

Faculdade de Engenharia da Universidade do Porto



**Aplicação Robotizada de Fibras com Programação
Rápida do Manipulador e Controlo de Qualidade**

Ana Sofia Barbosa

VERSÃO FINAL

Dissertação
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Automação

Orientadores: Prof. Dr. António Paulo Moreira
Prof. Dr. António Torres Marques

29 de Julho de 2016

© Ana Barbosa, 2016

Resumo

Dada a notória evolução, ao longo dos anos, dos processos de manufatura, o mercado tem vindo a poder afirmar-se como cada vez mais competitivo, sendo que, também e sobretudo, se torna premente a sua necessidade de atualização. Esta evolução apela à rentabilização de recursos, otimização dos processos, robotização de diversas etapas, reduzindo custos e melhorando ou garantindo o nível de qualidade dos produtos fabricados.

Entre os materiais com melhores propriedades, destacam-se os materiais compósitos poliméricos que têm maior resistência específica, resistência à corrosão e à fadiga, expansão térmica controlada, permitem moldagem de peças em formatos complexos e orientação das fibras em direções desejadas [4]. Assim, são crescentes os processos de manufatura que envolvem compósitos de fibras para produção de componentes construídos recorrendo quer ao enrolamento filamental, quer à projeção de pré-impregnados para diversas aplicações. A automatização destes processos não só os torna mais flexíveis como lhes confere maior repetibilidade e menor custo. No entanto, o aumento da complexidade das peças a produzir, bem como a exigência na qualidade do produto torna ambos os processos de enrolamento ou projeção de fibras mais complexo e requer maior flexibilidade do sistema automático.

Este projeto foi desenvolvido com o intuito de complementar o processo de enrolamento filamental CNC convencional com um sistema robotizado programado com ensinamento por demonstração e/ou programação baseada em CAD, a fim de fabricar estruturas multiorifícios, introduzir reforços localizados, produzir laminados com formas complexas, garantir a otimização estrutural e de processo, bem como permitir a colocação de fibras curvas com trajetórias complexas. Uma segunda abordagem da aplicação robotizada de fibras será a aplicação por projeção também com ensinamento do manipulador, por demonstração. Paralelamente, um sistema robotizado, dotado de visão artificial para o controlo de qualidade, garante a fiabilidade e reprodutibilidade das trajetórias predefinidas, bem como espessuras mínimas.

Abstract

The remarkable evolution of manufacturing processes over the years, makes the market increasingly competitive, and the need to update becomes critical. This development aims to maximize resources, optimize processes, robotization of various stages, reducing costs and improving or ensuring the level of quality of manufactured products.

The polymeric composites are included among the materials with improved properties such as high specific strength, resistance to corrosion and fatigue, controlled thermal expansion, allow modelling parts with complex shapes and fiber orientation in desired directions [4]. Thus, there are increasing manufacturing processes involving fiber composites for the production of components constructed using the filament winding and the projection prepregs for various applications. The automation of these processes not only makes more flexible but also gives them greater repeatability and lower cost. However, the increasing complexity of parts to produce, as well as the requirement on quality of the product makes both winding processes and projection requires more complex fibers and flexibility of the automatic system.

This project was developed with the intention of supplementing the conventional CNC filament winding process with a robotic system programmed by demonstration teaching and/or CAD programming-based in order to manufacture multi holes structures, introduce localized reinforcements, produce laminates with complex shapes, ensure the structural and process optimization as well as allow placement of curved fibers with complex trajectories. A second approach to robotic application of fibers will be applying for projection also with robot demonstration teaching. In addition, a robotic system with machine vision for quality control ensures the reliability and reproducibility of the predefined trajectory and minimum thicknesses.

Agradecimentos

Agradeço a todos aqueles que direta ou indiretamente contribuíram para que todo o meu percurso até aqui fosse possível e sobretudo aos que contribuíram para este último desafio superado.

Aos meus pais que me proporcionaram as melhores condições e me apoiaram sempre em decisões importantes.

Aos meus orientadores Prof. Dr. António Paulo Gomes Mendes Moreira e Prof. Dr. António Torres Marques pelo conhecimento partilhado e orientação ao longo deste projeto.

Aos meus amigos e familiares que sempre viveram e festejaram comigo as vitórias e objetivos alcançados.

Ao Vitor que foi incansável e que durante estes anos sempre me acompanhou e suportou emocionalmente.

Ao INESC que foi o meu segundo lar e a todos os que dele fazem parte por tudo o que me ensinaram.

Ao meu avô que esteja onde estiver teria gostado de ver o resultado deste percurso.

Acima de tudo, estou agradecida por hoje me sentir mais capaz e por ver realizado um dos meus maiores sonhos.

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice	viii
Lista de figuras	x
Lista de tabelas	xv
Abreviaturas e Símbolos	xvi
Capítulo 1	1
1.1 Motivação.....	1
1.2 Contexto	1
1.3 Objetivos	2
1.4 Revisão Bibliográfica.....	3
1.4.1 Propriedades mecânicas dos materiais	3
1.4.2 Materiais Compósitos.....	3
1.4.3 Enrolamento Filmentar	7
1.4.4 Programação do robô	15
1.4.5 PCL - <i>Point Cloud Library</i>	32
1.4.6 Princípios comuns de medição de distâncias.....	38
Capítulo 2	40
2.1 Caracterização do Problema	40
2.1.1 Definição do problema.....	40
2.1.2 Solução proposta.....	41
2.2 Projeção de compósito.....	42
2.2.1 Introdução	42
2.2.2 Arquitetura Funcional.....	42
2.2.3 Componentes de <i>Hardware</i>	43
2.2.4 Software.....	48
2.2.5 Componente mecânica.....	73
2.3 Enrolamento filamentar	76

2.3.1 Introdução.....	76
2.3.2 Arquitetura Funcional.....	77
2.3.3 Componentes de <i>Hardware</i>	77
2.3.4 <i>Software</i>	84
2.3.5 Componente mecânica.....	86
Capítulo 3	88
3.1 Testes ao material	88
Capítulo 4	94
4.1 Conclusões e Trabalho Futuro	94
Capítulo 5	97
5.1 Anexos.....	97
Referências	101

Lista de figuras

Figura 1.1 - Classificação dos Materiais Compósitos [8].	4
Figura 1.2 - Enrolamento circular [11].	8
Figura 1.3 - a) Enrolamento Polar e Helicoidal. b) Helicoidal - Enrolamento geodésico. c) Polar - Enrolamento Plano [11].	8
Figura 1.4 - Esquema de enrolamento de termoplásticos [11].	8
Figura 1.5 - a) Sistema convencional com 1-bobine de material, 2- tensor e 3- cabeça de deposição. b) Sistema com cabeça modular de deposição [14].	10
Figura 1.6 - Topologia de rede neuronal[2].	11
Figura 1.7 - Tipos de falhas obtidos com a aplicação de diferentes tensões tendo sido o modelo à esquerda alvo de maior tensão e o modelo à direita alvo de menor tensão [3].	11
Figura 1.8 - Representação dos parâmetros estudados por Polini e Sorrentino [19] influentes na trajetória de enrolamento.	12
Figura 1.9 - Sistema de controlo da tensão projetado por Akkus <i>et al.</i> [1].	14
Figura 1.10 - Simulação do enrolamento na peça em forma de cotovelo [5].	14
Figura 1.11 - Exemplos de digitalizadores [21].	16
Figura 1.12 - Ambiente de desenvolvimento do <i>Robot Studio</i> [21].	16
Figura 1.13 - Modelo 3D do robô da Universal Robots [21].	17
Figura 1.14 - Interface de programação para controlo por força [21].	18
Figura 1.15 - Robô de dois braços e sistema de seguimento ótico [23].	19
Figura 1.16 - Interface para ensinamento indireto do robô baseado em visão [24].	20
Figura 1.17 - Representação da posição do Kinect relativamente ao braço direito do utilizador [24].	20

Figura 1.18 - Interface para ensinamento de tarefas do sistema desenvolvido por Pedersen e Kruger[25].	21
Figura 1.19 - Ensinamento do robô pelo humano [26].	22
Figura 1.20 - Célula robótica desenvolvida por [27].	23
Figura 1.21: Modelo CAD de montagem da célula do robô [28]. O programa do robô é gerado a partir deste modelo.	24
Figura 1.22 - Conceito de programação <i>offline</i> [29].	25
Figura 1.23 - Ilustração da soldadura da interseção de tubos por um manipulador industrial [30].	26
Figura 1.24 - Fotografia de operador a conduzir o manipulador para uma coordenada específica [31].	27
Figura 1.25 - Marcador inserido na pistola de pintura [32].	28
Figura 1.26 - Ferramenta marcador utilizada no método SRP (Simplified Robot Programming) da ABB [33].	29
Figura 1.27 - Ilustração da configuração do Sistema SRP da ABB [34].	30
Figura 1.28 - Transferência de dados da simulação 3D para o robô real [35].	31
Figura 1.29 - Estatísticas apresentadas na interface do sistema desenvolvido por [35].	31
Figura 1.30 - Logotipo <i>Point Cloud Library</i> [36].	32
Figura 1.31 - Representação de nuvem de pontos antes a) e depois b) da aplicação de um filtro recorrendo à grelha voxel .	33
Figura 1.32 - Divisão de nuvem de pontos pela técnica octree.	34
Figura 1.33 - 3D k-d tree .	34
Figura 1.34 - Ponto de início da partição k-d tree.	35
Figura 1.35 - Partição do espaço em k-d tree e árvore resultante ⁴ .	36
Figura 1.36 - Pesquisa no ramo esquerdo da k-d tree ⁴ .	36
Figura 1.37 - 2D k-d tree.	37
Figura 1.38 - Sistema tipo da tecnologia TOF - <i>Time of Flight</i> [37].	38
Figura 1.39 - Sistema tipo da tecnologia de triangulação.	39
Figura 2.40 - Projeção de compósito.	42
Figura 2.41 - Arquitetura funcional relativa ao sistema de projeção de compósito.	43
Figura 2.42 - Representação do laser RF625 e respetivo referencial [38].	44
Figura 2.43 - Representação da ferramenta de demonstração.	45
Figura 2.44 - Interface gráfica da aplicação 6Dmimic.	46

Figura 2.45 - Interface da aplicação CodeGen no separador “Config”	47
Figura 2.46 - Interface da aplicação CodeGen no separador “Input Data”	47
Figura 2.47 - Apresentação inicial da interface gráfica.	49
Figura 2.48 - Diagrama geral de funcionamento da projeção robotizada de compósitos.	49
Figura 2.49 - Diagrama respetivo ao processo de calibração da ferramenta de deposição no manipulador.	50
Figura 2.50 - Calibração da ferramenta no robô.	51
Figura 2.51 - Demonstração de trajetória para projeção de compósito.	51
Figura 2.52 - Representação dos referenciais do Sistema robô, ferramenta e marcador luminoso.	53
Figura 2.53 - Sistema de coordenadas associado ao processo de aquisição de medidas do sensor.	54
Figura 2.54 - Diagrama respetivo à transformação dos pontos para gerar uma nuvem do molde.	56
Figura 2.55 - Visualização da nuvem de pontos gerada após transformação dos pontos adquiridos pelo sensor.	56
Figura 2.56 - Filtro de remoção de <i>outliers</i> aplicado com raio de pesquisa 7mm e testando 5 vizinhos.	57
Figura 2.57 - Filtro de remoção de <i>outliers</i> aplicado com raio de pesquisa 7mm e testando 50 vizinhos.	58
Figura 2.58 - Filtro de remoção de <i>outliers</i> aplicado com raio de pesquisa 4mm e testando 7 vizinhos.	58
Figura 2.59 - Filtro de remoção de <i>outliers</i> aplicado com raio de pesquisa 4mm e testando 10 vizinhos.	59
Figura 2.60 - Diagrama que representa a aplicação de filtros à nuvem de pontos.	60
Figura 2.61 - Visualização da nuvem de pontos gerada b) após a aplicação de filtros à nuvem original a).	60
Figura 2.62 - Imagem do processo de reprodução da trajetória demonstrada.	61
Figura 2.63 - Molde utilizado para teste do algoritmo antes e depois da deposição de material.	62
Figura 2.64 - Diagrama que descreve o processo de divisão da nuvem de pontos por áreas. ...	63
Figura 2.65 - Visualização da nuvem de pontos gerada após seleção de áreas por parte do utilizador.	63
Figura 2.66 - Diagrama que descreve a aquisição de nova digitalização após deposição de material e posterior processamento dos dados obtidos.	64
Figura 2.67 - Visualização da nuvem de pontos gerada após nova digitalização.	65

Figura 2.68 - Visualização dos valores de espessura calculados e introduzidos pelo utilizador.	65
Figura 2.69 - Representação da normal (a azul) a uma superfície de pontos contidos numa certa área (rodeada por circunferência a preto).	66
Figura 2.70 - Comparação das correspondências encontradas entre as duas nuvens usando uma área de 5 pontos para calcular as normais a) e usando uma área de 60 pontos b).....	67
Figura 2.71 - Representação de situação de deposição sendo o molde a linha preta e a camada de compósito a linha azul e as linhas amarela e vermelha, duas distâncias.	67
Figura 2.72 - Comparação dos resultados obtidos utilizando um mínimo de 60 pontos a) e 5 pontos b) para teste do intervalo de valores das distâncias calculadas.	68
Figura 2.73 - Pormenor de região em que as correspondências são feitas entre pontos que se afastam da direção normal.	69
Figura 2.74 - Comparação da seleção de pontos com espessuras insuficientes considerando 15 vizinhos a) e considerando 1 vizinho b).	70
Figura 2.75 - Visualização da nuvem com as áreas de espessura insuficiente destacadas a amarelo.	71
Figura 2.76 - Visualização dos pontos a azul da trajetória do robô para reforço em zonas com espessura insuficiente.	71
Figura 2.77 - Diagrama que representa a deteção de regiões de pontos com espessura insuficiente.	72
Figura 2.78 - Diagrama que representa seleção de pontos da trajetória do robô a serem novamente reproduzidos para reforço em regiões de espessura insuficiente.	73
Figura 2.79 - Vista frontal do modelo da peça de adaptação da pistola em <i>Solidworks</i>	74
Figura 2.80 - Vista lateral do modelo da peça de adaptação da pistola em <i>Solidworks</i>	74
Figura 2.81 - Vista de trás do modelo da peça de adaptação da pistola em <i>Solidworks</i>	75
Figura 2.82 - Vista lateral da peça de adaptação montada na pistola.	75
Figura 2.83 - Enrolamento Filamentar [10].	76
Figura 2.84 - Arquitetura funcional relativa ao sistema de enrolamento filamentar.	77
Figura 2.85 - Imagem da máquina de enrolamento instalada no INEGI.	80
Figura 2.86 - <i>Encoder</i> E6A2-CW5 da Omron.	81
Figura 2.87 - Ondas de saída do <i>encoder</i> E6A2 - CW5C.	82
Figura 2.88 - Sensor S51-pa nk da Datalogic.	82
Figura 2.89 - Esquema elétrico da ligação do <i>encoder</i> e do sensor de proximidade ao ATmega328p.	83
Figura 2.90 - Estrutura para enrolamento filamentar.	84

Figura 2.91 - Aplicação que permite visualizar os pontos adquiridos pelo sensor numa zona sem orifícios.	85
Figura 2.92 - Aplicação que permite visualizar os pontos adquiridos pelo sensor numa zona com orifícios.	85
Figura 2.93 - Modelo <i>Solidworks</i> da peça de adaptação do sensor laser VC Nano 3D ao manipulador.....	86
Figura 2.94 - Imagem real da peça de suporte para o sensor.....	87
Figura 3.95 - Simulação da aplicação de fibra de carbono em fita no molde.....	88
Figura 3.96 - Nuvem de pontos resultante da aplicação de fibra de carbono em fita no molde.....	89
Figura 3.97 - Imagem adquirida num ângulo de 90° com radiação laser a incidir perpendicularmente na fita de fibra de carbono.....	90
Figura 3.98 - Imagem adquirida na direção de aquisição da camera do sensor com radiação laser a incidir perpendicularmente na fita de fibra de carbono.....	90
Figura 3.99 - Imagem adquirida na direção de emissão da radiação laser que está a incidir na fita de fibra de carbono com um ângulo menor do que 90°.	91
Figura 3.100 - Imagem adquirida na direção de aquisição da camera do sensor com a radiação laser a incidir na fita de fibra de carbono com um ângulo menor do que 90°. .	91
Figura 3.101 - Imagem adquirida na direção capaz de captar a radiação laser refletida que está a incidir na fita de fibra de carbono com um ângulo menor do que 90°.	92
Figura 3.102 - Comparação de nuvem de pontos resultante da aplicação de fibra de vidro em fita no molde a) com imagem real b).	92
Figura 3.103 - Imagem adquirida num ângulo de 90° com radiação laser a incidir perpendicularmente na fita de fibra de vidro.	93
Figura 3.104 - Imagem adquirida na direção de aquisição da camera do sensor com radiação laser a incidir perpendicularmente na fita de fibra de vidro.	93

Lista de tabelas

Tabela 5.1– Lista de comandos para comunicação com o sensor laser VC Nano 3D.	97
Tabela 5.2 - Código para robô ABB, em RAPID, para digitalização de peça.	99

Abreviaturas e Símbolos

Lista de abreviaturas

CAD	<i>Computer Aided Design (texto não português em itálico)</i>
CNC	<i>Computerized Numerical Control</i>
DEEC	Departamento de Engenharia Electrotécnica e de Computadores
FEUP	Faculdade de Engenharia da Universidade do Porto
INEGI Industrial	Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial
IMU	<i>Inertial Measurement Unit</i>
PCL	<i>Point Cloud Library</i>
INESC	Instituto de Engenharia de Sistemas e Computadores
DLL	<i>Dynamic-link library</i>
PLC	<i>Programmable Logic Controller</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
DSP	<i>Digital Signal Processor</i>
CMOS	<i>Complementary metal-oxide-semiconductor</i>

Lista de símbolos

°C	Grau Celsius
°	Ângulo em grau

Capítulo 1

1.1 Motivação

Atualmente, no laboratório de materiais compósitos do INEGI existe um equipamento de enrolamento filamentar CNC de 6 eixos com controlo independente e que inclui também 3 módulos independentes de tensionamento unitário de filamentos de fibras. Este equipamento limita o formato das peças aqui produzidas a peças simétricas e sem concavidades. Por este facto, e havendo necessidade de serem criadas peças complexas, estruturas multiorifícios, de se introduzirem reforços localizados e possibilidade da aplicação da fibra por projeção, a solução passa por se complementar este equipamento com um sistema robotizado. Paralelamente a um manipulador robótico para projeção de fibra por meio de uma pistola, deve ser introduzido um sistema dotado de visão artificial para um controlo de qualidade que garanta a fiabilidade e resultado pretendido com as trajetórias predefinidas, bem como espessuras mínimas. No caso do enrolamento das fibras em fita este sistema de visão deve garantir a posição correta da fita.

1.2 Contexto

Dada a notória evolução, ao longo dos anos, dos processos de manufatura, o mercado tem vindo afirmar-se como cada vez mais competitivo, sendo que, também e sobretudo, se torna premente a sua necessidade de atualização. Esta evolução apela à rentabilização de recursos, otimização dos processos, robotização de diversas etapas, reduzindo custos e melhorando ou garantindo o nível de qualidade dos produtos fabricados.

Entre os materiais com melhores propriedades, destacam-se os materiais compósitos poliméricos que têm maior resistência e rigidez específica, resistência à corrosão e à fadiga, expansão térmica controlada, permitem moldagem de peças em formatos complexos e

orientação das fibras em direções desejadas [4]. Assim, são crescentes os processos de manufatura que envolvem compósitos de fibras para produção de componentes construídos recorrendo quer ao enrolamento filamental, quer à projeção de pré-impregnados para diversas aplicações. O enrolamento filamental é um processo para fabrico de produtos compósitos de formas variadas que consiste no enrolamento de reforços impregnados em resina sobre um molde em rotação [6]. A automatização destes processos não só os torna mais flexíveis como lhes confere maior repetibilidade e menor custo. No entanto, o aumento da complexidade das peças a produzir, bem como a exigência na qualidade do produto torna ambos os processos de enrolamento ou projeção de fibras mais complexo e requer maior flexibilidade do sistema automático.

1.3 Objetivos

O objetivo inerente a este projeto é o de complementar o processo de enrolamento filamental CNC convencional com um sistema robotizado, a fim de fabricar estruturas multiorifícios, introduzir reforços localizados, produzir laminados com formas complexas, garantir a otimização estrutural e de processo, bem como permitir a colocação de fibras curvas com trajetórias complexas. Uma segunda abordagem da aplicação robotizada de fibras será a aplicação por projeção com recurso a um robô com capacidade de produzir peças com concavidades/singularidades e efetuar reforços localizados. O robô integrado no sistema deve ser programado recorrendo ao seu ensinamento por demonstração e/ou programação baseada em CAD. Paralelamente, um sistema robotizado, dotado de visão artificial para o controlo de qualidade, deve garantir a fiabilidade e reprodutibilidade da posição com as trajetórias predefinidas, bem como espessuras mínimas aquando da deposição automática.

1.4 Revisão Bibliográfica

Os materiais estão intrinsecamente relacionados com o desenvolvimento da humanidade e, conseqüentemente, com o avanço tecnológico. Este é um conceito que teve origem remota em que a variedade dos materiais era imposta pelos que existiam na própria natureza, e que, culminou na criação de materiais sintéticos com propriedades adaptadas às nossas necessidades.

As soluções nas mais diversas aplicações de engenharia crescem no sentido de maximizar o rácio qualidade/custo focando-se em melhorar a relação peso/propriedades mecânicas dos produtos no menor tempo possível através de processos otimizados.

1.4.1 Propriedades mecânicas dos materiais

Normalmente, a decisão de escolha de um material reporta-se sobretudo às suas propriedades mecânicas, seguindo-se as propriedades químicas, térmicas, elétricas, magnéticas e óticas consoante a aplicação e o ambiente envolvente.

De acordo com Silva, Alves e Marques [7], a **elasticidade** dos materiais diz respeito à capacidade de recuperarem a sua forma quando a força aplicada é retirada. Então, os materiais com elevado módulo de elasticidade são intrinsecamente rígidos, enquanto os materiais com um baixo módulo são flexíveis. A **plasticidade** está relacionada com a resistência à deformação permanente dependendo da sua tensão de cedência. Os materiais com elevada tensão de cedência têm elevada resistência à deformação plástica, sendo difíceis de deformar plasticamente e os que apresentam uma baixa tensão de cedência, são muito mais fáceis de deformar permanentemente. A deformação plástica que o material consegue aguentar antes de romper está relacionada com a **ductilidade e tenacidade**, sendo os materiais que se rompem sem sofrer uma quantidade significativa de carga no regime plástico, materiais frágeis e os que se deformam plasticamente, materiais tenazes.

1.4.2 Materiais Compósitos

Os materiais compósitos são formados a partir da combinação de dois ou mais materiais macroconstituintes distintos que, geralmente, diferem em composição química e/ou física, com o objetivo de obter propriedades específicas, diferentes daquelas que cada um apresenta separadamente, tendo uma interface reconhecível entre os componentes [4].

Esta combinação de matérias-primas com boas propriedades culmina em materiais com características superiores o que torna os materiais compósitos tão atrativos. Existe uma extensa gama de materiais compósitos correspondendo a várias combinações de metais, polímeros e cerâmicos [7] e materiais compósitos naturais, como é o caso da madeira.

Na composição destes materiais estão presentes um componente macio chamado de **matriz** com a finalidade de conferir a estrutura ao material, transmitir a carga para as fibras e preencher espaços vazios e um outro elemento normalmente mais rígido e resistente que confere boas propriedades mecânicas ao compósito chamado de **reforço**.

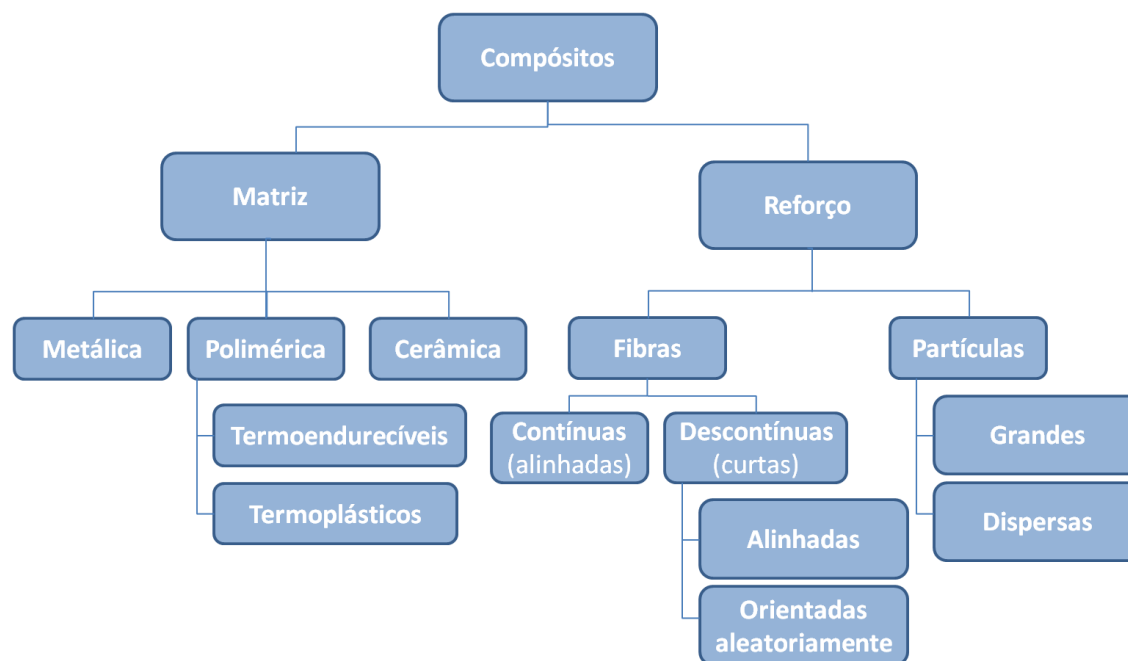


Figura 1.1 - Classificação dos Materiais Compósitos [8].

