colaboração possível com o operador humano e a localização permitida no local de trabalho. Tal como mencionado anteriormente, um robô tradicional está rodeado de barreiras de proteção, redes e sinalização, enquanto os robôs colaborativos não possuem limites de acesso e requerem um menor gasto com sistemas de proteção e segurança. Estes robôs são desenhados para poderem cooperar com o operador, possuindo os recursos e os sensores necessários para reconhecer a proximidade do ser humano. Numa eventual situação extrema, estes robôs possuem a inteligência para compreender que uma maior força ou resistência exercida estará associado a algum imprevisto e poderá tomar atitudes de segurança perante esta situação. Tudo isto faz com que seja possível eliminar a necessidade de gaiolas de segurança. Para além disto, estes robôs são facilmente programáveis. A maior parte dos robôs colaborativos podem ser programados por blocos, sendo apenas necessário ter o conhecimento de algumas variáveis fundamentais para a execução de determinada funcionalidade, enquanto os robôs industriais tradicionais precisam de conhecimento avançado e experiência de

programação. A Tabela 1 resume as vantagens e desvantagens dos robôs colaborativos face aos robôs industriais tradicionais [18].

Tabela 1 - Vantagens e desvantagens dos robôs colaborativos

Vantagens	Desvantagens
Programação mais fácil	Preço relativamente alto
Manutenção mais económica	Para atingir 100% do potencial, é necessário adquirir componentes extra
Maior sensibilidade em contato com o operador	Menor resistência
Dimensões relativamente pequenas	Na eventual colisão com o operador, o robô interrompe o processo
Pode incluir câmaras e sensores para deteção de obstáculos	Ainda possui baixa aplicação industrial, representando apenas cerca de 10% dos robôs em contexto industrial
Maior potencial na indústria atual	
Não há necessidade de utilizar barreiras de proteção	

Embora ainda pouco utilizados na indústria, a procura por uma maior precisão e uma maior velocidade de produção tem levado ao aumento gradual da utilização dos robôs colaborativos na indústria. Estes são programados para substituírem o operador humano, trabalhando sem pausas, na tentativa de obter um maior rendimento e produtividade, sendo utilizados em diversas aplicações, desde a coleta, embalagem e palatização, soldadura, montagem, no manuseamento de materiais ou equipamentos, e ainda na inspeção à qualidade dos produtos [19].

Para ser possível atingir o objetivo desta dissertação, e analisando todos os aspetos até então mencionados, para a célula de trabalho ser capaz de ser adaptável a diferentes componentes, ser possível operar em conjunto com o ser humano de forma a manter a segurança e procurando atingir o máximo de produtividade, rapidez e precisão na inserção de componentes, a escolha de um robô colaborativo aparenta ser essencial. Dentro das opções existentes neste mercado, e procurando especificamente por aspetos como elevada precisão, fluidez de movimentação, um alcance proporcional à célula de trabalho, um elevado número de entradas e saídas digitais para comunicação com sistemas externos, e ainda uma versatilidade elevada na possibilidade de escolha do *gripper* perante a ação a executar, o robô selecionado foi o TM5-700 da Techman Robot (Figura 14). Este robô será descrito em detalhe na secção 3.1.



Figura 14 - Robô colaborativo TM5-700 [20].

2.3 Fundamentos de Visão por Computador

Um dos objetivos desta dissertação é a correta inserção dos conectores nas placas de circuito impresso, assim como a validação da qualidade do conector, isto é, verificar se os pinos se encontram na sua totalidade e corretamente alinhados, sendo para tal necessário compreender alguns conceitos de visão por computador (ou visão artificial) e respectivas técnicas de processamento de imagem.

A inspeção visual é um conceito que existe desde que o ser humano foi capaz de fabricar os seus próprios objetos, sendo apenas necessário então observar a superfície do respetivo objeto para ser possível detetar a existência ou não de falhas. Na atualidade, grande parte das fábricas possuem um controlo de qualidade, e isso passa por fazer uma avaliação das peças produzidas, procurando por imperfeições ou defeitos. Inicialmente, os fabricantes recorriam frequentemente à inspeção visual utilizando um operador humano para a realização desta tarefa, utilizando a sua capacidade visual, num ambiente bem iluminado, e apenas ao manusear a peça manualmente, observando-a de todos os ângulos e com diferenças na incidência de luz, na tentativa de procurar a existência de defeitos prejudiciais ao objetivo funcional da respetiva peça.

Com a necessidade de aumentar a produtividade fabril, mas manter a qualidade de produção, foi essencial criar soluções automatizadas inovadoras capazes de efetuar esta tarefa. Isto levou à criação de *software* e câmaras que possuíam a capacidade de analisar, monitorizar e estudar a peça, bem como procurar pela existência de possíveis defeitos. Nas últimas décadas, tem-se observado um aumento da utilização de sistemas de visão artificial na indústria, aumentando a eficiência dos processos de inspeção. As principais vantagens de montar um sistema de visão artificial são:

- Funcionamento contínuo;**
- Objetividade e qualidade da inspeção das peças;**

- **Rapidez da avaliação;**
- **Obtenção de informações adicionais;**
- **Facilidade de (re)programação.**

Um sistema de visão abrange todas as aplicações (industriais ou não) nas quais hardware e *software* são combinados para fornecer orientação operacional a sistemas e/ou dispositivos com base na captura e processamento de imagens. No caso dos sistemas de visão industriais, estes necessitam de ser robustos, apresentar precisão elevada, alta confiabilidade e estabilidade, bem como um custo moderado. Estes sistemas dependem de sensores digitais existentes dentro das respectivas câmaras e lentes que captam as imagens, de modo a ser possível ao *software* processar, analisar e medir as características pretendidas.

Independentemente da tarefa que se pretenda realizar, o primeiro passo de qualquer sistema de visão passa pela captura do objeto em questão dentro do plano da imagem da câmara. Uma vez que todo o sistema depende desta etapa, o eventual insucesso nesta fase provoca uma maior dificuldade, ou mesmo impossibilidade, em identificar, medir, inspecionar, contar ou medir o objeto em questão. Na maioria dos sistemas de visão, o sistema é programado e configurado para encontrar o objeto com base em padrões, mas em alguns casos estes permitem alguma variação na aparência do objeto em estudo (Figura 15). De notar que, quanto maior a variação permitida, mais complexo terá tipicamente de ser o sistema de inspeção.

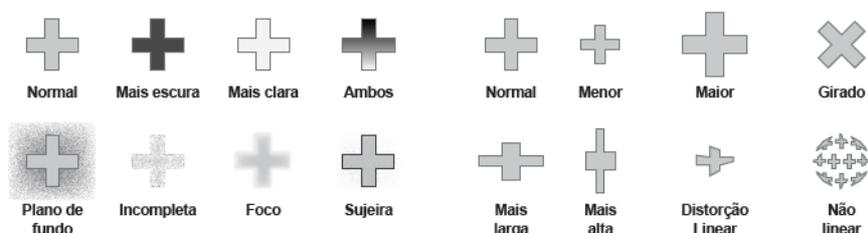


Figura 15 - Mudança de aparência de um objeto [21].

Posto isto, de seguida são mencionados alguns conceitos básicos dentro deste tópico, bem como alguns dos algoritmos de processamento de imagem mais utilizados em contexto industrial.

2.3.1 Conceitos básicos

Pinhole camera model

É o modelo mais simples e o mais útil no que toca à captura de uma imagem por uma câmara [22]. Este modelo descreve a relação matemática existente entre as coordenadas de um ponto no espaço

tridimensional e a sua projeção no plano de imagem de uma câmara *pinhole*. A luz refletida de um objeto é concentrada e projetada, através de um orifício, no plano da imagem de forma invertida (Figura 16).

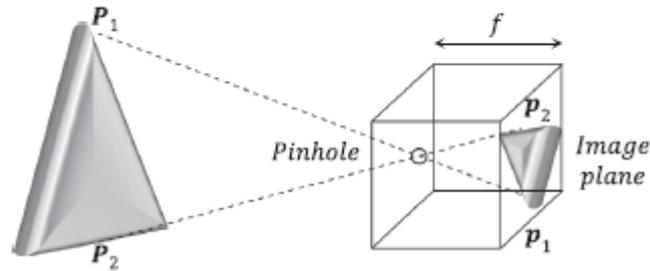


Figura 16 - Pinhole camera model [23].

Neste modelo não se leva em consideração distorções geométricas, bem como a desfocagem dos objetos devido às lentes e, como tal, apenas pode ser usado como uma aproximação do mapeamento de um espaço 3D numa imagem 2D. Eventualmente, parte destas limitações podem ser negligenciadas se a câmara em questão possuir uma elevada qualidade (nomeadamente se a distorção geométrica for muito reduzida) ou utilizando parâmetros adicionais para compensação das coordenadas da imagem. Esta flexibilidade faz com que este modelo seja bastante utilizado na visão computacional [24].

Dispositivos de deteção de imagem

Uma câmara inclui, além de outros componentes, um sensor eletrónico composto por uma matriz de elementos sensíveis capazes de capturar e quantificar a luz que incide na lente da câmara, os quais corresponderão posteriormente aos pixéis da imagem. O valor destes mapeia a intensidade da luz incidente no respetivo elemento sensível numa dada escala de intensidades, sendo normalmente definida entre 0 e 255 (Figura 17). Através deste mapeamento, obtém-se uma matriz bidimensional de M filas por N colunas (sendo M e N dependente da resolução do sensor).

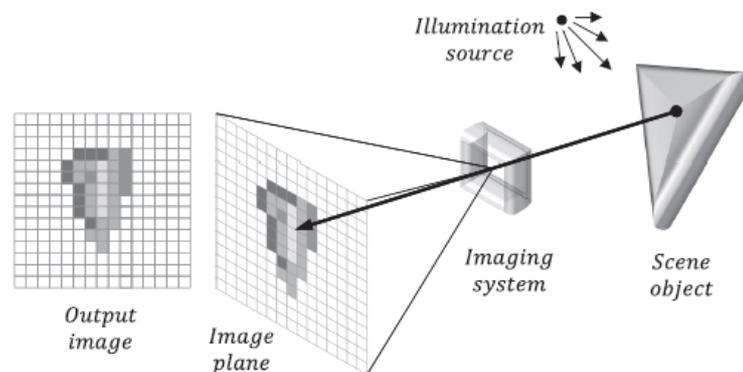


Figura 17 - Deteção de imagem [23].

Modelos de cores

Enquanto sensores monocromáticos apresentam apenas um valor de intensidade por pixel, uma câmara a cores utiliza filtros óticos para quantificar a intensidade de luz nas diferentes gamas de cores (ou espetros), sendo que geralmente utilizam filtros na gama do azul, verde e vermelho. As imagens a cores são normalmente descritas por 3 canais, um por cada uma das gamas acima definidas. Este modelo, RGB, é o mais conhecido e utilizado, sendo utilizado na fotografia convencional e em ecrãs de televisão, computadores, telemóveis, entre outros dispositivos eletrónicos [25]. A Figura 18 ilustra o mapeamento de todas as cores possíveis no modelo RGB, bem como um exemplo dos três canais independentes (vermelho, verde e azul) de uma imagem com o recurso a este modelo.



Figura 18 - Modelo RGB [23].

Apesar de ser o modelo mais utilizado, o modelo RGB não é único e outros podem ser utilizados para representar a mesma imagem, como o YUV, o CMYB, o HLS ou o HSV [23].

2.3.2 Algoritmos de processamento de imagem

O processamento de imagem consiste num conjunto de operações realizadas numa imagem, a fim de obter uma imagem melhorada ou extrair informação útil para uma determinada tarefa ou objetivo.

Ao longo do tempo, têm vindo a ser criados diferentes métodos/algoritmos para deteção, análise, reconstrução, restauração, compressão de dados ou aprimoramento de imagem, entre outros, os quais podem ser agrupados em três grupos principais:

- **Técnicas de baixo nível**, como o aprimoramento de cor ou a remoção de ruído;
- **Técnicas de médio nível**, como a compactação de dados;
- **Técnicas de alto nível**, como a deteção e/ou segmentação de objetos, algoritmos de reconhecimento e captura de dados, entre outros.

De seguida, são apresentados alguns dos algoritmos mais utilizados e relevantes para esta dissertação:

Aprimoramento de Imagem

Este tipo de algoritmo visa melhorar a qualidade da imagem, nomeadamente com vista a melhorar e/ou facilitar a análise computacional desta. Com este tipo de algoritmos é por vezes possível ultrapassar situações em que a qualidade de imagem não é inicialmente suficientemente boa, devido ao equipamento utilizado ou ao ambiente envolvente, permitindo na mesma a futura análise e estudo da imagem [26].

Uma das técnicas mais utilizadas dentro da melhoria da qualidade de imagem são funções de melhoramento do contraste da imagem. Neste caso as operações são geralmente realizadas com base no histograma da imagem [26], [27] (Figura 19). O histograma representa a distribuição das intensidades de uma imagem, isto é, trata-se de um gráfico onde é apresentado a quantidade de pixéis existente para cada tonalidade.



Figura 19 - Exemplo de aprimoramento de imagem [28].

Deteção

Esta é uma importante tarefa na visão computacional, cujo objetivo passa por desenvolver modelos capazes de detetar instâncias de objetos, pessoas ou afins, de diferentes classes, em imagens digitais [29].

Esta tarefa serve de base para diversas outras tarefas como segmentação, *tracking* de objetos, entre outros. Nos últimos anos, com os avanços tecnológicos, foi possível aprofundar este tópico, melhorando os algoritmos já existentes, de forma a torná-los mais rápidos, mais precisos e como tal, mais fiáveis. Deste modo, atualmente distingue-se estes algoritmos em duas classes distintas: as técnicas tradicionais e os métodos de *deep learning*.

Nas técnicas tradicionais não são, normalmente, necessários dados para treinar o sistema, sendo por isso métodos não supervisionados. Isto torna estas técnicas mais práticas de serem utilizadas, uma vez que não é preciso o trabalho manual de anotar os dados para posteriormente treinar o sistema. Por sua vez, estas técnicas são limitadas a diversos fatores, como iluminação, complexidade do cenário, sombras, entre outros.

Já nos métodos de *deep learning*, é necessário treinar e supervisionar o sistema, sendo o seu desempenho limitado pela capacidade do mesmo. Estes sistemas tornam-se geralmente mais robustos e precisos quanto maior a quantidade de dados (anotados) disponíveis para o treino do sistema, tornando a detecção de objetos mais precisa mesmo dentro de cenários mais complexos.

Segmentação

Neste tipo de algoritmo, o objetivo consiste em dividir a imagem em regiões ou objetos conforme o critério escolhido, obtendo como resultado, não uma imagem, mas sim um conjunto de regiões. A precisão deste algoritmo determina a qualidade da análise da imagem e posterior falha ou sucesso nas etapas seguintes. Existem duas estratégias genéricas:

- **Descontinuidade** → Consiste na partição da imagem com base em diferenças bruscas na intensidade do pixel;
- **Similaridade** → Consiste na partição da imagem com base na similaridade entre pixels, de acordo com um critério escolhido.

Um dos métodos mais utilizados é o método de *thresholding*. Neste método, após determinar ou simplesmente pré-definir o valor limiar de intensidade (o *threshold*), é analisado cada pixel da imagem em estudo. Se o valor desse mesmo pixel for superior ao valor fixado, então esse pixel passará a ser branco (ou preto); se o valor for menor, então o pixel passará a ser preto (ou branco). Esta comparação é realizada sempre com o mesmo valor de base e para todos os pixels existentes na imagem em estudo. Assim, iremos obter uma imagem binária de contraste máximo [30] (Figura 20). Para além desta abordagem global (com o recurso a um único valor de *threshold*), existem métodos adaptativos cuja ideia consiste em utilizar, para cada pixel, um valor limiar calculado na vizinhança do próprio pixel.



Figura 20 - Exemplo de segmentação por *thresholding* [31].

2.3.3 Principais aplicações dos sistemas de visão

Esta secção apresenta as quatro principais categorias nas quais se dividem as aplicações dos sistemas de visão artificial.

Orientação

Para ser possível analisar e estudar a peça em questão com a maior confiança possível é primeiramente necessário que o sistema se oriente de acordo com a peça em estudo. Este passo pode ser utilizado para situações em que é necessário retirar a posição e a orientação de um determinado objeto no espaço cartesiano ou, por exemplo, para permitir que o robô localize e se oriente de acordo com o alinhamento necessário.

Identificação

Neste tipo de funções, o sistema de visão artificial procura reconhecer e identificar peças a partir da leitura de códigos, matrizes, marcações diretas, etiquetas, conjuntos de caracteres, entre outros métodos. Para além da leitura de códigos ou caracteres, o sistema de visão também permite a identificação de um determinado objeto a partir da sua cor, forma, tamanho, estrutura, ou material, aumentando assim o leque de funcionalidades existentes para a identificação de objetos em estudo.

Aferição

Estes sistemas de visão são utilizados para realizar medições e calcular distâncias entre dois pontos ou zonas geométricas de forma a determinar se a distância corresponde ao valor pretendido. Neste caso, à medida que o objeto passa pelo campo de visão da câmara, esta captura imagens, e utiliza o seu *software* para calcular distâncias entre os pontos em estudo. Normalmente, os sistemas de visão conseguem ter uma precisão elevada em relação à inspeção visual (a qual se fixa na ordem dos 0.02 mm).

Inspeção

Nesta última categoria, o objetivo da utilização dos sistemas de visão artificial passa pela inspeção e procura de defeitos, falhas funcionais, contaminantes, ou outras irregularidades que ponham em questão o correto funcionamento do objeto em estudo.

2.3.4 Componentes do sistema de visão

Independentemente da funcionalidade destes, um sistema de visão é geralmente composto por:

Iluminação

Este é um dos componentes chave num sistema de visão artificial. É através da luz incidente na peça em estudo que o sistema será capaz de realizar a sua tarefa de orientação, deteção, inspeção ou outro. O objetivo passa por iluminar a peça a ser inspecionada da melhor forma, de maneira a realçar as características da mesma para que possam ser capturadas claramente pela câmara. De notar que, de acordo com a característica a avaliar, poderá ser utilizada luz incidente no objeto (isto é, posicionada do mesmo lado da câmara face ao objeto) ou posicionada por detrás deste. Assim, conforme o objetivo, existem diferentes tipos de iluminação disponíveis [21]:

- **Iluminação de fundo;**
- **Iluminação axial difusa;**
- **Luz estruturada;**
- **Iluminação indireta;**
- **Iluminação direta;**
- **Iluminação difusa indireta;**
- **Iluminação estroboscópica.**

Lentes

A lente é um dispositivo de transmissão ótico capaz de focar ou dispersar a luz (através do fenómeno de refração) com o intuito de controlar a forma como esta é capturada pelo sensor. Neste constituinte, a qualidade da ótica (e, geralmente, o preço da lente) irá ditar a qualidade e a resolução da imagem capturada. Este é um fator importante, uma vez que conforme o objetivo em questão, determinada ótica poderá não ser suficiente para detetar características mais pequenas na peça em estudo, podendo ser necessário obter uma lente de qualidade superior. De notar que a qualidade da imagem depende, para além da lente, de outros componentes, como o sensor empregue, a iluminação utilizada e a sua localização. As lentes utilizadas em sistemas de visão podem ser classificadas em dois tipos [21]:

- **Lentes intercambiáveis** → A combinação entre a lente e a extensão irá permitir obter uma melhor qualidade de imagem;
- **Lentes fixas** → São lentes que utilizam autofocus, ajustando-se de forma mecânica ou de forma automática.

Sensor de imagem

Como foi mencionado anteriormente, a lente tem um fator importante no que toca à qualidade de imagem, mas não é o único constituinte que possui esta importância. O sensor, que converte a luz que incide na lente numa imagem digital, também possui um papel importante na capacidade de a câmara captar uma boa imagem. A imagem criada será constituída por pixéis e, quanto maior a resolução do sensor maior será a resolução da matriz dimensional de pixéis na imagem resultante, aumentando deste modo o detalhe da imagem e, por conseguinte, a precisão das medidas realizadas na imagem. Aspetos como a peça em estudo, desde o seu tamanho ou estrutura, ou outros parâmetros, como as tolerâncias de inspeção, ditam a resolução necessária para realizar esta tarefa com maior sucesso.

Processamento

Este componente diz respeito aos componentes de *software* responsáveis pelo estudo e análise da imagem digital proveniente do sensor, aplicando algoritmos que analisam a imagem e extraem a informação necessária (tal como descrito sucintamente na secção 2.3.2), executando a inspeção e, por fim, tomando uma decisão.

Comunicação

Após tomada a decisão na fase de processamento, é necessário enviar esta resposta ao resto do sistema para se poder então prosseguir com a tarefa. Esta troca de informação pode ser realizada a partir de canais de I/O, comunicação série ou, em sistemas mais recentes, a partir de protocolos industriais de nível superior como o TCP/IP.

2.4 Montagem de PCBs

Uma vez que o objetivo principal deste projeto passa pela criação de um processo capaz de inserir componentes PTH de forma automática, com o apoio de um robô colaborativo, foi necessário efetuar uma revisão da literatura existente, dentro de tópicos como a manufatura de PCB, a utilização de robôs neste processo e, ainda, a inspeção automática através de sistemas de visão artificial.

A utilização de robôs no ambiente de montagem é uma das formas de aumentar a produtividade, e um dos setores a beneficiar com este aspeto é o setor da montagem de componentes eletrónicos. Neste âmbito, são normalmente utilizados robôs cartesianos ou SCARA. No entanto, apesar de não se recorrer a robôs de seis eixos, a verdade é que há a necessidade de implementar sistemas de produção mais flexíveis, com um investimento inicial mais baixo e competitivo. Por outro lado, a utilização de robôs

colaborativos permite a criação de um ambiente de cooperação entre o robô e o operador, beneficiando com a flexibilidade, repetibilidade e rapidez do robô e com as capacidades únicas do ser humano.

Ao longo dos anos, tem havido um esforço na melhoria destes processos e na respetiva utilização de robôs, existindo múltiplos estudos na área da manufatura de PCBs com o intuito de otimizar processos, reduzir tempos e maximizar a produtividade na linha de montagem. Em 1998, Crama *et al.* investigaram a possibilidade de com apenas uma linha de máquinas de inserção realizar a inserção de componentes em diferentes tipos de placas de circuito impresso. Os autores concluíram que através de métodos heurísticos eram capazes de resolver este problema e tornar a linha de montagem mais flexível por ser capaz de lidar simultaneamente com diferentes placas [32].

Desde então, outras áreas como a otimização do fabrico de placas de circuito impresso também têm vindo a ser estudadas. Em 1998, Kemal *et al.* investigaram problemas como a alocação de componentes e a sequência da colocação destes, na tentativa de determinar o melhor algoritmo para certas situações, comprovando que através de um algoritmo de aproximação de valor ϵ para o problema de roteamento apresentavam uma solução com um máximo erro de ϵ para o problema da PCB [33]. Já em 2018, também com o objetivo de otimizar a manufatura de PCBs, Karin *et al.* investigaram a possibilidade de colaboração entre humano e robô, na tentativa de aumentar a flexibilidade da célula de trabalho, diminuir o tempo gasto para o preenchimento de uma placa e ainda aumentar a diversidade no tipo de placas a montar. Estes autores propuseram uma abordagem matemática que permitiu obter um bom *makespan* geral, isto é, um bom tempo decorrido desde o início ao fim do trabalho, nomeadamente em comparação com uma abordagem heurística [34]. Também em 2018, Andrzejewski *et al.* investigaram a aplicação de um robô articulado de 6 eixos, mais especificamente o KUKA KR16, na inserção de componentes eletrónicos numa placa de circuito impresso [35]. O objetivo principal consistia na redução de tempo e custos, com vista a otimizar a linha de montagem. Para tal, foi utilizado um algoritmo genético, de forma a calcular qual seria a melhor sequência de inserção, bem como a melhor localização das respetivas peças a montar na PCB. Em 2019, os mesmos autores voltaram a utilizar os mesmos fundamentos, mas desta vez na tentativa de descobrir qual seria o melhor movimento, dentro dos diferentes tipos existentes num robô colaborativo, que provocaria o menor desgaste na estrutura do robô, sem esquecer fatores como a otimização do tempo de ciclo e a maximização do rendimento de produção [36].

Uma outra área de investigação foca na utilização de sistemas de visão artificial para a verificação da qualidade de componentes inseridos, para a inspeção do processo de inserção, entre outros. Em 2019, Fonseca e Jayasinghe realizaram um estudo sobre os diferentes algoritmos existentes e capazes

de serem implementados em sistemas de visão automáticos, para inspecionar a qualidade das juntas de solda de componentes THT nas placas de circuito impresso, propondo uma abordagem capaz de controlar o sistema robótico e ainda realizar o trabalho acrescido nas juntas de solda defeituosas, com uma diminuição acentuada na intervenção do operário [37]. Já em 2021, Celaschi *et al.* desenvolveram um sistema automático e inovador para a inspeção da montagem de componentes do tipo PTH em placas de circuito impresso. Este sistema era capaz de identificar defeitos críticos que podem ocorrer durante a inserção de componentes na PCB, através da aquisição de imagens e posterior interpretação das mesmas com recurso a algoritmos de inspeção básica [38].

3. REQUISITOS E ARQUITETURA DO SISTEMA

Tendo em consideração o objetivo de reduzir, ou até eliminar na totalidade, a necessidade de inserção manual de componentes eletrônicos em PCBs, mantendo a qualidade e a eficiência no processo de inserção, foi necessário definir um conjunto de requisitos, sendo que os mais relevantes são:

- A célula deve ter um sistema de transporte automático para as placas PCB;
- A célula deve ser capaz de identificar as PCBs, através da leitura de um código de barras;
- O sistema deve ser flexível o suficiente para facilmente adaptar o programa e a pinça de acordo com a PCB e os componentes a serem montados;
- A célula deve realizar uma pega e inserção segura e precisa dos diferentes componentes;
- O sistema necessita de validar cada componente antes de realizar a sua inserção;
- O sistema precisa assegurar que a PCB se encontra preenchida corretamente e na totalidade num tempo máximo de 30 segundos;
- É necessário a recolha de informações relevantes para o funcionamento da célula de trabalho, bem como a sua disponibilização num ficheiro para posterior análise e estudo.