1.4.2.1 Matrizes

Tal como está representado na Figura 1.1, as matrizes dos materiais compósitos podem ser de natureza metálica, polimérica ou cerâmica.

Na origem das matrizes metálicas estão, por norma, materiais tratados termicamente para melhorar as suas propriedades mecânicas tais como alumínio, magnésio, titânio ou mais frequentemente ligas de aços.

Os materiais cerâmicos normalmente usados na constituição dos materiais compósitos são misturas de silício e alumínio e também de outros óxidos como é o caso do zircónio, cromo e tório que conferem ao compósito uma baixa condutividade térmica, ou seja, a propriedade de serem isolantes para aplicações de altas temperaturas [6].

As matrizes poliméricas dividem-se em termoplásticas e termoendurecíveis. A essencial diferença entre os materiais compósitos de matrizes termoendurecíveis e termoplásticas revela-se no comportamento quando sujeitos ao calor. Os materiais de matrizes termoplásticas podem ser reprocessados pois quando aquecidos há quebra nas ligações moleculares passando a haver mobilidade para reformação. Os materiais de matrizes termoendurecíveis não podem ser reprocessados pois “Uma vez aquecidos assumem uma forma permanente” [9]. No entanto, hoje em dia, há técnicas de “reciclagem” que levam à despolimerização e recuperação da matéria prima.

Matrizes Termoplásticas

Estas matrizes normalmente são constituídas por materiais como o polipropileno, poliamida e o policarbonato. No entanto para compósitos de elevada resistência são usados materiais como o Poli-éter-éter-cetona, Poliimidaz e Polissulfureto de fenileno. Estes materiais possuem elevada viscosidade, apresentam maior resistência ao impacto e resistem a maiores temperaturas, têm a possibilidade de serem facilmente processados por injeção ou por termoformação e a possibilidade de serem reciclados.

- Poli-éter-éter-cetona: é um polímero semicristalino e a sua cristalinidade confere-lhe resistência à maioria dos solventes e estabilidade hidrolítica, no entanto, necessita de elevadas temperaturas e pressões no processo de impregnação [4, 9].
- Poliimida: é um polímero amorfo e transparente e é caracterizado pela grande resistência a altas temperaturas continuamente, é resistente a álcool, ácidos e solventes de hidrocarbonetos mas é solúvel em solventes parcialmente halogenados [4].
- Polissulfureto de fenileno: é um polímero semicristalino com elevada resistência mecânica, boa resistência química e excelente desempenho a altas temperaturas [9].
- Polipropileno: é um polímero termoplástico semicristalino com boa resistência ao impacto, elevada rigidez, boa resistência química e a sua temperatura típica de processamento torna-o bastante acessível ao processamento por técnicas tradicionais mais utilizadas.
- Poliamida: é um polímero semicristalino, normalmente denominado de Nylon®, tem boa resistência a produtos químicos, resistência térmica, resistência ao desgaste e em compósitos com fibras de carbono, por exemplo, a sua transcristalinidade possibilita boa adesão entre a fibra e a matriz, o que resulta em boas propriedades mecânicas.
- Policarbonato: é um termoplástico amorfo incolor com boa resistência térmica, química e aos agentes ambientais, sendo nomeadamente, dos termoplásticos com maior resistência ao impacto [9].

Matrizes Termoendurecíveis

As resinas normalmente utilizadas são as de epóxido, poliésteres insaturados, resinas de vinilester e resinas fenólicas. Estes materiais são mais resistentes e mais rígidos que os anteriores, no entanto são mais frágeis. De entre os mencionados, o que apresenta melhores propriedades é a resina de epóxido mas tem elevado custo.

- Poliésteres insaturados: são das resinas mais comuns no mercado dos compósitos por apresentarem boas propriedades mecânicas, elétricas e químicas. Têm ainda um baixo custo aliado a um processamento bastante acessível que acontece em duas etapas, uma em que a resina líquida se transforma num material gelatinoso e outra que resulta num endurecimento final que recorre a um processo de cura com libertação de calor.
- Resinas de vinilester: resultam de uma combinação de uma resina vinilester insaturada com uma resina de epóxido desenvolvidas para que as suas características fossem excelentes ao nível da resistência química à corrosão.

- Resinas fenólicas: estas resinas no estado líquido são usadas para impregnação de diversos materiais e no estado sólido apresentam-se como uma mistura de cargas, plastificantes e outros aditivos. Possuem ainda uma característica bastante útil pois em caso de fogo libertam pouco fumo e poucos tóxicos.
- Resinas de Epóxido: é uma resina bastante versátil que permite um vasto leque de possibilidades de processamento que originam resinas com propriedades variadas o que faz com que sejam mais caras. No processo de cura é-lhe adicionado um agente endurecedor apropriado que lhe confere resistência mecânica, resistência à abrasão, resistência química e boa adesão à fibra.

1.4.2.2 Reforços

Normalmente os materiais que estão na base do reforço têm elevada rigidez e uma densidade relativamente baixa e, tal como está presente na fig.1.1, os compósitos podem ser reforçados com fibras ou partículas. Para reforçar compósitos de matriz polimérica, são mais usadas fibras de vidro ou carbono, enquanto que os reforços de partículas estão mais associados a compósitos com matrizes de metal ou cerâmicas [7].

Fibras de vidro

As fibras de vidro têm como matérias-primas areia silicosa, carbonato de cálcio e espatoflúor adicionadas à sílica [10]. São normalmente amorfas, embora com altas temperaturas possa ocorrer cristalização, o que diminui a sua resistência [7]. Estas fibras são excelentes como isolantes elétricos mesmo em espessuras mínimas, não propaga chamas nem há libertação de fumos em caso de fogo. São indicadas em aplicações que exigem poucas perdas térmicas pois apresentam pequena condutibilidade térmica.

Fibras de Carbono

As fibras de carbono são menos utilizadas do que as fibras de vidro pelo facto de serem mais caras, mas possuem muito melhor rigidez e estabilidade dimensional, advindo daí o facto de serem indicadas para aplicações que exigem bom desempenho mecânico como é o caso da aeronáutica, a indústria espacial e muitos reservatórios de combustível e gás. Estas fibras podem ser fabricadas a partir da decomposição térmica da celulose ou do poliacrilonitrilo ou a partir do alcatrão. Normalmente são produzidas a partir do poliacrilonitrilo pois a produção a partir da celulose torna-se bastante cara e a partir do alcatrão não resulta numa fibra com tão boas propriedades mecânicas. De acordo com Silva, Alves e Maqrues [7], o processo de fabrico pelo qual passam as fibras produzidas a partir do poliacrilonitrilo decompõe-se em 4 fases: a oxidação, onde a fibra é aquecida a 300°C numa atmosfera rica em oxigénio; a carbonização, onde a cadeia molecular é aquecida até 1100°C num forno de atmosfera neutra; a grafitização, onde é feita a pirólise das fibras para se obter um grau elevado de orientação da estrutura e conferir à fibra um elevado módulo de elasticidade; um tratamento superficial, onde são mudadas as condições de superfície, em atmosfera de ácido nítrico ou sulfúrico, para que a fibra adira facilmente ao material empregue como matriz. Por norma, a dificuldade de adesão desta fibra revela-se em relação às matrizes poliméricas que é ultrapassada com tratamentos químicos à superfície para que se crie uma rugosidade que facilite a aderência [4].

É, também, importante salientar que os reforços podem surgir na forma de *roving*, ou seja, como um cordão de filamentos contínuos enrolados helicoidalmente em bobines que pode destinar-se à produção de fibras curtas para moldação por projeção, à produção de tecidos ou então pode ser processado para a produção de mantas, malhas ou entrançados. No caso das mantas, as fibras na forma de filamentos contínuos ou cortados são dispostas aleatoriamente e é-lhes colocado um ligante que lhes confere estabilidade. Nas malhas dá-se uma ligação sucessiva de camadas de fibras alinhadas que vão formando uma construção que permite uma distribuição equitativa da carga pelas fibras. Assim, esta topologia permite obter um reforço com elevado módulo de tração e flexão. Nos entrançados, a disposição das fibras é como o próprio nome indica entrançada, conseguindo-se maior resistência específica. Estão disponíveis numa configuração tubular ou plana, ideais para produtos como raquetes de ténis ou *sticks* de hóquei [9].

1.4.2.3 Pré Impregnados

Os pré-impregnados resultam de uma mistura de fibras e resinas posteriormente consolidada. Esta mistura é colocada entre folhas de silicone que são prensadas ou laminadas para uma boa molhagem das fibras e garantir espessuras mínimas. Posteriormente passa por um processo de cura para se poder manusear o pré-impregnado [7]. No entanto quando a matriz é termoplástica é necessária uma temperatura e pressão mais elevadas para este processo.

1.4.3 Enrolamento Filmentar

O enrolamento filmentar é um processo para fabrico de produtos compósitos de formas variadas que consiste no enrolamento do material sobre um molde em rotação [6]. Existem 3 padrões de enrolamento (helicoidal, polar e circular) de acordo com o ângulo que a trajetória do enrolamento faz com o eixo de rotação do mandril. Os enrolamentos polar e helicoidal são idênticos na medida em que cruzam o mandril em comprimento com um dado ângulo resultando num revestimento feito por caminhos alternados em orientações positivas e negativas. No entanto, a diferença entre os dois, como se pode ver na Figura 1.3 b) e c), caracteriza-se por uma trajetória reta no caso do enrolamento polar e em forma de hélice no caso do enrolamento helicoidal. No enrolamento circular (Figura 1.2), o revestimento do mandril é feito pela deposição das fibras perpendicularmente ao seu eixo [11].

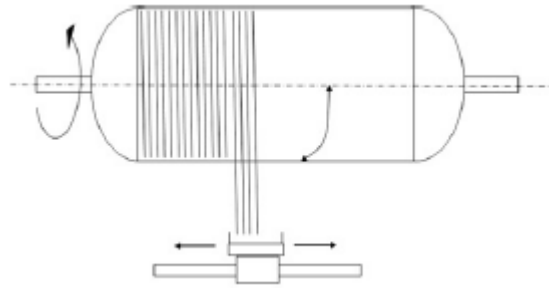


Figura 1.2 - Enrolamento circular [11].

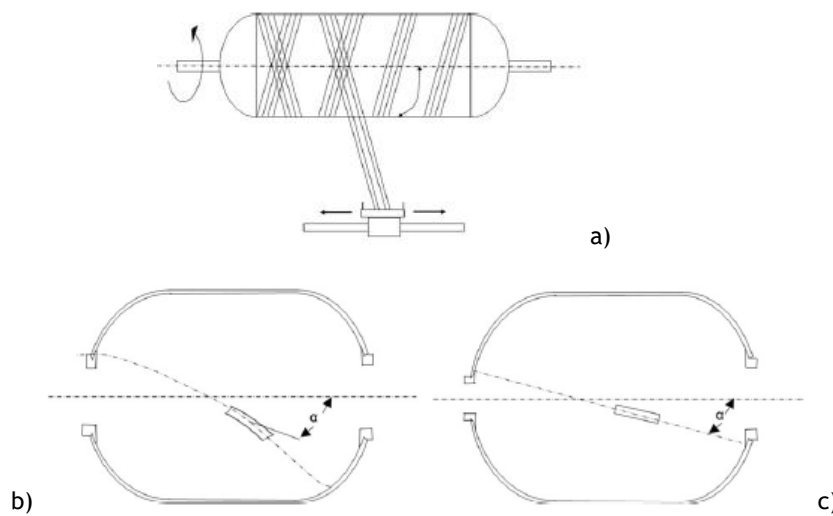


Figura 1.3 - a) Enrolamento Polar e Helicoidal. b) Helicoidal - Enrolamento geodésico. c) Polar - Enrolamento Plano [11].

No enrolamento de fibras com matrizes termoendurecíveis, há a passagem do material por uma banho de resina antes do enrolamento no mandril e, no final deste processo, a peça é submetida a um tratamento de cura [12]. Por outro lado, quando se trata do enrolamento de fibras com matrizes termoplásticas (Figura 1.4), o banho de resina é substituído por um sistema de aquecimento que torna o material moldável e, devido às propriedades específicas deste material, este processo pode ser executado com consolidação *in situ*, sem ser necessário o tratamento de cura [11].

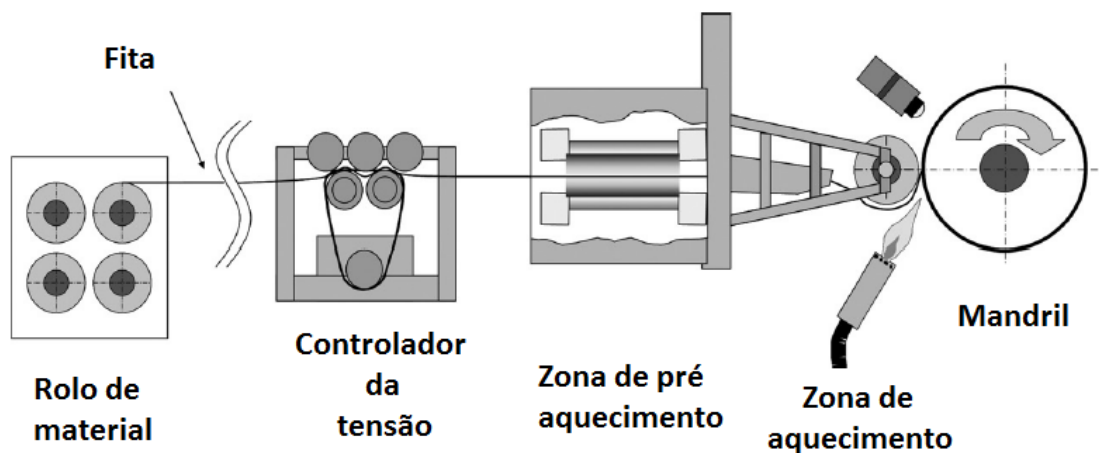


Figura 1.4 - Esquema de enrolamento de termoplásticos [11].

Tal como explicou Giacoletto [13], na revista *Reinforced plastics*, na década de 60 o enrolamento filamentar era visto como uma técnica de produção de produtos leves mas com grande força, o que merecia o respeito de ser considerado como uma forma de arte. O autor deste artigo compara mesmo o molde onde é enrolada a fibra como sendo a paleta, a fibra em resina como sendo a tinta, a máquina como sendo o pincel e o controlo da máquina como sendo o artista. Reforça ainda que todo o processo desde o planeamento do produto e das suas características, às etapas por que passa até se obter o produto final, acabam por ser formas de arte. Tudo é pensado e avaliado, o produto que a máquina desenvolverá, o ambiente em que esta trabalhará e os operadores que a controlarão. Com a evolução no tempo, os engenheiros foram redesenhando todo o processo do enrolamento filamentar, usaram todo o conhecimento que estava ao alcance para que inicialmente aparecessem as primeiras máquinas. Na década de 40, 50 estas eram bastante básicas, apenas tinham dois eixos de movimento, o de rotação e um eixo horizontal, no entanto era suficiente para as peças de filamentos que se pretendiam produzir na altura. Os avanços seguiram no sentido de permitirem ao operador controlar a máquina recorrendo a cintos, correias e engrenagens. Depois disso, a necessidade de um terceiro eixo foi modificando o aspeto das máquinas até que se começou a utilizar computadores para controlar estas máquinas. Este avanço melhorou bastante a precisão dos movimentos, a velocidade de controlo e levou a um aumento de eixos de movimento nas máquinas.

A automatização deste processo permitiu melhorar a robustez dos produtos e melhorar a sua qualidade dando maior repetibilidade do processo. Ao contrário das primeiras máquinas que apenas permitiam a produção de peças cilíndricas e sobretudo simétricas, a versatilidade das máquinas conferida pela sua automatização permite-lhes desenvolver peças com geometrias complexas e assimétricas. É a orientação de deposição das fibras que determina a resistência das peças e esta pode ser alterada consoante o ângulo de enrolamento, a velocidade de rotação do molde e consoante a velocidade de deslocamento da ferramenta de deposição das fibras [12].

É no sentido de se melhorar a qualidade das peças produzidas, aumentar a robustez e permitir maior liberdade na sua geometria que se desenvolvem diversos estudos sobre as trajetórias ótimas de enrolamento, bem como a inclusão de manipuladores robóticos para acrescentar graus de liberdade ao processo de produção.

O uso de robôs no enrolamento filamentar permite a produção de diferentes formas em pouco tempo, devido à sua reprogramação. A inserção de diversas ferramentas num manipulador robótico com o tipo do manipulador representado na Figura 1.5 a) e a facilidade de fazer chegar essas ferramentas ao ponto do molde que se pretende dá a liberdade de criar concavidades e outras formas assimétricas nas peças.

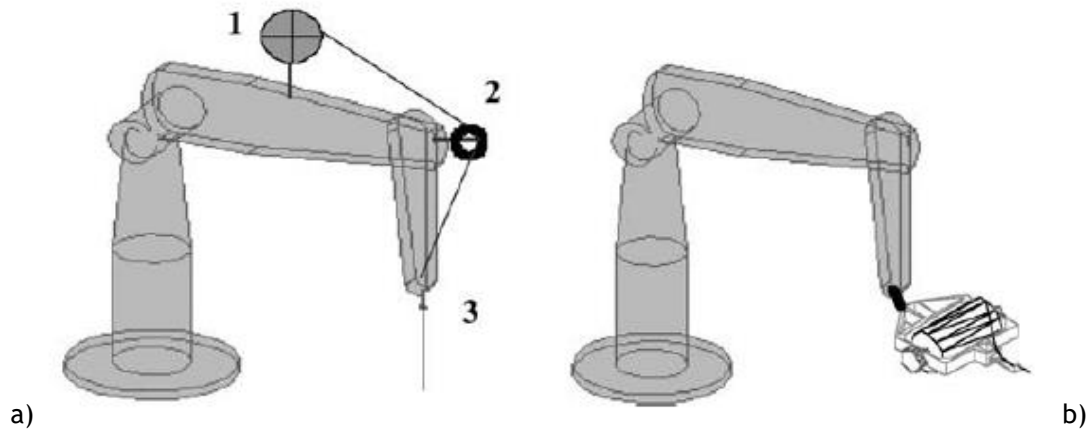


Figura 1.5 - a) Sistema convencional com 1-bobine de material, 2- tensor e 3- cabeça de deposição. b) Sistema com cabeça modular de deposição [14].

No entanto, este método tem limitações ao nível da dimensão da própria ferramenta que pode não permitir alcançar determinados pontos de concavidades acentuadas, o que pode desencadear movimentos de grande amplitude para pequenas mudanças de orientação. A utilização de uma ferramenta de deposição com diversos graus de liberdade daria também ao robô maior liberdade. De outra forma, um sistema tal como o representado na Figura 1.5 b) em que os graus de liberdade estão divididos entre o molde e o manipulador em que ambos funcionam sincronizadamente permite atenuar essa lacuna [14].

Em 2001, Anglani et al. [15] apontaram outra dificuldade no enrolamento filamenar robotizado que remete para a posição ótima da peça na área de trabalho do robô tendo em conta as suas limitações cinemáticas. Até aqui, a simulação utilizada estava restrita a ferramentas gráficas para representar o movimento do robô e para prever as propriedades mecânicas da peça final, o que levou ao interesse de desenvolverem uma simulação com uma aplicação mais específica que encontrasse a trajetória de enrolamento que otimizasse esse processo. Assim, estudaram uma solução que simula o comportamento do processo e monitoriza todos os parâmetros influentes para que se possa selecionar os valores dos ângulos das articulações que pertencem a um intervalo viável e para que se evite colisões durante o movimento do robô. Com o mesmo objetivo, moldaram a cinemática para que a variação dos ângulos das articulações em cada momento fosse a menor em todas as articulações por uma questão de minimizar o desgaste nas mesmas. Concluíram que há uma zona cinematicamente inalcançável, uma zona utilizável e uma zona de segurança não aconselhável tendo em conta as posições da peça que exigiam mais esforço das articulações e tendo em conta a importância de evitar colisões durante a trajetória para que as propriedades da peça final sejam ótimas.

Também com o objetivo de melhorar propriedades do enrolamento filamenar tais como a trajetória e a velocidade de enrolamento, em 2002 Heider et al. [2], demonstraram o uso de uma rede neuronal (Figura 1.6) *online* para calcular pontos de ajuste do processo para controladores PID usados na aplicação automatizada de termoplásticos. Os sistemas de controlo típicos para processos de manufatura de compósitos normalmente mantêm as entradas do processo numa janela desejada, no entanto os pontos de ajuste são baseados em experiências prévias que não são os mais adequados devido ao grau de não linearidade destes processos.

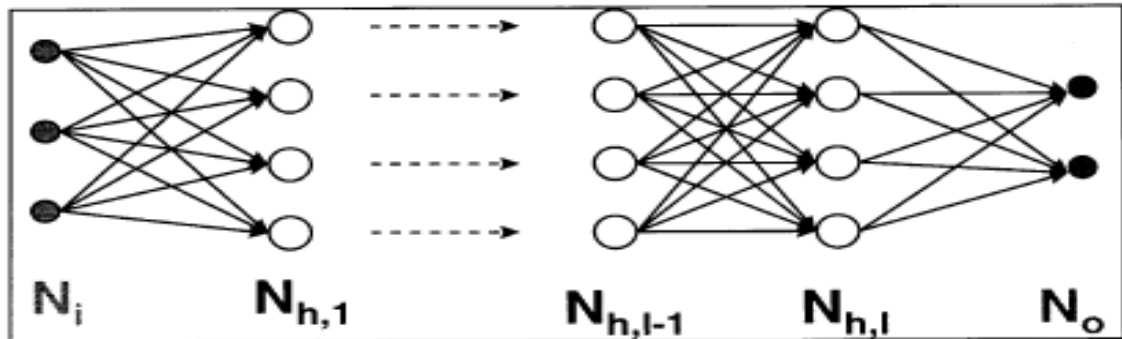


Figura 1.6 - Topologia de rede neuronal[2].

Então, o objetivo da inclusão das redes neuronais foi fazer com que o ajuste ao longo do processo fosse feito dinamicamente. Para isso inicialmente foram adquiridos dados para que se selecionasse apenas as entradas mais relevantes para o processo e assim poderem ter um espaço de entrada menor e uma rede menos complexa. A rede neuronal processa os pontos de ajuste do controlador de forma a que o rendimento seja máximo e seja cumprida a qualidade mínima desejada, tendo em conta que o histórico do processo influencia na qualidade das peças conseguintes mas que os ajustes são feitos dinamicamente ao longo do processo.

Uma das propriedades que mais influencia a qualidade da peça resultante do processo de enrolamento filamental é a tensão aplicada ao material. Em 2002 Mertiny e Ellyin [3] fizeram uma investigação experimental onde estudaram a influência da tensão aplicada durante o enrolamento filamental nas propriedades físicas e mecânicas de tubos de fibra de vidro reforçados com compósitos poliméricos especificamente. Estes tubos de transporte e armazenamento de fluidos são normalmente submetidos a uma variedade de condições de carga ao longo do seu tempo de vida, por isso foram sujeitos a diferentes cargas biaxiais para testes e a resposta ao stress destes componentes foi registada para que fossem tiradas conclusões. Os resultados experimentais (Figura 1.7) mostraram que a resistência dos componentes estava relacionada com o grau de tensão na fibra aquando do enrolamento filamental pois sob condições de tensão alta na fibra, o componente tubular respondia com uma resistência que resultava numa quebra do material.

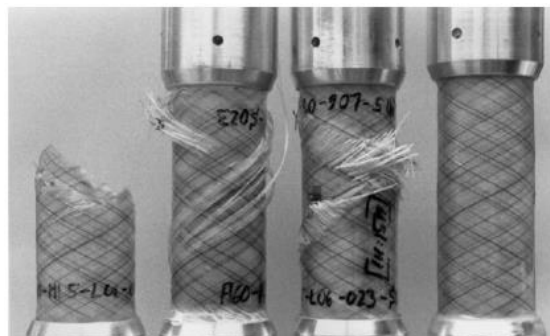


Figura 1.7 - Tipos de falhas obtidos com a aplicação de diferentes tensões tendo sido o modelo à esquerda alvo de maior tensão e o modelo à direita alvo de menor tensão [3].

Carrino et al. [16] mostraram também que o valor da tensão aplicada durante o enrolamento é dependente da velocidade a que o material é depositado. Para o concluírem, testaram um robô constituído por um suporte fixo, uma bobine de material de alimentação, um tensor e um sistema de deposição. A necessidade deste estudo deveu-se às trajetórias complexas necessárias para a produção de peças assimétricas que exigem tensão constante no material, ausência de derrapagens, torção e perdas de material durante o enrolamento [17]. As trajetórias que envolvem difíceis mudanças de direção da cabeça de deposição causam valores de tensão inconstantes se se mantiver o valor da velocidade de deposição. Isto origina variação no grau de alinhamento das fibras e conseqüente redução de qualidade das peças produzidas pois pouca tensão nas fibras origina enrugamentos ou dobras ao longo da direção de deposição. Por outro lado, demasiada tensão causa danos no material e fortes heterogeneidades ao longo da superfície da peça. Afirmaram ainda que a velocidade de enrolamento influencia a exatidão e repetibilidade das trajetórias pois o erro de localização da cabeça de deposição em todos os pontos da trajetória aumenta com a velocidade. Por outro lado, velocidades elevadas levam a puxões nas articulações devido ao aumento de *stress* nas fitas de fibra que se divide pelo rolo e por vibrações nas articulações do robô.

Em 2005, num estudo pormenorizado sobre este assunto, Polini e Sorrentino [18] concluíram quais os parâmetros fundamentais para o planeamento da trajetória de enrolamento e sobretudo quais os valores de velocidade que permitem que o valor de tensão aplicado no material se mantenha constante e adequado ao longo de todo o processo. Os parâmetros geométricos abordados e que estão representados na Figura 1.8 foram o número de pontos usados para aproximar a trajetória de enrolamento, o ângulo da trajetória e a distância de segurança entre a ferramenta de deposição e o material compósito enrolado para evitar colisões. Verificaram que aumentando o número de pontos usados para aproximar a trajetória de enrolamento ou aumentando o ângulo de trajetória acima dos 90° origina um movimento suave e contínuo do sistema de deposição anulando assim a ocorrência de perdas de tensão durante o enrolamento. Diminuir a distância de segurança implicou o desenrolamento de uma boa parte do material entre o sistema de deposição e o mandril e, conseqüentemente, uma diminuição no valor médio da tensão abaixo do valor nominal.

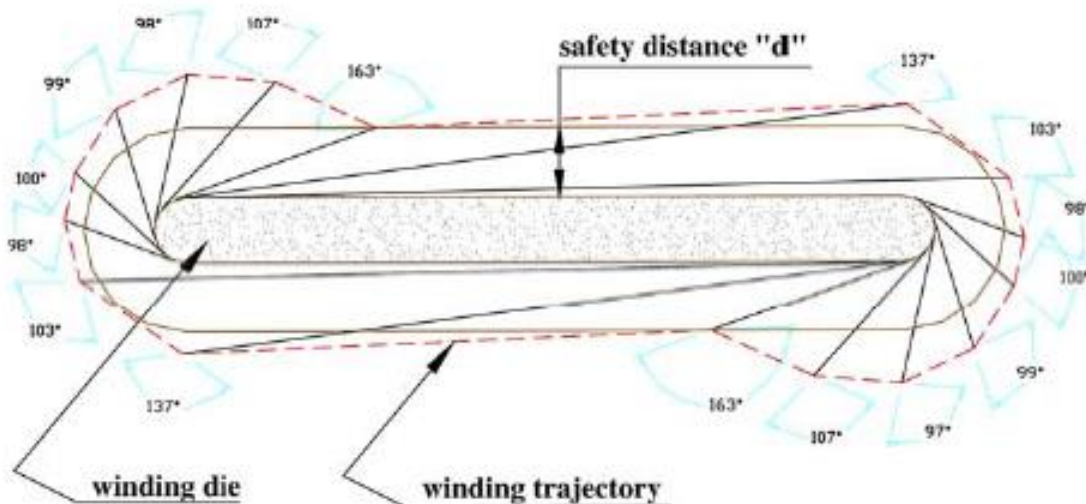


Figura 1.8 - Representação dos parâmetros estudados por Polini e Sorrentino [19] influentes na trajetória de enrolamento.

Concluíram ainda que o aumento do número de pontos de aproximação da trajetória reduzia o efeito da velocidade de deposição que afasta o valor médio da tensão do valor nominal. Por outro lado, o estudo [19] que trata da influência destes parâmetros no tempo de enrolamento revelaram que aumentar o ângulo da trajetória parece não influenciar significativamente o tempo de enrolamento mas o aumento do número de pontos para aproximação pode aumentar sensivelmente o tempo total de conclusão do processo. Um aumento da distância de segurança, quando combinado com os valores adequados tanto do ângulo da trajetória como do número de pontos de aproximação pode culminar numa diminuição no tempo de enrolamento, uma vez que origina uma trajetória mais sinuosa, que permite aumentar a velocidade do sistema de deposição entre os diferentes pontos da trajetória. No entanto, no que respeita ao valor ideal para a distância de segurança alegam que diminuindo este valor, diminui também a probabilidade de ocorrerem colisões durante o enrolamento e como nestas condições a quantidade de material desenrolado é menor, o valor da tensão ronda sempre valores mais próximos do valor nominal [20].

Com vista a enquadrar os valores de tensão sentidos nas fibras num intervalo adequado ao longo do processo de enrolamento dos filamentos, Akkus et al. [1] implementaram um sistema de controlo de tensão que simula o efeito da força da tensão nas máquinas de enrolamento filamentar (Figura 1.9), ou seja, um sistema que obtém valores teóricos numa zona de pré tensão para que se otimize o valor da tensão em tempo real que vai ser realmente sentido pelo filamento durante o enrolamento. Este sistema tem grande importância na medida em que a tensão aplicada nas fibras nesta fase de produção no enrolamento dos filamentos é decisiva na qualidade do produto final e no seu rácio força-peso. O sistema de controlo desenvolvido envolve um travão magnético, um servo motor para enrolamento das fibras, uma unidade de controlo PID usada como mecanismo de *feedback*, uma célula de carga para medir a tensão sentida na fibra e um conversor de dados. A tensão medida pela célula de carga é comparada com um valor nominal para a manter num intervalo de valores pré definidos.

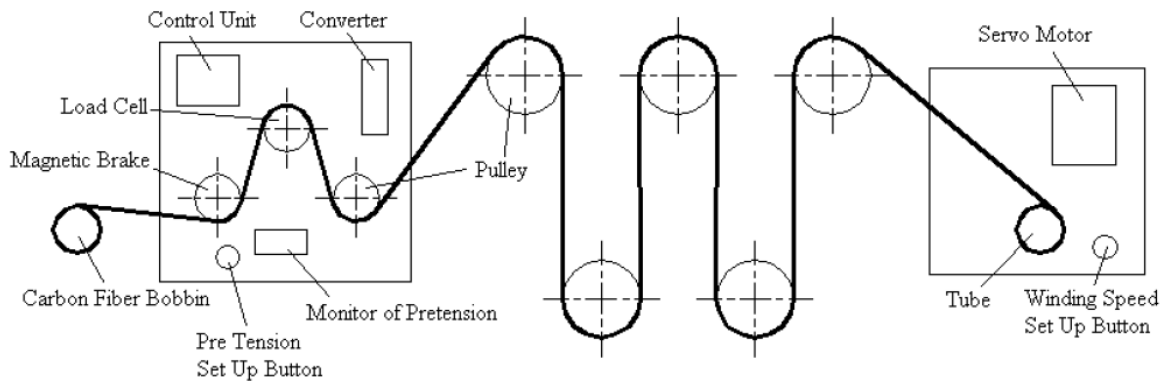


Figura 1.9 - Sistema de controlo da tensão projetado por Akkus *et al.* [1].

Nos dias de hoje, a utilização de robôs na produção de componentes utilizando materiais compósitos é trivial e cada vez mais em aplicações complexas e assimétricas. Exemplo disto foi a aplicação desenvolvida em 2013 por Zhong, W. *et al.* [5] para produção de tubos em forma de cotovelo. Projetaram uma trajetória estável para o enrolamento e simularam o processo no OpenGL (Figura 1.10) para se verificar a viabilidade do sistema. O *hardware* utilizado consistiu num computador industrial, um armário de controlo NX100 e um robô. O sistema de controlo do robô funciona com um computador *host* e um *slave* a trabalhar paralelamente. O computador *slave* envia dados para o computador central através da rede de dados para que a resposta do sistema acerca da posição e da postura do robô seja feita em tempo real sendo este um sistema de controlo modular e de alta velocidade. Os resultados obtidos revelaram que este método é bastante exato, confiável e que satisfaz os requisitos destas e de outras peças de forma complexa.

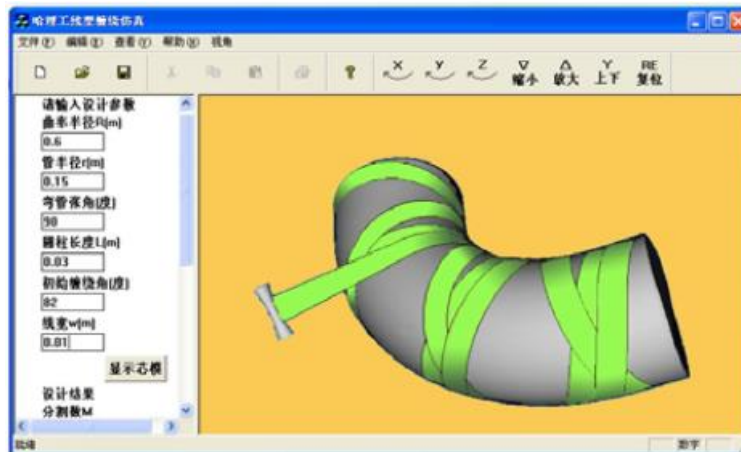


Figura 1.10 - Simulação do enrolamento na peça em forma de cotovelo [5].

1.4.4 Programação do robô

A programação de robôs industriais utilizados em sistemas de produção robustos de adequada qualidade é, muitas vezes, um desafio pois exige um elevado esforço e nível de especialização por parte do operário. Em muitos casos, a necessidade de um operário especializado, acresce o preço de funcionamento do sistema. Caso se verifique um erro durante a produção é necessário notificar o operário ou empresa externa que programou inicialmente o sistema. O que atualmente se exige também é o facto de um mesmo robô poder ser reprogramado para diversas tarefas ao longo do seu tempo de vida. Então, é no sentido de facilitar a interação do operário com o robô que se tem vindo a desenvolver formas fáceis de programação do robô para processos industriais.

Em 2013, Rossano *et al.* da ABB [21], apresentou numa conferência internacional um artigo onde aborda diversas dificuldades associadas à programação de robôs, especialmente em questões relacionadas com a criação e edição de trajetórias. Estas dificuldades advêm do facto da programação exigir diversos conhecimentos de lógica e matemática aliados a conceitos abstratos de variáveis e sistema de coordenadas no espaço, bem como o conhecimento da sintaxe de diversas linguagens de programação. No entanto, a dificuldade da programação de robôs integrados em sistemas industriais depende da aplicação e das tarefas a realizar, podendo ser atenuada pela consulta de manuais de instrução e pelo uso de blocos “*standard*” de algoritmos, tipos de dados e estruturas de controlo. A programação por demonstração é vista como uma aprendizagem supervisionada onde o operador executa a tarefa e o robô utiliza o sistema de sensores para adquirir informação da trajetória executada e, de seguida reproduz a ação. Neste sentido, a pesquisa tem-se acentuado na programação apoiada na visão 3D, utilizando diversos tipos de sensores desde os de baixo custo similares ao Kinect da *Microsoft* até sensores mais sofisticados, dependendo da precisão pretendida em cada aplicação.

A programação baseada num digitalizador (Figura 1.11) utiliza uma abordagem que regista a informação da posição 3D com um braço digitalizador fácil de mover [21]. O digitalizador atua como um substituto ao braço robótico. Sendo leve e tendo pouco atrito, este é mais fácil de manipular do que o braço do próprio robô utilizando uma consola de instruções. Este método é utilizado numa aplicação da ABB desde 2005 em que o utilizador pode definir rapidamente a trajetória 3D, arrastando a ponta do digitalizador ao longo dos contornos do caminho. Depois de calibrar o espaço do digitalizador com o espaço do robô, o robô consegue imitar os movimentos gravados.



Figura 1.11 - Exemplos de digitalizadores [21].

Na programação baseada em CAD, o utilizador insere dados CAD no sistema e este gera automaticamente um programa de controlo do robô. Muitos fabricantes de robôs industriais fornecem ferramentas de programação em ambiente simulado onde se pode incluir também informação proveniente de ficheiros CAD. No entanto a geração dos programas não é automática. Estas ferramentas permitem aos utilizadores recriarem o seu próprio ambiente de produção e criar programas de controlo do robô sem necessitarem do robô real. Uma vez que a maior parte da informação que é enviada ao robô virtual durante a programação *offline* são os dados de posição, é importante que o modelo seja uma representação precisa do ambiente real. Por exemplo, o *software Robot Studio* da ABB (Figura 1.12) inclui componentes que fornecem ferramentas específicas para gerar o programa do robô. Com estes componentes, o utilizador carrega o seu desenho CAD da peça numa ferramenta de simulação e consegue criar uma trajetória para o robô clicando em pontos da superfície 3D ou selecionando uma superfície inteira para que o *wizard* gere a trajetória [21]. No entanto não é obrigatório que o *software* tenha um simulador, a partir do ficheiro CAD pode gerar automaticamente o código para as trajetórias que cubram todo o desenho CAD.

Na ausência de um ficheiro CAD, pode ser utilizado um *scanner* 3D para capturar a geometria da peça [22]. A nuvem de pontos recolhidos pelo *scanner* é convertida para um modelo da superfície da peça que é submetido a um procedimento de filtragem/ suavização que remove o ruído. Estes dados são posteriormente utilizados por um *software* para gerar o programa do robô automaticamente, como descrito anteriormente.

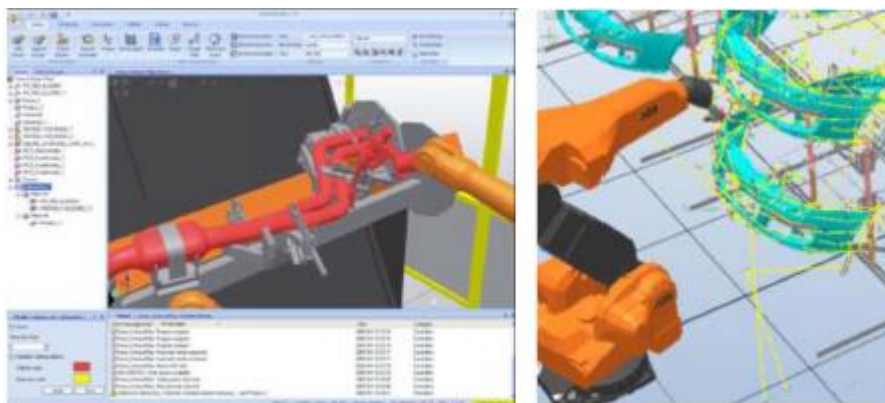


Figura 1.12 - Ambiente de desenvolvimento do *Robot Studio* [21].

A programação *Lead-Through* é baseada num sistema onde o utilizador pode conduzir o braço do robô a determinadas posições e/ou movimentos para o ensinar. A trajetória é armazenada no código do programa e é reproduzido durante a produção. A Universal Robots tem alguns braços robóticos que recorrem a esta abordagem, possuindo inclusivamente um dispositivo que também compreende uma representação 3D do robô (Figura 1.13), combinado assim aspetos da programação baseada em CAD com técnicas de programação *Lead-Through*. Este dispositivo de ensino permite ao operador mover o robô real ou mover o robô na simulação diretamente no *teach pendant* onde o modelo 3D do robô demonstra o movimento que o robô real irá executar. Isto oferece ao operador melhor perceção das consequências de um movimento antes que este seja executado.

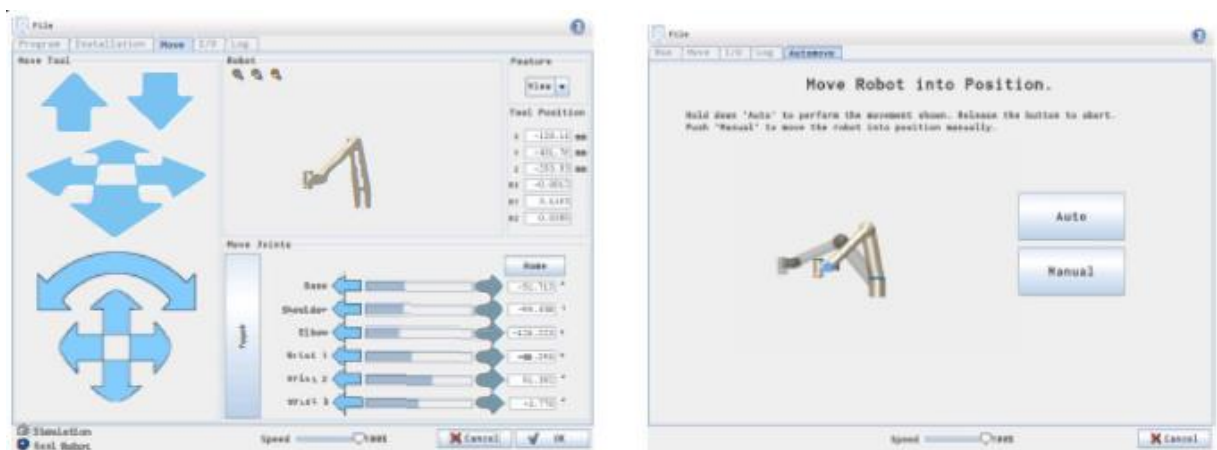


Figura 1.13 - Modelo 3D do robô da Universal Robots [21].

A programação por aprendizagem da trajetória é definida a partir da informação proveniente dos sensores externos [21]. Por exemplo, o sensor de força pode permitir ao robô tocar numa peça e determinar em que posições vai operar. Esta técnica normalmente é utilizada com complemento de outras em que o utilizador define diversos pontos-chave por onde deve ser descrita a trajetória do robô. Por exemplo, os produtos da ABB para controlo de máquinas por força, utilizam o sensor de força para permitir ao utilizador que este programe as trajetórias pela técnica de *Lead-Through* (Figura 1.14). Esta técnica permite que o utilizador tenha apenas que definir alguns pontos-chave da trajetória e o robô consegue determinar os restantes pontos e posições intermédias a partir do sensor de força.

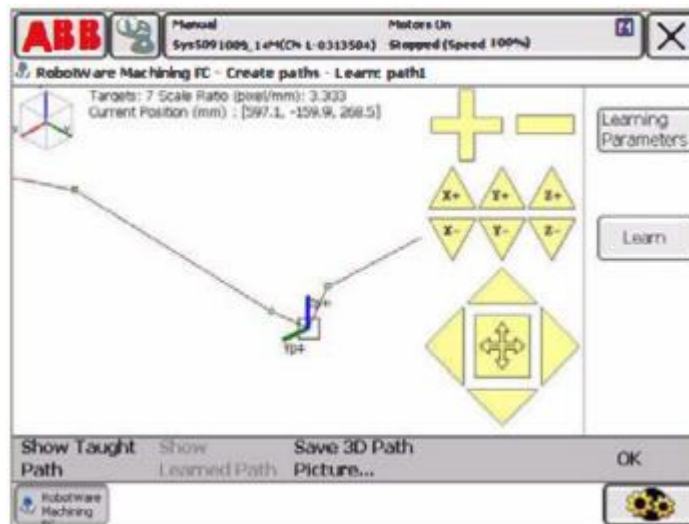


Figura 1.14 - Interface de programação para controle por força [21].

A ABB [23] desenvolveu também um método de programação por demonstração baseado num sistema de monitorização e testou-o num robô de 2 braços que se pode ver na Figura 1.15. Este método usa a monitorização de sensores de movimento para capturar os movimentos dos braços do utilizador e fazer com que o robô os imite. Paralelamente, o movimento do robô é registado, otimizado e é gerado automaticamente o programa pelo computador ou pelo controlador do robô. Isto proporciona uma redução no tempo de programação do robô e diminui a dificuldade em programa-lo. É possível ser implementado em manipuladores robóticos e que se programem vários robôs simultaneamente.

Este sistema da ABB inclui um ou mais robôs, sensores seguidores de movimento que capturam o movimento em 6D (translação e rotação) dos braços do utilizador e ações das suas mãos, um computador para processamento da informação fornecida pelo sensor, um dispositivo para controlar o robô e dispositivos de comunicação para conexão com o PC, com o controlador do robô e com os sensores, por exemplo um *router Ethernet*. O sistema seguidor dos movimentos utilizado é o OptiTrack da Natural Point.

Todo este processo passa por inicialmente se calibrarem os sensores, de seguida o operador demonstra uma tarefa, os seus movimentos são captados e transferidos ao robô. Este imita o movimento dos braços e as ações reproduzidas pelas mãos. Por fim, o computador armazena a trajetória do robô, otimiza-a e gera o programa automaticamente.

A versatilidade deste método proporciona uma cooperação entre vários robôs, uma programação fácil e eficiente tendo a resposta uma boa precisão e um tempo de atraso aceitável [23].



Figura 1.15 - Robô de dois braços e sistema de seguimento ótico [23].

Kim *et al.* [24] desenvolveram também uma interface para ensinamento indireto do manipulador baseado em visão com os sensores do Kinect e IMU (Figura 1.16). Através dos dados adquiridos pelos sensores são obtidos os dados do esqueleto do utilizador e calculados, assim os ângulos das articulações e o movimento do pulso para controlo do robô. Este sistema permite ao utilizador ensinar o robô sem um processo de pré-programação, o que melhora o tempo de ensinamento do robô e permite maior produtividade.

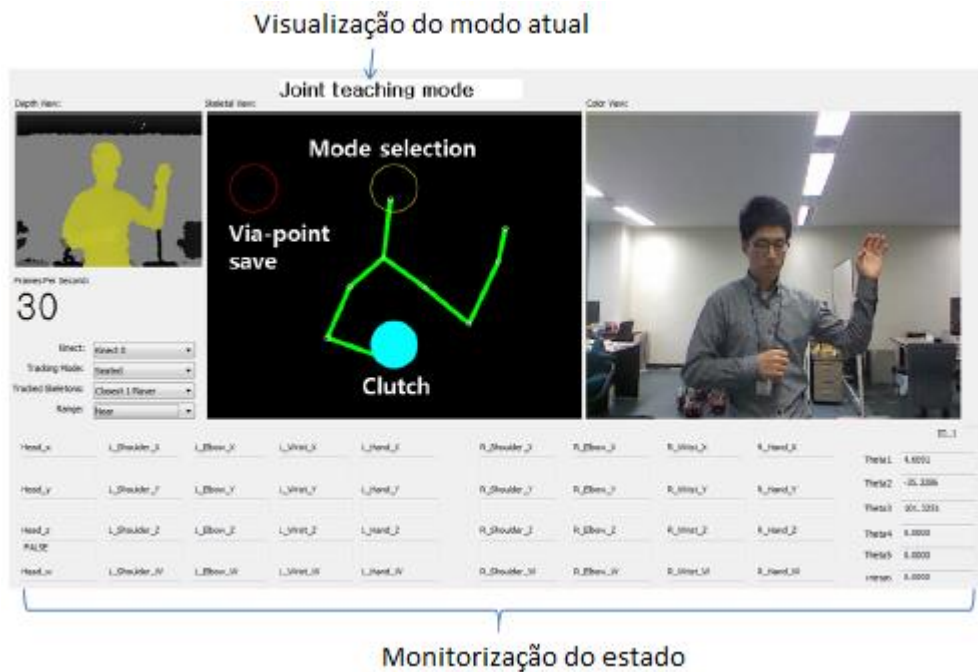


Figura 1.16 - Interface para ensinamento indireto do robô baseado em visão [24].

A partir do exoesqueleto que obtêm por meio dos sensores (Figura 1.17), é calculada a cinemática da posição do braço do utilizador a partir dos parâmetros de Denavit-Hartenberg para se obterem a posição e orientação exatas das articulações e serem reproduzidas no braço robótico.

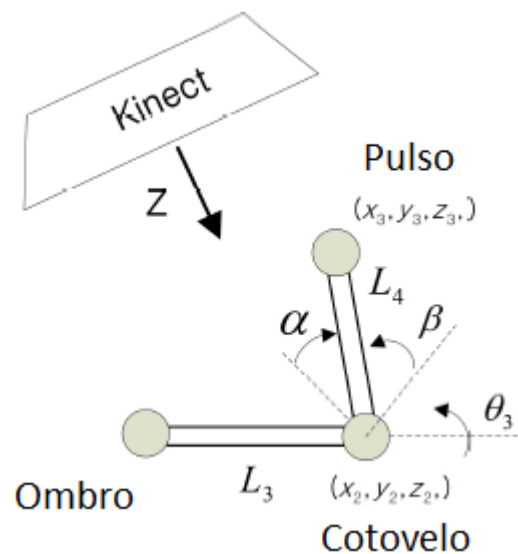


Figura 1.17 - Representação da posição do Kinect relativamente ao braço direito do utilizador [24].