De forma a assegurar o cumprimento destes requisitos, e como mencionado anteriormente, será necessário a utilização de três subsistemas: (1) o *cobot*, para permitir o trabalho colaborativo com o operador, bem como assegurar a tarefa de *pick and place*; (2) o sistema de visão, para validação dos componentes; e (3) um dispositivo externo para controlo e monitorização, o PLC. A Figura 21 ilustra a interação existente entre estes subsistemas. De uma forma resumida, o PLC recebe um sinal informando que a PCB chegou à área de montagem, dando início a este processo. De seguida, diferentes sinais são trocados entre o *cobot*, PLC e sistema de visão de forma a garantir a correta leitura do código de barras, da qualidade do componente e da tarefa de *pick and place*, possibilitando o avanço na tarefa principal. Toda a informação relevante será exibida numa interface para o operador, e os dados mais importantes serão guardados numa base de dados.

3.1 Robô TM5-700

Entrando em mais detalhe no subsistema robô, e tendo por base o tipo de componentes em estudo e das placas de assemblagem, os requisitos base passam pela rapidez e flexibilidade, a habilidade de trabalhar em segurança e em cooperação com o operador humano, ter um alcance mínimo de 700 mm de forma a cobrir a área de trabalho, possuir pelo menos 6 graus de liberdade, ter múltiplas interfaces

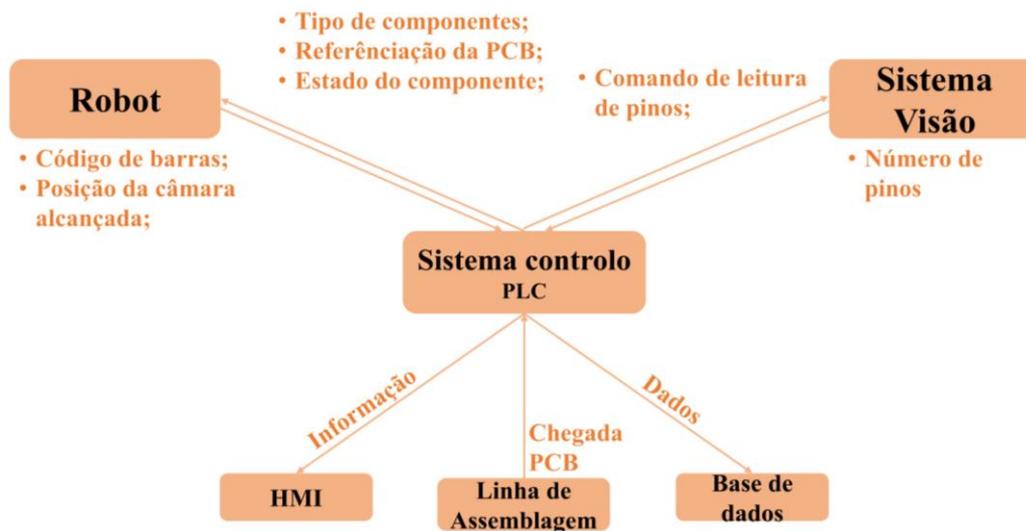


Figura 21 - Partilha de informação entre os três subsistemas da célula de trabalho desenvolvida.

de entrada/saída (como portas COM, HDMI, LAN e USB), e também apresentar vários protocolos de comunicação (como ligações RS232, Ethernet e Ethercat). Estes requisitos passam por assegurar a funcionalidade deste subsistema de uma forma totalmente automatizada, de forma a poupar tempo no processamento, sem abrir mão à qualidade e repetibilidade deste trabalho. Como mencionado anteriormente, o robô escolhido é o robô colaborativo TM5-700, da Techman Robot (Taiwan), sendo um robô desenhado para colaborar com o operador humano, mantendo um ambiente de trabalho totalmente seguro. Possui um sistema intuitivo e de fácil aprendizagem, possui também um vasto leque de informações, manuais, e tutoriais para apoio na implementação dos mais variados tipos de tarefas, desde as mais simples e básicas aos exemplos mais complexos aplicados à indústria. Para além disto, é um robô complementado com diversas ferramentas adicionais, como sensores, diferentes tipos de *grippers*, suportes móveis, entre outras, comprovando a sua enorme aplicabilidade.

Este subsistema possui um *software* próprio, o *TMflow*, o qual é um HMI (do inglês, *human-machine interface*) simples e intuitivo, capaz de oferecer ao operador um ambiente prático, baseado em blocos e ligações lógicas entre os mesmos, o que torna a tradução lógica de fluxogramas para a programação efetiva mais fácil (Figura 22).

Através deste *software*, é possível por parte do utilizador gerir o programa, bem como facilmente usufruir de todas as funcionalidades que este robô apresenta. Como é possível observar na Figura 22, do lado esquerdo apresentam-se os diferentes blocos de funções disponíveis, como a possibilidade de movimentação circular, retilínea, gerir a sua própria trajetória, realizar um movimento de paletização, entre outros, sendo ainda possível através deste *software* gerir as variáveis presentes no projeto, realizar



Figura 22 - Interface de programação do *software* TMflow [39].

o *debugging* do mesmo, e ainda colocar o robô na posição desejada, indicando diretamente o valor angular das suas juntas ou a posição cartesiana. Caso se pretenda deslocar o robô para uma posição desconhecida no seu referencial cartesiano, basta pressionar o botão *Free* enquanto se desloca, e chegando à posição desejada, pressiona-se o botão *Point*, guardando diretamente este ponto no projeto. Os dois botões apresentam-se situados na extremidade do robô.

O TM5-700, para além de possuir o *software* TMflow, possui ainda um *software* de visão, o *TMvision*, associado ao sistema de visão interno existente no robô (Figura 23). É este *software* que permite lidar com duas das tarefas de visão necessárias para este projeto: a leitura de um *part number* existente em cada PCB, para ser possível saber o seu tipo, e ainda a referenciação do robô face à posição e orientação da PCB, de forma a garantir o correto posicionamento dos componentes.



Figura 23 - Sistema de visão interno do robô TM5-700 [20].

Este *software* possui diversas funcionalidades, como o melhoramento da imagem (com a alteração do seu contraste, saturação, brilho, entre outros; Figura 24) ou o ajuste de parâmetros da própria câmara (como *shutter speed*, *focus* e também a abertura do diafragma).

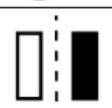
Function module	Function description
Contrast Enhancement 	Adjust image contrast.
Color Plane Extraction 	Obtain specific colors (such as red, blue, or green) or saturation.
Smoothing 	Filter out noise and increase the image's smoothness.
Thresholding 	Transform a raw image into a black and white one.
Morphology 	Erode, dilate, patch, or open the image.
Flip 	Flip the image.

Figura 24 - Funcionalidades de melhoramento de imagem existentes no *TMvision* [39].

Para além destas características mais direcionadas para a qualidade de imagem, o sistema oferece diferentes funcionalidades agrupadas em duas categorias diferentes:

Procura

Esta categoria contempla as funcionalidades existentes no *TMvision* para a deteção de objetos ou marcas dentro da imagem adquirida em tempo real, existindo num total cinco funções diferentes:

- ***Pattern Matching (shape)*** → Utiliza a forma geométrica do modelo para detetar objetos semelhantes, suportando alterações como a rotação ou a dimensão do objeto;
- ***Pattern Matching (Image)*** → Utiliza a distribuição dos valores dos pixels do modelo para detetar objetos semelhantes, sendo uma funcionalidade mais lenta que a anterior e que não suporta variações de tamanho do objeto;
- ***Blob Finder*** → Esta funcionalidade é mais adequada para objetos que não possuam uma forma geométrica fixa, utilizando a variação de cor existente entre o objeto e o seu fundo;
- ***Anchor*** → Esta funcionalidade define a posição e orientação inicial do objeto, permitindo depois manualmente ajustar o seu ponto âncora;

- **Fiducial Mark Matching** → Esta funcionalidade está desenhada para detetar e posicionar dois pontos nas PCBs, sendo rápida e precisa, mas apresenta algumas limitações no caso do objeto se encontrar rodado ou apresentar uma magnificação diferente na imagem adquirida.

Identificação

As funcionalidades desta categoria visam identificar aspetos específicos dentro da imagem adquirida ou do objeto pretendido, existindo um total de três funções:

- **Código de barras/QR code** → Esta funcionalidade permite ler códigos de barras e QRcodes, possibilitando por exemplo a identificação do tipo de placas ou componentes, caso estes possuam um deste tipo de códigos;
- **Color Classifier** → Esta funcionalidade, através de um processo de treino, é capaz de detetar as diferentes cores, sendo possível posteriormente tomar decisões consoante a cor detetada, para a separação por exemplo de objetos através de um código de cores;
- **String Match** → Com esta funcionalidade é possível por parte do utilizador comparar conjuntos de caracteres.

Através deste leque de funções, é possível detetar objetos, identificá-los, diferenciá-los, tomar decisões conforme as diferenças detetadas, realizar o *tracking* de objetos, entre outras funcionalidades.

Como é possível perceber, este robô possui as qualidades de *software* necessárias para este projeto, possuindo ainda as qualidades de *hardware* requisitadas, como entradas e saídas digitais e outras interfaces (contemplando portas COM, HDMI e LAN), e possui ainda o protocolo de comunicação Modbus e interfaces de comunicação, como *Socket* ou *Serial*. O protocolo Modbus especificamente será essencial no desenvolvimento da referida célula de trabalho, possibilitando a comunicação entre o robô e o PLC.

3.2 Sistema de visão

Quanto ao subsistema de visão externo, este contemplará uma das tarefas mais complicadas e críticas deste projeto: a inspeção de furos e a verificação da integridade de todos os pinos em cada componente, validando o seu estado para posterior inserção na PCB. Para tal ser possível, foi necessário seleccionar um sensor com resolução mínima de 2M pixéis que cobrisse a totalidade da área de inspeção (500x500 mm²), e que apresentasse ainda a capacidade de comunicação com dispositivos externos, para permitir a partilha de informação após realizar o estudo do componente.

Com base no anteriormente referido, o sistema de visão selecionado foi o controlador da OMRON, o FH1050, apoiado com a câmara FH-SC05R (Figura 25).



Figura 25 - Sistema de visão com controlador e sensor [40].

Este sistema implementa diversos protocolos/interfaces de comunicação, desde Ethernet, Serial, *Parallel Channels*, entre outros, possibilitando assim a troca de informação com os restantes subsistemas. É um *software* intuitivo, de fácil perceção, orientado a programação por blocos, possuindo um ambiente acessível a programadores menos experientes, e com a capacidade de execução necessária para este projeto. Para além disto, este sistema oferece um vasto leque de funcionalidades, desde funções de recolha de imagem, funções de inspeção e medida, funcionalidades de aprimoramento de imagem, funcionalidades de organização de fluxograma e funcionalidades de *output*. Seguidamente, é apresentada uma sucinta explicação de algumas destas funcionalidades, de forma a demonstrar as diferentes tarefas possíveis de se realizar com este sistema:

Inspeção e medida:

- **Search** → Após um modelo dado, o sistema procura por seções do objeto lido que diferenciem ao modelo em questão, dando um grau de similaridade entre ambas as imagens (Figura 26). Para questões mais críticas, existem funcionalidades que efetuam uma pesquisa com maior detalhe, sendo capazes de se abstrair do ruído, da qualidade ou do mau posicionamento do modelo perante a imagem.
- **Shape Search III** → Esta função regista um modelo padrão de imagem baseado no contorno do objeto, procurando posteriormente por objetos com elevado grau de similaridade (*threshold*). Esta funcionalidade pode detetar as semelhanças mesmo com sombras, reflexos, diferenças na posição e ainda ruído, sendo a mais precisa dentro destas funções (Figura 27).

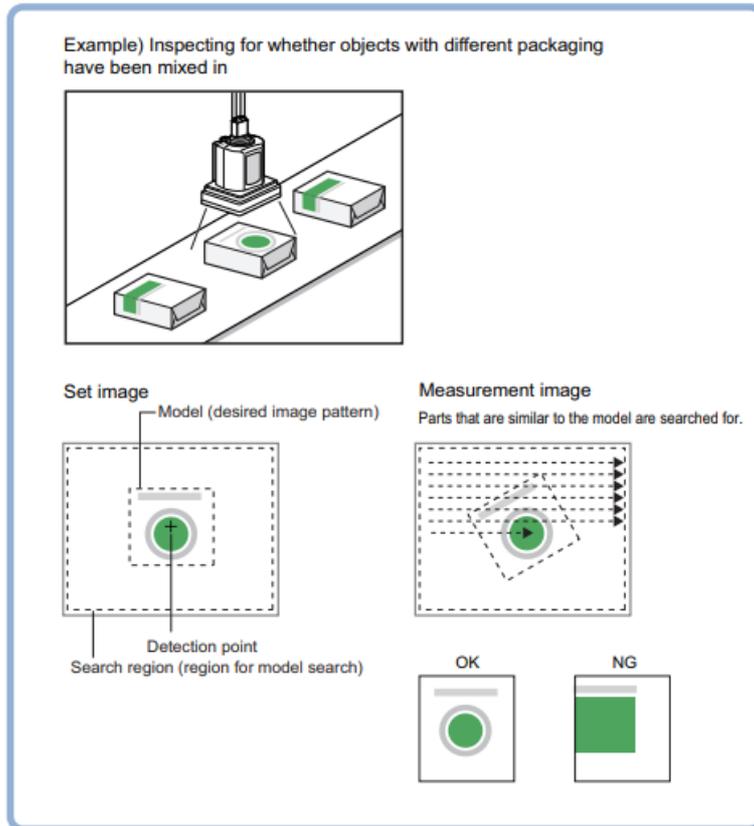


Figura 26 - Search function [41].

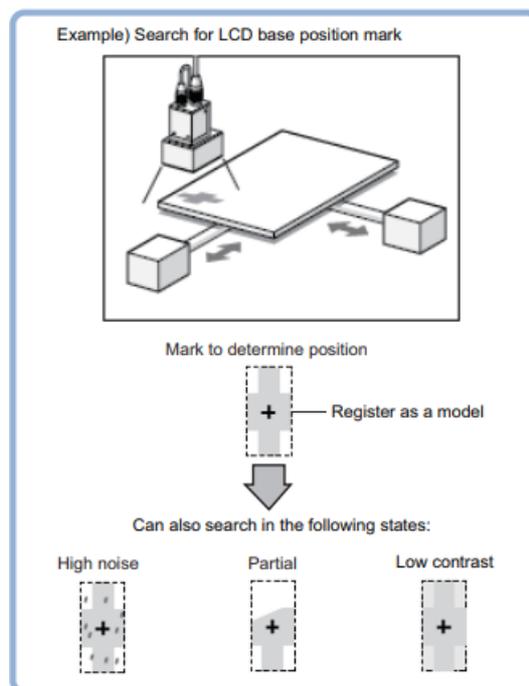


Figura 27 - Shape Search III [41].

- **Edge Pitch** → É capaz de encontrar e contar as arestas através da mudança de cor existente na região de medida. Esta função pode servir por exemplo para contar o número de pinos num circuito integrado ou então para calcular a distância entre dois pinos (Figura 28).

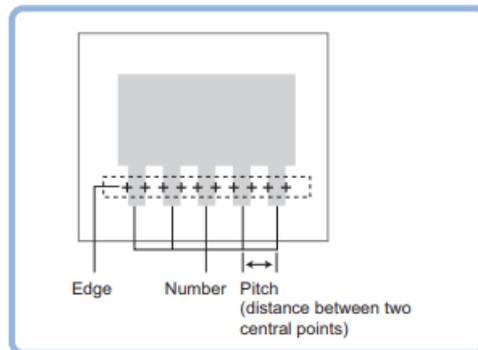


Figura 28 - Edge Pitch [41].

- **Defect** → Deteta defeitos, contaminações ou manchas no modelo, através da variação de cor dentro da região de medida (Figura 29).

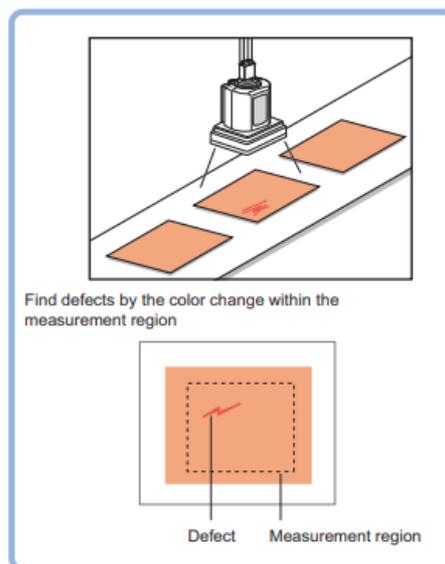


Figura 29 - Defect [41].

- **Fine Matching** → Procura semelhanças através da imagem modelo e da sobreposição de imagens registadas de entrada, garantindo uma procura rápida, mas altamente precisa e confiável, de diferenças (Figura 30).

Compensação de Imagem

- **Position Compensation** → Esta funcionalidade é capaz de reposicionar o objeto na posição desejada, movendo a distância calculada pela diferença entre o modelo dado e a leitura efetuada. Mesmo com posições diferentes do objeto, é possível reposicioná-lo sem haver a necessidade de mover o objeto fisicamente.

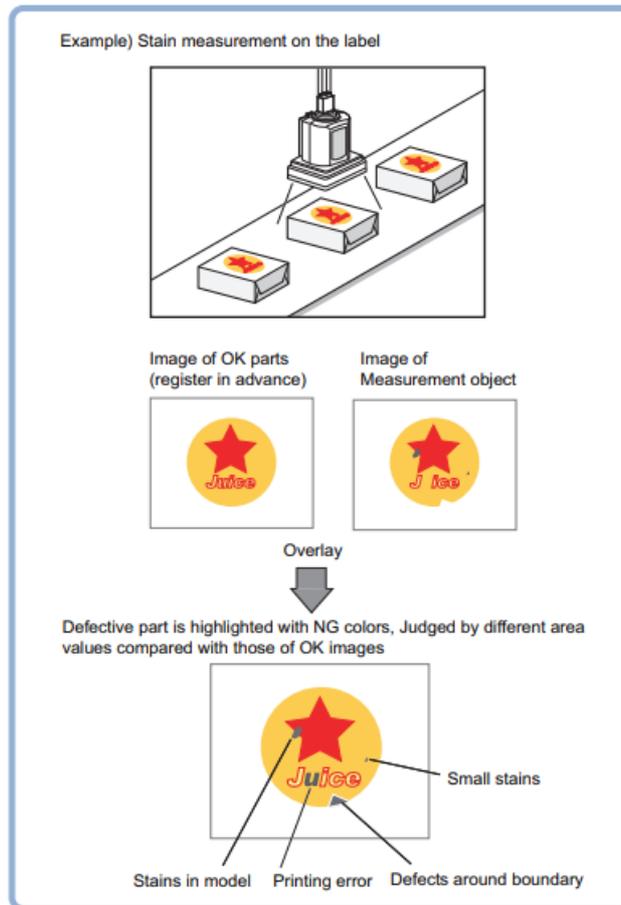


Figura 30 - Fine Matching [41].

- **Filtering** → Processa a imagem adquirida aplicando filtros capazes de tornar as leituras e a execução de medidas bastante mais precisas e fáceis (Figura 31).

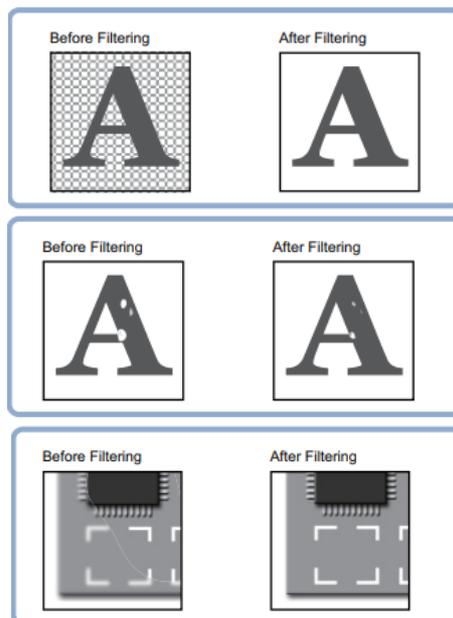


Figura 31 - Filtering [41].

Suporte à Inspeção e Medida

- **Calculation** → Esta funcionalidade permite aplicar funções aritméticas a dados obtidos em diferentes blocos de funções ao longo do projeto (Figura 32).

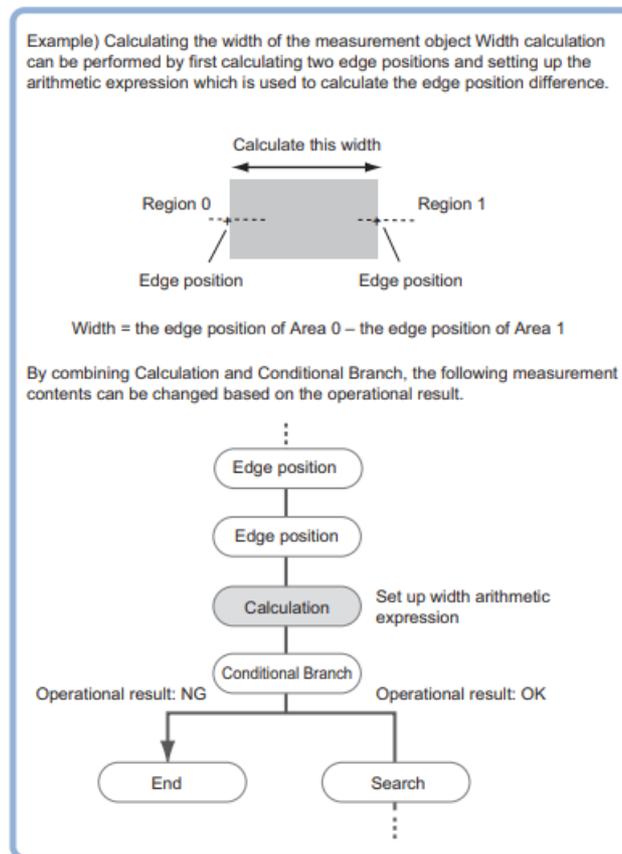


Figura 32 - Calculation [41].

- **Get Unit Data** → Esta função permite ir buscar dados aos blocos utilizados.

Output

- **Parallel Judgement Output** → Permite enviar dados, através da interface paralela, após realizar uma análise escolhida pelo utilizador com base nos dados recolhidos (Figura 33).

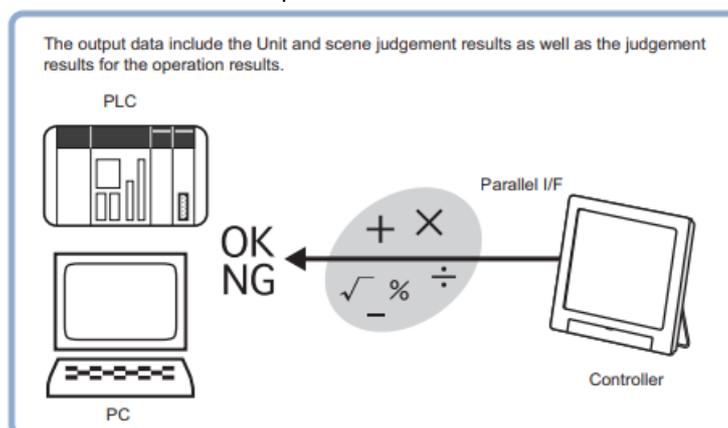


Figura 33 - Parallel Judgement Output [41].

Para além disto, oferece uma vasta gama de funcionalidades de análise e aquisição de imagem, desde: funcionalidades de melhoria de imagem como contraste, brilho, *shutter speed*, *white balance*, aplicação de filtros de imagem, entre outros; e ainda funcionalidades de inspeção e medida, como procura de figuras, marcas, objetos, medidas entre extremos, procura de defeitos, entre muitas outras. Este *software* oferece ainda, um leque de funções de tratamento de dados e de aplicação de funções matemáticas, possibilitando a construção de fluxogramas complexos, desde a aquisição de imagem, até ao tratamento e partilha de informação dos dados obtidos e analisados.