De acordo com a Figura 1.17, θ_3 , L_4 e α são parâmetros de Denavit-Hartenberg, (x_2, y_2, z_2) são as coordenadas do cotovelo e (x_3, y_3, z_3) são as coordenadas do pulso.

A informação das coordenadas ao longo do tempo é enviada ao controlador do robô para que este reproduza toda a trajetória. O envio da informação é permitido devido à comunicação que se pode estabelecer com o controlador do robô por *sockets*. E, por fim são considerados erros de posição inerentes a todo o processo.

Pedersen e Kruger [25] apuraram a crescente necessidade de reprogramação rápida dos manipuladores móveis e flexíveis para desempenho de novas tarefas e propuseram um novo método de programação simples. Ao contrário do ensino por demonstração convencional que, normalmente visa reconhecer a ação e onde esta é aplicada, propuseram uma divisão deste trabalho. Primeiro o operador especifica manualmente a ação que o robô deve executar e, paralelamente, são especificados parâmetros da ação usando gestos para, por exemplo, indicar qual o objeto em que se vai aplicar a ação. Assim, mesmo mudando o ambiente, este método adapta-se pois não exige que os objetos estejam em locais pré-definidos. Permite ainda que o robô seja controlado por pessoas inexperientes ou sem formação, pois foram feitas experiências com diferentes pessoas em diversos ambientes e revelou-se efetivamente intuitivo. Este sistema tem incorporados dois sensores *Primesense* RGB-D, especialmente o *Kinect* da *Microsoft* para identificação dos objetos e o *Asus Xtion Pro Live* para identificação dos gestos do instrutor. Integra ainda duas interfaces gráficas, uma para feedback do controlo e reconhecimento dos gestos e outra usada para especificações e ensino de tarefas (Figura 1.18).

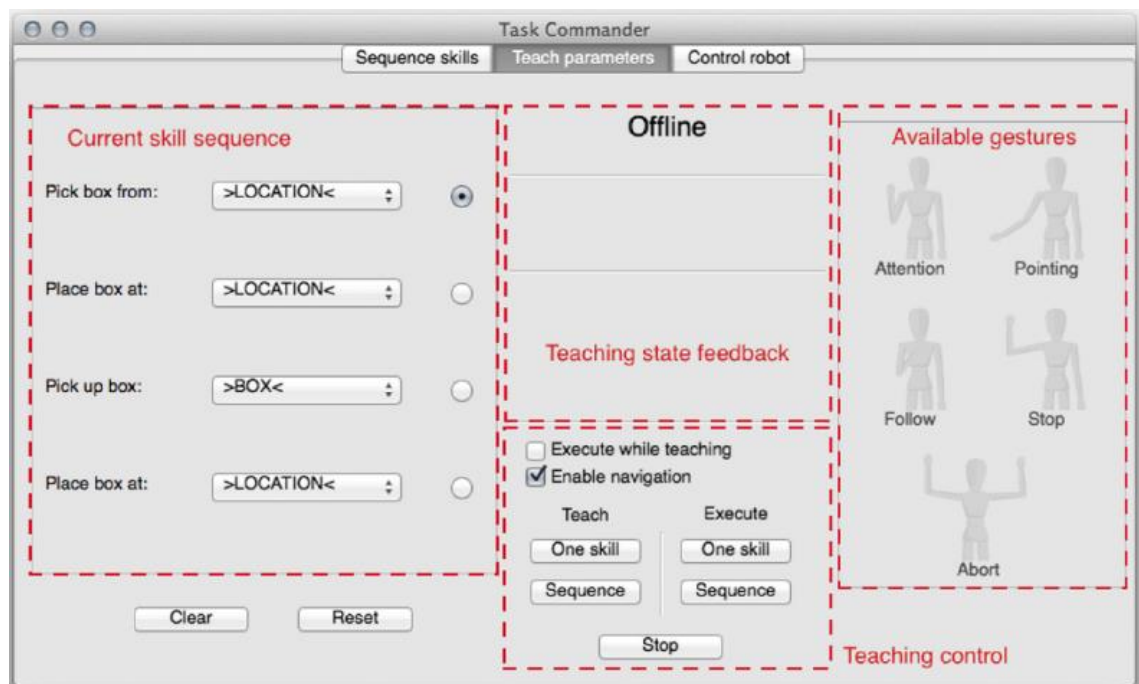


Figura 1.18 - Interface para ensino de tarefas do sistema desenvolvido por Pedersen e Kruger[25].

Maeda e Aburata [26] também estudaram o método de programação por ensinamento baseado em visão. Este método inclui o processamento da imagem adquirida pelo sistema de visão que consegue alcançar mais robustez em caso de mudança das condições da tarefa do que um método de ensinamento convencional. Posteriormente integraram um reforço na fase de ensinamento deste método para ser mínima a exigência do operador e aplicaram-no ao manipulador robótico num ambiente virtual. No entanto, o processo de ensinamento reforçado *online* requer muito tempo de computação. Além disso, a baixa taxa de sucesso do ensinamento reforçado faz com que o tempo de ensinamento seja maior. Então, com o objetivo principal de acelerar o processo de ensinamento reforçado incluíram neste projeto o reforço do ensinamento.

Este método segue os seguintes passos:

- Aprendizagem supervisionada por sistema de visão feita pelo ensinamento humano como está representado na Figura 1.19. O operador comanda o robô para que este desempenhe a tarefa, enquanto todas as imagens de ensinamento e todos os movimentos do robô são gravados. É obtido um mapeamento das imagens dos movimentos do robô como uma rede neuronal artificial em que basicamente os seus dados de entrada são as cenas das imagens e forma como saídas um movimento do robô correspondente à imagem.

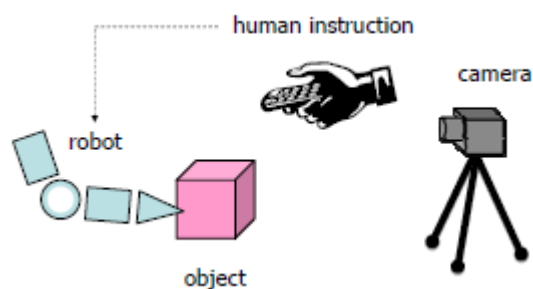


Figura 1.19 - Ensino do robô pelo humano [26].

- Aprendizagem reforçada baseada em visão para melhoria do desempenho. O movimento do robô é determinado pela saída da rede neuronal e inicialmente, o movimento pode não ser apropriado às condições da nova tarefa, mas o mapeamento das imagens dos movimentos do robô é gradualmente atualizado com o agente de reforço na aprendizagem.

O agente de reforço na aprendizagem caracteriza-se pelas variáveis que são melhoradas para reduzir atrasos temporais, por exemplo sabendo o atraso que se verifica, sendo a rede neuronal ajustada para modificar o seu *output*. Estas melhorias no processamento do mapeamento das imagens resultaram num método mais rápido cerca de 50% e a melhoria na aprendizagem das redes neuronais resultou num rácio de sucesso no ensinamento reforçado cerca de 30% maior do que o método de ensinamento sem reforço [26].

Focados numa programação fácil e acessível a utilizadores não especializados, Neto, Pires e Moreira [27] desenvolveram um sistema robótico de programação mostrando ao robô o que ele deve fazer, por meios naturais, ou seja, com elevado nível de abstração da linguagem de programação do robô. Isto através de gestos e fala, controlo de força e técnicas de geração de código. Destes, deram especial atenção ao reconhecimento de gestos obtido por extração de dados por meio de um sensor de movimento (acelerómetro de 3 eixos) embutido no comando de controlo remoto da Wii. Tanto os gestos como as posições da mão foram reconhecidos através de uma abordagem estatística e redes neuronais artificiais.

O protótipo do sistema (Figura 1.20) contém um robô industrial, um comando de controle remoto da Wii para capturar o movimento das mãos, um microfone para capturar a voz do operador, um sensor de força e um computador para correr a aplicação que controla todo o sistema. Usando gestos, o operador pode conduzir o robô para uma posição desejada e por comandos voz pode gravar as configurações do robô e gerar o código. Repetindo este processo, consegue gerar uma trajetória.

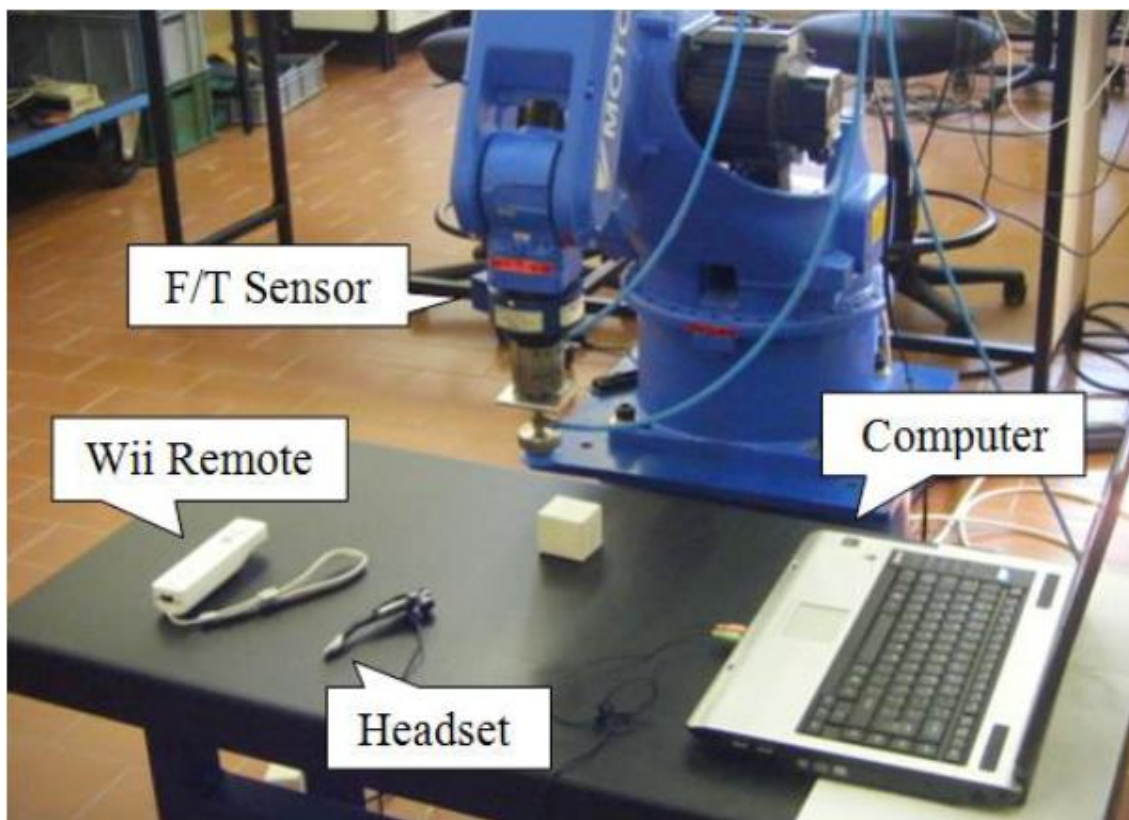


Figura 1.20 - Célula robótica desenvolvida por [27].

Para avaliarem este sistema, fizeram várias experiências num ambiente não controlado e compararam o seu desempenho com uma abordagem similar que em vez de gestos utiliza um sistema de orientação da mão baseado numa estratégia de controlo de força. Mostraram também que o sistema desenvolvido pode ser aplicado em diferentes plataformas robóticas e adaptado a diversos utilizadores, fazendo uma demonstração em dois robôs diferentes (ABB e MOTOMAN).

Neto e Moreira [28], também focados numa programação intuitiva e direta do robô, desenvolveram um sistema baseado em CAD para programar um robô a partir de um ambiente CAD 3D, permitindo a utilizadores com conhecimentos básicos de CAD, gerarem programas de robôs de modo *offline*, sem interromper o processo de produção do robô (Figura 1.21).

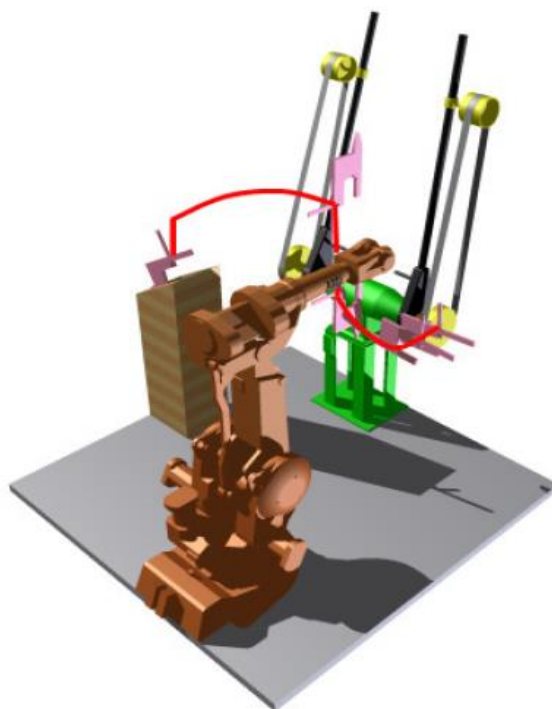


Figura 1.21: Modelo CAD de montagem da célula do robô [28]. O programa do robô é gerado a partir deste modelo.

Este sistema funciona como uma interface homem-robô, onde o utilizador consegue gerar programas para o robô através de um pacote CAD de baixo custo e comercialmente disponível. O grande objetivo coloca-se no facto desta interface ser algo simples, intuitiva, com curta curva de aprendizagem, barata e permitir o ensinamento do robô num curto período de tempo. Utilizaram, então, o pacote *Autodesk Inventor* para fazer essa interface entre o homem e o robô e as informações necessárias para a programação são extraídas sob a forma de modelos CAD. Essa informação deve conter a posição e também a orientação no espaço de todos os pontos da trajetória. A geração automática do código é feita através de um método o mais genérico possível, dividido em duas fases: uma primeira em que se definem e parametrizam as posições e orientações do robô, eixos de referência, ferramentas, etc e uma segunda em que se constrói o corpo do programa que contém predominantemente instruções de movimentos do robô. Posteriormente, o código gerado é testado e alvo de ajustes se for necessário. Para provar a simplicidade e baixa dificuldade da programação utilizando este método, Neto e Moreira fizeram experiências num robô da ABB com pessoas que nunca tinham trabalhado com o mesmo. Concluíram assim que este método é intuitivo, tem uma pequena curva de aprendizagem e permite a pessoas não especializadas gerarem um programa para o robô em poucos minutos.

Neto e Mendes [29], desenvolveram também um sistema que permite uma programação *offline* (Figura 1.22), a partir de um desenho CAD em execução no pacote CAD 3D utilizado no sistema descrito anteriormente. Exploraram a forma mais adequada de representar o movimento do robô num desenho CAD, de extrair automaticamente os dados de movimento do desenho, de fazer o mapeamento dos dados do ambiente virtual (modelo CAD) para o ambiente real e de fazer a geração automática das trajetórias/programas do robô.

A programação *offline* não é um processo completamente automático, envolve edição do código e definição dos programas por meio de um *software* de computador que simula o ambiente real do robô. No entanto, esta programação permite:

- Que o robô seja programado paralelamente à execução de tarefas;
- Que a programação possa ser feita por alguém especializado que pode ter o seu local de trabalho no escritório e não no espaço de fabrico;
- Os programas do robô podem ser previamente testados em simulação para previsão do comportamento do robô. O que permite otimizar os processos de fabrico.

Por outro lado a programação *offline* surge associada a:

- Erros de calibração do robô;
- Necessidade de testar os programas desenvolvidos *offline* no robô real para verificar se funciona corretamente;
- Necessidade de informação precisa sobre a modelação da célula robótica.

Utilizaram um processo idêntico ao anteriormente descrito para geração do código, uma vez que existe um ponto comum em todos os programas gerados. No geral, os robôs executam tarefas de manipulação e o processo de geração do programa não difere muito de aplicação para aplicação tendo pontos comuns tais como o agarrar, mover e colocar. Então, a geração automática dos programas para o robô não é mais do que a escrita de comandos num ficheiro de texto, linha por linha, que neste caso específico é gerido pela interface de *software* que extrai os dados de um desenho CAD, interpreta-os e gera os programas.

Neto e Mendes, para validar o sucesso deste método, usaram um robô *Motoman* e sujeitaram-no a duas experiências. Uma primeira em que o robô transportou objetos de um sítio para outro e a segunda em que o robô transportou objetos entre obstáculos. Neste, tal como em todos os métodos que recorrem à representação visual dos objetos no espaço, é verificado um erro de posição que muitas vezes vem da falta de importância dada ao processo de calibração do robô. É ainda um processo pouco sofisticado mas preza pelo facto de ser simples e intuitivo de usar.

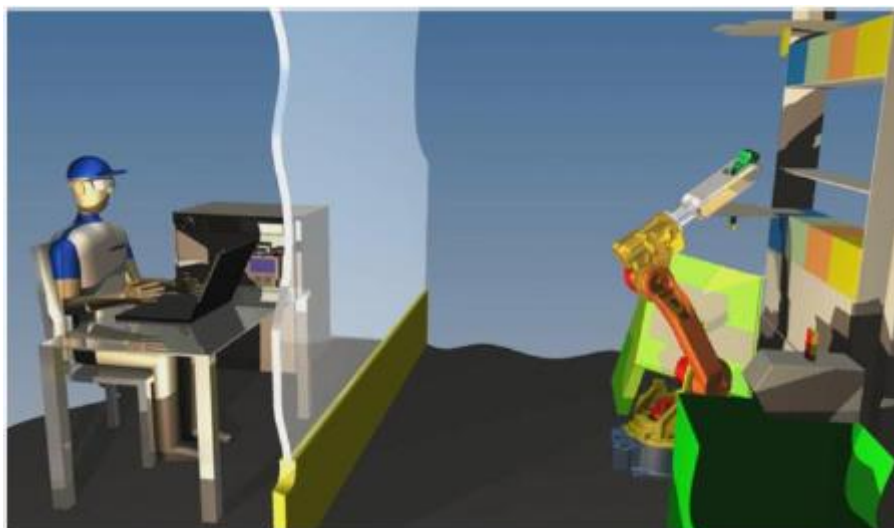


Figura 1.22 - Conceito de programação *offline* [29].

Shi *et al.* [30] também recorreram à programação *offline* para desenvolverem um método de programação para soldar interseções de tubos (Figura 1.23), com vista a evitar erros que acontecem no processo manual de soldadura. Para isso, utilizaram um robô articulado em cooperação com um posicionador e planearam uma trajetória de soldadura baseada no modelo geométrico da interseção de tubos mais complexo. Foi então utilizada uma programação *offline* que integra um algoritmo genérico baseado num modelo suficientemente complexo para que possa ser aplicado em muitos casos diferentes. O algoritmo é facilmente implementado através de um programa de computador de acordo com o fluxograma do algoritmo. Neste tipo de programação consegue-se evitar vários defeitos na soldadura por o algoritmo ser minuciosamente pensado previamente. No entanto, este método não garante fiabilidade em desvios acentuados na trajetória devido à deformação dos tubos.

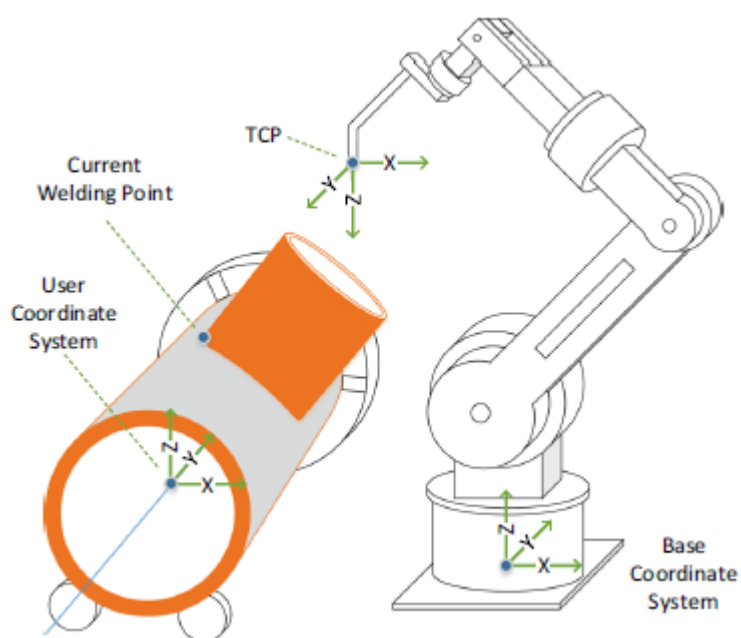


Figura 1.23 - Ilustração da soldadura da interseção de tubos por um manipulador industrial [30].

Também para responder à exigência por parte da indústria relativamente à flexibilidade e agilidade da produção, principalmente em áreas como a robótica onde ainda se vêm muitos robôs industriais que estão aparafusados no chão e dedicados a uma só tarefa específica, Schou *et al.* [31] propuseram um método que se adapta a novas abordagens para operar e programar novas tarefas de forma rápida. Denominaram este método de *task-level* onde a programação do robô é generalizada numa seleção de habilidades. Desenvolveram uma interface humano-robô baseada na programação *task-level* e como o objetivo é o robô ser programado por um operador de fábrica com conhecimentos de robótica limitados, a interface desenvolvida é fácil e rápida de usar. Por isso, a programação partiu de um nível de comandos específicos de movimentos ou funcionalidades básicas para um nível focado na tarefa. A abordagem da programação *task-level* surge dividida em três camadas que consistem em movimentos simples, habilidades e tarefas. Os movimentos simples são caracterizados por movimentos e funcionalidades básicas tais como agarrar ou mover para um ponto. As habilidades são capacidades orientadas a um objeto tais como rodar, “pick”, “place”, etc. As tarefas são descritas como sequências de habilidades associadas a parâmetros específicos da tarefa, sendo a tarefa em si um arquivo que contém a sequência de habilidades e os parâmetros associados a cada uma.

Da interface fazem parte as funcionalidades de executar e programar novas tarefas. A parte da programação é dividida na fase de especificação, onde é escolhida e parcialmente parametrizada a sequência de habilidades, e na fase de ensinamento onde o operador interage diretamente com o manipulador para o conduzir ao alvo específico (Figura 1.24).

Depois de alguns testes feitos, esta interface mostra ser promissora embora lhe faltem ainda algumas instruções de apoio ao operador durante a fase de ensinamento e questões de robótica tais como evitar os limites das articulações e a escolha de trajetórias livres de colisões.



Figura 1.24 - Fotografia de operador a conduzir o manipulador para uma coordenada específica [31].

Contrariamente à programação convencional do robô que é feita através da especificação de um conjunto de pontos definidos no espaço, foi desenvolvido um novo método de programação por demonstração em que “ o operador humano demonstra uma determinada tarefa e o robô é automaticamente programado de modo a que consiga imitar o movimento humano ” [32].



Figura 1.25 - Marcador inserido na pistola de pintura [32].

Este método engloba um sistema de visão estereoscópica e um marcador luminoso (Figura 1.25) que são acionados simultaneamente por uma técnica de *sincrovision*, patenteada pela FEUP e licenciada à empresa FLUPOL, S.A., para que a leitura dos dados (orientação e posição) seja precisa e robusta mesmo em ambientes industriais. O seguimento do movimento do operador é, assim, baseado em marcadores na gama do visível e capturado por câmeras. Esta técnica tem inúmeras vantagens pois é um método sem contacto, os operadores não têm que usar dispositivos adicionais e a calibração é permanente enquanto as câmeras não são deslocadas. É possível adaptar o número de câmeras a usar conforme a aplicação e dependendo da precisão desejada. A trajetória feita pelo operador é reconstruída por meio de algoritmos capazes de a reproduzir com 6 graus de liberdade e que a suavizam antes de ser inserida no robô.

O funcionamento com este sistema exige reduzida interação do humano com o computador, apenas é necessário que indique o início e o fim da demonstração bem como para calibrar o sistema e reproduzir os programas desejados. Para se efetuar a programação é indispensável que se atravesse os seguintes passos:

- Calibração: Passo onde é feita a calibração da câmera com o robô e a calibração entre a ferramenta do operador com o marcador e com a ferramenta do robô. Esta calibração apenas é necessária uma vez desde que as câmeras permaneçam no mesmo local em relação ao robô.

- Demonstração: Esta etapa é iniciada a partir da aplicação e, de seguida o operador apenas tem que executar a tarefa enquanto os dados vão sendo capturados e processados. Concluída a trajetória, o operador apenas tem que finalizar o processo através da interface de forma natural, sem qualquer ferramenta extra ou alteração na rotina do operador.

- Reprodução: Nesta fase, o operador apenas tem que selecionar na interface o programa desejado. O código é transferido para o controlador do robô e este executa-o.

Tanto o processamento de imagem, como a análise da trajetória completa e a geração do código são feitos depois de finalizada a demonstração mas demoram apenas alguns segundos, daí a imitação do movimento ser alcançado num curto período de tempo.

Este sistema foi testado numa aplicação de pintura em que o operador demonstrou várias trajetórias inserindo apenas na interface de *software* qual o ponto inicial e o final do processo, tendo-se obtido resultados de sucesso aprovados pelos próprios pintores [32]. O projeto está em funcionamento, desde 2014 na FLUPOL, S.A. atuando, precisamente como interface de alto-nível para um robô industrial de pintura em *spray*.