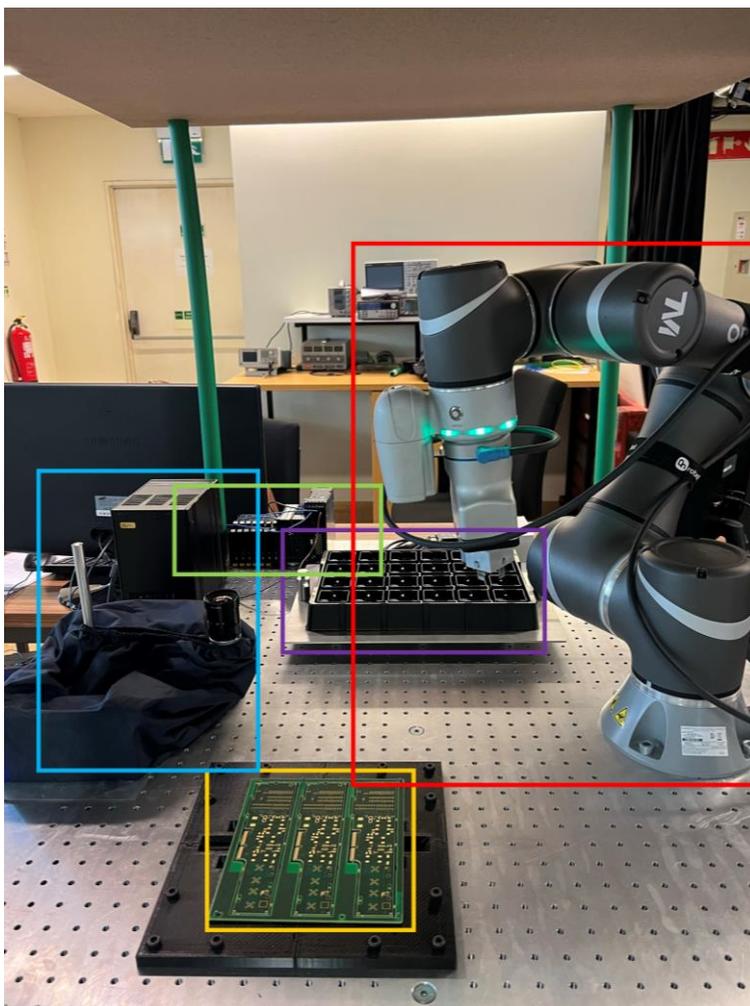
3.3 Controlador Lógico Programável, PLC

Sendo que um dos aspetos mais importantes de qualquer processo automatizado é a sua monitorização e registo de atividade, um subsistema capaz de realizar tais tarefas, como um PCL, tem de ser aplicado. Este subsistema tem como função a gestão de toda a célula de trabalho, desde o comando, o controlo e ainda a monitorização. Para ser possível desempenhar tais funcionalidades, o PLC tem de possuir os seguintes requisitos: diferentes protocolos de comunicação como Ethernet, RS232 e EtherCat; entradas e saídas digitais; facilidades de interação com bases de dados e, por fim, ser modular. Entre estas características, uma das mais essenciais passa pela possibilidade de trocas de informação com bases de dados com disponibilização de variáveis segundo o protocolo OPC-UA.

4. MÉTODOS

De modo a ser possível implementar a célula de trabalho e cumprir com os requisitos listados previamente, foi essencial desdobrar o problema em múltiplas tarefas individuais, implementando cada subsistema de forma independente. Após as tarefas individuais de cada subsistema estarem completas e a cumprirem os seus objetivos, estes são então integrados como um só, focando-se na precisão e velocidade, e respeitando o objetivo chave: a substituição do operador na montagem de PCBs com este tipo de componentes (isto é, PTH), tentando acelerar o processo e reduzir os custos sem nunca abrir mão da qualidade.

Para se compreender o processo e os métodos aplicados na implementação da referida célula de trabalho (Figura 34), será explicado em detalhe os seus três subsistemas principais: o robô colaborativo, o sistema de visão externo e o PLC.



- Vermelho** → Sistema robô
- Verde** → PLC
- Roxo** → Placa de componentes
- Azul** → Sistema de visão
(câmara+controlador)
- Amarelo** → PCB

Figura 34 - Célula de trabalho proposta.

4.1 Implementação do sistema robô

O sistema robô apresenta duas tarefas principais: em primeiro lugar, é o sistema responsável pelo movimento na tarefa principal, isto é, assegurar a correta inserção dos componentes, a validação dos componentes junto ao sistema de visão e, ainda, o preenchimento das placas de assemblagem nas posições corretas; em segundo lugar, relativamente ao sistema de visão interno que possui, este vai ser essencial para a execução de duas tarefas, uma passa pela leitura do código de barras das PCBs de modo a identificar o seu tipo, e a outra passa pela referência da posição da PCB antes de realizar a inserção dos componentes (Figura 35).

Estas tarefas são desenvolvidas nas duas aplicações oferecidas pelo próprio *cobot*, sendo o *TMflow* usado para a movimentação e o desenvolvimento da tarefa, e o *TMvision* para as duas tarefas de visão.



Figura 35 - Fluxograma de operação do subsistema robô.

Resumidamente, o *cobot* irá aguardar pelo sinal fornecido por um sensor de que a placa de assemblagem se encontra na área de montagem. Posteriormente, realiza então a leitura do código de barras da PCB e efetua a respetiva referência da mesma, concluindo aqui as suas tarefas de visão. Por fim, prossegue com o preenchimento da PCB. Aqui, o robô desloca-se até à caixa de componentes, realiza a pega do mesmo e leva-o até à zona de inspeção na área do sistema de visão externo. Caso o

componente se encontre em boas condições, o *cobot* realiza então a inserção. Caso contrário, irá descartar este componente. Este processo com os componentes é repetido até a PCB se encontrar totalmente preenchida.

4.1.1 Movimentação

Relativamente a esta parte da tarefa, o robô tem de se deslocar em torno da célula de trabalho para alcançar as diferentes posições necessárias de ordem a cumprir o seu objetivo, garantindo sempre a maior precisão possível e velocidade, mas sem comprometer a segurança da própria célula, dos diferentes componentes, e ainda do operador.

De forma a garantir a segurança dos componentes e das placas de assemblagem, os movimentos de aproximação a estas peças mais sensíveis são realizados em dois passos, respeitando uma movimentação *waypoint* (Figura 36).

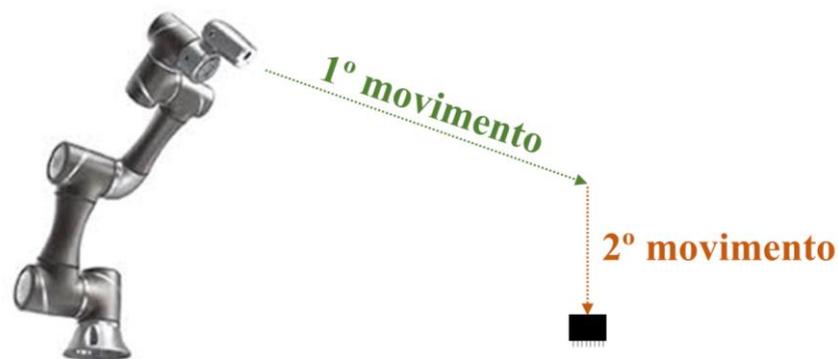


Figura 36 - Movimentação em dois passos, *waypoint motion*.

Ao longo da execução da sua tarefa, existem dois momentos críticos que poderão causar danos e posteriores gastos, nomeadamente a realização da pega do componente a inserir e a inserção do mesmo na PCB. Através do tipo de movimentação demonstrado anteriormente é possível ter um maior controlo sobre a posição do robô e ainda diminuir a velocidade de deslocação do mesmo, garantindo uma aproximação mais cautelosa e precisa, assegurando que a integridade dos pinos do componente não é alterada aquando da inserção na PCB.

De forma a garantir a segurança do operador durante a movimentação do *cobot* e a colaboração entre ambos, foi necessário definir uns valores de velocidade e aceleração capazes de garantir o funcionamento da célula de trabalho, sem negligenciar a saúde do operador. Para tal, foram definidas as zonas que poderiam entrar em contacto com o robô colaborativo, sendo estas as mãos, dedos, pulsos e antebraços, sendo então ajustados os valores de velocidade máxima, bem como do tempo que o robô demora atingir esses mesmos valores (Figura 37). De forma a aumentar ainda mais a segurança do

operador, foi ativado o G-Sensor, sendo este um sensor que aumenta a sensibilidade do *cobot* em colisões, diminuindo a força de impacto e reduzindo o tempo que o mesmo demora a parar.

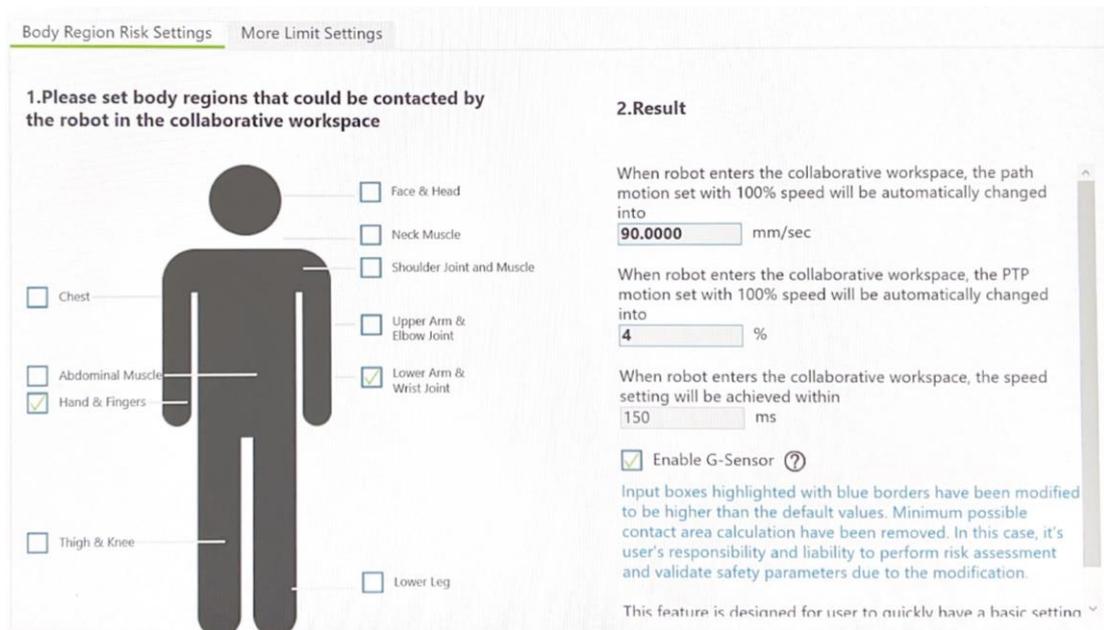


Figura 37 - Definições de segurança homem-máquina.

4.1.2 Visão interna do sistema robô

Nesta tarefa, tal como mencionado anteriormente, o robô recorre ao *software TMvision*, a partir do qual é capaz de identificar o tipo de placa que acabou de chegar à área de montagem (através da leitura do seu código de barras) e ainda realizar a referênciação da mesma, garantindo sempre a correta inserção dos componentes independentemente das alterações de posição das PCBs.

Este *software* oferece dois tipos de funcionalidades principais: funcionalidades de inspeção e procura, e funcionalidades de identificação (Figura 38) [39].

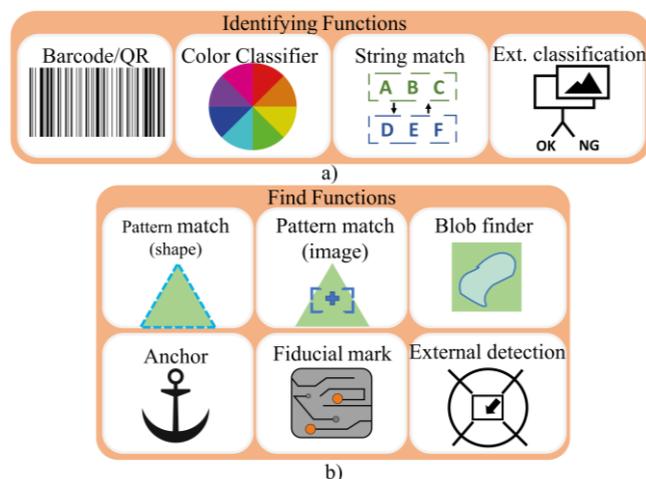


Figura 38 - Funcionalidades disponíveis no sistema de visão interno: a) funcionalidades de identificação; b) funcionalidades de procura.

Tal como mencionado anteriormente, todas as PCBs possuem um código de barras único, de forma a ser possível distingui-las entre os diferentes grupos existentes. Este processo permite ao robô identificar o tipo e a quantidade de componentes que é necessário inserir na respetiva PCB (Figura 39). Neste caso, e como é possível observar na Figura 38, o *software TMvision* já possui uma funcionalidade específica para esta questão, pelo que esta foi utilizada. Resumidamente, foi necessário identificar a área onde o código de barras pode aparecer, tendo em conta dois fatores: primeiro, a placa de montagem pode, por vezes, vir desviada da posição base, sendo que o código de barras encontrar-se-ia então deslocado da sua posição normal; segundo, quanto maior a área de inspeção e procura pelo código de barras, mais tempo demorará o sistema a percorrer esta etapa, podendo provocar atrasos no preenchimento da placa.



Figura 39 - Exemplo de um código de barras na PCB.

Na Figura 40, é possível observar o aspeto da interface de configuração e implementação desta funcionalidade no software *TMvision*.

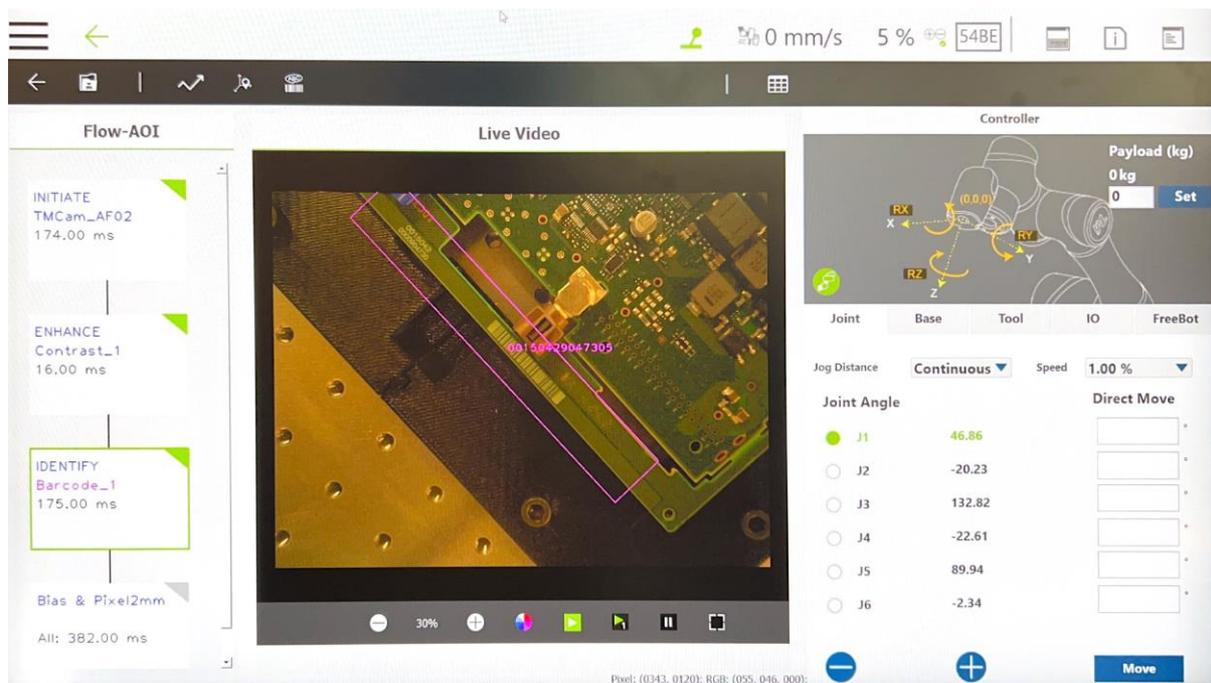


Figura 40 - Funcionalidade código de barras implementada no *TMvision*.

À esquerda na Figura 40, é possível observar três blocos principais: INITIATE_TMCam_AF02, ENHANCE Contrast_1 e, ainda, IDENTIFY Barcode_1. Os primeiros dois blocos servem para iniciar a aquisição de imagem e alterar a qualidade desta, nomeadamente alterando o ganho, o nível de focagem da imagem, e ainda utilizar filtros para melhoria do contraste. Este conjunto de operações tem como objetivo tornar a imagem mais clara e nítida para facilitar a procura e identificação do código de barras. Por fim, o terceiro e último bloco define a área onde o robô deve procurar pelo código de barras e efetua a respetiva pesquisa. Através da imagem adquirida em tempo real (zona central da Figura 40; em que a área selecionada está identificada a rosa), é então possível visualizar onde se encontra a placa e o respetivo código de barras.

Para a segunda tarefa deste sistema de visão interno, a referência à placa, existe também uma funcionalidade própria oferecida pelo software, a funcionalidade de procura através de *Fiducial Marks*. Estas marcas fiduciais, são pequenas marcas existentes nas placas de assemblagem especificamente para os sistemas de visão serem capazes de reconhecer onde se encontra a PCB (Figura 41).

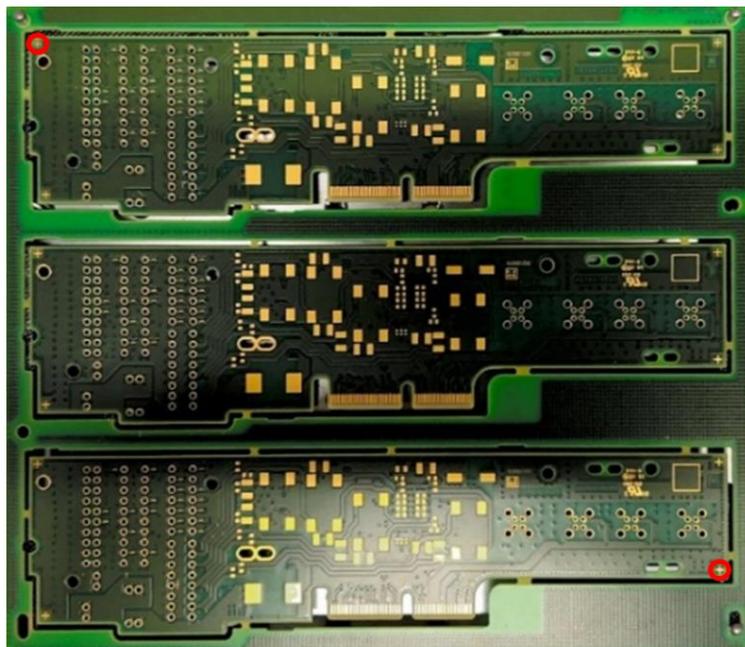


Figura 41 - Exemplo das *marcas fiduciais* numa PCB.

Na Figura 42 é possível observar o aspeto da implementação desta funcionalidade através do *software TMvision*.

Mais uma vez, é possível observar três blocos: os dois primeiros já foram explicados anteriormente e servem para aquisição da imagem obtida pela câmara do próprio robô e melhoria na qualidade e nitidez desta; e o último, FIND FiducialMarks, é onde decorre o processo principal desta funcionalidade. Aqui, são definidos aspetos como as marcas fiduciais a identificar, a área de estudo e o limiar de similaridade,

e é, de seguida, efetuada a procura por estas mesmas marcas. O processo de definição da marca fiducial e da área de procura pela mesma passa pela seleção desses mesmos aspetos através de uma imagem adquirida em tempo real (Figura 43).

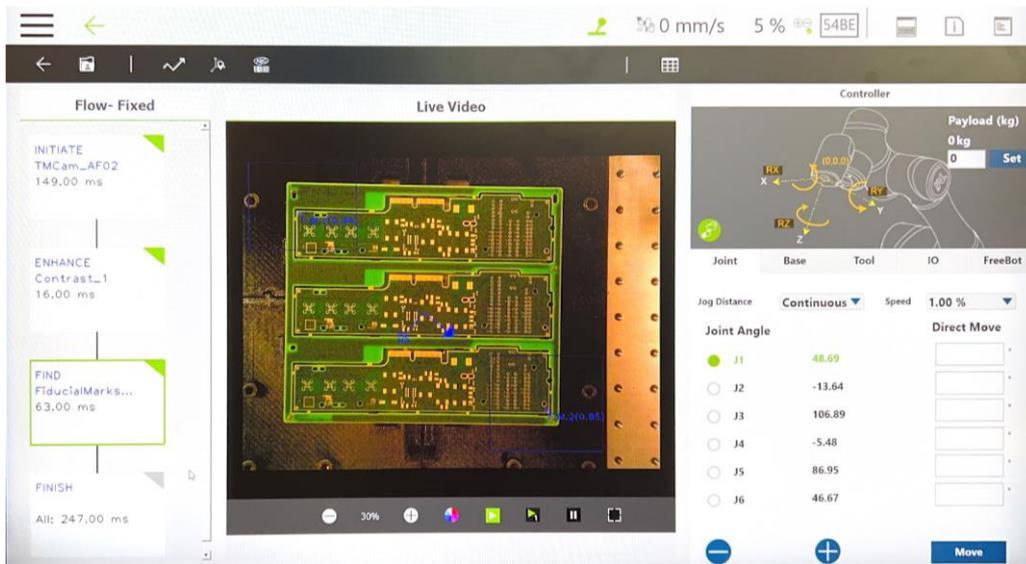


Figura 42 - Funcionalidade *Fiducial Marks* implementada no *TMvision*.

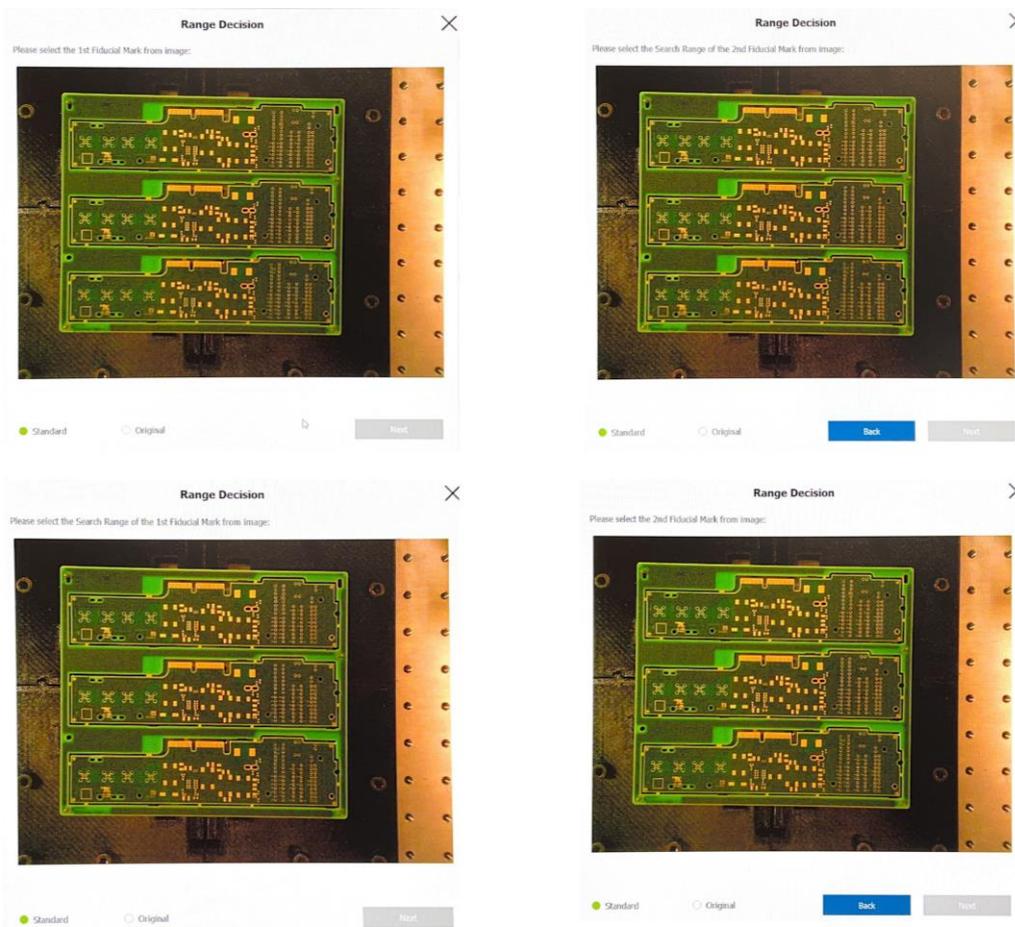


Figura 43 - Identificação das marcas fiduciais e da área de procura.

Após esta seleção é então necessário definir o limiar de similaridade mínimo entre o modelo e a nova marca fiducial obtida, para o sistema ser então capaz de encontrar e selecionar as marcas de acordo com a semelhança entre cor, tamanho e posição. O resultado é ilustrado na Figura 44, onde é possível observar que o software foi capaz de identificar as marcas fiduciais (identificadas a azul).

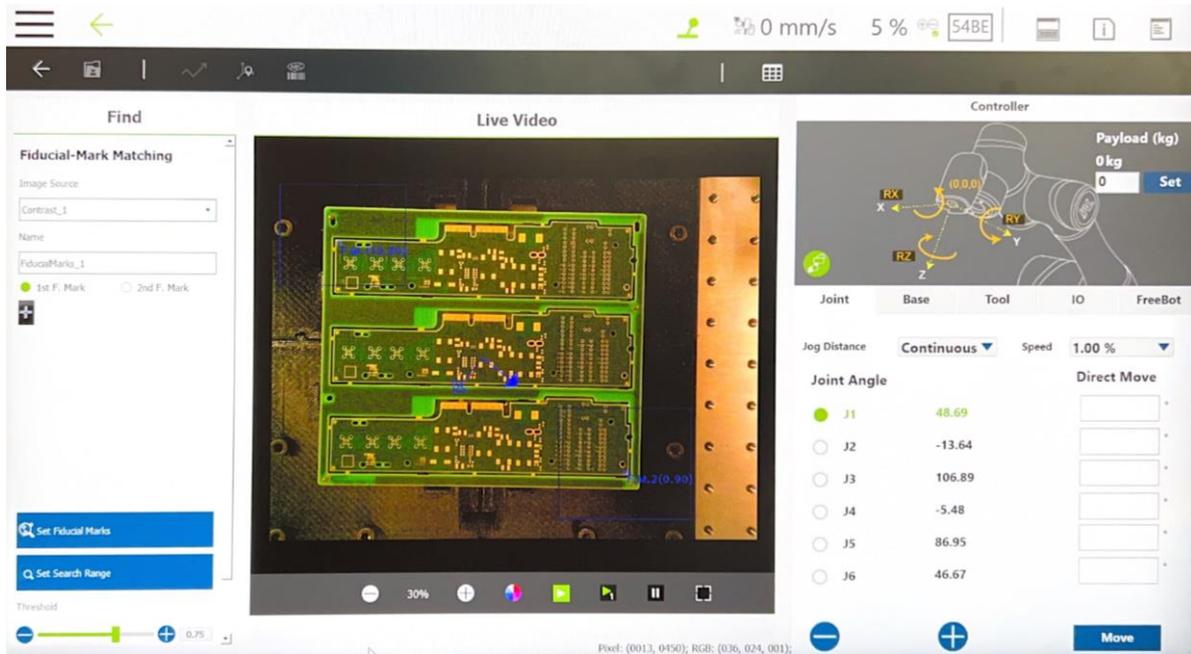


Figura 44 - Detecção das marcas fiduciais na placa de montagem.

Por fim, estando a funcionalidade especificada para a tarefa, após o robô atingir a posição para realizar a referenciação, e após encontrar as marcas fiduciais, este irá criar um novo referencial, sendo o ponto de origem (0,0,0) o ponto intermédio entre as duas marcas fiduciais. A partir deste momento, independente da posição da PCB, a deslocação que o *cobot* tem que fazer para inserir os componentes será sempre a mesma, assegurando desta forma o correto e seguro preenchimento da PCB.

4.2 Implementação do sistema de visão

Quanto ao sistema de visão externo, o seu objetivo passa por assegurar que o componente que vai ser inserido na PCB se encontra em perfeitas condições de ser utilizado. Normalmente, esta tarefa era resolvida por inspeção humana, mas o objetivo é substituir o operador por um sistema de visão de forma a tornar esta célula de trabalho mais autónoma, fiável, precisa e rápida. Para isso, e tal como mencionado anteriormente, o sistema de visão da Omron, composto pelo controlador FH-1050 e a câmara FH-SC05R, foi utilizado.

A Figura 45 demonstra sucintamente o fluxograma da tarefa implementada por este subsistema.

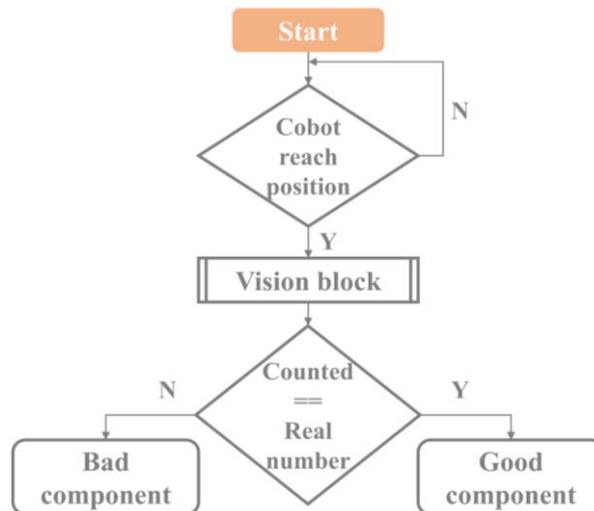


Figura 45 - Fluxograma de operação do subsistema visão.

Sucintamente, o subsistema de visão aguarda pela chegada do *cobot* à posição de inspeção, onde irá então realizar o seu código de leitura e verificação dos pinos do componente (bloco *Vision block* ilustrado na Figura 45). Posteriormente, a informação obtida deste bloco é guardada e enviada para ser possível prosseguir com a tarefa principal da célula de trabalho.

Entrando em detalhe no bloco *Vision block*, após a chegada do robô à posição de inspeção, o sistema de visão faz uma única aquisição de imagem para contar o número de pinos numa determinada área e então concluir se estes se encontram danificados ou não. A ideia passou por dividir o componente em estudo em pequenas secções, de forma a facilitar o trabalho de inspeção, e cada secção com um número ou grupo/linha de pinos pré-definidos (Figura 46).



Figura 46 - Componente dividido em secções e número de pinos por secção.

O procedimento realizado dentro do *Vision block* é demonstrado na figura seguinte (Figura 47).

Nesta figura observa-se três tipos de blocos diferentes: Camera Image Input FH, Shape Search III, e ainda o Calculation. O bloco Shape Search III é o bloco responsável pela procura e análise da



Figura 47 - Vision block.

integridade dos pinos. Aqui é necessário definir a região de interesse (ROI; retângulo grande a vermelho na Figura 46, por exemplo), e o modelo que se está à procura nessa mesma área (retângulo pequeno vermelho na Figura 46, por exemplo). Através de um algoritmo de deteção de objetos, este bloco regista um modelo padrão de imagem baseado na informação de contorno, detetando partes da imagem de entrada que mais se aproximam ao modelo em estudo, acrescentando cada objeto semelhante à contabilização total. Um aspeto importante a definir é o limiar de similaridade entre o modelo padrão e os modelos detetados pelo sistema, tendo sido realizados testes para identificar qual o valor mais adequado de modo a garantir a correta identificação da integridade de cada pino.

Posteriormente, o bloco Calculation é utilizado para calcular o número total de pinos identificados em cada bloco Shape Search III. Se o número for igual ao número teórico (44 neste caso), o componente encontra-se em bom estado, pelo que o cobot prossegue com a sua inserção. Caso contrário, o cobot descarta este componente, colocando-o numa caixa específica para o efeito.

De notar que os outros dois blocos presentes na Figura 47, Camera Image Input FH, são utilizados para calibrar a câmara, e melhorar a imagem entre as análises, facilitando o processo de identificação dos pinos nos restantes blocos.

4.3 Implementação do PLC

Este subsistema visa estabelecer uma ligação entre os outros dois subsistemas, sendo responsável pelo controlo e pela monitorização da célula de trabalho, bem como pelo registo dos dados relevantes durante a execução da tarefa principal para serem posteriormente armazenados numa base de dados. Os diferentes dados a registar incluem, por exemplo, o número de componentes danificados, o número

de PCBs preenchidos num determinado período de tempo, o tempo necessário para preencher cada placa, entre outros. A Figura 48 resume a informação trocada entre os subsistemas, sendo este processo dividido em três partes principais.

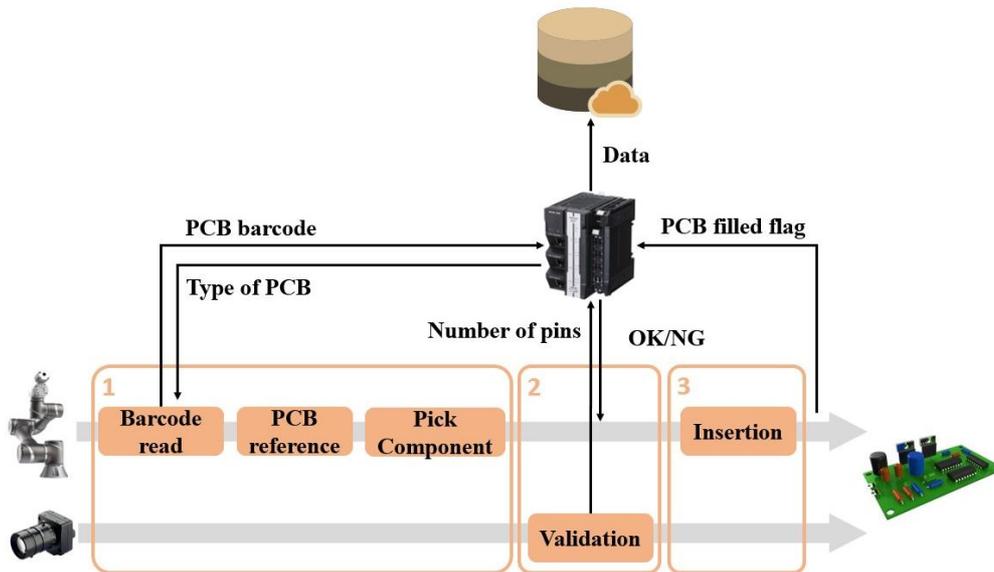


Figura 48 - Fluxograma de operação da célula de trabalho, e informação partilhada entre subsistemas.

Em primeiro lugar, o robô espera que o PLC indique a chegada de uma nova PCB, informação essa obtida através de um sensor infravermelho. Depois, como mencionado anteriormente, o robô utiliza o seu sistema de visão interno para realizar duas tarefas: a leitura do código de barras e a referenciação da placa, enviando o sinal de leitura para o sistema de controlo. Aqui, o PLC decide então que tipo de PCB se trata e informa o robô sobre o tipo e número de componentes a inserir.

Ao fazer a pega de um dos componentes em questão, o *cobot* leva o mesmo para a área do sistema de visão externo. Aqui, o PLC envia dois comandos: (1) o comando *Measure*, onde o sistema executará o código demonstrado na Figura 47, efetuando a contagem do número de pinos e avaliando o estado do componente; e (2) o comando *Get Data*, onde o PLC recebe do sistema de visão externo o número de pinos que o software detetou. Posteriormente, o PLC compara este valor com o valor esperado, e se o resultado for positivo, informa o *cobot* que pode prosseguir com a inserção. Se não for, o PLC informa o *cobot* para descartar este componente e escolher um novo.

Finalmente, o *cobot* procede com a inserção do componente bom, repetindo estes passos até que o PCB esteja preenchido. Entretanto, o PLC irá armazenar dados relevantes durante estas etapas, tais como o número de PCBs preenchidos, o número de componentes danificados, e o tempo para preencher um PCB, entre outros. A Tabela 2 explica como a informação flui entre os três subsistemas.

Tabela 2 - Troca de informação entre o PLC e os restantes subsistemas.

PLC recebe	PLC envia
Sensor da PCB- informa quando a PCB chegou à área de montagem	Sinal/Mensagem para o robô iniciar o processo
Código de barras da PCB através da leitura do robô	Tipo de PCB para robô saber o tipo de componentes a utilizar
Chegada ao sistema de visão externo	Comando de <i>Measure</i> para o sistema de visão
_____	Comando <i>Get Data</i> para o sistema de visão
_____	Componente OK/NG para o <i>cobot</i>
PCB preenchida	_____

Inicialmente foi necessário adicionar todas as cartas existentes no PLC no software, de forma que no *Sysmac Studio*, software utilizado para a implementação do PLC, ficasse implementado o mesmo PLC utilizado na prática. De seguida, foi preciso garantir que o endereço IP era o mesmo para ser possível implementar este código. Posto isto, após assegurar estes dois pontos, é possível passar à implementação do programa. Inicialmente começa-se por um bloco de comunicação direta com o robô, através de Modbus, onde é possível obter dados internos do mesmo, como coordenadas, variáveis, manipulação de zonas de memória, controlo de registos, entre outros (Figura 49), e dois blocos para realizar o set ou o clear a variáveis do robô. Assim que a comunicação estiver estabelecida, é possível iniciar a execução da tarefa principal.

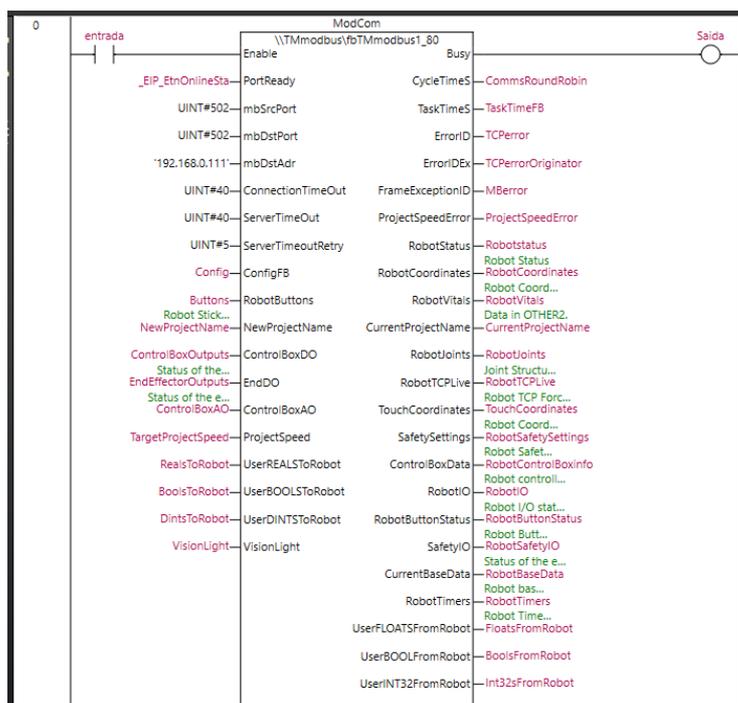


Figura 49 - TModbus function, bloco de comunicação direta com o robo TM5-700.

Primeiramente, o PLC irá aguardar pela chegada de uma nova PCB, informação essa que será registada na variável FlagNBA (*flag new barcode*). Após a leitura do código de barras, o PLC irá retornar ao robô a informação de qual PCB se trata, para que este possa executar o código correto para o preenchimento dessa determinada PCB (Figura 50). Posto isto, após a chegada do robô à área de inspeção por parte do sistema de visão externo, um registo novo irá ser acionado, a FlagCamPos. Assim que este registo seja acionado e a execução do programa do sistema de visão seja realizado, o PLC irá fazer uma série de aquisição de dados, bem como interpretação dos mesmos, em relação ao preenchimento da PCB em estudo (Figura 51). Em primeiro lugar, incrementa o número de PCBs trabalhadas desse tipo, e verifica a quantidade de componentes existentes ainda na caixa para inserção, para que, caso a caixa fique vazia, o sistema aguarde pela reposição da mesma.

```

6  S10S 1 //UPDATE Flags
      2 FlagNBA:=BoolsFromRobot[0];
      3 FlagCamPos:=BoolsFromRobot[1];

7  FlagNBA 1 Barcode:=FloatsFromRobot; //Leitura da memória do robot onde se encontra número de nova PCB
      2
      3 IF Barcode[0] >= 150400000000 AND Barcode[0] <= 150499999999 THEN
      4   PCB1:=TRUE;
      5   PCB2:=FALSE;
      6   PCB3:=FALSE;
      7   PCB4:=FALSE;
      8 END_IF;
      9
     10 IF Barcode[0] >= 432700000000 AND Barcode[0] <= 432799999999 THEN
     11   PCB2:=TRUE;
     12   PCB1:=FALSE;
     13   PCB3:=FALSE;
     14   PCB4:=FALSE;
     15 END_IF;
     16
     17 IF Barcode[0] >= 349500000000 AND Barcode[0] <= 349599999999 THEN
     18   PCB3:=TRUE;
     19   PCB2:=FALSE;
     20   PCB1:=FALSE;
     21   PCB4:=FALSE;
     22 END_IF;
     23
     24 IF Barcode[0] >= 111100000000 AND Barcode[0] <= 111199999999 THEN
     25   PCB4:=TRUE;
     26   PCB2:=FALSE;
     27   PCB3:=FALSE;
     28   PCB1:=FALSE;
     29 END_IF;
     30
     31 ExecuteSQLInsert:=FALSE;
     32 InsertData.DamagedComp:=0;
     33 //Desativar NBA no cobot
  
```

Figura 50 - Interpretação do código de barras, e retorno da informação.

De seguida, um novo registo irá receber a informação da qualidade do componente, OK_NG. Caso o componente esteja em condições de ser inserido na placa, este registo irá então encontrar-se com o valor de *true*, realizando ainda o incremento do contador de componentes já inseridos na placa. Caso contrário, o número de componentes danificados é incrementado. Assim que o número de componentes registados como inseridos por parte do PLC seja igual ao número de componentes total que a PCB deste tipo leva, um registo será colocado a *true*, Flag_FullPCB (linha 21, Figura 51) para a avisar o robô que o preenchimento da placa terminou e que o sistema voltará a aguardar pela chegada de uma nova PCB.

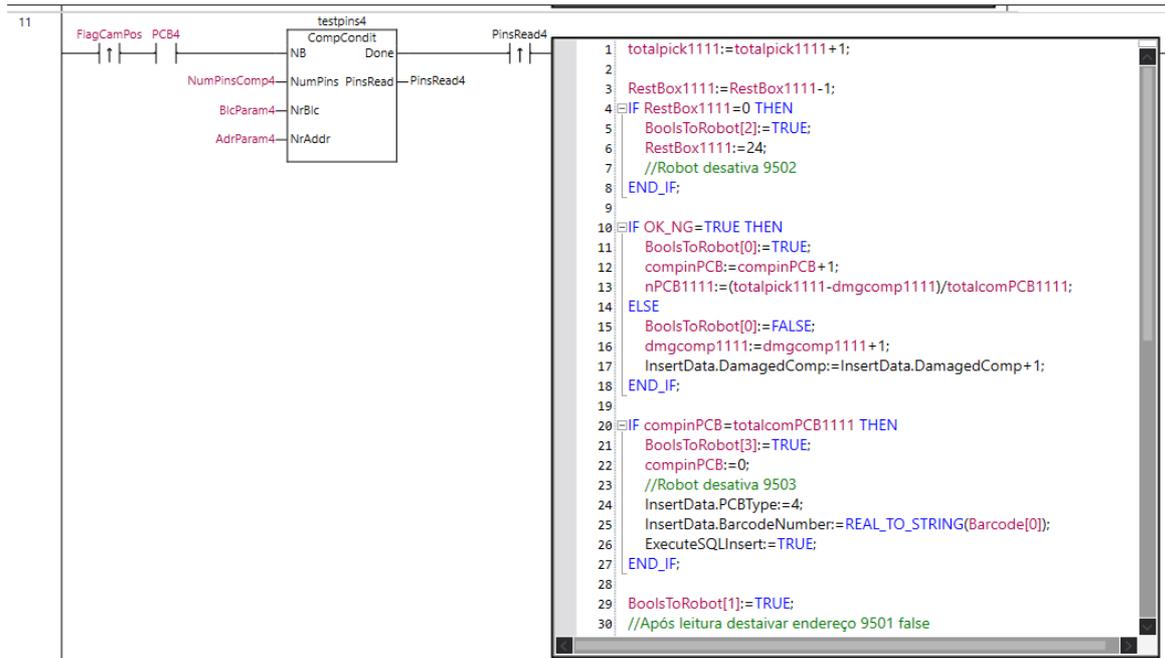


Figura 51 - Preenchimento e interpretação de dados da PCB em estudo.

De notar que os registos utilizados para o controlo da célula de trabalho tiveram de ser criados no *software TMflow*, respeitando a posição de memória designada à *user area* (Figura 52).

Others 4	FC	Address ₁₀	Address ₁₆	Type	R/W	Note
User Define Area	01/03/05/06	9000~9999	2328~270F	User-define	R/W	

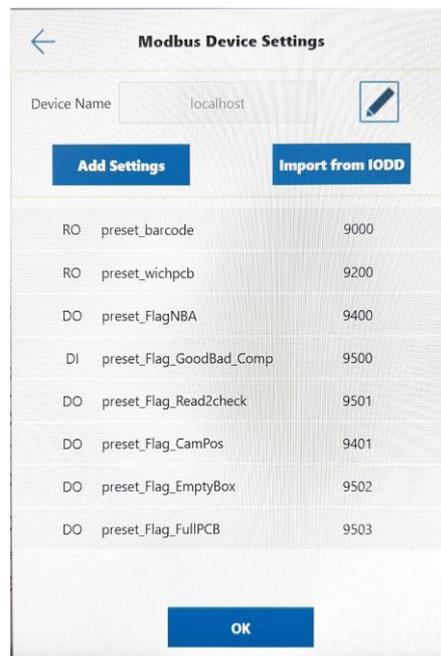


Figura 52 - Variáveis Modbus.

O bloco Modbus apresentado na Figura 49 divide esta *user area* em secções, de forma a facilitar o acesso às respetivas variáveis (Figura 53).

.UpdateUserFLOATS	BOOL	Read 100 float/real – User Defined Area [9000-9199] Write 100 float/real – User Defined Area [9200-9399]
.UpdateUserINT32	BOOL	Read 50 int32/dint – User Defined Area [9600-9699] Write 50 int32/dint – User Defined Area [9700-9799]
.UpdateUserCoils	BOOL	Read 64 Coil/Bool – User Defined Area [9400-9463] Write 64 Coil/Bool – User Defined Area [9500-9563]

a)

Input Name	Data Type	Comment (Read Only)
UserFLOATSFromRobot	REAL[100]	Read from User Defined Area [9000 to 9199] NOTE: Each FLOAT occupies 2 modbus registers.
Input Name	Data Type	Comment (Read Only)
UserINT32FromRobot	DINT[50]	Read from User Defined Area [9600 to 9699] NOTE: Each INT32 occupies 2 modbus registers.
Input Name	Data Type	Comment (Read Only)
UserBOOLFromRobot	BOOL[64]	Read from User Defined Area [9400 to 9463] NOTE: Each BOOL occupies 1 modbus register.

b)

Figura 53 - User area: a) Input; b) Output.

4.3.1 Integração do sistema de visão externo

Para ser possível o controlo do sistema de visão externo por parte do PLC, foi necessário realizar uma comunicação via *tag datalinks*, que é um tipo de comunicação fácil, rápida, que permite a troca de grandes quantidades de informação e funciona a partir do princípio de Produtor/Consumidor (*Publish/ Subscriber*). Inicialmente foi necessário adicionar o sistema externo de visão no *Symac Studio*, indicando o seu endereço IP e o tipo de dispositivo que se tratava (FH Series). Posteriormente foi necessário criar um tipo de dados união para controlo de sinais e os seus estados, e ainda duas estruturas de dados, uma de entrada, para aceder à área de comandos, e outra de saída, para ter acesso à área de resposta (Figura 54).

Structures	Name	Base Type
Union	▼ S_EIOutput	STRUCT
Enumerated	ControlFlag	U_EIPFlag
	CommandCode	DWORD
	CommandParam1	DINT
	CommandParam2	DINT
	CommandParam3	DINT
	▼ S_EIInput	STRUCT
	StatusFlag	U_EIPFlag
	CommandCodeEcho	DWORD
	ResponseCode	DWORD
	ResponseData	DINT
	OuputData	ARRAY[0..7] OF DINT

Figura 54 - Duas estruturas de dados para acesso aos comandos e à área de resposta.

De seguida, foi necessário criar duas variáveis globais (indicando o tipo de dados a que correspondem), sendo que cada um irá corresponder a uma das duas estruturas mencionadas anteriormente. Posteriormente, basta apenas realizar o registo destas *tags*, sendo necessário identificar a respetiva variável global como sendo de entrada ou saída (Figura 55).

Tag Set Name	Bit Selection	Size (Byte)	Size (Bit)	Instance ID	Controller Status
EIP002_IN	<input type="checkbox"/>	48		Auto	Not included
EIPIInput	<input type="checkbox"/>	48	0		

Tag Set Name	Bit Selection	Size (Byte)	Size (Bit)	Instance ID	Controller Status	Output at Fatal Err
EIP002_OUT	<input type="checkbox"/>	20		Auto	Not included	
EIPOutput	<input type="checkbox"/>	20	0			Cleared

Figura 55 - Registo das *tags* de entrada e saída.

Por fim, é necessário estabelecer as conexões, definindo a variável que requisita o pedido e aquela que recebe o pedido (Figura 56).

Target Device	Connection Name	Connection I/O	Input/Out	Target Variable	Size [Byte]	Originator Variable	Size [Byte]	Connection Type	RPI [ms]	Timeout V1
192.168.0.220 FH Series Re	default_001	Consume Data f	Input	101	48	EIP002_IN	48	Point to Point c	50.0	RPI x 4
			Output	100	20	EIP002_OUT	20	Point to Point c		

Figura 56 - Estabelecer as conexões das respetivas *tags*.

Posto isto, e como é possível observar na Figura 51, existe um bloco chamado “testpins4”, onde o PLC tem controlo sob o sistema de visão externo. Aqui, o PLC manda comandos específicos para ser possível primeiramente realizar o código implementado no sistema de visão externo (*MeasureCMD*), e o segundo comando serve apenas para ir buscar a informação sobre o componente estar ou não em boas condições (*UnitDataCMD*) (Figura 57).

MeasureCMD	DWORD	1052688	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Do not publish
UnitDataCMD	DWORD	4198400	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Do not publish
OkayCMD	DWORD	00000000	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Do not publish

Figura 57 - Comandos específicos para o sistema de visão externo.

Para isto, é necessário o PLC enviar um comando e de seguida ativar a *flag ControlFlag*, de forma que este seja enviado para o sistema de visão. Posteriormente, envia um segundo comando que irá ler a um endereço especificado pela *flag CommandParam2* de um bloco especificado pela *flag CommandParam1* do código no sistema de visão, de forma a obter o valor presente nessa memória

(corresponde ao bloco *Calculation* mencionado em 4.2). Se esse valor for igual ao valor teórico de pinos do componente em questão, então o registo OK_NG é colocada a *true* (Figura 58).

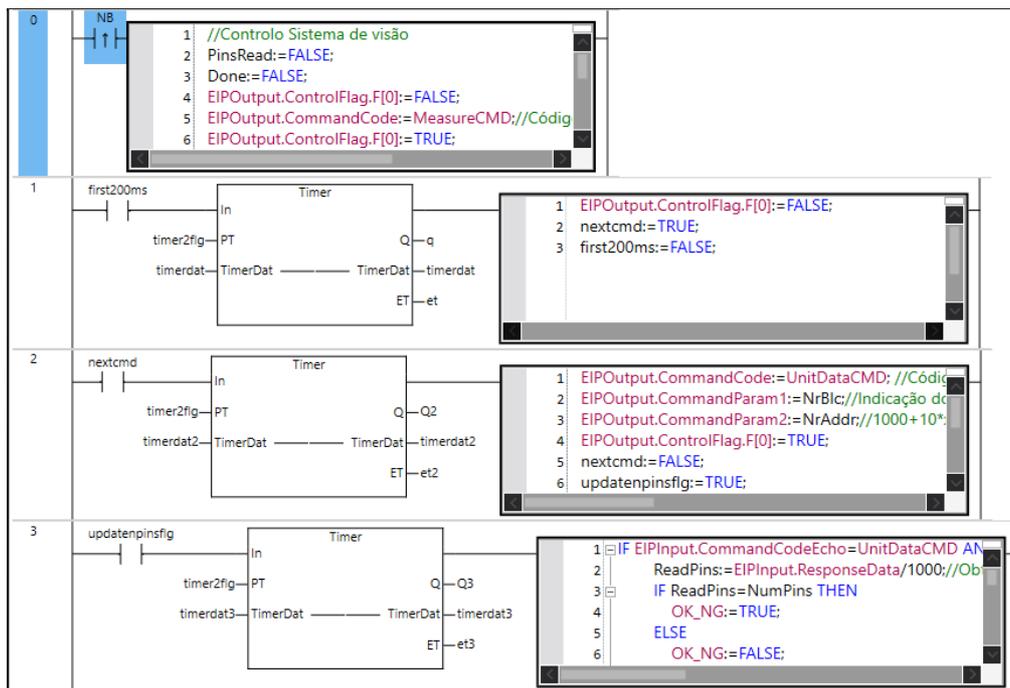


Figura 58 - Bloco *testpins4*.