Similar a este sistema sob o ponto de vista de programação simples e rápida do robô, a ABB desenvolveu também um sistema capaz de replicar os movimentos do humano e traduzi-los num programa para o robô especialmente para aplicações de pintura [33]. Este processo inicia-se com a gravação do movimento do operador, usando um sensor portátil semelhante a uma pistola de pulverização de tinta convencional como se pode ver na Figura 1.26. A captura do movimento é feita através um sistema baseado em campos magnéticos artificiais designado de *Polhemus Liberty* e a sua gravação é iniciada através da pressão do botão da pistola do qual também resulta a projeção de uma linha de guia.



Figura 1.26 - Ferramenta marcador utilizada no método SRP (Simplified Robot Programming) da ABB [33].

Os dados capturados contém a posição, orientação e marcas de tempo o que permite a recriação do movimento do operador muito próximo do exato mas no ambiente virtual. Os dados do movimento capturados são processados e otimizados para serem utilizados no programa da aplicação em específico. O sistema elimina dados que não são relevantes como é exemplo a definição de uma trajetória em linha reta em que adquire uma série de pontos quando apenas são necessários dois pontos. Os dados capturados relativamente ao tempo servem para determinar a velocidade da trajetória entre pontos consecutivos portanto tem-se especial atenção aos dados que são eliminados para que não se perca informação útil para se saber a velocidade. Tendo o processamento dos dados finalizado, a trajetória criada pode ser vista e editada no *software* RobView da ABB que está configurado para aplicações de pintura. Por fim, a trajetória é transferida para o controlador do robô para que este a execute dando-lhe um movimento mais complexo e fluido. Este sistema, em relação ao descrito anteriormente, tem a desvantagem de se basear em campos magnéticos o que o torna vulnerável a objetos metálicos que influenciam o campo magnético. Isto reduz a sua precisão em caso de proximidade destes objetos. Tem também a desvantagem de no processo de demonstração, o operador não estar efetivamente a realizar a operação de pintura, apenas está a simular o que pode não emergir nos mesmos resultados.

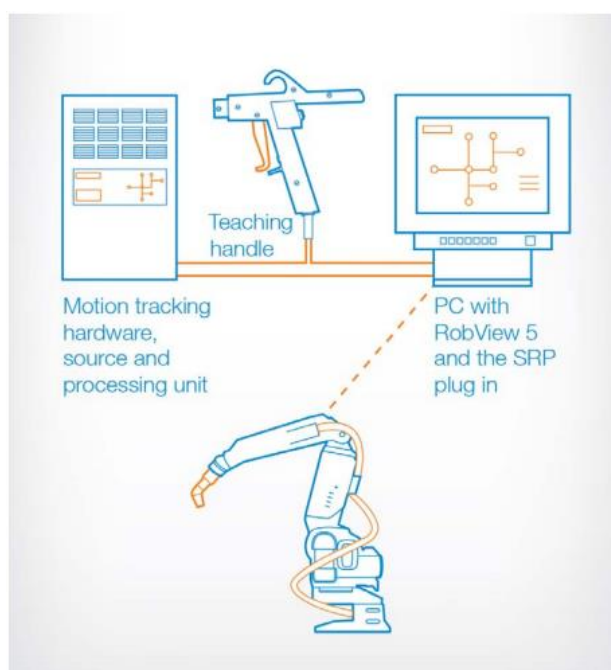


Figura 1.27 - Ilustração da configuração do Sistema SRP da ABB [34].

Em operações desempenhadas na indústria por robôs como soldar, aparafusar ou perfurar, otimizar o tempo do movimento do manipulador entre diferentes pontos torna-se importante para maximizar o rácio de produção. No entanto, por vezes, a programação do robô é uma tarefa complexa e, por isso, Tahriri *et al.* [35] desenvolveram um *software* de simulação 3D integrado e um sistema de realidade virtual para a simplificação e aceleração de tarefas ou seja, para melhorar a qualidade dos processos de manufatura. Este sistema tem a capacidade de comunicar, transferir, otimizar e testar os dados obtidos dos sistemas de realidade virtual e do ambiente 3D para o robô real de forma rápida e eficiente (Figura 1.28).

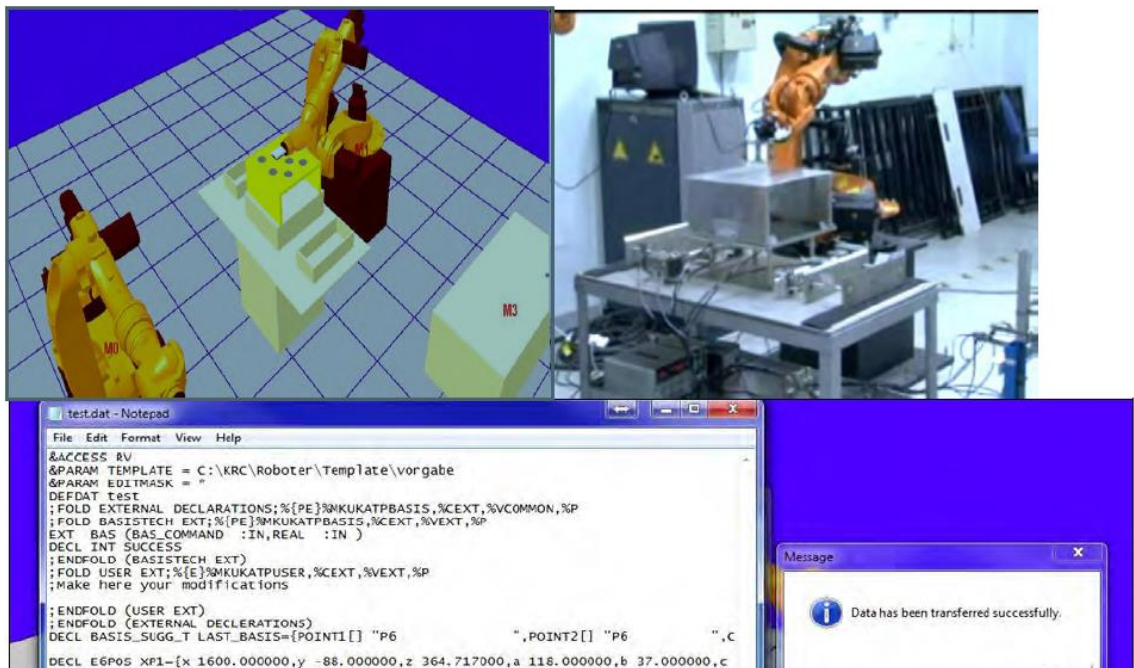


Figura 1.28 - Transferência de dados da simulação 3D para o robô real [35].

A realidade virtual é a interface humano-computador que responde de forma interativa aos comandos inseridos pelo utilizador. Usada em aplicações de manufatura como planeamento, agendamento da linha de montagem, simulação e manutenção, previne erros caros nos processos de planeamento e execução.

Este modelo de otimização do tempo de movimento do braço robótico foi testado no robô KUKA num processo de aparafusar. O sistema de realidade virtual adquire do utilizador os dados da tarefa e estes são transferidos para o *software* de simulação 3D. Este *software* otimiza a sequência de pontos da tarefa no sentido de recuperar algum tempo no movimento. Assim, foi possível concluir que o rácio de produção aumentou e o tempo total de realização da tarefa diminuiu tal como se pode ver nas estatísticas da Figura 1.29.

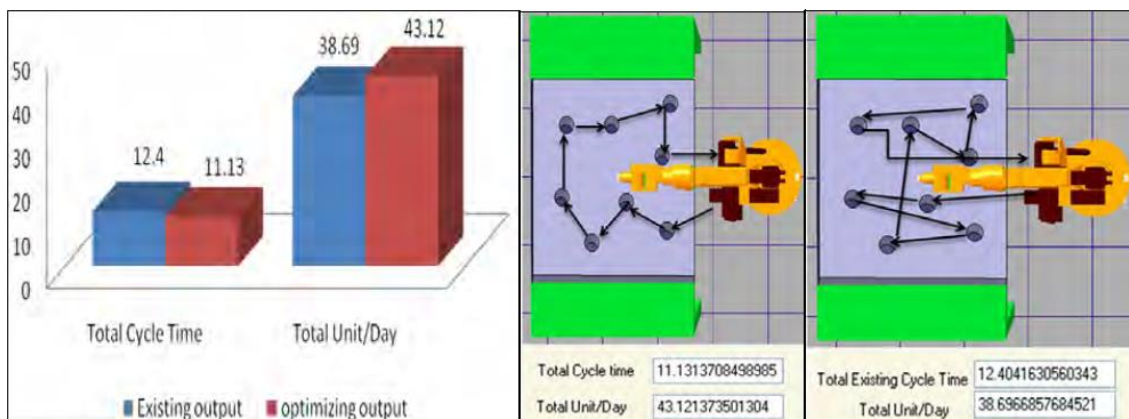


Figura 1.29 - Estatísticas apresentadas na interface do sistema desenvolvido por [35].

1.4.5 PCL - *Point Cloud Library*

PCL é um projeto aberto de grande escala para imagens 2D/3D e para processamento de nuvens de pontos. Esta biblioteca disponibiliza algoritmos para filtrar, reconstruir superfícies e para segmentação úteis na remoção de outliers, na extração de pontos chave, no tratamento de regiões de interesse, no reconhecimento de objetos e na própria visualização das nuvens de pontos. PCL é dividida em pequenas bibliotecas de código que podem ser compiladas separadamente, o que reduz a exigência computacional das plataformas por onde pode ser distribuída.



Figura 1.30 - Logotipo *Point Cloud Library* [36].

Estruturas de dados para pesquisa PCL

Nos algoritmos de pesquisa de pontos vizinhos é frequente o uso de estruturas de dados que se baseiam na análise do ambiente de um dado ponto. O objetivo destas estruturas é o de fazer esta pesquisa o mais eficientemente possível e o de, por vezes, também compactar dados.

Voxel grids

A grelha voxel é uma estrutura de dados para partição do espaço 3D que divide o espaço Euclédiano em voxels regulares. Esta estrutura é, muitas vezes, utilizada em filtros que diminuem a resolução da nuvem de pontos para reduzir ruído associado a medições na aquisição (Figura 1.31).

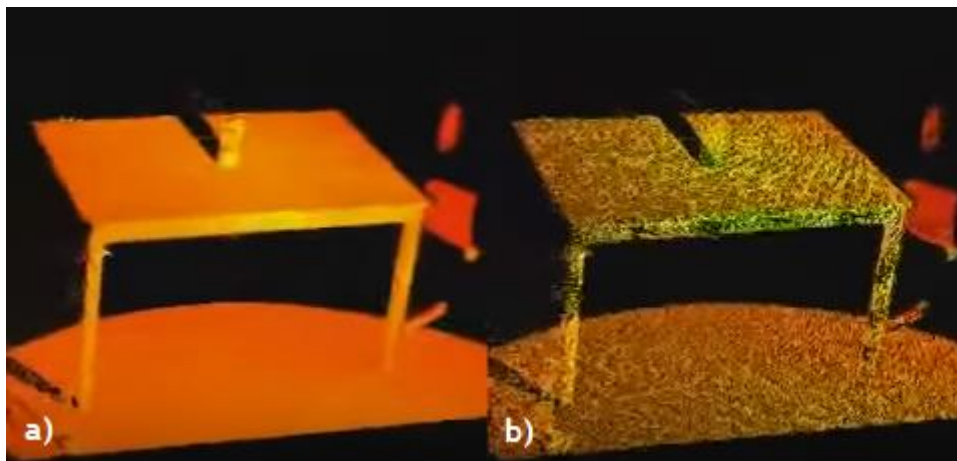


Figura 1.31 - Representação de nuvem de pontos antes a) e depois b) da aplicação de um filtro recorrendo à grelha voxel ¹.

Octrees

O octree é uma técnica de partição hierárquica do espaço que adapta a sua estrutura de dados em árvore à distribuição dos pontos pela nuvem (Figura 1.32). Isto é feito recursivamente dividindo cada voxel em 8 octantes percorrendo a árvore em profundidade nível a nível até ao fim ou até não haver mais pontos para pesquisar.

¹ http://pointclouds.org/documentation/tutorials/voxel_grid.php

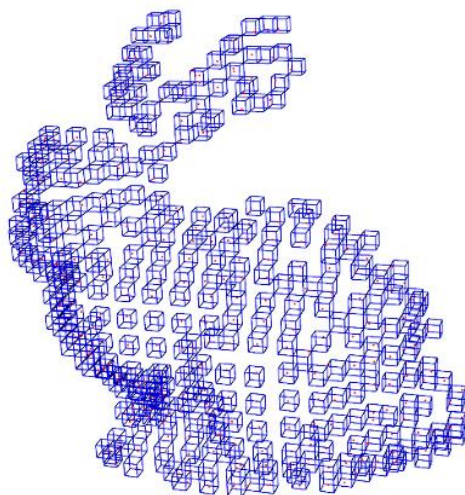


Figura 1.32 - Divisão de nuvem de pontos pela técnica octree².

K-d tree

Kd-tree é uma estrutura de dados para partição do espaço que armazena um conjunto de pontos de k dimensões numa estrutura em árvore que permite pesquisa eficiente em intervalos de pontos e pesquisas de vizinhos mais próximos (Figura 1.33). K-d tree pode também ser usado para encontrar correspondências entre nuvens ou características de descritores.

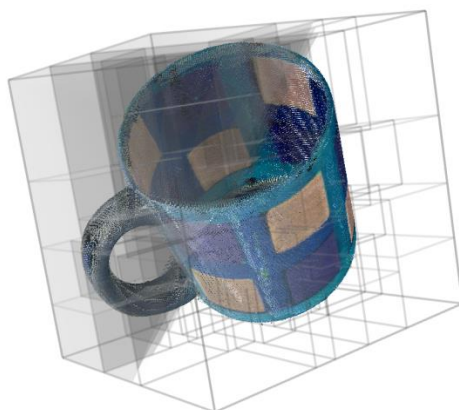


Figura 1.33 - 3D k-d tree³.

Na pesquisa de vizinhos mais próximos, as árvores de k dimensões em cada nível dividem-se ao longo de uma dimensão específica usando um hiperplano que é perpendicular a um eixo. No exemplo da Figura 1.34 em que à esquerda se podem ver o ponto que se pretende pesquisar e os respetivos vizinhos e à direita a árvore de pesquisa, o ponto raiz da árvore é aquele mais distante do ponto que está a ser pesquisado (assinalado com um x) de modo a que o círculo

² http://docs.pointclouds.org/trunk/group__octree.html

³ http://docs.pointclouds.org/trunk/group__kdtree.html

com centro nesse ponto contenha todos os pontos candidatos a vizinhos mais próximos. Neste caso, o ponto raiz da árvore é o ponto A.

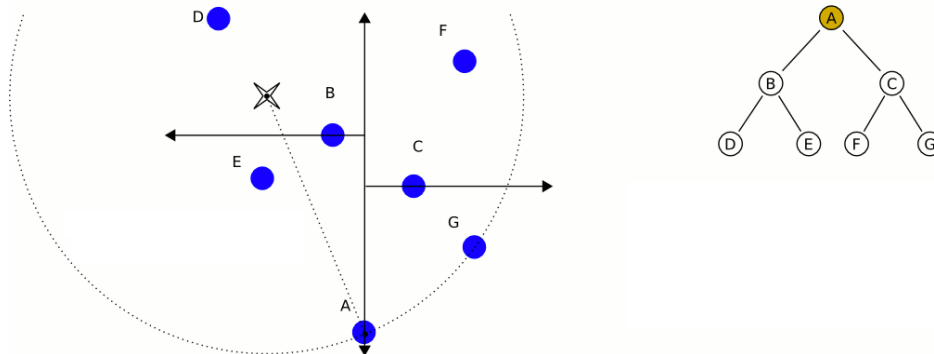


Figura 1.34 - Ponto de início da partição k-d tree⁴.

A subdivisão da árvore (Figura 1.35) é feita com base nas coordenadas do primeiro elemento, ou seja se a coordenada do próximo elemento é menor do que a da raiz, ele deverá estar na sub-árvore esquerda e se for maior do que a da raiz, deverá estar na sub-árvore da direita. Cada nível abaixo na árvore se vai dividindo, retornando à primeira dimensão assim que as outras estejam esgotadas. A divisão seguinte é sempre feita com hiperplanos perpendiculares ao eixo considerado anteriormente. O método mais eficiente de utilizar a árvore de k dimensões é usando um método de partição de ordenação rápida que coloca em cada nível o valor mediano e todos os valores de coordenadas menores à esquerda e os maiores à direita. Ou seja, após se ter encontrado o ponto A dividiu-se em x o plano, colocando à esquerda na árvore o ponto mediano com coordenada menor nesse eixo (B) e maior à direita (C). De seguida a divisão é feita em y, perpendicularmente à divisão anterior, colocando à esquerda na árvore o ponto o ponto com menor coordenada (D no ramo de B e F no ramo de C) em relação à raiz e à direita o ponto com maior coordenada (E no ramo de B e G no ramo de C) em x.

⁴ https://en.wikipedia.org/wiki/K-d_tree#/media/File:KDTree-animation.gif

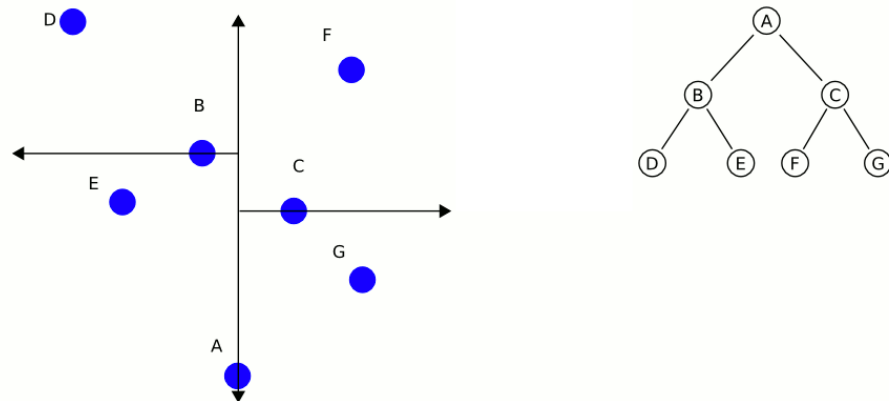


Figura 1.35 - Partição do espaço em k-d tree e árvore resultante⁴.

Esta partição é feita sucessivamente em ambos os ramos até que as partições são constituídas apenas por um elemento ⁵. Começando a pesquisa pelo ramo de B (Figura 1.36), calcula-se a distância do ponto que está a ser pesquisado a B e verifica-se que o círculo gerado por estes dois pontos já não contém mais pontos.

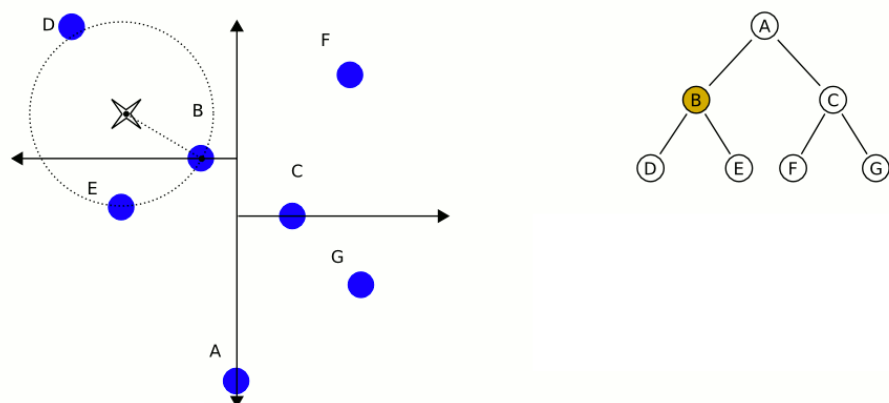


Figura 1.36 - Pesquisa no ramo esquerdo da k-d tree⁴.

D e E são descartados pois estão mais distantes do ponto pesquisado. Teoricamente o vizinho mais próximo é o que resulta da interseção deste círculo com o ramo da árvore resultante da última partição possível. Na figura Figura 1.37 pode ver-se outro exemplo de uma divisão de duas dimensões em k-d tree.

⁵ https://en.wikipedia.org/wiki/K-d_tree

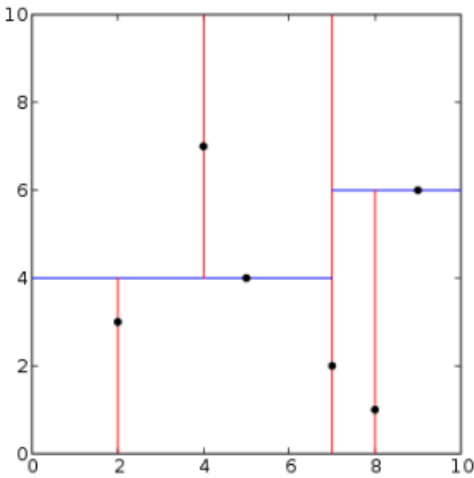


Figura 1.37 - 2D k-d tree⁶.

⁶ https://en.wikipedia.org/wiki/K-d_tree#/media/File:Kdtree_2d.svg

1.4.6 Princípios comuns de medição de distâncias

TOF - Time of Flight

O funcionamento deste sistema (Figura 1.38) inicia-se com os leds presentes na camera a incidirem a radiação no objeto, que é refletida e posteriormente captada pelo sensor. Cada pixel da imagem captada mede o tempo que a luz necessita para viajar entre a camera e o objeto. De seguida, a camera calcula a distância ao objeto para cada pixel e gera uma nuvem de pontos que é enviada ao PC para posterior processamento.

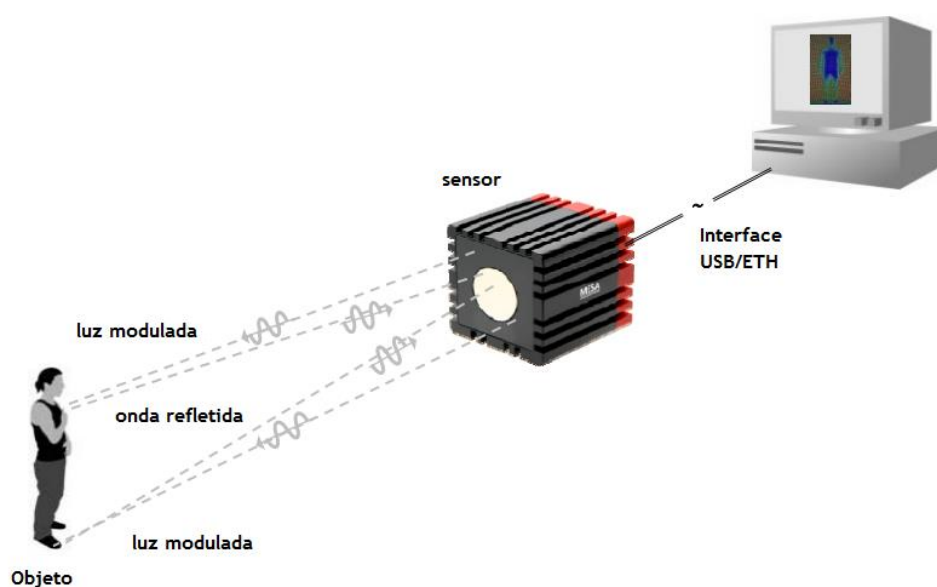


Figura 1.38 - Sistema tipo da tecnologia TOF - *Time of Flight* [37].

Conhecida a velocidade v de propagação do feixe emitido pelo led e medido o tempo Δt que o feixe refletido leva a retornar ao sensor depois de embater no objeto, a distância do sensor ao objeto pode ser determinada pela expressão 1.1.

$$d = \frac{1}{2} v \times \Delta t, \quad (1.1)$$

Triangulação

O funcionamento deste sistema (Figura 1.39) inicia-se com o laser presente no sensor a incidir a radiação no objeto, que é refletida e posteriormente captada pela camera que se encontra num ângulo conhecido.

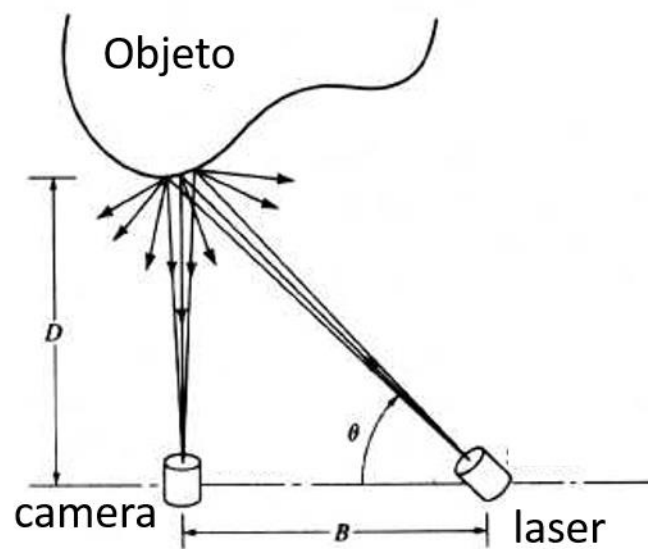


Figura 1.39 - Sistema tipo da tecnologia de triangulação⁷.