4.3.2 Integração da base de dados

Para ser possível implementar uma base de dados foi necessário a utilização do software *SQL Server Management Studio* e *SQL Server Configuration Manager*, em conjunto com o software do PLC, *Sysmac Studio*. Inicialmente, foi preciso ativar o TCP/IP nos protocolos para a base de dados SQL, configurando também o endereço IP e a porta em questão no *SQL Server Configuration Manager*. Quanto ao *SQL Server Management Studio*, criou-se uma base de dados, onde dentro desta criou-se uma tabela com os dados que pretendemos registar (Figura 59).

Column Name	Data Type	Allow Nulls
DateTime1	datetime	<input type="checkbox"/>
BarcodeNumber	text	<input type="checkbox"/>
PCBType	int	<input type="checkbox"/>
DamagedComp	int	<input type="checkbox"/>

Figura 59 - Tabela com as variáveis a guardar do sistema.

Por fim, no que toca a configurações, no *Sysmac Studio* é necessário realizar dois passos: criar uma estrutura com as variáveis da tabela criada (Figura 60), e seguidamente criar a conexão à base de dados,

indicando o endereço IP, a porta, o nome, e o login e palavra-chave usados na criação da mesma. Na Figura 61 é possível ver a conexão que foi estabelecida.

▼ SQLData	STRUCT
DateTime	DATE_AND_TIME
BarcodeNumber	STRING[256]
PCBType	INT
DamagedComp	INT



Figura 60 - Estrutura com as variáveis da base de dados e variável do tipo *SQLData*.

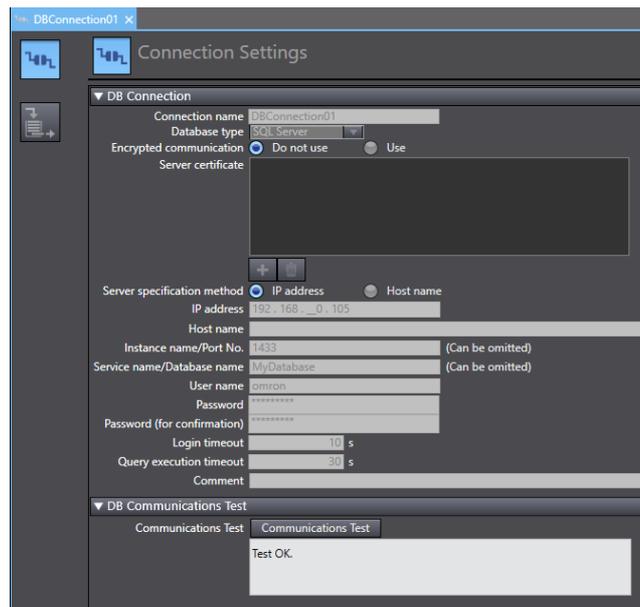


Figura 61 - Conexão à base de dados no software Sysmac Studio.

Finalmente, foi criado os blocos de funções para a conexão e inserção de dados na base de dados criada (Figura 62).

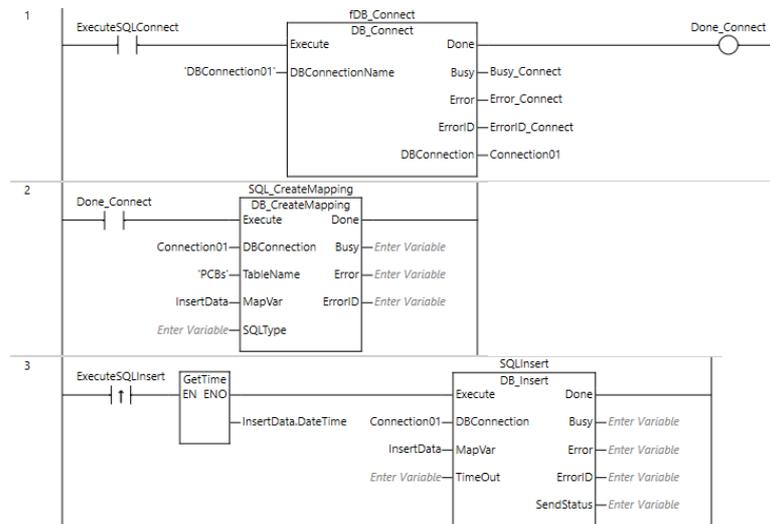


Figura 62 - Blocos de funções para implementação da base de dados.

Em primeiro lugar, estabelece-se a conexão com a base de dados criada, sendo de seguida efetuado o mapeamento da mesma, indicando a tabela e o tipo do conjunto de variáveis que pretendemos inserir. Após estes passos, já é possível estabelecer a comunicação com esta e enviar os dados para a mesma. Como é possível observar na Figura 51, após o preenchimento da placa, na linha 26 é executada a inserção de dados na base de dados, incluindo o tipo de PCB, o seu código de barras e ainda a quantidade de componentes danificados.

5. TESTES E DISCUSSÃO

Neste capítulo são apresentados alguns testes realizados para validação da célula de trabalho, sendo também apresentada uma discussão relativa aos resultados obtidos.

5.1 Testes

Esta secção demonstra alguns exemplos práticos da referida célula de trabalho, tendo sido realizados os seguintes testes:

- Teste A: Quanto tempo leva para o *cobot* preencher uma PCB;
- Teste B: Fiabilidade do sistema de visão externo;
- Teste C: Monitorização e aquisição de dados;
- Teste D: Validação do fluxo de informação para a base de dados;
- Teste E: Implementação do protocolo OPC-UA para disponibilização de variáveis.

Estes testes foram realizados no ambiente ilustrado na Figura 63.

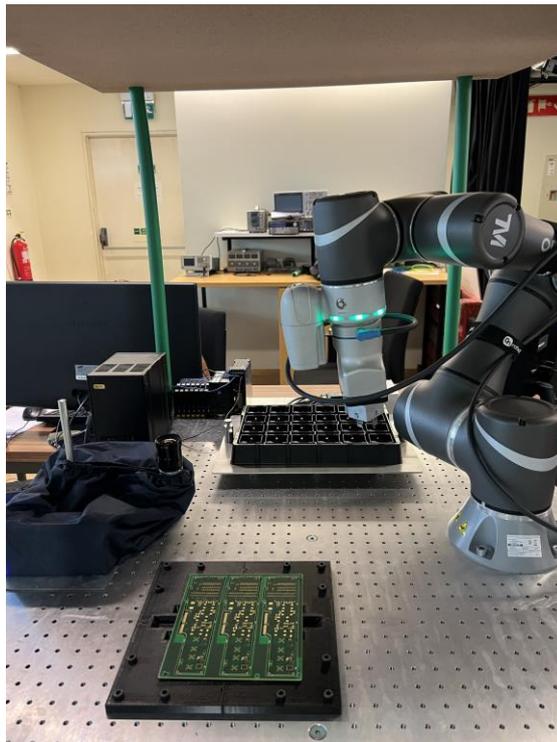


Figura 63 - Ambiente da célula de trabalho.

5.1.1 Quanto tempo leva para o *cobot* preencher uma PCB

Como foi apresentado inicialmente na lista de requisitos desta célula de trabalho, o robô deve ser rápido o suficiente para preencher corretamente uma PCB em menos de 30 segundos. Neste teste, a

ideia passa por executar a tarefa total do sistema, isto é, a realização de *pick and place*, referência da placa para garantir a correta e segura inserção de componentes, e verificação da integridade de cada componente e respetiva inserção, registando o tempo que a célula de trabalho demora a cumprir tal tarefa, repetindo vários ensaios a diferentes velocidades. Na Figura 64 encontra-se o fluxograma sucinto que explica esta tarefa.

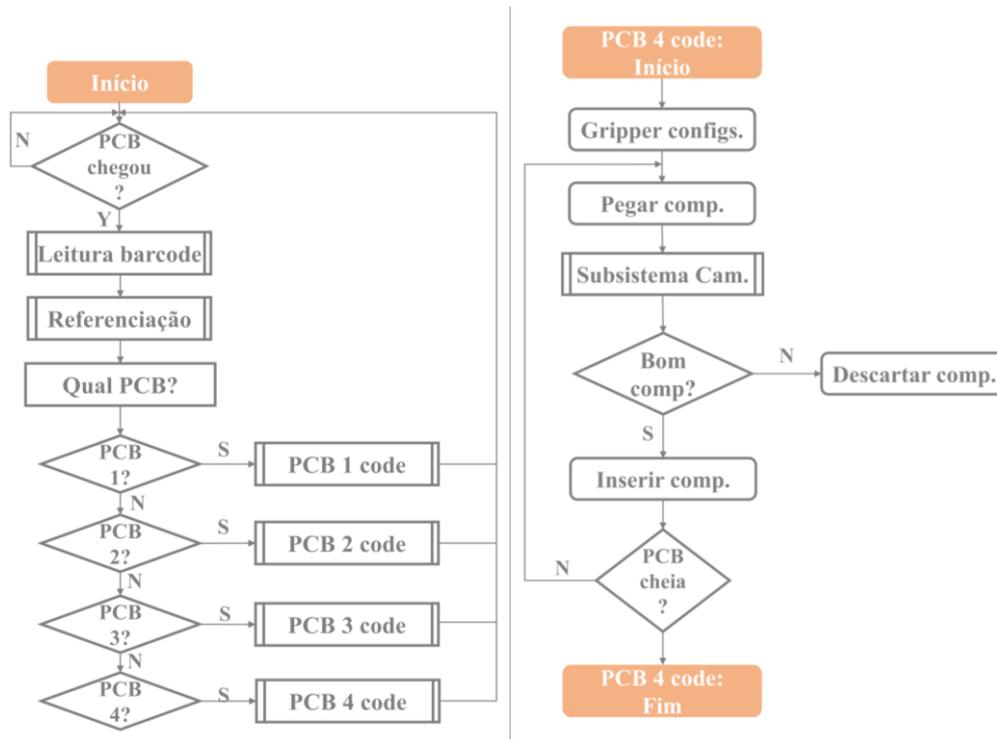


Figura 64 - Fluxograma da tarefa principal da PCB tipo 4.

A velocidade máxima do robô foi colocada em 1.5 m/s, e os ensaios foram realizados com diferentes percentagens dessa velocidade, nomeadamente a 20%, 40%, 60%, 80% e 100%. De notar dois aspetos: primeiro foi possível constatar que o robô nunca atinge a velocidade máxima uma vez que os movimentos necessários para esta tarefa não possuem a distância suficiente para tais velocidades serem atingidas; e ainda, em certas trajetórias, devido às medidas de segurança implementadas, o robô não conseguia executar o movimento, sendo que era necessário estabelecer um valor inferior específico de velocidade para essa trajetória. A Tabela 3 mostra os tempos registados para cada definição de velocidade.

Tabela 3 - Tempo necessário para preencher a PCB a diferentes percentagens de velocidade.

Velocidade	Tempo
20%	0:59,260
40%	0:38,672

60%	0:31,313
80%	0:29,241
100%	0:28,039

Através dos resultados presentes na tabela acima, é possível verificar que o robô é capaz de preencher uma PCB do tipo 4 em menos de 30 segundos, sendo uma PCB que leva três componentes.

Assim, demonstrou-se que o robô pode de facto cumprir os requisitos mencionados, tanto em termos de velocidade como de eficiência. Contudo, ainda não se demonstrou o seu valor acrescentado quando comparado com o operador humano. Para tal, foram realizados testes adicionais, onde o operador apenas tinha de pegar no componente e tentar inserir na placa de montagem. Apesar de inicialmente o ser humano demorar cerca de 20 segundos a preencher uma placa, após algum treino, reparou-se que este tempo descia para uma média de 15 segundos por placa. De forma a comparar isto com o robô colaborativo, foi realizada uma pequena tarefa, demonstrada na Figura 65.

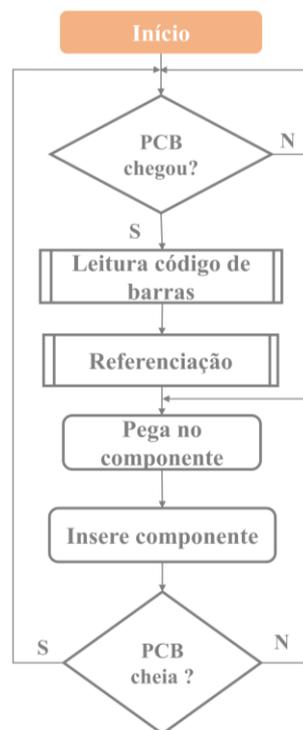


Figura 65 - Fluxograma da tarefa comparativa ao ser humano.

Realizando o mesmo tipo de teste (isto é, sem verificação da integridade do componente), registou-se um tempo médio de 18,780 segundos para que o robô preenchesse na totalidade a placa de montagem. Embora comparando diretamente os dois tempos se verifique que o ser humano é mais rápido, globalmente, é possível concluir que o robô é capaz de manter este ritmo continuamente, não apresentando alterações significativas no resultado final, nem atrasos devido a erros ou pela necessidade

de pausas para descanso. Acrescenta-se ainda o facto de o robô, com apenas mais 10 segundos (Tabela 3), ser capaz de verificar a integridade de cada componente inserido, garantindo uma maior fiabilidade no que toca à placa no seu estado final.

5.1.2 Fiabilidade do sistema de visão externo

Este teste visa confirmar a precisão do sistema de visão externo para a validação da integridade de cada componente. Para tal, diferentes componentes foram estruturalmente comprometidos em áreas diferentes, e a tarefa foi executada 10 vezes para cada tipo de componente danificado, de forma a contabilizar o número de verdadeiros positivos que o sistema de visão poderia detetar (Figura 66). Além disso, foi também utilizado um componente em bom estado para verificar se no meio dos componentes danificados, o sistema de visão era capaz de o identificar.

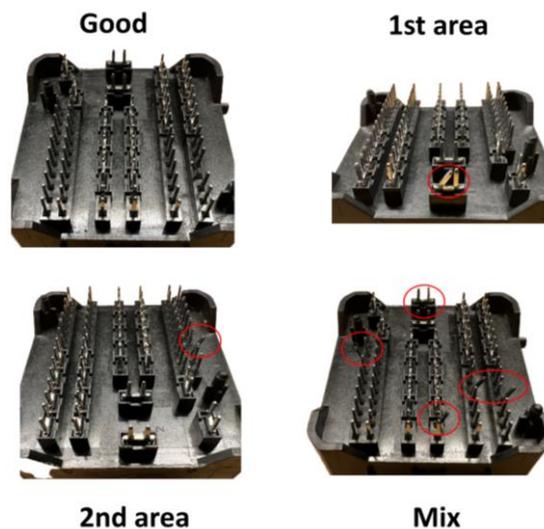


Figura 66 - Exemplos dos componentes utilizados.

Durante os 10 ensaios de cada teste, o sistema de visão demonstrou a capacidade necessária, validando todos os componentes que se encontravam em bom estado, e rejeitando aqueles que se encontravam danificados (Figura 67 e Figura 68). Além disso, através do *software* do próprio sistema de visão, é possível verificar onde se encontra o dano nos componentes comprometidos, apresentando aqui também, um resultado de 10 em 10 (Figura 68).

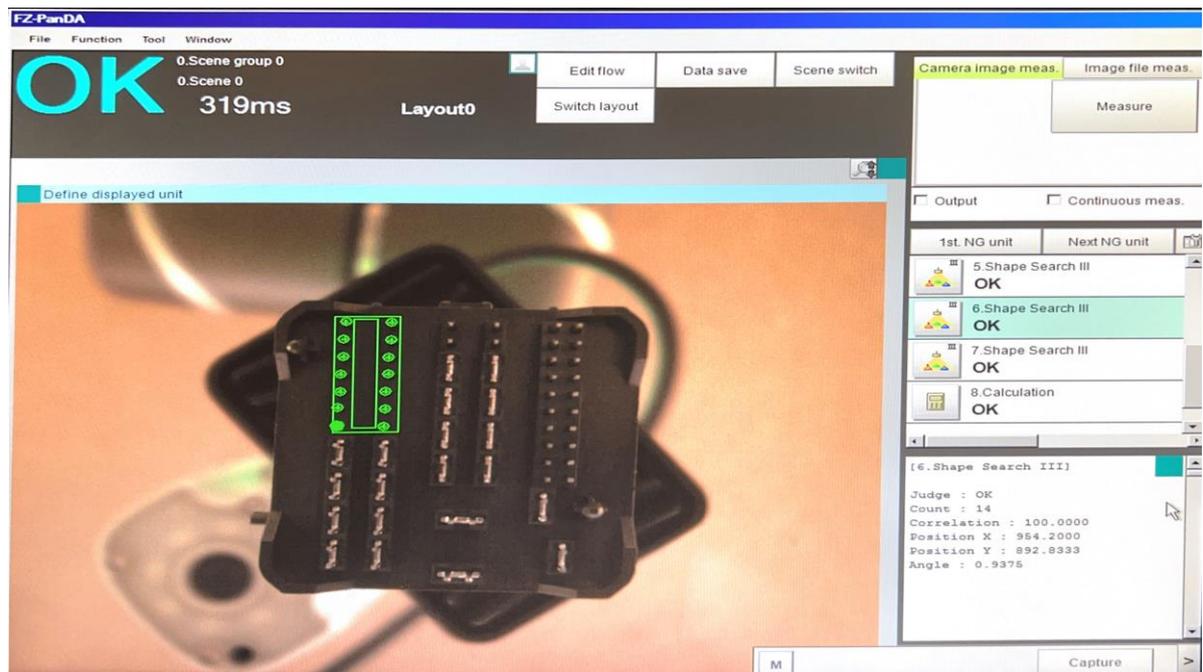
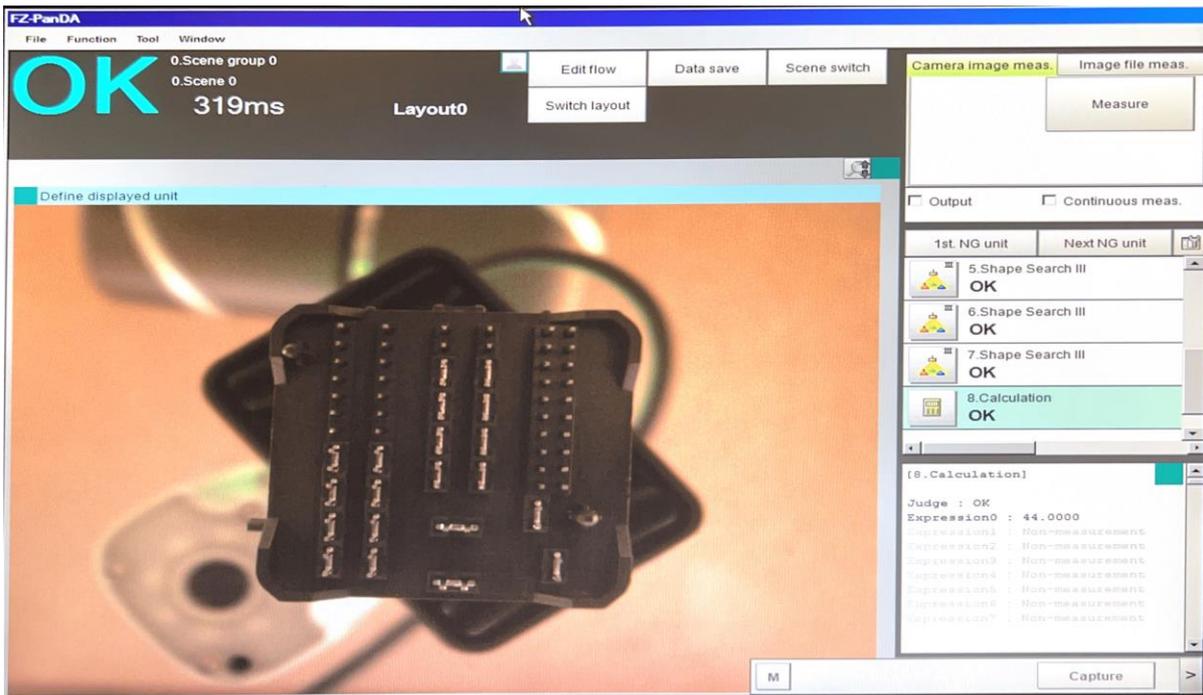


Figura 67 - Componente em bom estado, apresentando uma contagem de 44 pinos.

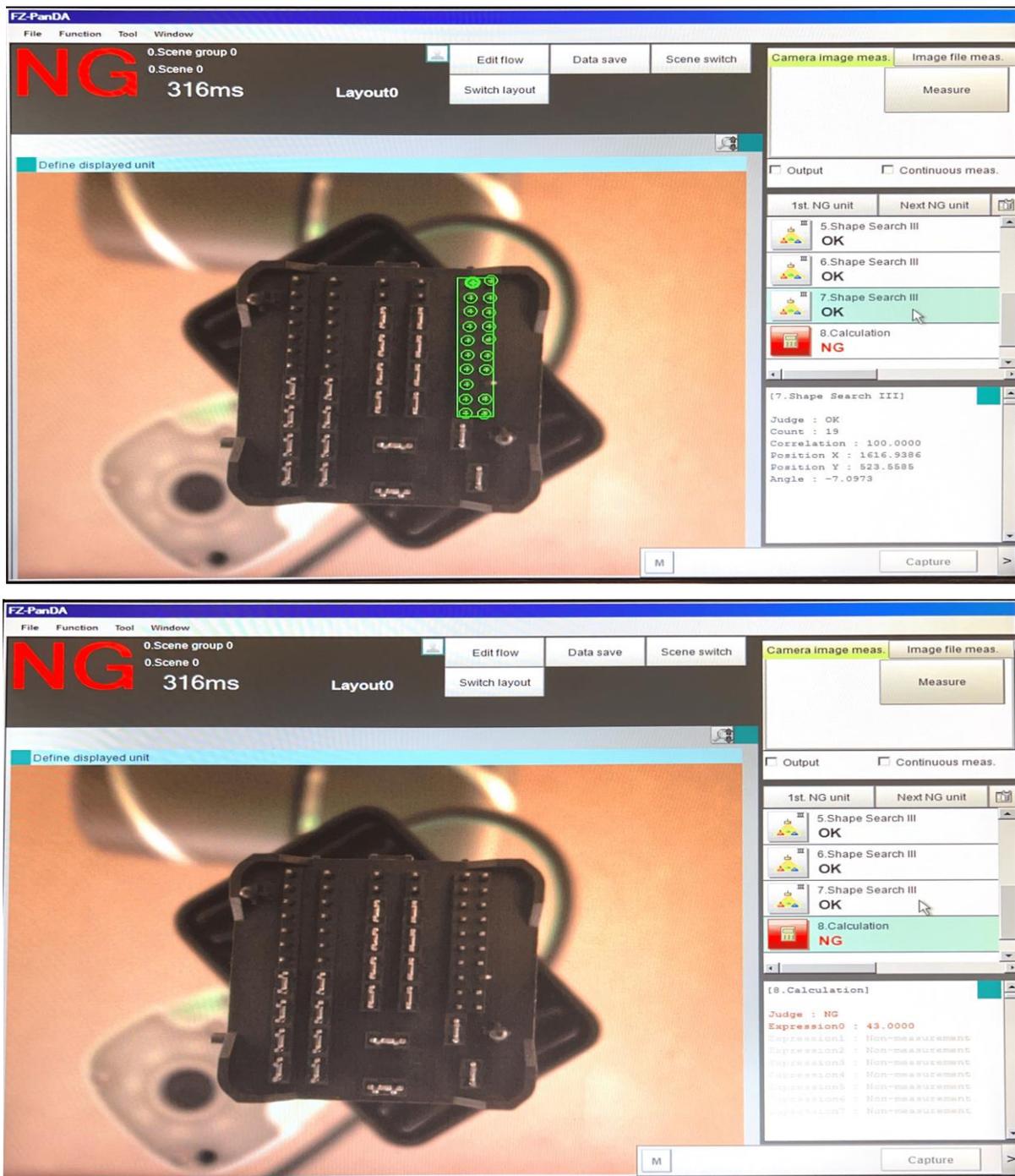


Figura 68 - Componente em mau estado, sendo possível observar qual o pino que se encontra danificado.

De notar que, a partir do momento que o robô atinge a zona de inspeção, até ao momento em que o sistema principal (PLC) recebe a resposta por parte do sistema de visão, a câmara apenas precisa de 0.8-1s para processar e analisar a informação. É possível constatar este facto analisando a Figura 58, a Figura 67 e a Figura 68, onde se pode verificar que o processo de análise por parte do sistema de visão externo possui três *timers* de 200 ms e um tempo de 319 ms a concluir o próprio código, dando um total de 919 ms.

5.1.3 Monitorização e aquisição de dados

Este teste tem como objetivo verificar que o PLC é capaz de recuperar a informação e os dados necessários durante a execução da tarefa acima mencionada, para posterior visualização, documentação, e realização de estudos sobre estes mesmos dados.