Conhecido o ângulo θ que o feixe emitido pelo laser faz com a linha em que se encontra o emissor e o recetor, a distância D do sensor ao objeto pode ser determinada pela expressão

$$D = \tan(\theta) \times B , \quad (1.2)$$

sendo B a distância entre o emissor e o recetor do feixe.

⁷ <https://paginas.fe.up.pt/~eol/SP-leec/TRABALHOS/ROB/sensores.html>

Capítulo 2

2.1 Caracterização do Problema

Nesta secção será descrito o cerne do objetivo de desenvolvimento deste trabalho sob a forma de definição do problema, bem como a descrição de abordagens a serem consideradas como possíveis soluções do problema.

2.1.1 Definição do problema

A definição do problema, inserida na motivação deste trabalho compreende vários aspetos, neste caso relacionados com situações específicas do processo de enrolamento filamentar e aplicação de fibra por projeção. O problema afirma-se em aspetos como, a dificuldade inerente ao enrolamento de fibras para produção de peças complexas, estruturas multiorifícios, a dificuldade de aplicação de reforços localizados em concavidades, a dificuldade associada à programação de robôs e a necessidade de um sistema que permita o controlo da qualidade das peças desenvolvidas pelo robô. Faz parte também deste desafio, o objetivo de se propor uma possível solução para um sistema genérico aplicável ao enrolamento de peças complexas.

Atualmente, no laboratório de materiais compósitos do INEGI existe um equipamento de enrolamento filamentar CNC de 6 eixos com controlo independente e que inclui também 3 módulos independentes de tensionamento unitário de filamentos de fibras. Este equipamento limita o formato das peças aqui produzidas a peças simétricas, com concavidades limitadas e com limitação no número de orifícios, surgindo então a necessidade do complemento com um equipamento capaz de realizar rotações, pontos inalcançáveis pelo sistema CNC convencional e apoio na criação de orifícios e formas assimétricas.

Na aplicação de fibras por projeção, a exigência põe-se na espessura e qualidade da peça e na capacidade do operador produzir peças de idêntica qualidade repetidamente. A peça produzida deve ter uma determinada espessura tão precisa quanto o cliente exija, o que pode não ser uma tarefa óbvia para o humano. No enrolamento de fibras em fita exige-se que não aconteçam situações de derrapagem, ou seja, que a fita fique na posição definida com a compactação desejada. Em casos de produção em série é também uma tarefa árdua a produção de peças com boa repetibilidade. Portanto é necessário um sistema para controlo de qualidade em paralelo.

Por outro lado, os equipamentos industriais automáticos associados a este tipo de processos tornam-se difíceis de serem programados por qualquer operário, exigindo alguém especializado. Este facto acarreta custos normalmente difíceis de serem suportados por pequenas e médias empresas culminando na aquisição de sistemas que são pobremente aproveitados. Isto desacredita a potencialidade destes sistemas e diminuição de investimento neles.

2.1.2 Solução proposta

A solução proposta, que abarca os objetivos deste trabalho visa resolver a questão do enrolamento filamentar para peças complexas como complemento do equipamento CNC convencional com um manipulador robótico havendo comunicação entre ambos. A aplicação robotizada de fibras permite maior liberdade para trajetórias complexas e, conseqüentemente peças de mais variadas formas. A inclusão de um manipulador neste sistema permite também detetar e apoiar o equipamento CNC convencional em situações inerentemente complicadas de enrolamento filamentar como é o caso da criação de orifícios nas mais variadas peças. Nesta situação, a fibra em fita torna a tarefa de enrolamento uma tarefa árdua necessitando do apoio de uma ferramenta que proporcione o desvio da fita. Ou seja, o robô atuaria nesta situação de forma a detetar o obstáculo (orifício) e fazer com que a fita o contorne ou, sendo programado com conhecimento prévio da localização do obstáculo, atuará de forma adequada. Por outro lado, a utilização de um manipulador em aplicações de projeção de fibra permite a produção de peças com maior repetibilidade face à capacidade de produção humana.

O facto dos sistemas robotizados para este fim serem, muitas vezes, difíceis de programar pode ser ultrapassado pela programação de robôs por demonstração. Esta solução passa por aplicar um método de simples programação em que o robô aprende a realizar a tarefa por meio de uma prévia demonstração do operador. O código gerado para programação do robô deve ser genérico para ser aplicado a diferentes modelos de manipuladores de diferentes fabricantes. Isto permite ao operador programar o robô de forma intuitiva, simples e prática alterando a trajetória sempre que necessário e aplicando este método qualquer que seja o manipulador instalado.

Para monitorização da espessura da fibra na sua aplicação por projeção, deve introduzir-se um sistema dotado de visão artificial. Este sistema permitirá garantir, para além da espessura desejada, a qualidade da deposição. Para isso, o sistema de visão deve responder a requisitos de precisão e fiabilidade. No caso do enrolamento filamentar, este sistema de visão deve auxiliar na deteção de orifícios e garantir a posição correta da fita.

2.2 Projeção de compósito

2.2.1 Introdução

A projeção robotizada de fibra (Figura 2.40) caracteriza-se por um processo de moldação através da projeção de fibras curtas com uma pistola guiada por um manipulador robótico. Associado a este está integrado um sistema de visão para aquisição do perfil do molde a ser revestido e para deteção da espessura de compósito depositado. Automaticamente, este sistema é capaz de efetuar reforços localizados em zonas com lacunas no preenchimento conforme comparação da espessura de compósito depositado com a espessura pretendida pelo operador. Através de uma interface gráfica é possível o utilizador introduzir a espessura que pretende que seja depositada por áreas específicas do molde bem como os avanços durante o processo.



Figura 2.40 - Projeção de compósito⁸.

2.2.2 Arquitetura Funcional

A arquitetura funcional do sistema de projeção de compósito descrita na Figura 2.41 tem como entradas os comandos do utilizador no PC através da interface gráfica e os pontos do sistema 6DMimic obtidos na demonstração da aplicação do compósito. Como saídas, o utilizador tem acesso, através da interface gráfica, às espessuras calculadas pelo programa e o robô

⁸ <http://www.inegi.up.pt/experienciasprojetos.asp?idm=4&idsubm=4&LN=>

executa os processos de projeção de compósito e/ou reforço localizado em zonas de deposição em falta. As ligações entre o PC e o sensor bem como entre o PC e o robô são feitas por Ethernet, utilizando o protocolo TCP/IP.

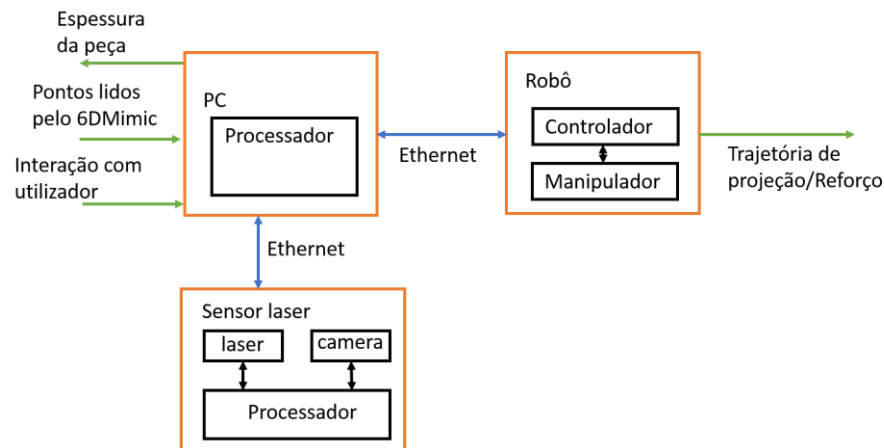


Figura 2.41 - Arquitetura funcional relativa ao sistema de projeção de compósito.

2.2.3 Componentes de *Hardware*

Sensor laser *Riftek*

O laser utilizado para obter digitalizações dos moldes e sucessivas deposições foi um laser 2D RF625 da Riftek. Este equipamento opera a partir da radiação em linha de um laser semiconductor formada por uma lente e projetada para um objeto. A radiação refletida pelo objeto é recolhida pela lente e dirigida para um sensor de imagem CMOS bidimensional. A imagem de perfil do objeto é assim formada e analisada por um processador de sinal, que calcula a distância ao objeto (coordenada Z) para cada ponto do conjunto ao longo da linha de laser sobre o objeto (coordenada X) (Figura 2.42).

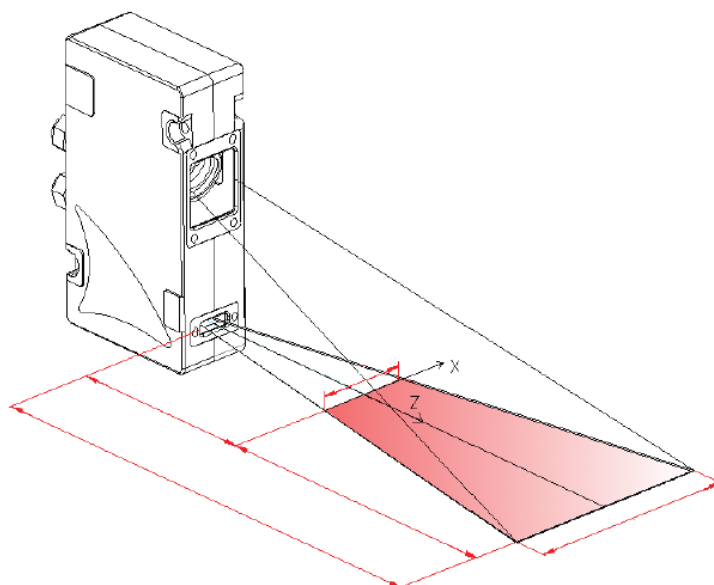


Figura 2.42 - Representação do laser RF625 e respetivo referencial [38].

Comunicação com o laser Riftek

A comunicação com o laser Riftek é feita por sockets por meio de um cabo Ethernet utilizando o protocolo TCP/IP. Todas as funções de comunicação já estavam implementadas numa biblioteca fornecida pelo fabricante.

Manipulador Motoman

Programação do manipulador

A programação do manipulador para que este efetue a trajetória de projeção é feita por demonstração.

O operador executa o movimento de cobertura de toda a peça, tal como o movimento necessário para a projeção manual da fibra, com uma ferramenta de demonstração. Esta ferramenta é uma pistola de projeção de fibra que inclui o marcador do sistema 6D Mimic para que sejam adquiridos os pontos referentes à trajetória como se pode ver na Figura 2.43.



Figura 2.43 - Representação da ferramenta de demonstração.

O sistema utilizado na demonstração engloba uma componente de visão estereoscópica e um marcador luminoso que são acionados simultaneamente por uma técnica de *sincrovision*, para que a leitura dos pontos (posição dos leds do marcador) seja precisa e robusta mesmo em ambientes industriais. A captura dos pontos ao longo da trajetória é feita através de duas cameras que detetam a presença de marcadores na gama da luz visível presentes no icosaedro. Estes marcadores luminosos são LEDs de alta potência/ alto brilho que são fáceis de controlar e têm grande versatilidade. Para captura da orientação 3D são precisos 3 LEDs de cores diferentes não colineares visíveis pelas cameras em cada momento da demonstração. As duas cameras adquirem uma imagem do marcador simultaneamente que é processada de modo a que apenas os pontos luminosos sejam analisados. A calibração é permanente desde que as cameras não sejam deslocadas.

A trajetória feita pelo operador é reconstruída por meio de um algoritmo capaz de reproduzir com 6 graus de liberdade e que a suavizam antes de ser inserida no robô.

Para funcionamento do sistema, apenas é necessário que o utilizador indique o início e o fim da demonstração na interface de utilização do 6Dmimic que está representada na Figura 2.44.

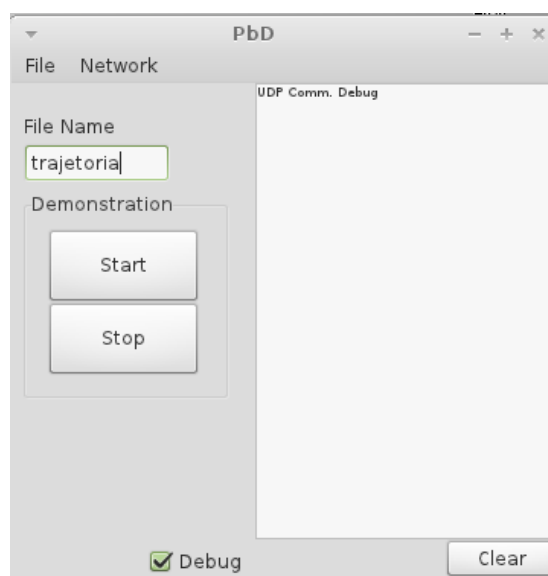


Figura 2.44 - Interface gráfica da aplicação 6Dmimic.

Este sistema que está instalado no computador do laboratório inclui uma interface que tem dois botões, um de “START” para informar o início da demonstração e um de “STOP” para terminar a aquisição de dados como se pode ver na Figura 2.44. Tem também um campo de preenchimento que corresponde ao nome a atribuir aos ficheiros onde são guardados os pontos obtidos. Após este processo, são gerados dois ficheiros com o nome introduzido pelo utilizador que estarão guardados na pasta “PseudoCodigo” com as extensões .traj, que contém todos os pontos adquiridos e .segs, que contém apenas os pontos resultantes da aplicação de um filtro de suavização aos pontos da trajetória eliminando alguns pontos intermédios. Posteriormente, após obtenção dos pontos da trajetória a reproduzir e de se ter feito as transformações necessárias aos pontos, é feita uma conversão dos pontos num ficheiro de código para o robô MOTOMAN através da aplicação CodeGen (Figura 2.45 e Figura 2.46) que se encontra na pasta “siiari-deltamatic”. Para isso, na interface deve-se carregar o ficheiro .segs gerado anteriormente pressionando o botão “LoadWayPoints que faz aparecer na janela de visualização do separador “Input Data” os pontos. Deve-se ainda, no separador “Config”, atribuir um nome ao ficheiro no campo “job name” e gerar o programa em “Generate Code”.

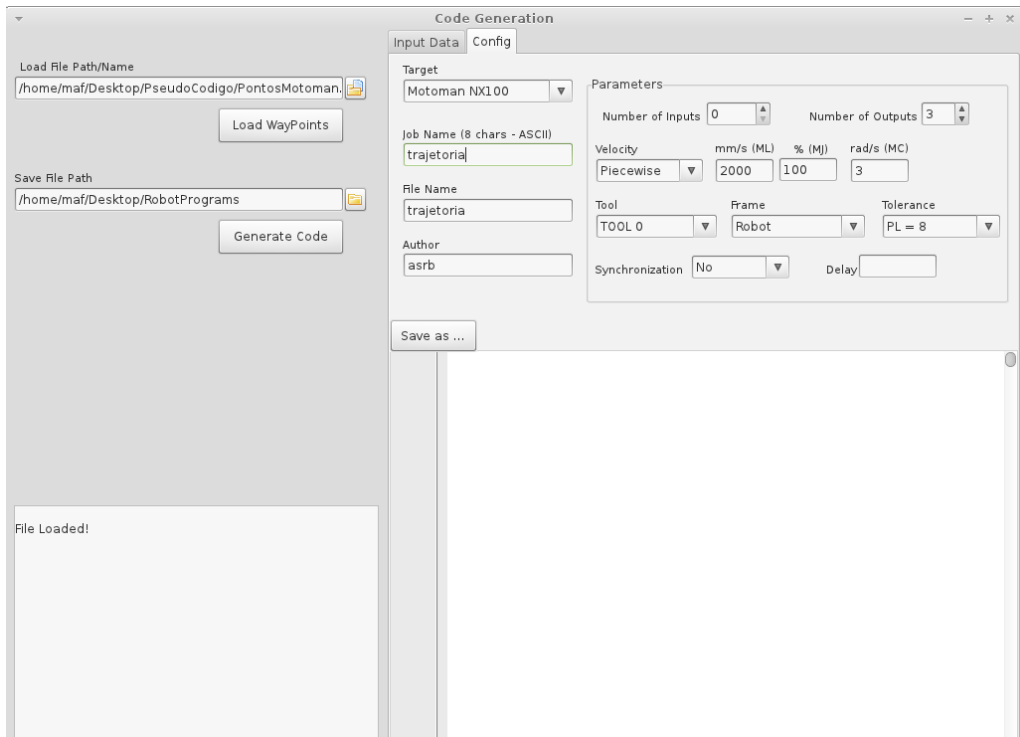


Figura 2.45 - Interface da aplicação CodeGen no separador “Config”.

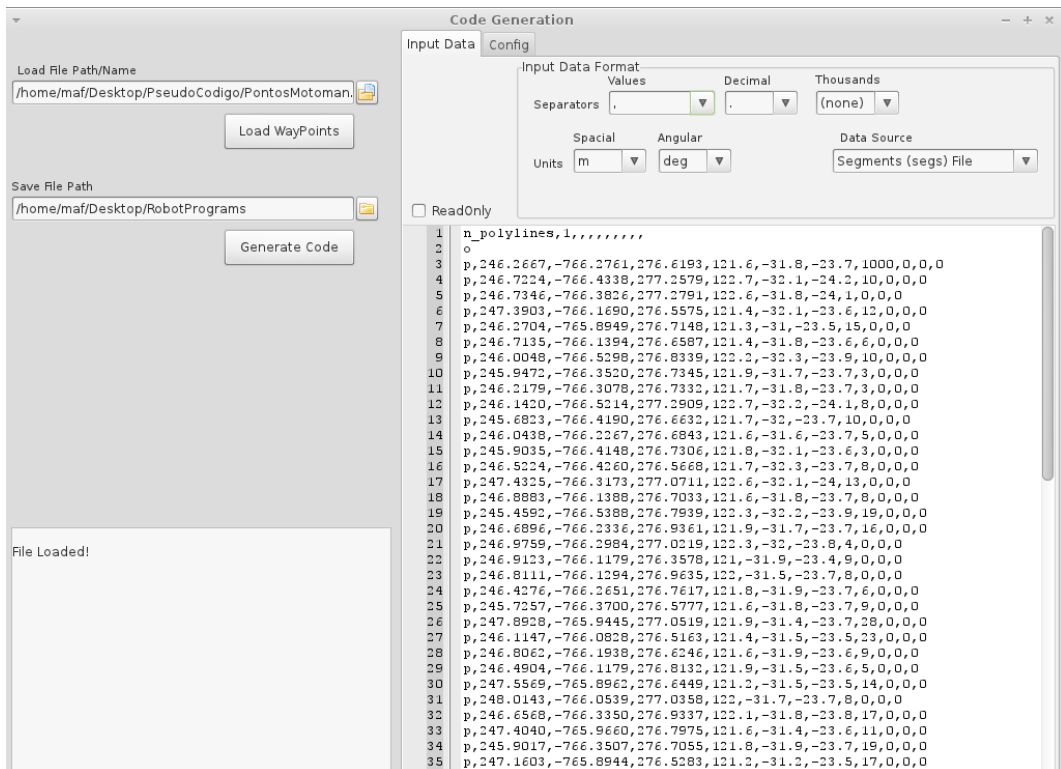


Figura 2.46 - Interface da aplicação CodeGen no separador “Input Data”.

Comunicação com robô MOTOMAN

A comunicação com o robô é feita por Ethernet baseada na documentação fornecida⁹.

Aplicação MotomanCom

Esta aplicação permite descarregar programas de extensão .JBI para o controlador NX100 para serem executados pelo manipulador MOTOMAN. Executando a aplicação, deve-se selecionar “open” no menu “window” e assim que o ficheiro do programa é aberto deve-se selecionar “send” para que o programa seja enviado para o robô.

2.2.4 Software

Para que o utilizador pudesse definir a espessura de deposição de fibra, criou-se uma interface gráfica (Figura 2.47) através da qual se pode visualizar um primeiro *scan* do molde da peça adquirido por meio do sensor laser e utilizando um manipulador. De seguida é aplicado um filtro aos dados para remoção de *outliers* e estruturas não pertencentes ao molde. O utilizador define áreas na superfície do molde para que possa estipular espessuras adequadas à forma. De seguida, é feita uma primeira deposição da fibra por meio de um braço robótico com a ferramenta de deposição. Posto isto, obtém-se novo *scan* da peça para que se avalie se a deposição é suficiente. O utilizador pode através da interface ter conhecimento das áreas que vão ser posteriormente reforçadas. Este reforço é automaticamente planeado e executado pelo manipulador.

⁹ <http://www.motoman.com/motomedia/manuals/docs/152388-1.pdf>

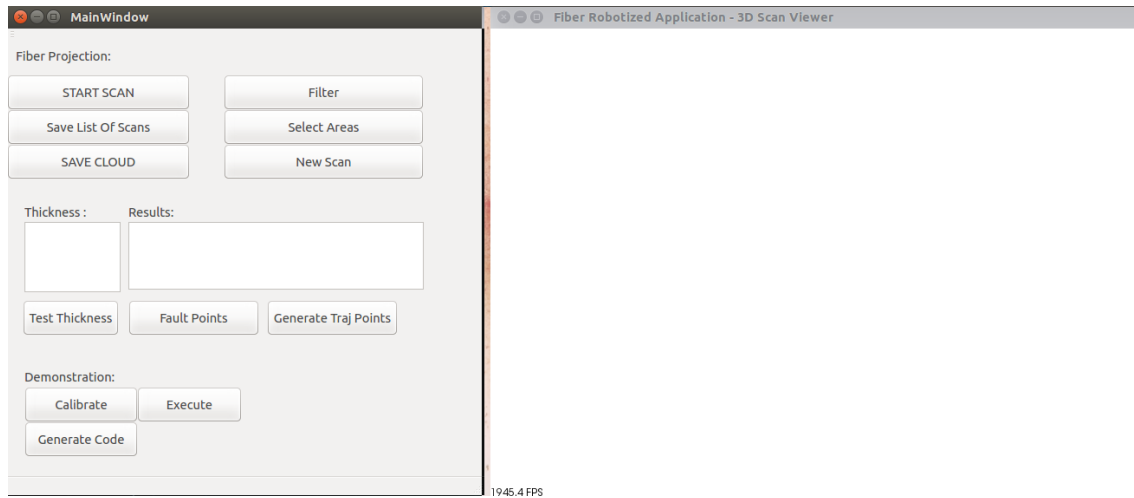


Figura 2.47 - Apresentação inicial da interface gráfica.

A interface gráfica tem várias funcionalidades associadas aos requisitos estipulados para a projeção robotizada de fibra (Figura 2.48) e para o processo de demonstração que permite a programação do manipulador através das rotinas “Calibrate”, “Generate Code” e “Execute” (Figura 2.47).



Figura 2.48 - Diagrama geral de funcionamento da projeção robotizada de compósitos.

Programação por demonstração

A programação do robô para executar o processo de projeção de fibra ou para executar a digitalização das peças procede-se essencialmente em dois passos: demonstração/aquisição de pontos e execução.

Previamente deve ser feita uma calibração da ferramenta no robô (Figura 2.50) através do botão “Calibrate” que executa o processo representado no diagrama da Figura 2.49. Este processo coloca a ferramenta de demonstração, presa no manipulador, num ponto conhecido em que se sabe tanto as coordenadas da ferramenta calibrada no robô como o ponto que está a ser adquirido através do marcador luminoso nessa posição. Adquire-se uma medição com o sistema 6DMimic através do marcador luminoso que é guardada num ficheiro .segs e posteriormente lida pela aplicação para que a pose adquirida através do marcador luminoso representada numa matriz de transformação seja utilizada para determinar a matriz que relaciona o referencial do marcador luminoso com o referencial da ferramenta do robô. Este processo é executado inicialmente uma vez e não é necessário repetir desde que o marcador luminoso não seja movido em relação à ferramenta de projeção.

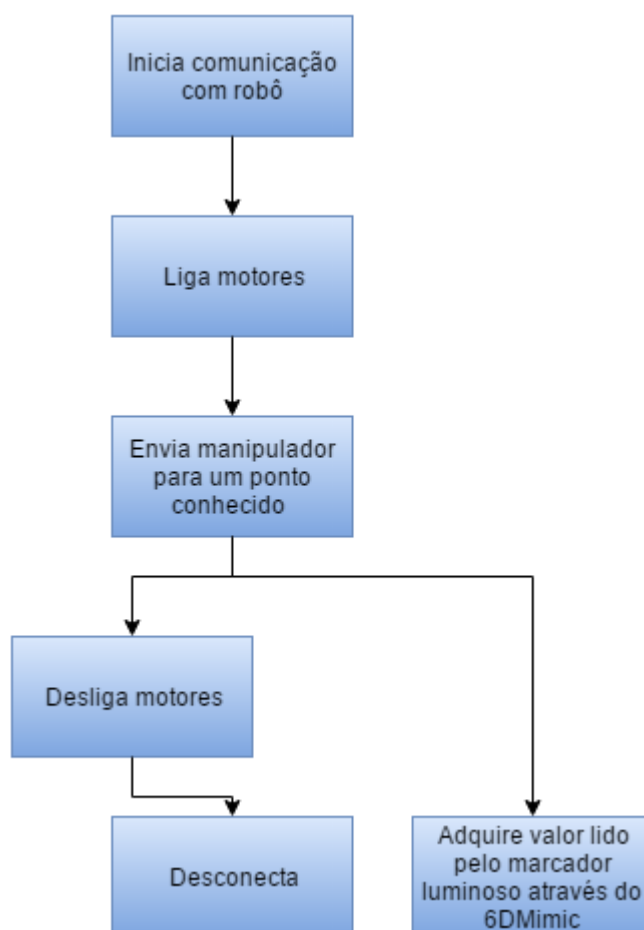


Figura 2.49 - Diagrama respetivo ao processo de calibração da ferramenta de deposição no manipulador.