O número de componentes, número de PCBs preenchidas, entre outros, são exemplos de dados essenciais a retirar, de forma a ser mais fácil por parte do operador saber quando é necessário repor uma nova caixa de componentes, conhecer a percentagem de componentes danificados num determinado lote, ou ainda, ter noção de quantas PCBs de um determinado tipo é possível preencher ao longo do dia.

De forma a verificar a informação registada por parte do PLC, foi desenhada uma interface no *software CX-Designer* (Figura 69).

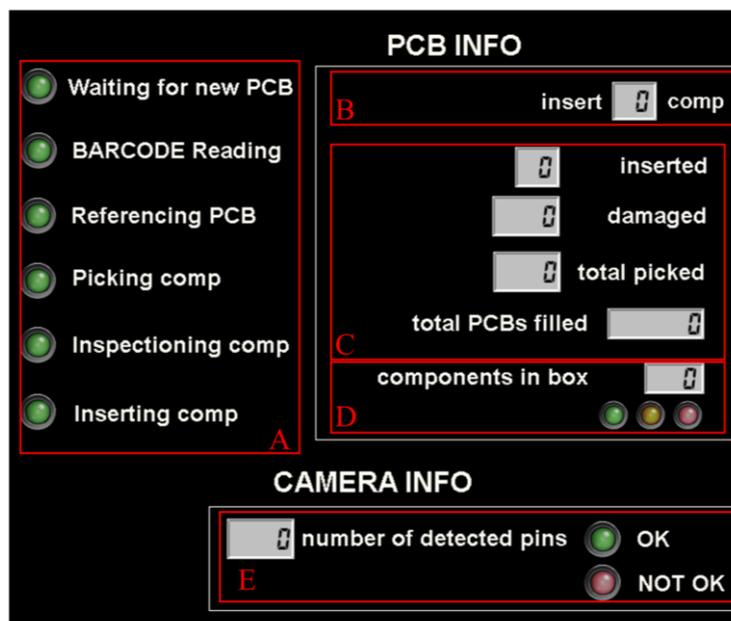


Figura 69 - Interface construída no *software CX-Designer*.

A Figura 69 ilustra as cinco secções principais desta interface:

- Secção A: notifica o operador sobre as etapas a serem realizadas durante a rotina, bem como a tarefa em execução no momento;
- Secção B: informa o tipo de PCB que foi detetado (através da leitura do código de barras), e o número de componentes que leva;
- Secção C: apresenta o número de componentes inseridos, o número de componentes danificados, e o número total de componentes utilizados;
- Secção D: informa o número de componentes que a caixa contém, possuindo um conjunto de LEDs de forma a facilitar a interpretação por parte do operador. O LED amarelo indica que a

caixa se encontra praticamente sem componentes, e o LED vermelho indica que é necessário repor a caixa de componentes;

- Secção E: permite saber o número de pinos que se encontram em bom estado, mostrando o resultado do sistema de visão externo.

A Figura 70 ilustra uma possível sequenciação do que pode ser observado nesta interface ao longo de várias iterações de montagem de PCBs.

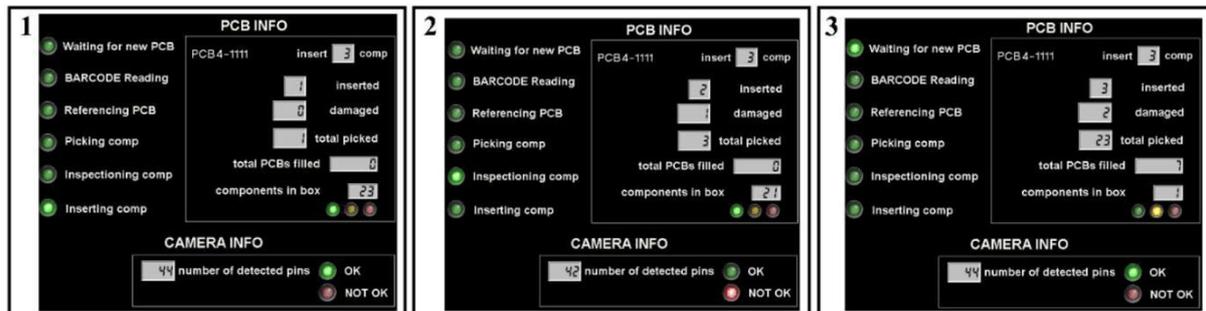


Figura 70 - Interface durante iterações sequenciadas.

As duas primeiras imagens foram gravadas durante o preenchimento da primeira PCB (Figura 70-1 e Figura 70-2). Na primeira imagem, mostra-se a tarefa de *pick and place* e a validação de um componente, onde é possível ver o número de pinos contabilizados pelo sistema de visão externo (44), fases estas que ocorrem durante a etapa de “*Insert comp*”. Já a segunda imagem mostra a última inserção esperada, onde se verifica que se trata de um componente danificado, uma vez que o número de pinos contabilizados é inferior ao número teórico ($42 < 44$), apresentando este resultado tanto na secção E, como na secção C (com o respetivo incremento no número de componentes danificados). A última figura (Figura 70-3) ilustra uma possível iteração da interface ao longo de um dia de trabalho, demonstrando o número total de componentes utilizados até então, o número de PCBs preenchidas, entre outros dados.

5.1.4 Validação do fluxo de informação para a base de dados local

Este teste visa validar a correta implementação da base de dados, demonstrando o fluxo da informação mais importante registada por parte do PLC.

Como se pode observar na Figura 62, existe um conjunto de blocos que inicializa e mapeia a base de dados, possibilitando o envio dos dados necessários. Este envio é efetuado no final de cada PCB preenchida, como é possível observar na Figura 51 na linha 26, sendo que apenas os dados existentes na tabela criada na base de dados (Figura 59) são enviados. É possível a observação destes dados através do software em questão, o *Microsoft SQL Server Management Studio* (Figura 71).

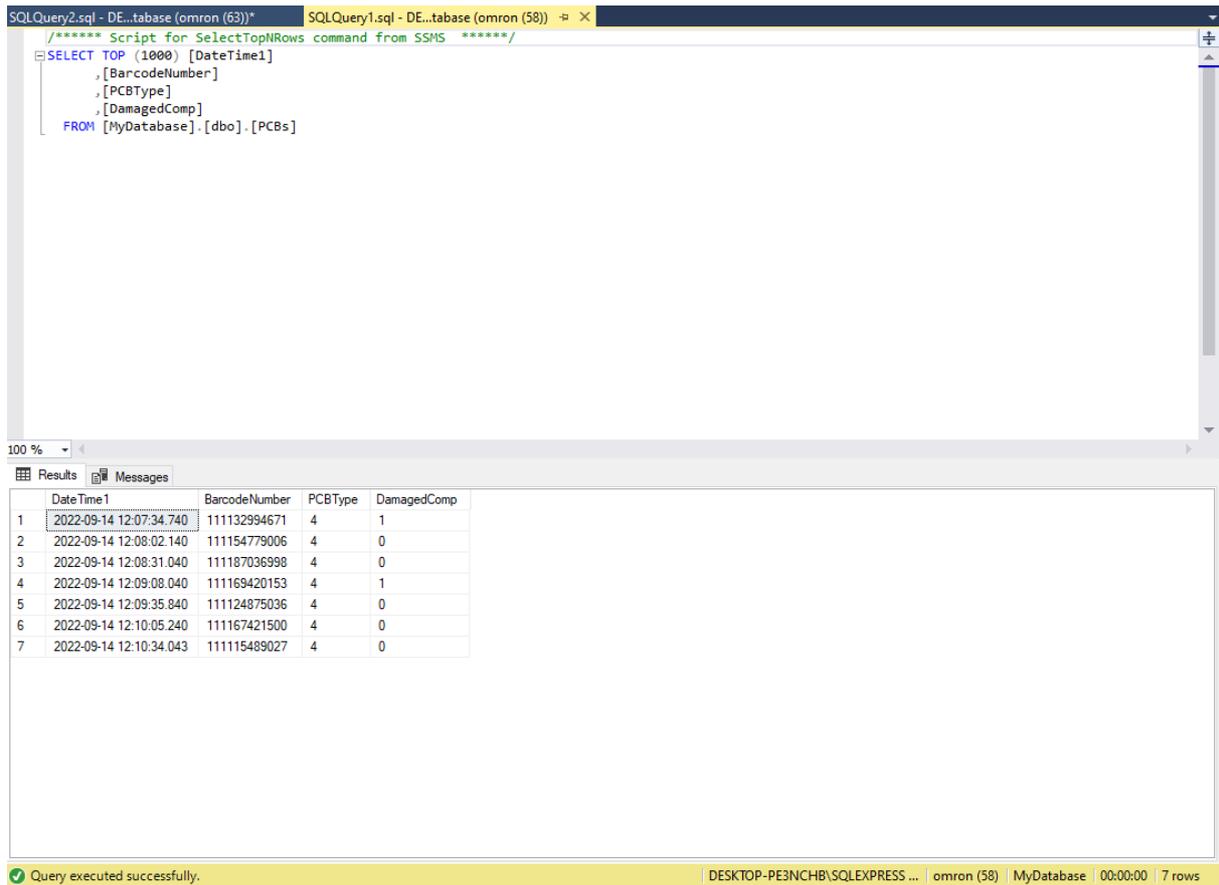


Figura 71 - Observação das variáveis da tabela criada na base de dados.

Com a observação da tabela, é possível verificar que o robô demora, em média, 27 a 30 segundos para preencher uma placa do tipo 4, excetuando quando encontra um componente danificado, sendo obrigado a repetir o processo (linha número 4, onde se observa um tempo total de 37 segundos).

5.1.5 Implementação do protocolo OPC-UA para disponibilização de variáveis

Neste último teste pretende-se demonstrar a correta implementação do protocolo OPC-UA, possibilitando a opção de visualização de certas variáveis em tempo real.

Este teste foi realizado em simultâneo com o teste C (5.1.3), sendo possível observar os mesmos valores da Figura 70 (Figura 72).

Comparando ambas as figuras, Figura 70 e Figura 72, é possível verificar que numa primeira iteração o robô apenas tinha realizado a pega num componente, estando esse em bom estado. De seguida, verifica-se que o último componente a colocar na placa se encontra danificado, impossibilitando o preenchimento da mesma (mantendo o valor de $nPCB1111$ igual a 0). Por fim, no final de sete placas preenchidas, o total de componentes utilizados é igual a 23, uma vez que durante este processo havia dois componentes danificados.

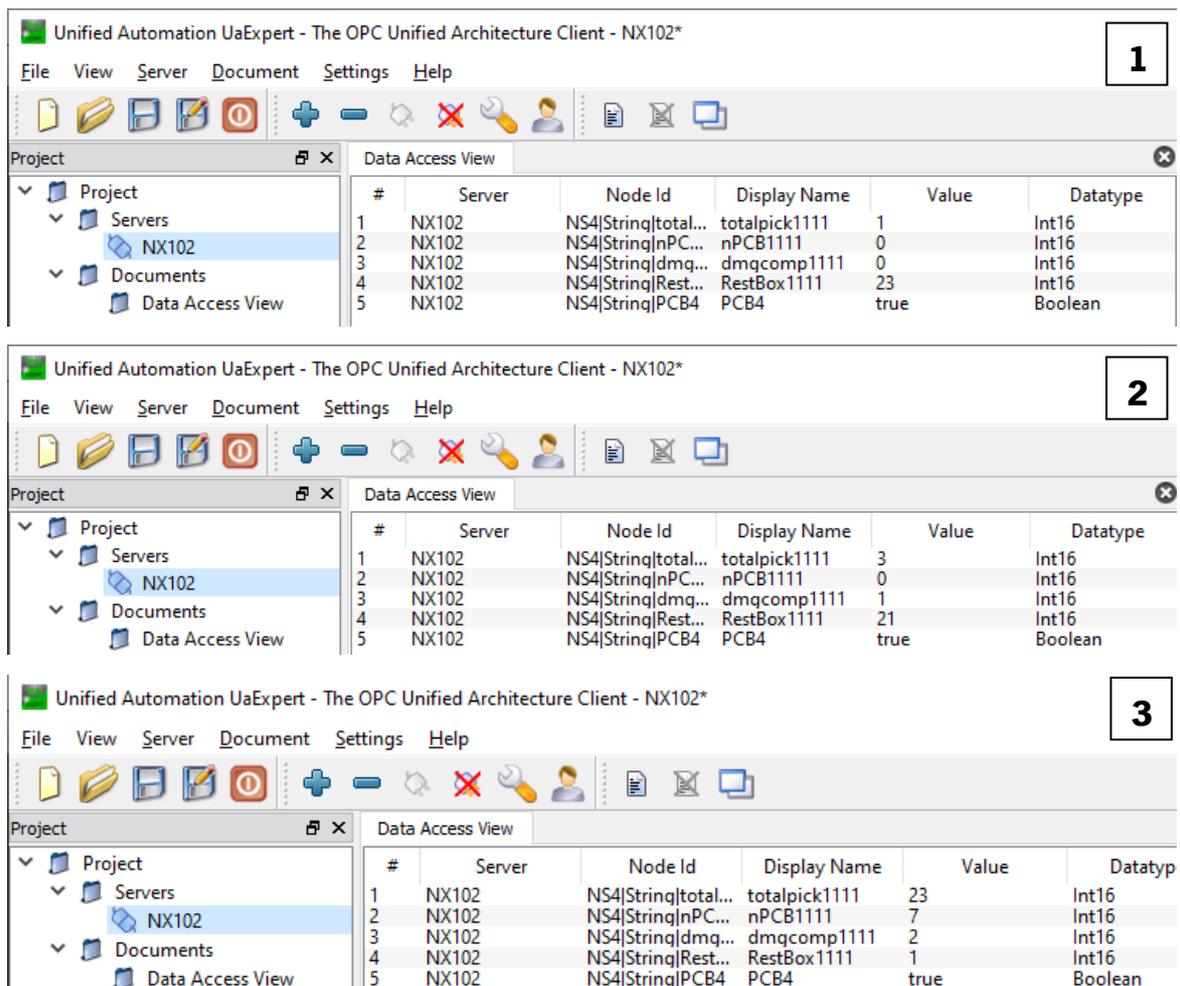


Figura 72 - Visualização em tempo real das variáveis.

5.2 Discussão

Durante o Teste A, confirmámos o cumprimento do requisito dos 30 segundos como tempo máximo para preenchimento de uma PCB. Realizámos alguns ensaios de inserção para comparar com um operador humano, concluindo que o operador pode ser mais rápido, em média, 3,780 segundos. No entanto, outros requisitos devem ser considerados, como a eficiência, a repetibilidade e a precisão. Os seres humanos são suscetíveis ao cansaço, e após horas de trabalho, a sua precisão e eficiência não será a mesma, levando a atrasos na produção ou, nos piores casos, PCBs danificadas. Em oposição, o *cobot* atua sempre ao mesmo nível, sustentando estes requisitos ao longo do tempo. Estes requisitos também são possíveis com robôs industriais, mas eles têm algumas desvantagens, tais como: são utilizados para cargas úteis mais pesadas, e também requerem uma área de trabalho isolada, indo contra a ideia de trabalho colaborativo.

Além disso, foram testadas duas posições diferentes da câmara externa. Inicialmente colocou-se a câmara apontada para baixo, demonstrando uma desvantagem enorme uma vez que obriga o braço robótico a executar um movimento de rotação extremo e lento. Por sua vez, ao colocar a câmara a apontar para cima, concluiu-se que esta seria a forma mais rápida e fácil de conseguir um trabalho eficiente, uma vez que as variações do meio ambiente não afetavam a precisão e decisão rápida por parte deste sistema. Pensando em possibilidades diferentes deste sistema de visão, ponderou-se duas ideias: (1) a utilização da câmara do *cobot* para efetuar a inspeção, sendo que esta não apresentava os requisitos mínimos para este processo; e (2) a utilização de uma câmara externa acoplada ao *cobot*, com os requisitos de qualidade e resolução, mas que aumentaria os tempos de ciclo de todo o processo, uma vez que o robô necessitaria de colocar o componente numa posição intermédia para realizar a leitura.

Uma das tarefas nesta célula de trabalho que ainda requer um operador consiste na substituição das caixas de componentes vazias. No teste B, concluiu-se que é vantajoso a apresentação deste tipo de informação com o recurso a três LEDS, para além de um rótulo com o número de componentes restantes, dando assim um aviso luminoso explícito ao operador sobre a quantidade de componentes existentes na caixa e a necessidade da sua substituição. Para além desta tarefa, para reduzir o tempo de ciclo, o operador pode cooperar com o *cobot*, limitando a área de pesquisa dos códigos de barras por parte do robô (pelo facto de cada tipo de PCB possuir uma zona específica para o código de barras). Esta colaboração dá-se através de botões, que ao serem acionados especificam a área de pesquisa que o robô deve realizar.

O teste C confirmou o elevado nível de precisão do sistema de visão externo na validação de menores deslocamentos e defeitos nos pinos dos componentes, proporcionando assim uma sensibilidade e exatidão que o olho humano não consegue alcançar.

Como mencionado anteriormente, o operador humano poderia substituir algumas funcionalidades implementadas na célula de trabalho, como a leitura do código de barras e as funções de referenciação. Aqui, o trabalhador poderia ser responsável por informar o *cobot* do tipo de componentes que devem ser inseridos. No entanto, a ideia é minimizar a interação, e através da função de leitura do código de barras é possível efetuar esta tarefa de forma autónoma, sem perda de tempo significativo. Além disso, uma vez que não há garantias que a placa chegue ao local de inserção sempre da mesma forma e na mesma posição, devido à possibilidade de ocorrerem pequenos deslocamentos ao longo do tempo, ao utilizar a função de referenciação da placa, garantimos a correta colocação de cada componente,

Para além da solução proposta ao longo do trabalho, outras duas abordagens foram consideradas. A primeira abordagem considerava uma maior colaboração entre o humano e o robô, sendo que este último executaria a tarefa de validação do componente através do seu sistema de visão, e o humano seria responsável pela sua inserção. Por outro lado, na segunda hipótese, o operador realizaria a tarefa de validação do componente e, depois de ter sido validado, seria o *cobot* a efetuar a sua inserção. Ambas as soluções têm desvantagens uma vez que aumentam o tempo de ciclo do processo, com a transição do componente entre o operador e o *cobot*, bem como introduzem uma dependência de processos que não garantem o mesmo nível de precisão e eficiência (o operador).

6. CONCLUSÃO

Neste trabalho, foi apresentada uma solução para um protótipo de uma célula de trabalho autónoma capaz de montar PCBs com componentes PTH. A referida célula consiste em três sistemas principais: (1) um *cobot*; (2) um sistema de visão externo; e (3) um PLC, trocando informações entre eles através do protocolo TCP/IP.

O primeiro subsistema, o *cobot*, é responsável principalmente pelo movimento na tarefa principal, incluindo pegar nos componentes, levá-los para a área de inspeção, e inseri-los na PCB. O subsistema de visão assegura o posicionamento e a identificação precisa da PCB, e verifica a condição dos diferentes componentes. O último subsistema, o PLC, gere as interações tanto com o *cobot* como com o sistema de visão externo, e é responsável pela monitorização da célula de trabalho, registando e armazenando numa base de dados os dados mais relevantes gerados durante a execução da tarefa.

O presente documento descreve a implementação da referida célula de trabalho, e mostra os aspetos-chave na criação de cada subsistema. Por fim, foram apresentados cinco testes para validação do protótipo proposto, focando-se nos aspetos mais importantes como a troca e aquisição de informação, a validação de componentes, e a precisão e a velocidade durante a montagem da PCB em comparação com um operador humano, todos eles apresentando um resultado positivo.

Em suma, este trabalho apresenta uma célula de trabalho operacional que visa otimizar um processo que é atualmente efetuado manualmente, demonstrando que é possível a substituição desta tarefa por um sistema capaz de manter altos níveis de desempenho num longo período de tempo.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Sarah Yasin. (2018, 30/12). *Surface Mount Electronic Components and their Types*. Available: <https://www.electroniclovers.com/2018/10/surface-mount-electronic-components-and-their-types.html>
- [2] T. Kalsoom, N. Ramzan, S. Ahmed, and M. Ur-Rehman, "Advances in sensor technologies in the era of smart factory and industry 4.0," *Sensors*, vol. 20, no. 23, p. 6783, 2020.
- [3] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *2016 49th Hawaii international conference on system sciences (HICSS)*, 2016, pp. 3928-3937: IEEE.
- [4] Altus. (2019, 30/12). *Conheça os nove pilares da Indústria 4.0 e a sua relevância para a atividade industrial*. Available: <https://www.altus.com.br/post/212/conheca-os-nove-pilares-da-industria-4-0-e-sua-relevancia-para-a-atividade-industrial>
- [5] A. Ustundag and E. Cevikcan, *Industry 4.0: managing the digital transformation*. Springer, 2017.
- [6] Matt Reed. (2019, 30/12). *Innovation 4.0: A Digital Revolution for R&D*. Available: <https://www.newstatesman.com/spotlight/2019/09/innovation-40-digital-revolution-rd>
- [7] S. L. Anderson, "Asimov's "three laws of robotics" and machine metaethics," *Ai & Society*, vol. 22, no. 4, pp. 477-493, 2008.
- [8] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics. Industrial Robotics: Theory, Modelling and Control, Sam Cubero," ed: InTech, 2006.
- [9] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.
- [10] L. R. B. d. Oliveira, "Cinemática inversa para simulação de figuras articuladas," 2018.
- [11] Atomoclast. (2017, 30/12). *Forward and Inverse Kinematics, an Introduction*. Available: <https://realitybytes.blog/2017/06/16/forward-and-inverse-kinematics-an-introduction/>
- [12] Leandro, "Estrutura e características gerais dos robôs," 2018, Available: https://professorleandro.weebly.com/uploads/3/8/6/3/3863436/4_estrutura.pdf, Accessed on: 3/1.
- [13] B. ISO, "8373: 2012-Robots and robotic devices. Vocabulary," *BSI*. <https://shop.bsigroup.com/ProductDetail>, 2012.
- [14] Fernando Ribeiro, "Robótica: configuração de robôs," 2020, Available: https://elearning.uminho.pt/webapps/blackboard/execute/content/file?cmd=view&content_id=1094413_1&course_id=47899_1, Accessed on: 30/12.
- [15] A. Dobra, "General classification of robots. Size criteria," in *2014 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD)*, 2014, pp. 1-6: IEEE.
- [16] R. Lousada. (2020, 30/12). *O que é Robótica: conceito, história e evolução*. Available: <https://blog.eletrogate.com/o-que-e-robotica-conceito-historia-e-evolucao/>
- [17] I. F. o. Robotics, "Demystifying Collaborative Industrial Robots," *Frankfurt, Germany*, 2018.
- [18] P. Poór, T. Broum, and J. Basl, "Role of collaborative robots in industry 4.0 with target on education in industrial engineering," in *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*, 2019, pp. 42-46: IEEE.
- [19] S. Vojčić, "Applications of collaborative industrial robots," *Machines. Technologies. Materials.*, vol. 14, no. 3, pp. 96-99, 2020.
- [20] TM-Robot. (2019, 30/12). *Regular Payload Series*. Available: <https://www.tm-robot.com/en/regular-payload/>
- [21] Cognex, "Introdução ao sistema de visão," 2021, Available: <https://www.cognex.com/pt-pt/what-is/machine-vision>, Accessed on: 30/12.
- [22] J. P. David A. Forsyth, *Computer Vision - A Modern Approach*. 2002, p. 793.
- [23] P. R. G. Kurka and A. A. D. Salazar, "Applications of image processing in robotics and instrumentation," *Mechanical Systems and Signal Processing*, vol. 124, pp. 142-169, 2019.
- [24] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*. McGraw-hill New York, 1995.
- [25] R. Hirsch, *Exploring colour photography: a complete guide*. Laurence King, 2005.
- [26] E. R. Davies, *Computer and machine vision: theory, algorithms, practicalities*. Academic Press, 2012.
- [27] G. J. Aguiar, B. B. Zarpelao, and S. BARBON JUNIOR, "Melhoria de Contraste em Imagens Digitais baseado em Inteligência Artificial," ed: Londrina, 2017.
- [28] MathWorks. (30/12). *Enhancement methods in image processing*. Available: <https://www.mathworks.com/discovery/image-enhancement.html>
- [29] Gaudenz Boesch, "Object Detection in 2021: The Definitive Guide," 2021, Available: <https://viso.ai/deep-learning/object-detection/>, Accessed on: 30/12.
- [30] S. Zhu, X. Xia, Q. Zhang, and K. Belloulata, "An image segmentation algorithm in image processing based on threshold segmentation," in *2007 third international IEEE conference on signal-image technologies and internet-based system*, 2007, pp. 673-678: IEEE.