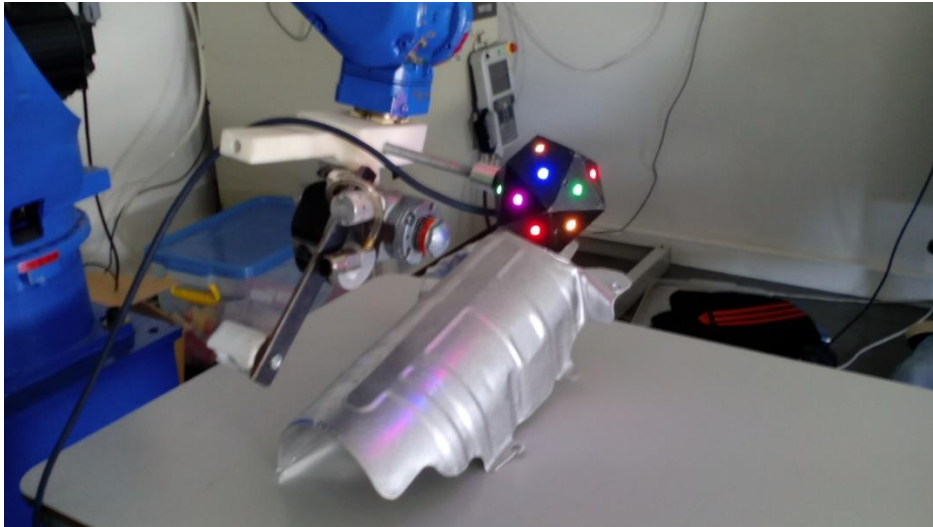


Figura 2.50 - Calibração da ferramenta no robô.

Na fase de demonstração (Figura 2.51), as poses adquiridas através do marcador luminoso estão calibradas no referencial de base do robô e sofrem posterior transformação de modo a serem convertidas num programa para o robô e sejam reproduzidas por este. Todas as poses adquiridas neste processo através da aplicação 6Dmimic com o procedimento descrito no capítulo 2.2.3 são guardadas num ficheiro .segs que é lido por esta aplicação que procede à transformação destas poses através de uma matriz de transformação que relaciona o referencial do marcador luminoso com o referencial da ferramenta do robô. Este procedimento é despoletado pelo botão “Generate Code”.

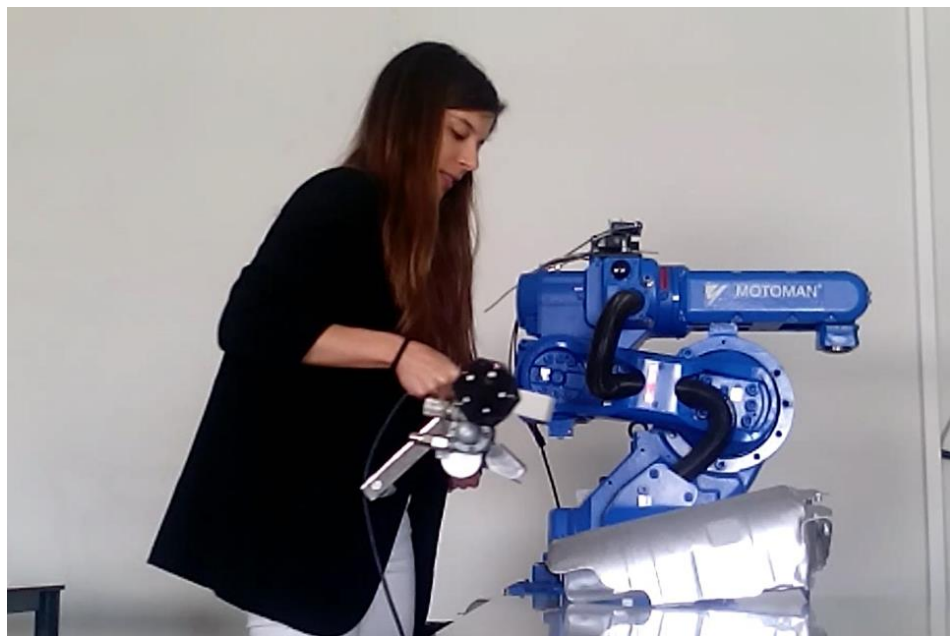


Figura 2.51 - Demonstração de trajetória para projeção de compósito.

O referencial 0 corresponde ao referencial de base do robô, o referencial m corresponde ao referencial com origem no ponto adquirido através do marcador luminoso e o referencial f corresponde ao referencial da ferramenta do robô (Figura 2.52). A pose adquirida através do marcador luminoso expresso no referencial de base do robô pode ser representada pela matriz H_m^0

$$H_m^0 = \begin{bmatrix} R_m^0 & d_m^0 \\ 0 & 1 \end{bmatrix}, \quad (2.3)$$

sendo $R_m^0 = R_m^0_{z,\phi} \cdot R_m^0_{y,\theta} \cdot R_m^0_{x,\psi}$ dada pela orientação da pose e $d_m^0 = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}$ dada pelas coordenadas da pose.

A pose dada pela ferramenta que está calibrada no robô expressa no referencial de base do robô pode ser representada pela matriz H_f^0 .

$$H_f^0 = \begin{bmatrix} R_f^0 & d_f^0 \\ 0 & 1 \end{bmatrix}, \quad (2.4)$$

sendo $R_f^0 = R_f^0_{z,\phi} \cdot R_f^0_{y,\theta} \cdot R_f^0_{x,\psi}$ dada pela orientação da pose e $d_f^0 = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix}$ dada pelas coordenadas da pose.

H_f^m é a matriz de transformação homogénea que relaciona os referenciais do marcador e da ferramenta.

$$H_f^m = (H_m^0)^{-1} \times H_f^0, \quad (2.5)$$

Então a matriz H_f^m corresponde à matriz de calibração que transforma todas as poses adquiridas através do marcador luminoso durante a demonstração em poses da trajetória a ser reproduzida pelo manipulador. Assim, a transformação das poses é feita multiplicando esta matriz pela matriz H_m^0 que é gerada pelas poses que vão sendo adquiridos na demonstração.

$$H_f^0 = H_m^0 \times H_f^m, \quad (2.6)$$

Obtém-se assim a matriz H_f^0 de onde se podem tirar as coordenadas x, y, z e os ângulos de Euler R_x , R_y , R_z ou quaterniões das poses que dão origem à trajetória a ser reproduzida pelo manipulador. Estas poses são guardadas num ficheiro MotomanPoints.segs para serem convertidas num programa para o robô MOTOMAN através da aplicação “CodeGen” descrita no

capítulo 2.2.3. São também guardadas as poses com a orientação em quaterniões e diretamente convertidas num programa para o robô ABB.

Esta fase do projeto que compreende o processo de transformação dos pontos foi realizada em colaboração com o Nuno Rodrigues, no âmbito da sua dissertação cujo tema era "Ensinamento por demonstração com controlo de força e inspeção por visão em células de polimento robotizadas". Desta colaboração, surgiu alguma troca de ideias, que acabaram implementadas da mesma forma em ambos os projetos.

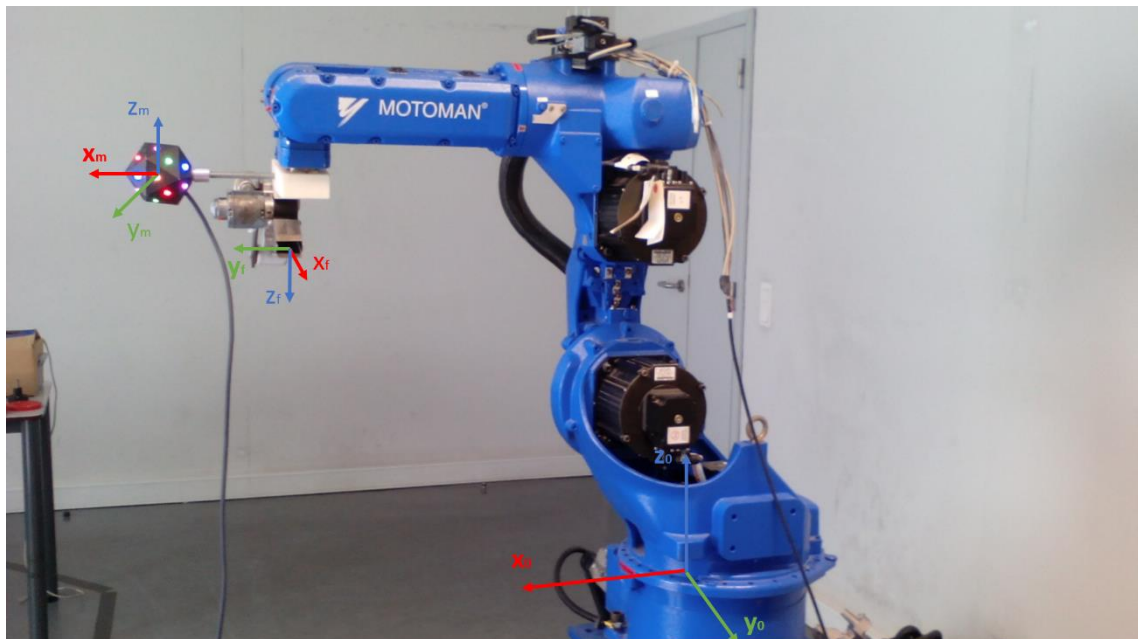


Figura 2.52 - Representação dos referenciais do Sistema robô, ferramenta e marcador luminoso.

De seguida, é efetuado o procedimento "Execute" para que o manipulador execute o programa que contém a trajetória demonstrada e que pode ser descarregado para o robô através da aplicação MotomanCom. Ou o programa pode ser descarregado para o controlador do manipulador ABB através da interface FileZilla.

Algoritmo de processamento dos dados obtidos

O primeiro passo na utilização da interface é o "Start Scan" que faz um *reset* às variáveis que guardam os valores medidos pelo sensor. Assim que se pretender iniciar a digitalização do molde com o sensor deve-se selecionar "Save List of Scans" para serem guardados os pontos lidos num ficheiro ListofScans.txt. Durante a digitalização são também guardadas as poses da ferramenta do robô calibrada na origem do referencial do laser. Estas poses estão expressas no referencial base do robô e são representadas numa matriz de transformação homogênea que ao ser multiplicada aos pontos adquiridos do sensor os transforma também para o mesmo

referencial (Figura 2.53). Para isso, o programa para o manipulador, em RAPID, que está no Anexo B contém já o algoritmo que retém todos os pontos e que no final os guarda num ficheiro PointsFile.txt. É também no programa do manipulador que está incluída uma rotina que define quando o *trigger* é ativado e onde a aquisição de pontos do sensor é sincronizada com a aquisição dos pontos do robô.



Figura 2.53 - Sistema de coordenadas associado ao processo de aquisição de medidas do sensor.

As poses adquiridas do manipulador podem conter informação das coordenadas x , y , z e do quaternião que indica a orientação da ferramenta ou das coordenadas x , y e z e dos ângulos de Euler.

É necessária a representação das poses em matrizes de transformação de referenciais e como foi considerada a utilização deste algoritmo tanto para uma situação como para outra, foram consideradas as duas topologias e respetivas conversões.

Os quaterniões são uma das ferramentas mais usadas para definir orientações. Eles são uma extensão dos números complexos no espaço 4D: $q=[q_r, q_i, q_j, q_k]^T$. Os quaterniões unitários representam orientações no espaço 3D.

A passagem de ângulos de Euler para matriz de rotação, pode ser efetuada da seguinte forma:

$$R_{z,\phi} = \begin{bmatrix} \cos \phi & -\text{sen} \phi & 0 \\ \text{sen} \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

$$R_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \text{sen} \theta \\ 0 & 1 & 0 \\ -\text{sen} \theta & 0 & \cos \theta \end{bmatrix}, \quad (2.8)$$

$$R_{x,\psi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\text{sen}\psi \\ 0 & \text{sen}\psi & \cos\psi \end{bmatrix}, \quad (2.9)$$

Que culmina na matriz de rotação (utilizada nos manipuladores da Motoman):

$$R = R_{z,f} \times R_{y,\theta} \times R_{x,\psi}, \quad (2.10)$$

$$R = \begin{bmatrix} \cos(f)\cos(\theta) & \cos(f)\sin(\theta)\sin(\psi) - \sin(f)\cos(\psi) & \cos(f)\sin(\theta)\cos(\psi) + \sin(f)\sin(\psi) \\ \sin(f)\cos(\theta) & \sin(f)\sin(\theta)\sin(\psi) + \cos(f)\cos(\psi) & \sin(f)\sin(\theta)\cos(\psi) - \cos(f)\sin(\psi) \\ -\sin(\theta) & \cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) \end{bmatrix}, \quad (2.11)$$

A conversão de quaterniões ($q = [q_r, q_i, q_k]^T$) para matriz de rotação (utilizada nos manipuladores da ABB) corresponde a:

$$R = \begin{bmatrix} q_r^2 + q_i^2 - q_j^2 - q_k^2 & 2q_iq_j - 2q_rq_k & 2q_rq_j + 2q_iq_k \\ 2q_rq_k + 2q_iq_j & q_r^2 - q_i^2 + q_j^2 - q_k^2 & 2q_jq_k - 2q_rq_i \\ 2q_iq_k - 2q_rq_j & 2q_rq_i + 2q_jq_k & q_r^2 - q_i^2 - q_j^2 + q_k^2 \end{bmatrix}, \quad (2.12)$$

A relação entre o referencial da ferramenta (sensor laser) e o referencial de base do manipulador é dada pela matriz de transformação homogénea H_s^m .

$$H_s^m = \begin{bmatrix} R_s^m & d_s^m \\ 0 & 1 \end{bmatrix}, \quad (2.13)$$

Sendo $d_s^m = \begin{bmatrix} x_s^m \\ y_s^m \\ z_s^m \end{bmatrix}$ dado pelas coordenadas do ponto central da ferramenta do manipulador

(sensor laser) e R_s^m dado pela orientação do referencial dessa mesma ferramenta.

Os pontos no referencial de base do manipulador são calculados a partir dos pontos no referencial da ferramenta utilizando-se a equação 2.14.

$$p^m = H_s^m \times p^s, \quad (2.14)$$

Feita a conversão dos pontos através da operação “SAVE CLOUD” que executa um algoritmo que transforma os pontos adquiridos pelo sensor expressos no seu referencial s em pontos expressos no referencial m de base do robô (Figura 2.54), consegue-se construir uma nuvem de pontos que corresponde à superfície da peça digitalizada como se pode ver na Figura 2.55.

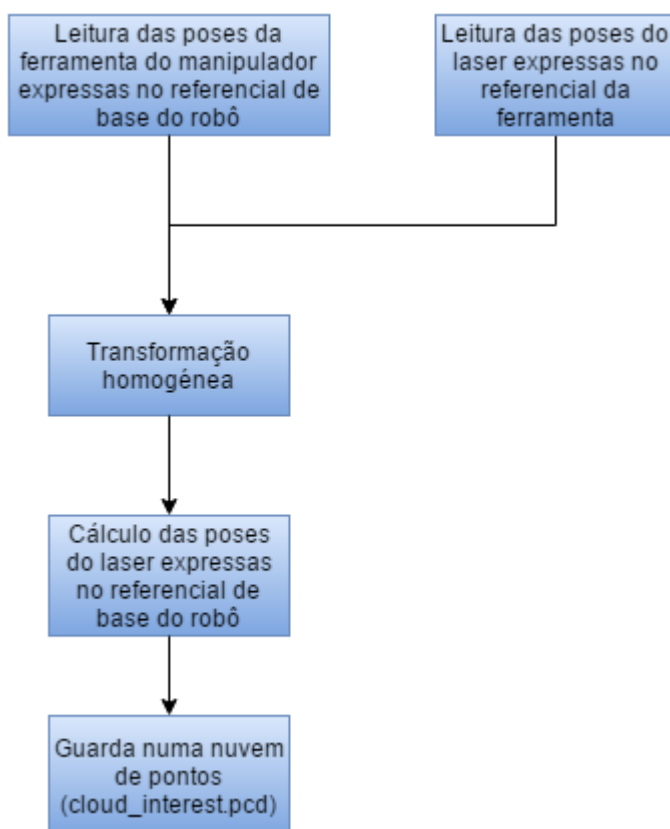


Figura 2.54 - Diagrama respetivo à transformação dos pontos para gerar uma nuvem do molde.

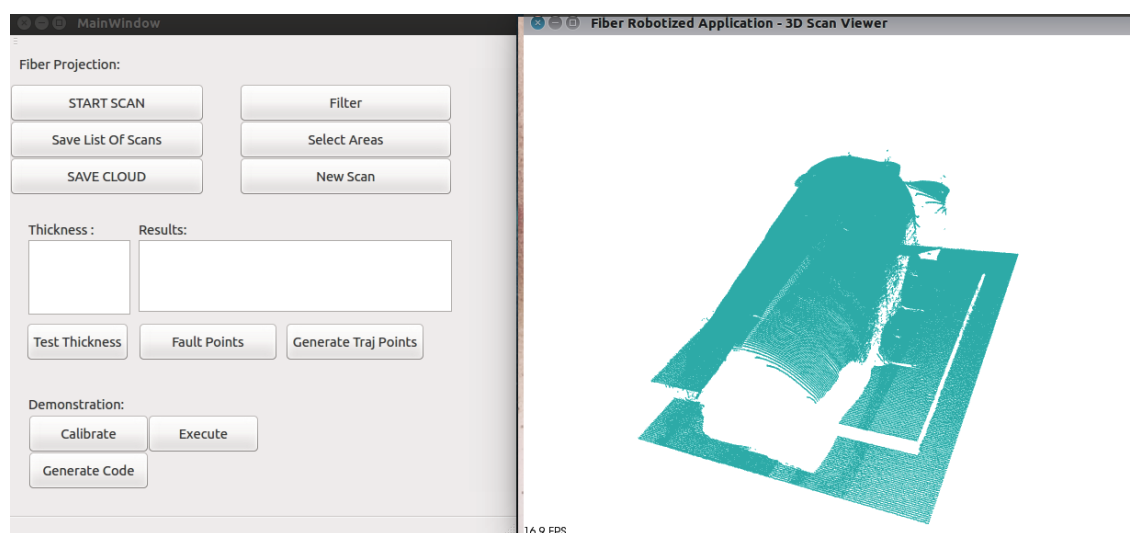


Figura 2.55 - Visualização da nuvem de pontos gerada após transformação dos pontos adquiridos pelo sensor.

Após a obtenção da nuvem de pontos representativa do molde, verifica-se que o varrimento do laser associa erros às medições e que mantém a estrutura de suporte, o que complica a estimativa das características da nuvem de pontos.

Este tipo de erros podem ser eliminados através da realização de uma análise estatística sobre cada ponto em relação aos seus vizinhos sendo eliminados aqueles que não satisfazem

determinados critérios. Então, de seguida prevê-se a execução do processo “Filter” representado na Figura 2.60 para a aplicação de filtros à nuvem de pontos. A filtragem feita na secção “Remoção de *Outliers*” da Figura 2.60, baseia-se no número de vizinhos de cada ponto fazendo-se uma pesquisa num raio de 7 mm ao seu redor usando uma técnica que recorre a métodos de pesquisa na nuvem descritos acima no Capítulo 1.4.5. Neste caso, cada ponto pesquisado tem que ter no mínimo 27 vizinhos num raio de 7mm. Estes valores foram determinados com base nos resultados obtidos, depois de se experimentarem várias combinações de raio e número de vizinhos. Sendo esta a que culminou numa nuvem com menor número de *outliers*. Como se pode ver na Figura 2.56 onde foi usado um número demasiado baixo de vizinhos mas o mesmo raio de pesquisa de 7mm, muitos *outliers* não foram removidos, como se verifica nas zonas assinaladas com círculos. A Figura 2.57 foi obtida usando maior número de vizinhos mas o mesmo raio de pesquisa e conclui-se que foram removidas partes da superfície da peça importantes para posterior processamento.

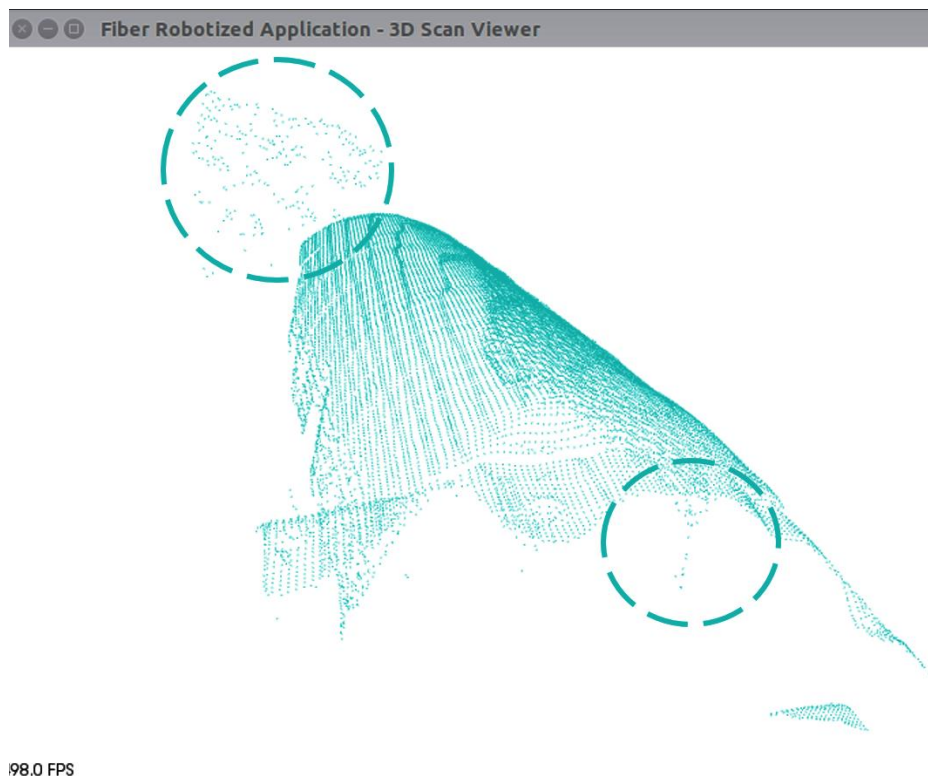


Figura 2.56 - Filtro de remoção de *outliers* aplicado com raio de pesquisa 7mm e testando 5 vizinhos.

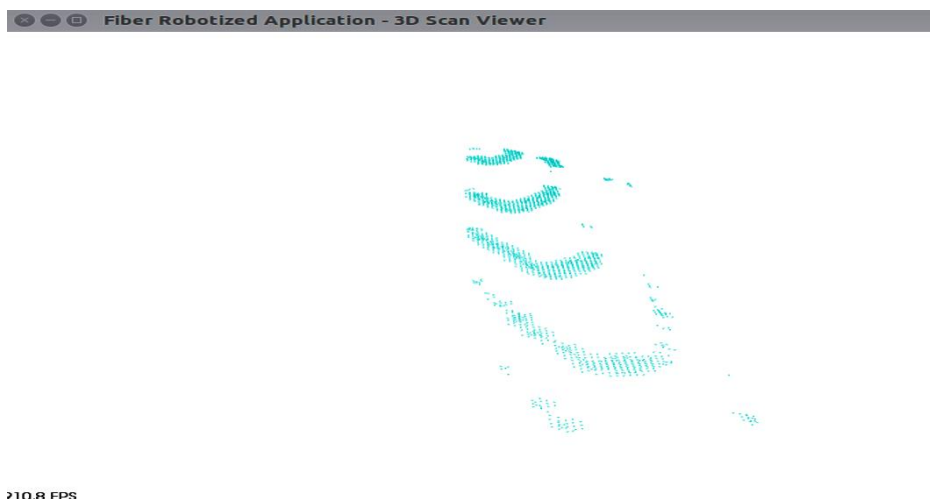


Figura 2.57 - Filtro de remoção de *outliers* aplicado com raio de pesquisa 7mm e testando 50 vizinhos.

O raio de pesquisa escolhido baseia-se na resolução dos dados adquiridos do sensor, ou seja, tendo em conta a distância entre uma aquisição e outra deve-se escolher um raio que contenha alguns pontos.

Foi, também, testado um raio de pesquisa de 4mm e menor número de vizinhos. A hipótese de se baixar ou até aumentar tanto o valor do raio de pesquisa como o número de vizinhos mínimo nesse raio seria também viável, no entanto neste teste feito verificaram-se pequenos conjuntos de pontos distantes do resto da nuvem (Figura 2.58). A Figura 2.59 foi obtida utilizando-se um raio de 4mm também mas um maior número de vizinhos do que no caso anterior concluindo-se que algumas áreas da superfície do molde eram removidas.