- [31] Wikipedia. (2021, 30/12). *Thresholding (image processing)*. Available: [https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))
- [32] Y. Crama, O. E. Flippo, J. Van De Klundert, and F. C. Spijksma, "The assembly of printed circuit boards: A case with multiple machines and multiple board types," *European journal of operational research*, vol. 98, no. 3, pp. 457-472, 1997.
- [33] K. Altinkemer, B. Kazaz, M. Köksalan, and H. Moskowitz, "Optimization of printed circuit board manufacturing: Integrated modeling and algorithms," *European Journal of Operational Research*, vol. 124, no. 2, pp. 409-421, 2000.
- [34] K. Bogner, U. Pferschy, R. Unterberger, and H. Zeiner, "Optimised scheduling in human–robot collaboration—a use case in the assembly of printed circuit boards," *International Journal of Production Research*, vol. 56, no. 16, pp. 5522-5540, 2018.
- [35] K. Andrzejewski, M. Cooper, C. Griffiths, and C. Giannetti, "Optimisation process for robotic assembly of electronic components," *The International Journal of Advanced Manufacturing Technology*, vol. 99, no. 9, pp. 2523-2535, 2018.
- [36] M. Cooper, C. Griffiths, K. Andrzejewski, and C. Giannetti, "Motion optimisation for improved cycle time and reduced vibration in robotic assembly of electronic components," *AIMS Electronics and Electrical Engineering*, vol. 3, no. 3, pp. 274-289, 2019.
- [37] C. Fonseka and J. Jayasinghe, "Implementation of an automatic optical inspection system for solder quality classification of THT solder joints," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 2, pp. 353-366, 2018.
- [38] S. Celaschi, M. S. de Castro, A. Fernandes, and A. Xavier, "Machine vision for automatic inspection of pin through hole components assembled on a PCB," *Global Journal of Engineering and Technology Advances*, vol. 6, no. 1, pp. 091-098, 2021.
- [39] Omron. (2020). *Software Manual TMvision*. Available: <https://www.edata.omron.com.au/eData/manuals.html>
- [40] Omron. (8/01). *AI-based defect detection that exceeds the ability of expert inspectors*. Available: <https://docs.rs-online.com/a45b/A700000007741624.pdf>
- [41] Omron. *FH/FHV/FZ5 Series Vision System - Processing Item Function Reference Manual*. Available: <https://www.edata.omron.com.au/eData/manuals.html>

ANEXO A - WORK CELL FOR ASSEMBLING SMALL COMPONENTS IN PCB

Work cell for assembling small components in PCB

1st Mauro Queirós

Algoritmi Center, University of Minho
Guimarães, Portugal
<https://orcid.org/0000-0001-5896-7423>

4th José Meireles

MEtRICs Research Center, University
of Minho
Guimarães Portugal
<https://orcid.org/0000-0003-0881-2348>

2nd João Lobato Pereira

Algoritmi Center, University of Minho
Guimarães, Portugal
<https://orcid.org/0000-0002-1053-7405>

5th Jaime Fonseca

Algoritmi Center, University of Minho
Guimarães, Portugal
<https://orcid.org/0000-0001-6703-3278>

3rd Valdemar Leiras

Algoritmi Center, University of Minho
Guimarães, Portugal
<https://orcid.org/0000-0002-8365-7889>

6th João Borges

Algoritmi Center, University of Minho
Guimarães, Portugal
Polytechnic Institute of Cávado and
Ave, Barcelos, Portugal
<https://orcid.org/0000-0002-5880-033X>

Abstract— Flexibility and speed in the development of new industrial machines are essential factors for the success of capital goods industries. When assembling a printed circuit board (PCB), since all the components are surface mounted devices (SMD), the whole process is automatic. However, in many PCBs, it is necessary to place components that are not SMDs, called pin through hole components (PTH), having to be inserted manually, which leads to delays in the production line. This work proposes and validates a prototype work cell based on a collaborative robot and vision systems whose objective is to insert these components in a completely autonomous or semi-autonomous way. Different tests were made to validate this work cell, showing the correct implementation and the possibility of replacing the human worker on this PCB assembly task.

Keywords— Cobot; PCB assembly; work cell;

I. INTRODUCTION

In today's industry, any manual operation in repetitive production lines that causes delays in the production process needs to be semi-automated or fully automated so that the factory remains competitive in an increasingly aggressive and fast market. Printed circuit boards (PCBs) are a fine example of such necessity, as they are in virtually all electronic components of everyday life. Over the years, assembling electronic components on PCBs has undergone several changes, many of which aim to make this process increasingly automated, efficient, fast, and economically efficient [1, 2]. This is how Surface Mount Technology (SMT) arose, with surface mount devices (Surface Mount Devices) being mounted directly on the surface of the printed circuit board, generally in an automatic way using an SMT machine [3].

Despite the advances seen with SMDs, there are other components (Figure 1) that still need to be manually assembled on PCBs, such as diodes, capacitors, and connectors named Pin Through Hole (PTH) components [4]. This limitation considerably decreases the production efficiency of PCBs assembled with this type of components when compared to those that only have SMD components.



Figure 1 - Example of non-SMD components.

Aiming to mitigate the limitation mentioned above and take advantage of the rise of collaborative robots (cobots) [5, 6], this work proposes a novel industrial work cell for the automatic insertion of non-SMD components into PCBs. The proposed cell integrates three main systems from the literature: (1) a cobot to enable either collaborative or autonomous work [7]; (2) a vision system for component validation [8, 9]; and (3) an external device for controlling and monitoring the work cell [10].

II. REQUIREMENTS OF THE WORK CELL

Taking into consideration the goal of reducing or eliminating the need for manually inserting different components into PCB, but maintaining quality and efficiency during the insertion process, a set of requirements were defined, of which the most relevant are: (1) the cell must have an automatic transport system for the PCB boards; (2) it must have the ability to identify the different PCBs; (3) adapt the program and gripper according to the PCB and components to be assembled; (4) it must perform accurate and secure gripping and insertion for the different components; (5) the system needs to validate each component prior to insertion, using for example, a vision system; (6) needs to assure that the PCB is correctly and fully assembled within 30 seconds; and for last (7) it also requires collecting information relevant to the operation of the work cell, making it available in a database for future analysis and study.

III. WORKFLOW

To meet the requirements mentioned above, one can propose the use of: (1) the cobot TM5-700, a 6 DOF collaborative robot capable of using different grippers and that integrates its own vision system, with a range of 700mm and less than 0.1mm of accuracy and repeatability; (2) the FH1050 vision system by Omron, which has various image acquisition and processing functionalities, with a minimum 2M pixels as resolution to cover the entire insertion area; and (3) the NX102-9020 Omron, a PLC acting as an external device capable of monitoring and controlling all tasks and functionalities of the work cell, with multiple communication facilities, specially OPC-UA, and must have database interaction capabilities. These proposed systems were used and are connected, using the TCP/IP protocol for data exchange. Figure 2 illustrates how the subsystems interact throughout the task. In short, the PLC receives a signal from the assembly line to start the flow, and, after that, different signals are exchanged between the cobot, the PLC, and the vision system to proceed

with the task. All the essential information to be seen by the human worker is visible in an interface and then sent to a database.

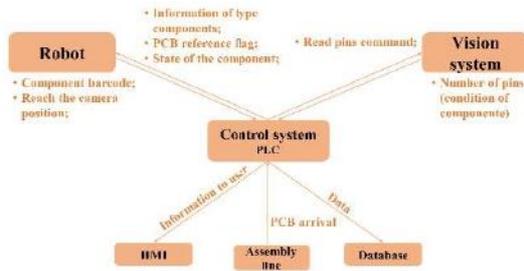


Figure 2 - Data exchanged between the three subsystems.

In short, the cobot will start by reading the barcode on the PCB to know what components must be inserted and do the referencing task to know where the PCB is. After that, it is time for the external vision system to read the number of pins each component has to validate its quality. Depending on the information received, the cobot will discard the component or place it in the proper PCB position upon the validation task. After placing the component, if the PCB is not yet fully assembled, the cobot will repeat the operation for another component. When finished, it will wait for the arrival of a new PCB.

IV. METHODOLOGY

To execute the workflow explained above, one should consider each subsystem individually and how to integrate them to focus on precision and speed, aiming to respect the following key goal: replace the worker in the assembly of PCBs with this type of components, trying to speed up the process and reduce costs without ever losing quality.

To understand the full process of this work cell, each subsystem will be explained in a more detailed way.

A. Cobot subsystem

The cobot will have two major tasks: first, it is the subsystem responsible for the movement in the main task, including picking up a component, taking it to the vision subsystem, and, finally, placing it in the right location on the PCB. The other part is regarding the internal vision system that this cobot has. Here, the cobot will perform two analyses: read the PCB's barcode to know which component to insert and reference itself with respect to the assembly board before inserting the first component. Both tasks will be solved using the TMFlow software, whereas, for the vision tasks, the TMvision software is employed.

1) Cobot vision system

To identify the PCB that just arrived and place the components in the correct position, even if the PCB comes tilted or just crooked, the cobot's internal vision system has two types of functionalities [11]: identifying functions, like barcode reading, color classifier and string match; and find functions, like pattern match, blob finder, fiducial marks, and others.

Every PCB comes with an associated barcode to know precisely which type (and amount) of component that PCB takes. Here, it is necessary to specify the region where the barcode can appear, and once the camera reads a barcode, it

returns the specific number of that PCB. Knowing the group to which that particular number belongs, it is possible to tell which components must be inserted (Figure 3).



Figure 3 - Example of a barcode and fiducial marks in a PCB.

Once the cobot reaches the position to do the referencing, it will find the fiducial marks (Figure 3) and create a new coordinate frame, where the middle point between the two fiducial marks is the origin. From this point on, regardless of the position of the PCB, the cobot will always be able to make the same movement to the insertion positions, ensuring the correct filling of the board.

B. FH vision subsystem

Here, the objective is to ensure that the component that the cobot will insert is in good condition to be possible to insert into the PCB without wasting it. Usually, this is solved by human inspection, but the goal is to replace it with a vision system to make the work cell fully automatic and faster.

The FH-1050 vision system is used, taking advantage of the FZ-PanDA software [12]. In brief, upon waiting for the cobot with the component to reach the point where the camera is pointing, the system will make one single acquisition to count the number of pins in a certain area to check if they are damaged. The idea is to divide the component into small sections, each with a specific number of pins or groups/lines of pins (Figure 4).



Figure 4 - Component divided into sections and number of pins of each section.

This procedure is done using the "Shape Search III" block, where we must select the region of interest (ROI; big red box) and the subject we want to look for in that exact area (small red box, for example). Through an object detection algorithm, this function block registers a model of an image pattern based on its contour information, detects parts of inputted images that most closely match the model, and adds each similar object to the total count. An experimental study was performed to select the suitable threshold value for the degree of similarity between the template and the new image. Afterwards, the "Calculation" block is employed to find the total number of pins that were identified in each "Shape Search III" block. If the number equals those expected for the

component in question (44 in the example in Figure 4), the component is in good condition, and therefore the cobot will proceed with its insertion. If not, the cobot will discard this component, placing it in a specific box for damaged components.

C. PLC subsystem

This subsystem aims to establish a connection between the other two subsystems, being responsible for the control and monitoring of the work cell and registering the relevant data during the execution of the task, to be later stored in a database. The whole process of this task is divided into three main parts.

First, the cobot waits for the PLC to tell it that a new PCB has arrived. Then, as mentioned before, the cobot will use its internal vision system to do the referencing task and to read the barcode, sending the reading signal to the control system. Here, the PLC will then decide what type of PCB it is and then tells the cobot what type of components to insert.

Upon picking up the component, the cobot will take it to the external vision system area. Here, the PLC will send two commands: (1) first the command *measurement*, where the system will run the code to count the number of pins to assess the component's condition; and (2) the *get data* command, where the PLC will receive from the external vision system the number of pins that the software detected. Then, the PLC compares this value with the expected value, and if the result is positive, it informs the cobot that it can proceed with the insertion. If it is not, the PLC informs the cobot to discard this component and pick a new one.

Finally, the cobot will proceed with the insertion of the good component, repeating these steps until the PCB is filled. Meanwhile, during these steps, the PLC will store relevant data, such as the number of PCBs filled, the number of damaged components, and the time to fill one PCB, among others.

V. TESTS

This section demonstrates some practical examples of this work cell, showing the correct implementation of the main systems and the compliance with the requirements. The tests were: how much time does it take for the cobot to fill a PCB and the reliability of the external vision system.

These tests were employed in the following environment, where it is possible to see: (A) the external vision system for inspection; (B) the cobot with the internal vision system for barcode reading and referencing; (C) the box of components; (D) the PCB. (Figure 5).

A. How much time does it take for the cobot to fill a PCB

In this test, the idea was to run the full task of this work cell and see how long it takes to fill a PCB, repeating this trial for different velocities.

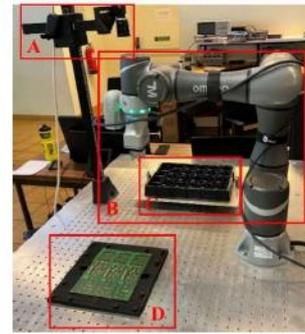


Figure 5 - Environment of the work cell.

The max velocity was set to 1.5 m/s, and the trials were done for 20%, 40%, 60%, 80% and 100% of that speed. Of note, it was possible to see that the cobot never reaches the max speed because the movements that it makes during this task do not have the range needed to reach that velocity. Table 1 shows the time recorded for each speed setting.

Table 1 - Time to fill a PCB with different speeds

Velocity	Time (m/s)
20%	0:59,260
40%	0:38,672
60%	0:31,313
80%	0:29,241
100%	0:28,039

It is possible to conclude that the cobot needs less than 30 seconds to fill a PCB of type 4 (one that takes three components).

Comparing to a human worker, which initially takes around 20 seconds to fill a PCB, without inspection, this time was reduced to an average of 15 seconds after a few attempts, making it a bit faster than the cobot. But in this case, the operator is susceptible to tiredness and distractions, making their precision and efficiency decrease over time, unlikely the cobot that is capable of maintaining his pace at his highest level.

B. Reliability of the external vision system

This last test aims to confirm if the external vision system has the capacity and reliability to validate different types of damaged components. For that, different components were structurally compromised in different areas, and the task was run ten times to inspect how many true positives the vision system could detect. Additionally, the same test was run for one good component to validate if the system could validate its integrity either (Figure 6).

During the ten trials of each test, the external vision system showed its reliability when validating the components, verifying that it is in bad condition in all of the examples mentioned above, except for the case of a "good" component. Moreover, it has also the capability of showing in which area the damage is, presenting a result of 10 out of 10 successful validations.



Figure 6 - Examples of the components used.

VI. DISCUSSION

During Test A, we confirmed the accomplishment of the 30 second requirement. We performed some insertion trials to compare this with a human worker, concluding that the worker can be faster by 1.280 seconds on average. However, some major requirements need to be considered, like efficiency, repeatability, and precision. Humans are susceptible to tiredness, and after hours of work, their precision and efficiency will not be the same, leading to delays in production or, in worst cases, damaged PCBs.

Moreover, different camera positions were tested during this test to find the easiest and fastest way to achieve a more efficient work. When the camera is on top, the environment lightning variations do not affect the external vision system precision and fast decision making. There were still two other possibilities: one being the cobot's camera performing the inspection, however this camera did not present the minimum requirements for this process; and the other was an external camera attached to the cobot, which met the quality and resolution requirements, but increased the cycle time of the entire process since the robot would need to place the component in an intermediate position to take the reading.

As mentioned before, the human worker could replace some functionalities of this work cell, like the barcode read and the referencing functions. Here, the worker could be responsible for informing the cobot of what type of components should be inserted. However, the idea is to minimize the human interaction, and through the barcode read function is possible to pass this information in an autonomous way without wasting much time. Also, when using the referencing function, we guarantee the correct placement of each component, as even when using a mechanical interlock, there is no guarantee of the correct PCB orientation, which over time can cause displacements in the order of millimeters.

VII. CONCLUSION

In this work, was presented a solution to a prototype autonomous work cell capable of assembling PCBs with PTH components. This work cell consisted of three main systems: (1) the cobot; (2) the vision system; and (3) the PLC, exchanging information between them via TCP/IP protocol.

The first one is mainly responsible for the movement in the main task, including picking up the components, taking them to the inspection area, and inserting them into the PCB. The vision subsystem ensures the precise positioning and identification of the PCB and verifies the components'

condition. The last subsystem manages both the cobot and external vision system's interactions and is responsible for monitoring the work cell by registering the relevant data generated during the execution of the task and storing it in a database.

Hereto, the three subsystems were demonstrated and explained, always showing the key aspects of the creation of this work cell. The performance of the proposed work cell was demonstrated through three examples, focusing on the most important aspects, namely exchange and acquisition of information, the validation of components, and the gain in precision and speed during PCB assembly over a human operator, all of them showing positive and successful results.

ACKNOWLEDGMENT

This work is co-funded by the European Regional Development Fund (ERDF) through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) of the Portugal 2020 Program [Project No. 45070, "FlexASComp"; Funding Reference: POCI-01-0247-FEDER-045070].

REFERENCES

- [1] K. Bogner, U. Pferschy, R. Unterberger, and H. Zeiner, "Optimized scheduling in human-robot collaboration—a use case in the assembly of printed circuit boards," *International Journal of Production Research*, vol. 56, pp. 5522-5540, 2018.
- [2] Y. Crama, O. E. Flippo, J. Van De Klundert, and F. C. Spieksma, "The assembly of printed circuit boards: A case with multiple machines and multiple board types," *European journal of operational research*, vol. 98, pp. 457-472, 1997.
- [3] B. Iftikhar, M. Malik, S. Hadi, O. Wajid, M. Farooq, M. Rehman, *et al.*, "Cost-effective, Reliable, and Precise Surface Mount Device (SMD) on PCBs," in *IOP Conference Series: Materials Science and Engineering*, 2020, p. 012007.
- [4] Wendy Jane Preston. (2018, 4/01). *The difference between THT and SMT*. Available: <https://www.harwin.com/blog/the-difference-between-tht-and-smt/>
- [5] P. Poór, T. Broum, and J. Basl, "Role of collaborative robots in industry 4.0 with target on education in industrial engineering," in *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*, 2019, pp. 42-46.
- [6] S. Vojić, "Applications of collaborative industrial robots," *Machines. Technologies. Materials.*, vol. 14, pp. 96-99, 2020.
- [7] I. F. o. Robotics, "Demystifying Collaborative Industrial Robots," *Frankfurt, Germany*, 2018.
- [8] J. P. David A. Forsyth, *Computer Vision - A Modern Approach*, 2002.
- [9] P. R. G. Kurka and A. A. D. Salazar, "Applications of image processing in robotics and instrumentation," *Mechanical Systems and Signal Processing*, vol. 124, pp. 142-169, 2019.
- [10] R. Langmann and L. F. Rojas-Peña, "A PLC as an Industry 4.0 component," in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2016, pp. 10-15.
- [11] Omron. (2020). *Software Manual TMvision*. Available: <https://www.edata.omron.com.au/eData/manuals.html>
- [12] Omron. *FH/FHV/FZ5 Series Vision System - Processing Item Function Reference Manual*. Available: <https://www.edata.omron.com.au/eData/manuals.html>

**ANEXO B - HUMAN-ROBOT COLLABORATION (HRC) WITH VISION INSPECTION FOR PCB
ASSEMBLY**

Human-Robot Collaboration (HRC) with Vision Inspection for PCB Assembly

Mauro Queirós^{1,2}^a, João Lobato Pereira^{1,2}^b, Nuno M. C. da Costa²^c, S. Marcelino⁶,
José Meireles³^d, Jaime C. Fonseca²^e, António H. J. Moreira^{4,5}^f and João Borges^{2,5}^g

¹University of Minho, Guimarães, Portugal

²Algoritmi Center, University of Minho Guimarães, Portugal

³MEtRICs Research Center, University of Minho, Guimarães, Portugal

⁴2.Ai – School of Technology, IPCA, Barcelos, Portugal

⁵Polytechnic Institute of Cávado and Ave, Barcelos, Portugal

⁶DIB4T, Marinha Grande, Portugal

Keywords: Cobot, PCB Assembly, Work Cell.

Abstract: Flexibility and speed in the development of new industrial machines are essential factors for the success of capital goods industries. When assembling a printed circuit board (PCB), since all the components are surface-mounted devices (SMD), the whole process is automatic. However, in many PCBs, it is necessary to place components that are not SMDs, called pin through-hole components (PTH), having to be inserted manually, which leads to delays in the production line. This work proposes and validates a prototype work cell based on a collaborative robot and vision systems whose objective is to insert these components in a completely autonomous or semi-autonomous way. Different tests were made to validate this work cell, showing the correct implementation and the possibility of replacing the human worker on this PCB assembly task.

1 INTRODUCTION

In today's industry, any manual operation in repetitive production lines that causes delays in the production process needs to be semi-automated or fully automated so that the factory remains competitive in an increasingly aggressive and fast market. Printed circuit boards (PCBs) are a fine example of such necessity, as they are in virtually all electronic components of everyday life. Over the years, assembling electronic components on PCBs has undergone several changes, many of which aim to make this process increasingly automated, efficient, fast, and economically efficient (Altinkemer, Kazaz, Köksalan, & Moskowitz, 2000; Andrzejewski, Cooper, Griffiths, & Giannetti, 2018; Bogner, Pferschy, Unterberger, & Zeiner, 2018; Crama,

Flippo, Van De Klundert, & Spieksma, 1997). This is how Surface Mount Technology (SMT) arose, with surface mount devices (Surface Mount Devices) being mounted directly on the surface of the printed circuit board, generally in an automatic way using an SMT machine. Typically, these components are smaller in size, have better electrical performance, must withstand higher soldering temperatures, and must be selected, positioned, and soldered more carefully to achieve an acceptable manufacturing yield (Iftikhar et al., 2020).

Despite the advances seen with SMDs, there are other components (Figure 1) that still need to be manually assembled on PCBs, such as diodes, capacitors, and connectors named Pin Through Hole (PTH) components (Wendy Jane Preston, 2018). This limitation considerably decreases the production efficiency of PCBs assembled with this type of

^a  <https://orcid.org/0000-0001-5896-7423>

^b  <https://orcid.org/0000-0002-1053-7405>

^c  <https://orcid.org/0000-0002-8425-3501>

^d  <https://orcid.org/0000-0003-0881-2348>

^e  <https://orcid.org/0000-0001-6703-3278>

^f  <https://orcid.org/0000-0002-2148-9146>

^g  <https://orcid.org/0000-0002-5880-033X>

components when compared to those that only have SMD components.

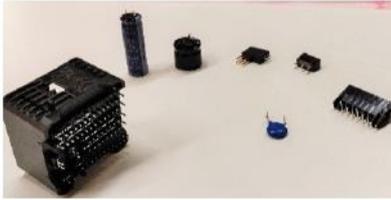


Figure 1: Example of non-SMD components.

Aiming to mitigate the limitation mentioned above and take advantage of the rise of collaborative robots (cobots) (Poór, Broum, & Basl, 2019; Vojić, 2020), this work proposes a novel industrial work cell for the automatic insertion of non-SMD components into PCBs. The proposed cell integrates three main systems from the literature: (1) a cobot to enable either collaborative or autonomous work (Robotics, 2018); (2) a vision system for component validation (David A. Forsyth, 2002; Kurka & Salazar, 2019); and (3) an external device for controlling and monitoring the work cell (Langmann & Rojas-Peña, 2016).

Throughout this paper, are going to be presented the fundamental requirements of this work cell, the layout and the main workflow of this task. It is also explained the key aspects of each subsystem methodology, and to validate this work cell, three tests were made and demonstrated. Finally the results and main ideas drawn throughout this work are discussed and concluded.

2 REQUIREMENTS OF THE WORK CELL

Taking into consideration the goal of reducing or eliminating the need for manually inserting different electronic components into PCBs but maintaining quality and efficiency during the insertion process, a set of requirements were defined, of which the most relevant are the following:

- The cell must have an automatic transport system for the PCB boards;
- It must have the ability to identify the different PCBs;
- Adapt the program and the gripper according to the PCB and components to be assembled;
- It must perform accurate and secure gripping and insertion for the different components;
- The system needs to validate each component prior to insertion, using, for example, vision

software to read the number of pins and make sure it is in good condition;

- Needs to assure that the PCB is correctly and fully assembled within 30 seconds;
- It also requires collecting information relevant to the operation of the work cell and making it available in a database for future analysis and study.

2.1 Cobot Subsystem

Going into more detail about the robot subsystem and taking into consideration aspects such as the components under study and the PCBs in question, the requirements of the robot are: speed and flexibility, as well as the ability to operate in conjunction with a human being; at least 700 mm range; at least 6 degrees of freedom (DOF); accuracy and repeatability of less than 0.1 mm; multiple I/O interfaces; and support for various communication protocols.

2.2 Vision System

To validate the quality of each component, one of the most frequently used approaches in the industry is the use of a vision system. In this case, for the proper functioning of the work cell, a minimum sensor resolution of 2M pixels is required to cover the entire insertion area of the component (500x500 mm²) and still allow accurate hole inspection and pin integrity verification. In addition, this system must permit the sharing of data with other subsystems.

The system must also be able to read the barcodes in the PCBs and inspect their correct position to reference the robot with respect to it doing the correct insertion of all components. For these tasks, the camera used may have a lower resolution.

2.3 Programmable Logic Control (PLC) System

Monitoring and recording the activity of an automated process is paramount, and thus this subsystem will be responsible for managing the entire work cell (command, control, and monitoring). For this to be possible, the external device must comply with the following requirements: ethernet, RS232, and EtherCAT communication facilities; digital inputs and outputs; database interaction facilities, have OPC-UA; and be modular. As main characteristics, the same PLC must have database interaction capabilities and the capacity to make process-relevant variables available through the OPC-UA protocol.

3 LAYOUT AND WORKFLOW

To meet the requirements mentioned above, one can propose the use of: (1) the cobot TM5-700, a 6 DOF collaborative robot capable of using different grippers and that integrates its own vision system; (2) the FH1050 vision system by Omron, which has various image acquisition and processing functionalities; and (3) the NX102-9020 Omron, a PLC acting as an external device capable of monitoring and controlling all tasks and functionalities of the work cell. These proposed systems were used and are connected, using the TCP/IP protocol for data exchange. Figure 2 illustrates how the subsystems interact throughout the task. In short, the PLC receives a signal from the assembly line to start the flow, and, after that, different signals are exchanged between the cobot, the PLC, and the vision system to proceed with the task. All the essential information to be seen by the human worker is visible in an interface and then sent to a database.

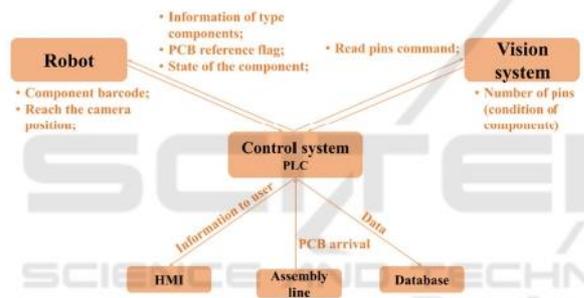


Figure 2: Data exchanged between the three subsystems.

The work cell further includes one box for the damaged components, the PCB allocation zone, and other zones for the boxes of the different types of components. Figure 3 shows a sketch of the work cell, illustrating all the components mentioned above distribution.

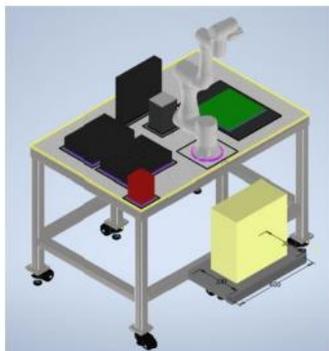


Figure 3: Sketch of the work cell.

With this distribution, the cobot can reach every position needed, independently of the velocity chosen for the task, without too much torque effort.

In a simplified way, the general task flow of this work cell is shown in Figure 4.

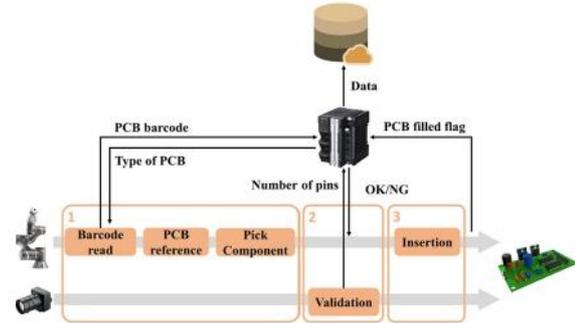


Figure 4: Flow of the work cell.

In short, the cobot will start by reading the barcode on the PCB to know what components must be inserted and do the referencing task to know where the PCB is. After that, it is time for the external vision system to read the number of pins each component has to validate its quality. Depending on the information received, the cobot will discard the component or place it in the proper PCB position upon the validation task. After placing the component, if the PCB is not yet fully assembled, the cobot will repeat the operation for another component. When finished, it will wait for the arrival of a new PCB.

4 METHODOLOGY

To execute the workflow explained above, one should consider each subsystem individually and how to integrate them to focus on precision and speed, aiming to respect the following key goal: replace the worker in the assembly of PCBs with this type of components, trying to speed up the process and reduce costs without ever losing quality.

To understand the entire process of this work cell, each subsystem will be explained in a more detailed way.

4.1 Cobot Subsystem

The cobot will have two major tasks: first, it is the subsystem responsible for the movement in the main task, including picking up a component, taking it to the vision subsystem, and, finally, placing it in the right location on the PCB. The other part is regarding

the internal vision system that this cobot has. Here, the cobot will perform two analyses: read the PCB's barcode to know which component to insert and reference itself with respect to the assembly board before inserting the first component (Figure 5). Both tasks will be solved using the TMFlow software, whereas, for the vision tasks, the TMvision software is employed.

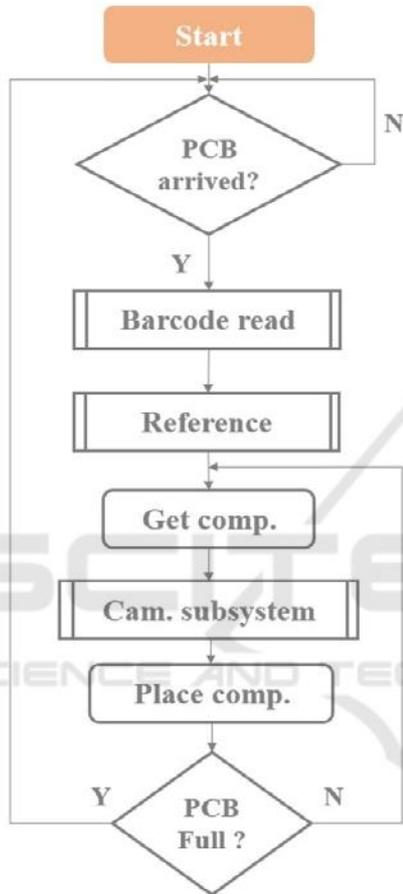


Figure 5: Workflow of the cobot subsystem.

4.1.1 Motion

Regarding this part of the task, the robot will have to move around the work cell to reach the desired positions, with the highest possible precision and speed, without compromising the safety of the environment and operator.

First, it will start in its home position, and as soon as the PCB sensor is activated, the robot moves to the zone where it will do the barcode reading and the referencing task. Upon gathering the information needed, the cobot picks the specific PCB component and moves it to the zone where the external camera will check the integrity of the pins. After validating

the integrity of the component, the robot proceeds to the insertion of the same.

4.1.2 Cobot Vision System

To identify the PCB that just arrived and place the components in the correct position, even if the PCB comes tilted or just crooked, the cobot's internal vision system has two types of functionalities (Figure 6) (Omron, 2020).

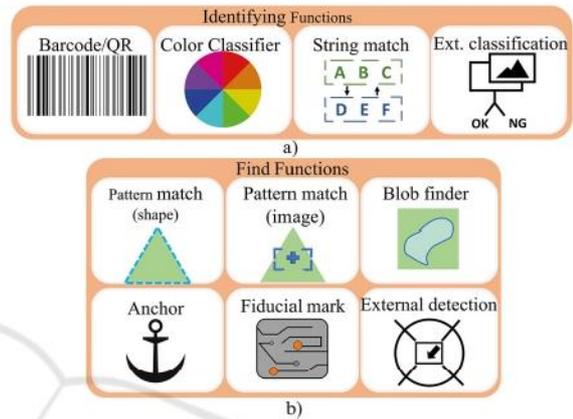


Figure 6: Functionalities available in the cobot's internal vision system: a) identify functions; b) find functions.

Every PCB comes with an associated barcode to know precisely which type (and amount) of component that PCB takes. To this end, the Barcode/QR code identify function is employed. Here, it is necessary to specify the region where the barcode can appear, and once the camera reads a barcode, it returns the specific number of that PCB. Knowing the group to which that particular number belongs, it is possible to tell which components must be inserted (Figure 7).



Figure 7: Example of a barcode in a PCB.

For PCB referencing, the Fiducial mark find function is employed. Fiducial marks are small targets in an assembly board placed on the top copper layer (and bottom if it is 2-layers) and allow the vision system to recognize where the PCB is (Figure 8).

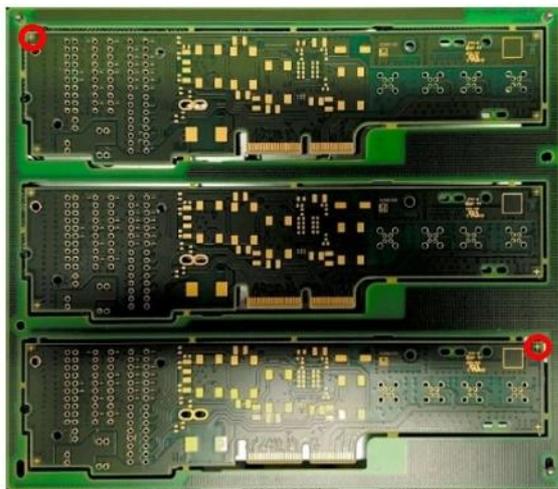


Figure 8: Examples of fiducial marks in a PCB.

The fiducial marks, the target study area, and the similarity threshold need to be selected to use this function properly. Once the cobot reaches the position to do the referencing, it will find the fiducial marks and create a new coordinate frame, where the middle point between the two fiducial marks is the origin. From this point on, regardless of the position of the PCB, the cobot will always be able to make the same movement to the insertion positions, ensuring the correct filling of the board.

4.2 FH Vision Subsystem

Here, the objective is to ensure that the component that the cobot will insert is in good condition to be possible to insert into the PCB without wasting it. Usually, this is solved by human inspection, but the goal is to replace it with a vision system to make the work cell fully automatic and faster.

The FH-1050 vision system is used, taking advantage of the FZ-PanDA software (Omron). Figure 9 summarizes the tasks performed by the vision system.

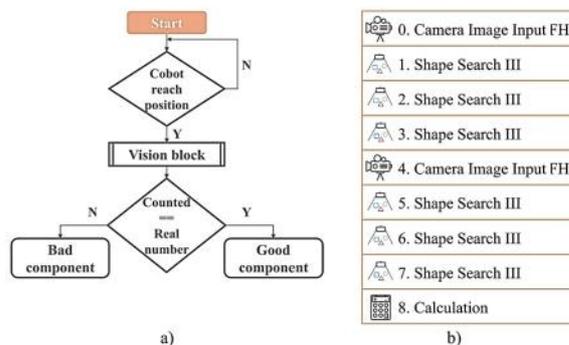


Figure 9: a) Workflow; b) vision block.

In brief, upon waiting for the cobot with the component to reach the point where the camera is pointing, the system will make one single acquisition to count the number of pins in a certain area to check if they are damaged. The idea is to divide the component into small sections, each with a specific number of pins or groups/lines of pins (Figure 10).



Figure 10: Component divided into sections and number of pins of each section.

This procedure is done using the "Shape Search III" block, where we must select the region of interest (ROI; big red box) and the subject we want to look for in that exact area (small red box, for example). Through an object detection algorithm, this function block registers a model of an image pattern based on its contour information, detects parts of inputted images that most closely match the model, and adds each similar object to the total count. An experimental study was performed to select the suitable threshold value for the degree of similarity between the template and the new image. Afterwards, the "Calculation" block is employed to find the total number of pins that were identified in each "Shape Search III" block. If the number equals those expected for the component in question (44 in the example in figure 10), the component is in good condition, and therefore the cobot will proceed with its insertion. If not, the cobot will discard this component, placing it in a specific box for damaged components.

Of note, the other two blocks present in figure 10, the "Camera Image Input FH", are used to calibrate the camera and enhance the image between analyses.

4.3 PLC Subsystem

This subsystem aims to establish a connection between the other two subsystems, being responsible for the control and monitoring of the work cell and registering the relevant data during the execution of the task, to be later stored in a database. The kind of

data to keep track of includes, for example, the number of damaged components, the number of PCBs filled, the time it takes to fill each board type, and so on. Figure 4 summarizes the information exchanged between subsystems, and as shown, the whole process of this task is divided into three main parts.

First, the cobot waits for the PLC to tell it that a new PCB has arrived. Then, as mentioned before, the cobot will use its internal vision system to do the referencing task and to read the barcode, sending the reading signal to the control system. Here, the PLC will then decide what type of PCB it is and then tells the cobot what type of components to insert.

Upon picking up the component, the cobot will take it to the external vision system area. Here, the PLC will send two commands: (1) first the command measurement, where the system will run the code to count the number of pins to assess the component's condition; and (2) the get data command, where the PLC will receive from the external vision system the number of pins that the software detected. Then, the PLC compares this value with the expected value, and if the result is positive, it informs the cobot that it can proceed with the insertion. If it is not, the PLC informs the cobot to discard this component and pick a new one.

Finally, the cobot will proceed with the insertion of the good component, repeating these steps until the PCB is filled. Meanwhile, the PLC will store relevant data during these steps, such as the number of PCBs filled, the number of damaged components, and the time to fill one PCB, among others.

Table 1 explains how the information will flow between the three subsystems.

Table 1: Information exchange.

PLC takes	PLC sends
PCB sensor - PCB arrived at the assembly area	Signal/Message for cobot to start
PCB barcode from the cobot reading.	Type of PCB for cobot to know which type of components it takes
External vision system reached	Measure command for vision system
-----	Get data command for vision system
-----	OK/NG component for cobot
PCB assembled	-----

5 TESTS

This section demonstrates some practical examples of this work cell, showing the correct implementation of the main systems and the compliance with the requirements. The tests were:

- Test A: How much time does it takes for the cobot to fill a PCB;
- Test B: Reliability of the external vision system;
- Test C: Monitoring and data acquisition.

These tests were employed in the following environment, where it is possible to see: (A) the external vision system for inspection; (B) the cobot with the internal vision system for barcode reading and referencing; (C) the box of components; (D) the PCB (Figure 11).

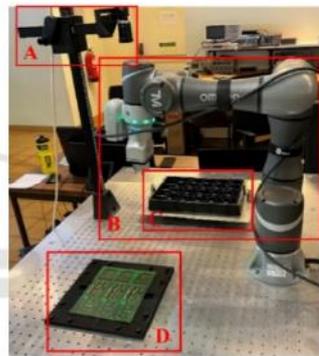


Figure 11: Environment of the work cell.

5.1 How Much Time Does It Take for the Cobot to Fill a PCB

In this test, the idea was to run the full task of this work cell and see how long it takes to fill a PCB, repeating this trial for different velocities (Figure 12).

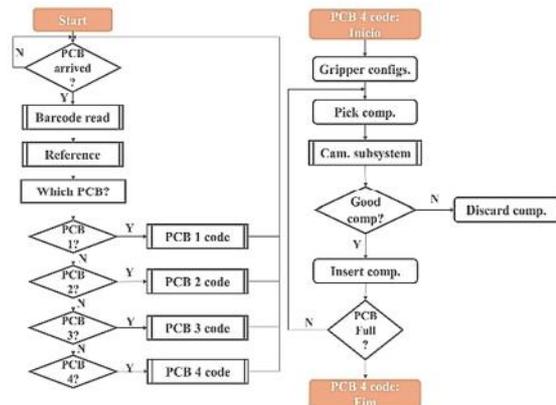


Figure 12: Workflow of the main task for PCB type 4.

The max velocity was set to 1.5 m/s, and the trials were done for 20%, 40%, 60%, 80%, and 100% of that speed. Of note, it was possible to see that the cobot never reaches the max speed because the movements that it makes during this task do not have the range needed to reach that velocity. Table 2 shows the time recorded for each speed setting.

Table 2: Time to fill a PCB with different speeds.

Velocity	Time (m/s)
20%	0:59,260
40%	0:38,672
60%	0:31,313
80%	0:29,241
100%	0:28,039

It is possible to conclude that the cobot needs less than 30 seconds to fill a PCB of type 4 (one that takes three components).

So far, one has shown that the robot can indeed meet the mentioned requirements, both in terms of speed and efficiency, but we have not observed its added value over the human operator. Thus, some additional tests were performed, where an operator tried to insert the components manually without validation of any kind. Despite initially taking around 20 seconds to fill a PCB, this time was reduced to an average of 15 seconds after a few attempts. The same type of test was executed using the cobot, observing a time of 16.280 seconds. Overall, it is possible to conclude that the cobot, besides being able to maintain this pace continuously, does not present significant delays over an operator, as it takes only 12 seconds more to fill the PCB while also inspecting it as well as the components.

5.2 Reliability of the External Vision System

This last test aims to confirm if the external vision system has the capacity and reliability to validate different types of damaged components. For that, different components were structurally compromised in different areas, and the task was run ten times to inspect how many true positives the vision system could detect. Additionally, the same test was run for one good component to validate if the system could validate its integrity either (Figure 13).

During the ten trials of each test, the external vision system showed its reliability when validating the components, verifying that it is in bad condition in all of the examples mentioned above, except for the case of a "good" component. Moreover, it also has the capability of showing in which area the damage is,

presenting a result of 10 out of 10 successful validations. From the moment the cobot reaches the inspection zone, and the moment the main system (PLC) receives the answer, the camera only needs around 0,8-1s to process all this.

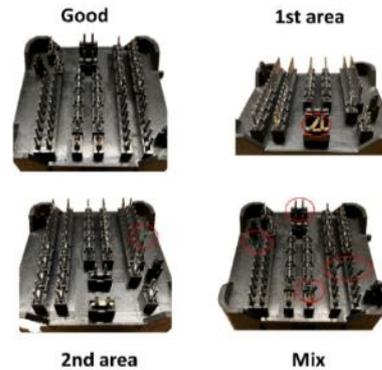


Figure 13: Examples of the components used.

5.3 Monitoring and Data Acquisition

The goal here is to test whether the PLC can retrieve the necessary information during the execution of the task mentioned above to be later able to visualize and document this data. The number of components and the number of PCBs filled with each type are examples of data that needed to be retrieved to be easier for the co-worker to know, for example, when to change the empty box of components, to know the percentage of damaged components or the number of PCBs of type 3 filled in one day.

To check the retrieved information, an interface was built using the CX-Designer software (Figure 14).

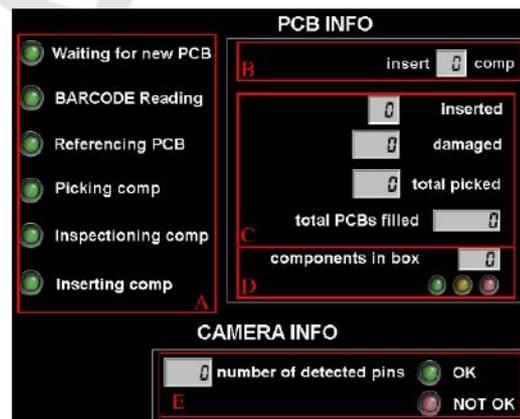


Figure 14: CX-Designer interface.

Figure 14 illustrates the five main parts of the interface:

- Section A notifies about the steps being performed during the routine;
- Section B informs the type of PCB detected during the barcode read function and the number of components it takes;
- Section C presents the number of components inserted, the number of damaged components, and the total number of picked components;
- Section D informs the number of components left in the box, showing a yellow LED when it is almost empty, and a red LED when it is empty;
- Section E lets know the number of pins read, showing the result of the external vision system.

Figure 15 shows a possible sequencing of what can be observed in the interface over several iterations.

Two images were recorded during the filling of the first PCB (Figure 15-1 and Figure 15-2). The first one shows the pick and place task and the validation task of the first component, where it is possible to see the number of pins read by the external vision system validating the component and resulting in its insertion. The second shows the expected last component of that PCB, where it was a damaged component, showing 42 read pins instead of 44, and a total of one damaged component out of a total of three. The last figure (Figure 15-3) illustrates an example of a possible iteration of the interface during a day of work.

6 DISCUSSION

During Test A, we confirmed the accomplishment of the 30 second requirement. We performed some insertion trials to compare this with a human worker, concluding that the worker can be faster by 1.280 seconds on average. However, some major requirements need to be considered, like efficiency, repeatability, and precision. Humans are susceptible to tiredness, and after hours of work, their precision and efficiency will not be the same, leading to delays in production or, in worst cases, damaged PCBs. In

opposition, the cobot always performs at his highest level, sustaining these requirements throughout hours of work. These requirements are also possible with industrial robots, but they have some disadvantages, such as: they are usually more expensive, they are used for heavier payloads, and they also require an isolated work area.

Moreover, different camera positions were tested during this test to find the easiest and fastest way to achieve a more efficient work. When the camera is on top, the environment lightning variations do not affect the external vision system precision and fast decision making. There were still two other possibilities: one being the cobot's camera performing the inspection, however this camera did not present the minimum requirements for this process; and the other was an external camera attached to the cobot, which met the quality and resolution requirements, but increased the cycle time of the entire process since the robot would need to place the component in an intermediate position to take the reading.

Test B confirms the high level of inspection by the external vision system when validating minor displacements and defects of the components' pins. It thus delivers a sensibility and accuracy that a human eye cannot achieve.

One of the few tasks in this work cell that a human worker needs is replacing an empty component box. In Test C, one concluded that this type of information cannot be presented as a mere label but also needs to be displayed using three LEDs, representing the quantity of components left in the box, giving an explicit luminous warning to the human worker. In addition to this task, to reduce the cycle time of this process, the operator cooperates with the cobot by informing it of the limited barcode search area since each type of PCB has its barcode in different areas.

Test C confirms the high level of inspection by the external vision system when validating minor displacements and defects of the components' pins. It thus delivers a sensibility and accuracy that a human eye cannot achieve.

As mentioned before, the human worker could replace some functionalities of this work cell, like the

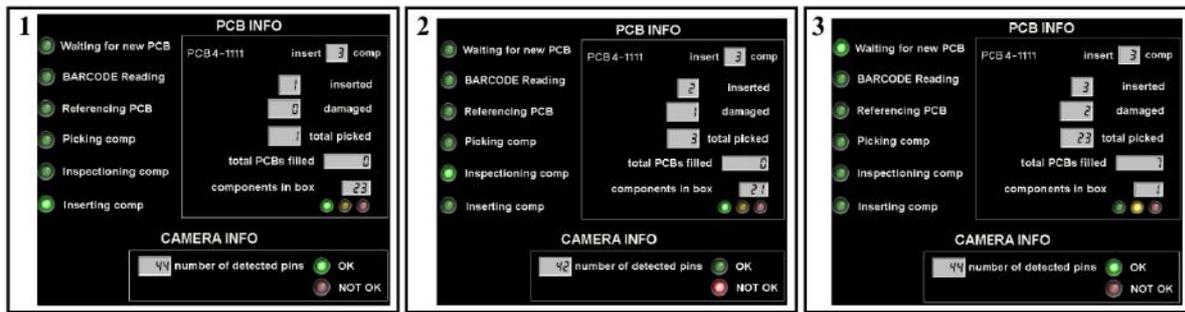


Figure 15: Interface during sequenced iterations.

barcode read and the referencing functions. Here, the worker could be responsible for informing the cobot of what type of components should be inserted. However, the idea is to minimize the human interaction, and through the barcode read function is possible to pass this information in an autonomous way without wasting much time. Also, when using the referencing function, we guarantee the correct placement of each component, as even when using a mechanical interlock, there is no guarantee of the correct PCB orientation, which over time can cause displacements in the order of millimeters.

In addition to the solution proposed throughout the paper, other two approaches were considered. The first one considered the collaboration between human and robot in a way that the robot would perform the pick task and take it to the vision system, and after the component validation, it would be inserted by the human, having the transference of it between the cobot and the operator. On the other hand, in the second hypothesis, the operator would perform the validation task of the component, and after it was validated, it would be transferred between the operator and the cobot, and the cobot would perform the insertion part. Both solutions have disadvantages since they increase the cycle time of the process, with the transition of the component between the operator and the cobot, as well as relying on less accurate and precise systems.

7 CONCLUSION

In this work, was presented a solution to a prototype autonomous work cell capable of assembling PCBs with PTH components. This work cell consisted of three main systems: (1) the cobot; (2) the vision system; and (3) the PLC, exchanging information between them via TCP/IP protocol.

The first one is mainly responsible for the movement in the main task, including picking up the

components, taking them to the inspection area, and inserting them into the PCB. The vision subsystem ensures the precise positioning and identification of the PCB and verifies the components' condition. The last subsystem manages both the cobot and external vision system's interactions and is responsible for monitoring the work cell by registering the relevant data generated during the execution of the task and storing it in a database.

Hereto, the three subsystems were demonstrated and explained, always showing the key aspects of the creation of this work cell. The performance of the proposed work cell was demonstrated through three examples, focusing on the most important aspects, namely exchange and acquisition of information, the validation of components, and the gain in precision and speed during PCB assembly over a human operator, all of them showing positive and successful results.

In sum, this work proves the optimization of a human process. It proves the possible replacement of the main human tasks with systems that are capable of maintaining a high level of performance throughout long periods of time.

ACKNOWLEDGEMENTS

This work is co-funded by the European Regional Development Fund (ERDF) through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) of the Portugal 2020 Program [Project No. 45070, "FlexASComp"; Funding Reference: POCI-01-0247-FEDER-045070].

REFERENCES

Altinkemer, K., Kazaz, B., Köksalan, M., & Moskowitz, H. (2000). Optimization of printed circuit board

- manufacturing: Integrated modeling and algorithms. *European journal of operational research*, 124(2), 409-421.
- Andrzejewski, K., Cooper, M., Griffiths, C., & Giannetti, C. (2018). Optimisation process for robotic assembly of electronic components. *The International Journal of Advanced Manufacturing Technology*, 99(9), 2523-2535.
- Bogner, K., Pferschy, U., Unterberger, R., & Zeiner, H. (2018). Optimised scheduling in human-robot collaboration—a use case in the assembly of printed circuit boards. *International Journal of Production Research*, 56(16), 5522-5540.
- Crama, Y., Flippo, O. E., Van De Klundert, J., & Spieksma, F. C. (1997). The assembly of printed circuit boards: A case with multiple machines and multiple board types. *European journal of operational research*, 98(3), 457-472.
- David A. Forsyth, J. P. (2002). *Computer Vision - A Modern Approach*.
- Iftikhar, B., Malik, M., Hadi, S., Wajid, O., Farooq, M., Rehman, M., & Hassan, A. (2020). *Cost-effective, Reliable, and Precise Surface Mount Device (SMD) on PCBs*. Paper presented at the IOP Conference Series: Materials Science and Engineering.
- Kurka, P. R. G., & Salazar, A. A. D. (2019). Applications of image processing in robotics and instrumentation. *Mechanical Systems and Signal Processing*, 124, 142-169.
- Langmann, R., & Rojas-Peña, L. F. (2016). *A PLC as an Industry 4.0 component*. Paper presented at the 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV).
- Omron. FH/FHV/EZ5 Series Vision System - Processing Item Function Reference Manual. Retrieved from <https://www.edata.omron.com.au/eData/manuals.html>
- Omron. (2020). Software Manual TMvision. Retrieved from <https://www.edata.omron.com.au/eData/manuals.html>
- Poór, P., Broum, T., & Basl, J. (2019). *Role of collaborative robots in industry 4.0 with target on education in industrial engineering*. Paper presented at the 2019 4th International Conference on Control, Robotics and Cybernetics (CRC).
- Robotics, I. F. o. (2018). Demystifying Collaborative Industrial Robots. *Frankfurt, Germany*.
- Vojić, S. (2020). Applications of collaborative industrial robots. *Machines. Technologies. Materials.*, 14(3), 96-99.
- Wendy Jane Preston. (2018). The difference between THT and SMT. Retrieved from <https://www.harwin.com/blog/the-difference-between-tht-and-smt/>