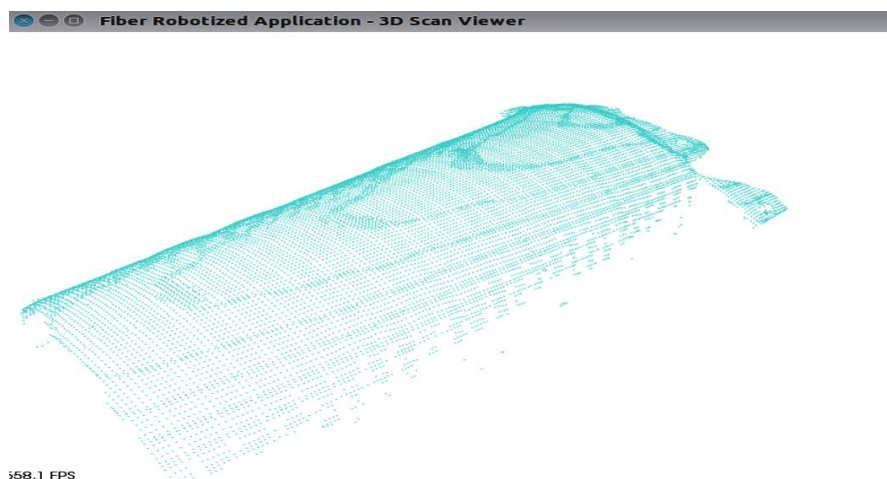


Figura 2.58 - Filtro de remoção de *outliers* aplicado com raio de pesquisa 4mm e testando 7 vizinhos.

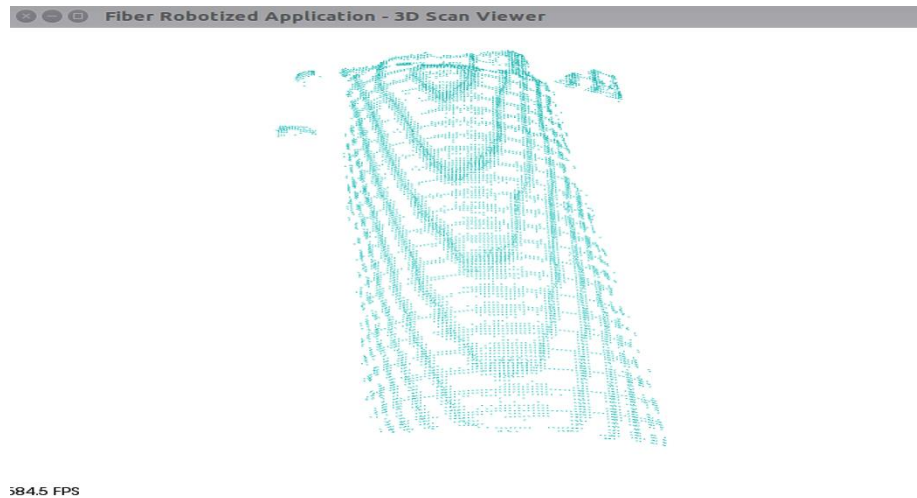


Figura 2.59 - Filtro de remoção de *outliers* aplicado com raio de pesquisa 4mm e testando 10 vizinhos.

A aquisição dos dados com o laser posicionado em ângulos diferentes introduz um erro nas medições que é visível em zonas em que a digitalização é sobreposta e das quais resultam distâncias diferentes. Para minimizar este erro, utilizou-se um filtro de grelha voxel. Este filtro, intitulado de “Redução da resolução” na Figura 2.60, cria uma partição do espaço sobre a superfície da peça em pequenos voxels que se caracterizam por pequenos volumes 3D de 2x2x8 mm. Estas dimensões foram escolhidas por permitirem que cada voxel inclua maioritariamente pontos com coordenada z afastada mas com coordenadas x e y muito próximas, ou seja, pontos que remetem para a mesma zona da superfície mas que por erro de medição têm coordenadas z bastante diferentes. De seguida, os pontos contidos em cada voxel são aproximados pelo seu centróide, diminuindo a resolução da nuvem de pontos.

Para que se removam pontos que fazem parte da estrutura de apoio à peça, fez-se uma remoção dos pontos, intitulada no diagrama da Figura 2.60 de “Remoção Condicional”, que pertencem à condição estabelecida com as cotas correspondentes à estrutura de apoio que como se pode ver no diagrama são cotas maiores do que 20mm. Ou seja, sabe-se que distâncias maiores do que 20 mm correspondem à estrutura que apoia a peça. A nuvem de pontos resultante da aplicação destes filtros está presente na Figura 2.61.

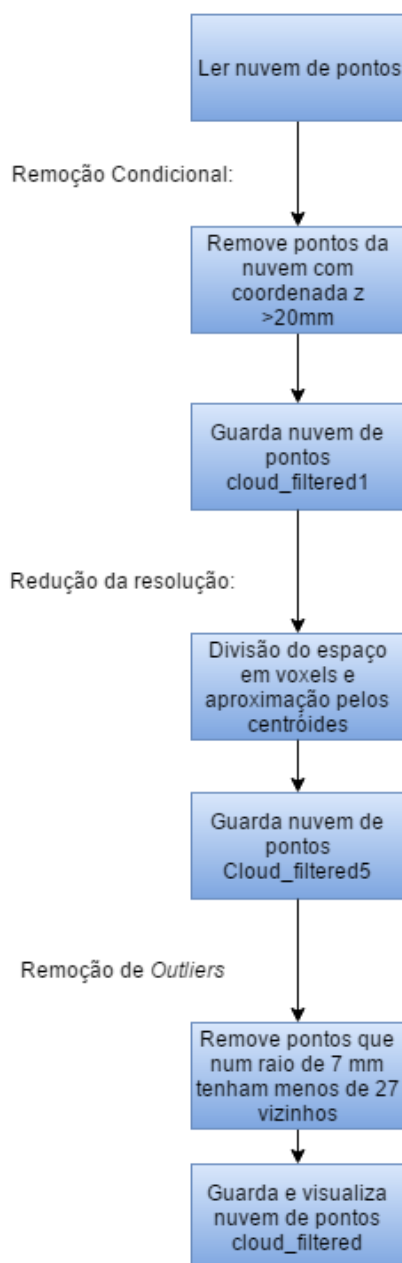


Figura 2.60 - Diagrama que representa a aplicação de filtros à nuvem de pontos.

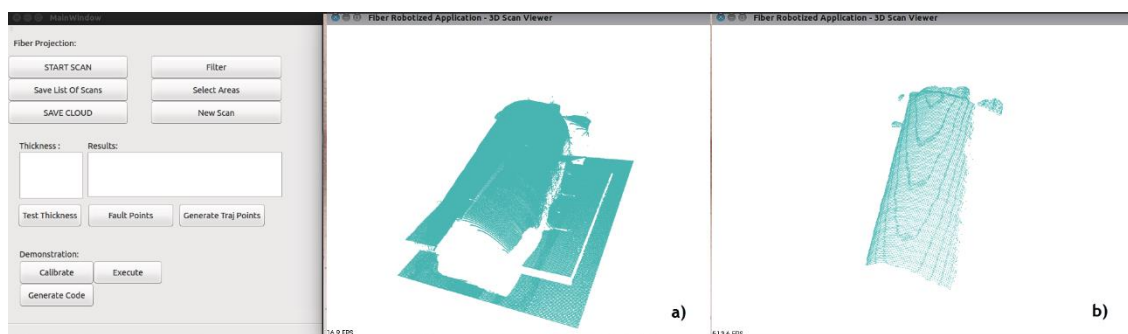


Figura 2.61 - Visualização da nuvem de pontos gerada b) após a aplicação de filtros à nuvem original a).

Feita esta aquisição dos dados respetivamente ao molde, programa-se o manipulador para que este execute uma trajetória que cubra todo o molde. A programação é feita através do processo de programação por demonstração descrito anteriormente. A demonstração da trajetória pode ser feita em qualquer altura, no entanto é nesta altura que o robô deve executar os movimentos tendo a ferramenta de projeção presa na extremidade tal como na Figura 2.62.

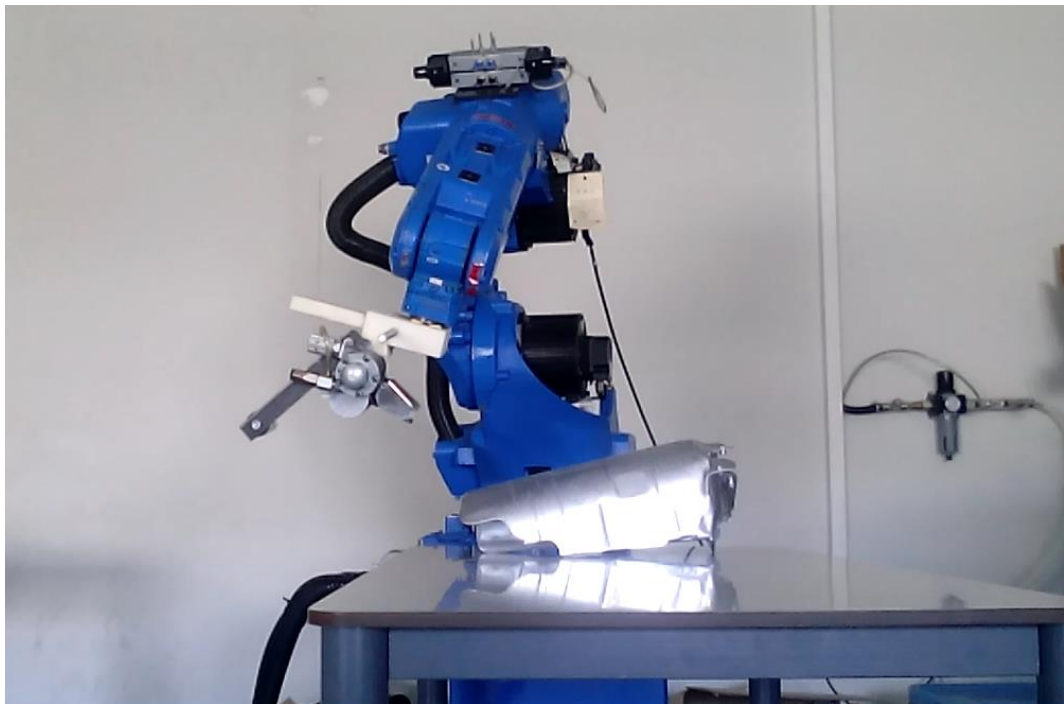


Figura 2.62 - Imagem do processo de reprodução da trajetória demonstrada.

Para se simular a deposição de compósito no molde, criou-se uma camada de um material esponjoso com aproximadamente 3 mm como se pode ver na Figura 2.63.



Figura 2.63 - Molde utilizado para teste do algoritmo antes e depois da deposição de material.

Para que o utilizador pudesse dividir a nuvem em áreas de interesse e associar-lhes espessuras que pretende obter, foi desenvolvido um algoritmo que divide a nuvem geral de pontos da peça em n nuvens conforme a divisão solicitada e que está descrito na Figura 2.64. Permite como *input* valores de espessura em mm correspondentes a cada área que o utilizador pode seleccionar na janela de visualização da nuvem de pontos. A seleção pelo utilizador é feita pressionando Shift + seleção com o rato na janela de visualização.

Com essas coordenadas, divide a peça ao longo do eixo dos yy (Figura 2.65) através de um filtro passa banda¹⁰ guardando os pontos em estruturas distintas. Este filtro, selecciona os pontos de uma banda de valores e guarda-os numa estrutura de dados, de seguida selecciona os pontos que surgem na banda de valores imediatamente seguinte e guarda noutra estrutura e assim sucessivamente até que a nuvem de pontos fica dividida e armazenada em várias nuvens para serem tratadas individualmente.

¹⁰ <http://pointclouds.org/documentation/tutorials/passthrough.php>

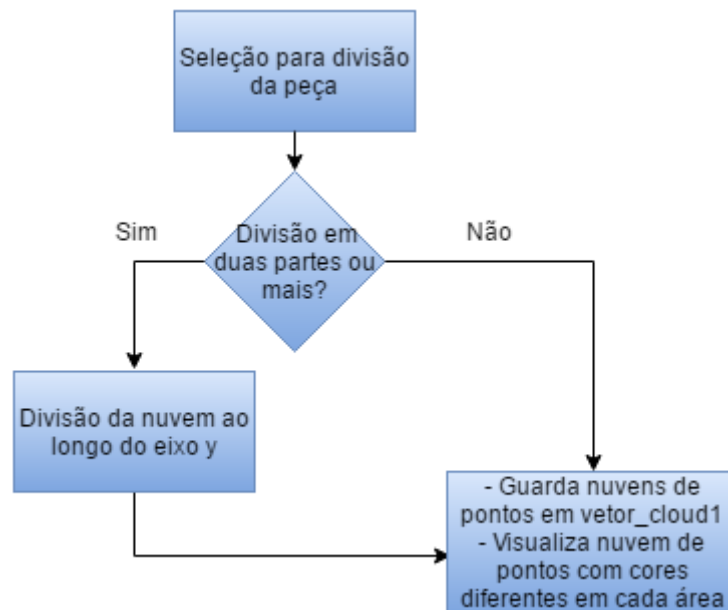


Figura 2.64 - Diagrama que descreve o processo de divisão da nuvem de pontos por áreas.

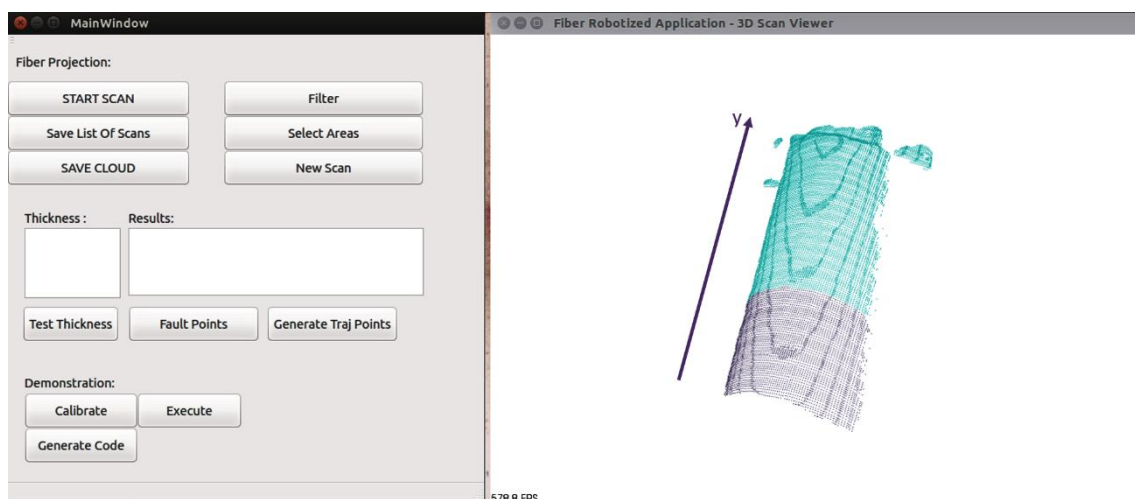


Figura 2.65 - Visualização da nuvem de pontos gerada após seleção de áreas por parte do utilizador.

De seguida, foi feita uma nova digitalização da peça para avaliar a espessura do material depositado. Para representação da nuvem de pontos adquirida nesta condição, é usado o botão “New Scan” que executa o algoritmo descrito na Figura 2.66 e que mostra a visualização dos novos dados já submetidos a uma filtragem e com a nuvem dividida por áreas como se pode ver na Figura 2.67 a rosa e verde. O processo de filtragem e divisão da nuvem é o mesmo descrito e explicado anteriormente a que se submeteu a nuvem que corresponde à digitalização do molde. Ou seja, submete-se a nuvem gerada por nova digitalização da peça também ao filtro de remoção de *outliers*, à remoção condicional para retirar cotas que correspondem à estrutura de apoio, redução de resolução e posterior divisão em áreas para que tanto esta nuvem como a nuvem anteriormente processada estejam nas mesmas condições para serem comparadas com

o intuito de calcular as distâncias entre elas. Estas distâncias corresponderão à espessura de material que foi depositado posteriormente à primeira digitalização feita.

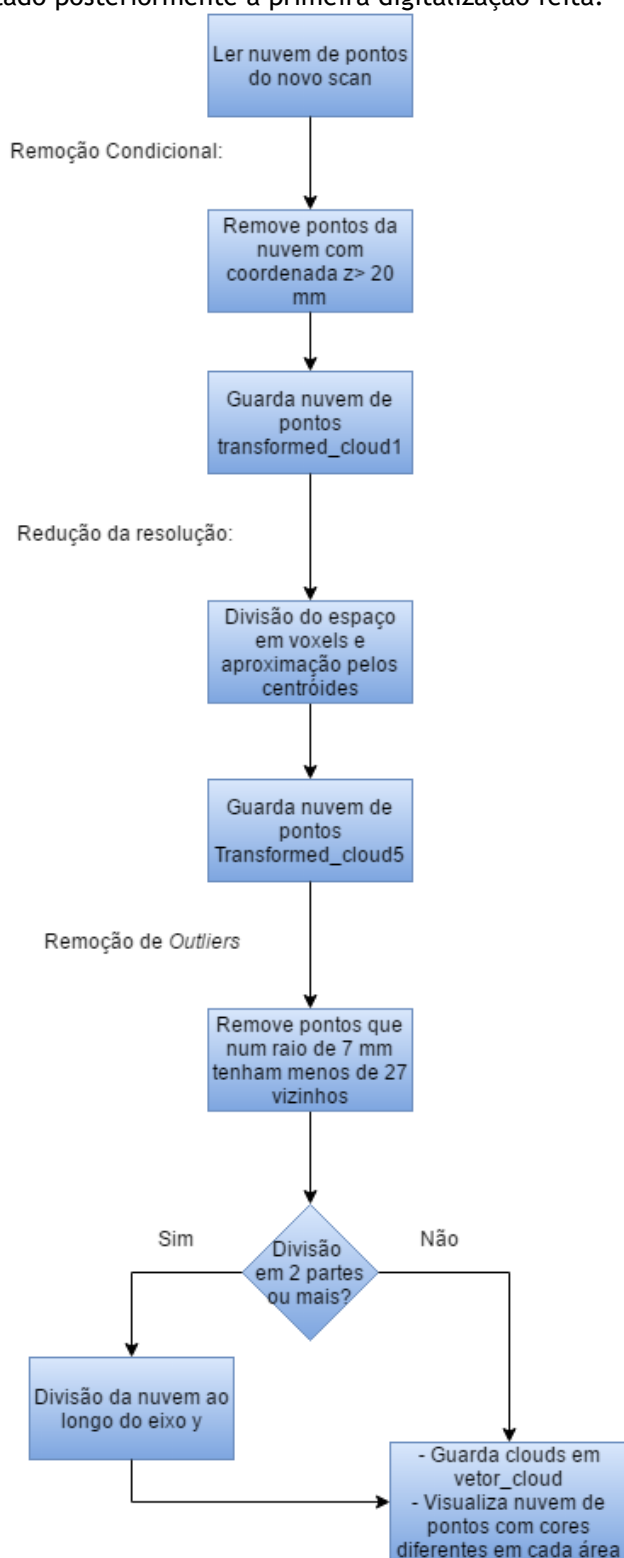


Figura 2.66 - Diagrama que descreve a aquisição de nova digitalização após deposição de material e posterior processamento dos dados obtidos.

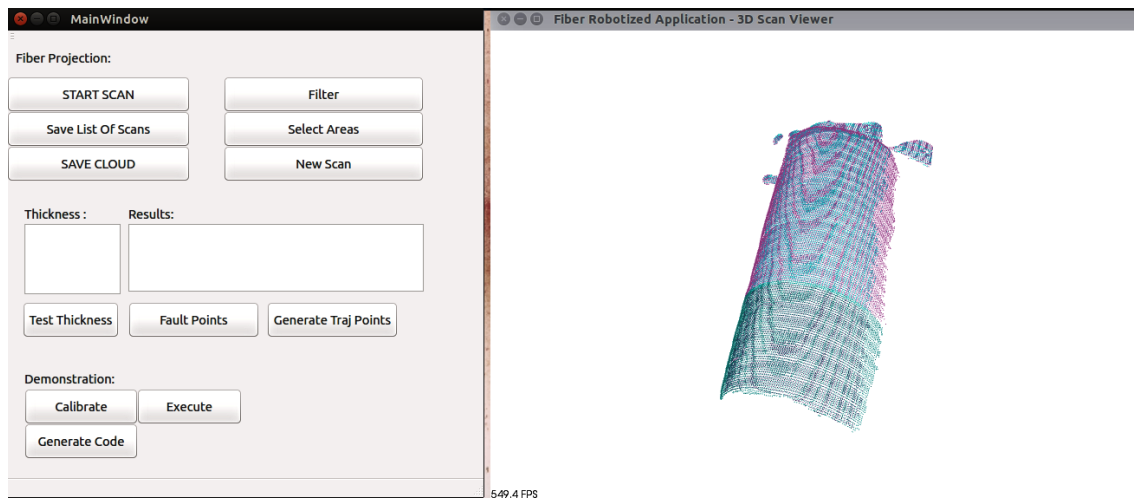


Figura 2.67 - Visualização da nuvem de pontos gerada após nova digitalização.

A espessura de material depositado é calculada com base na distância entre o *scan* feito ao molde e o *scan* feito após a deposição. Para cada área da peça é determinada a espessura de forma a que a espessura mínima encontrada seja comparada com a espessura introduzida pelo utilizador no campo “Thickness” (Figura 2.68), seguido do botão “Test Thickness” que executa o procedimento para cálculo da espessura e faz avaliação da espessura depositada como sendo “Boa” ou “Insuficiente”.

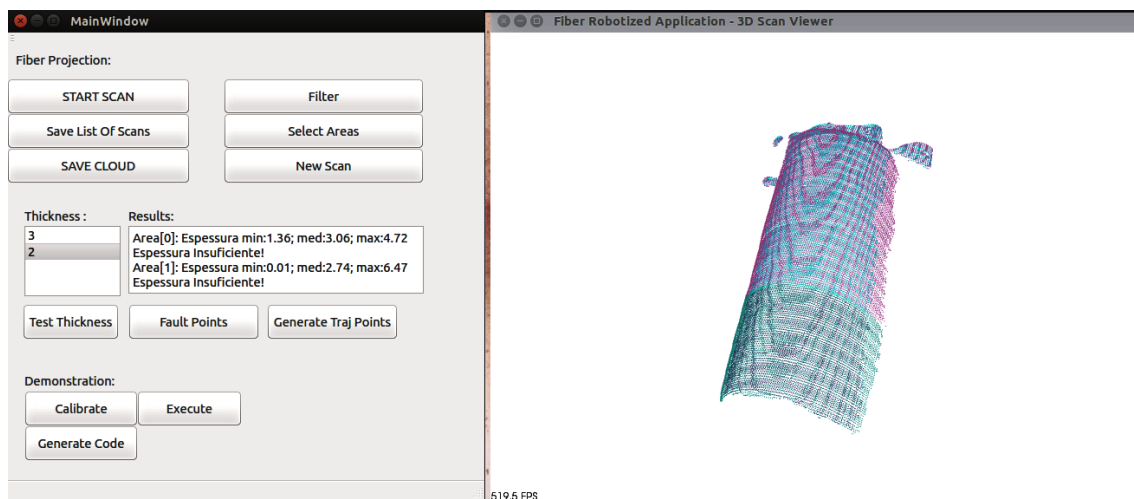


Figura 2.68 - Visualização dos valores de espessura calculados e introduzidos pelo utilizador.

Foram testados para cálculo da espessura, dois algoritmos que detetavam a distância entre as duas nuvens de pontos.

Cálculo da espessura (algoritmo 1): Inicialmente começou-se por se implementar um algoritmo de cálculo da distância entre as duas nuvens de pontos através de um método de pesquisa do ponto mais próximo de uma em relação à outra. O método de Hausdorff é utilizado para calcular a distância entre pontos de dois conjuntos distintos, X e Y e considera que todos

os pontos do conjunto X têm uma correspondência de, pelo menos um ponto do conjunto Y. A distância de Hausdorff pode ser calculada como a distância MAX-MIN de X para Y, isto é encontrar o máximo da distância a partir de um elemento em X para o seu vizinho mais próximo em Y. Esta abordagem utiliza uma varredura linear ao longo de X e utiliza um índice para ajudar a calcular o vizinho mais próximo em Y. A partir desta pesquisa são gerados pares de correspondências e são explorados os índices dos dois conjuntos, melhorando o desempenho em termos de pesquisa na nuvem de pontos e do tempo total de resposta. A pesquisa é feita de X para Y e de Y para X para que a máxima distância seja o máximo calculado nos dois sentidos.

Cálculo da espessura (algoritmo 2): O algoritmo que de facto foi usado para calcular a espessura depositada percorre a superfície da nuvem correspondente ao molde previamente adquirido, vai pesquisando para cada área de 60 pontos qual a normal a essa área (Figura 2.69) e nessa direção é calculada a distância entre as duas nuvens. Definiu-se 60 pontos porque fazendo vários testes se revelou o valor que permite determinar correspondências na direção normal com menor número de correspondências erradas. Como se pode ver na Figura 2.70 em a) foram usados 5 pontos para calcular a normal à superfície por eles gerada e em b) foram usados 60 pontos tendo-se obtido melhores resultados.

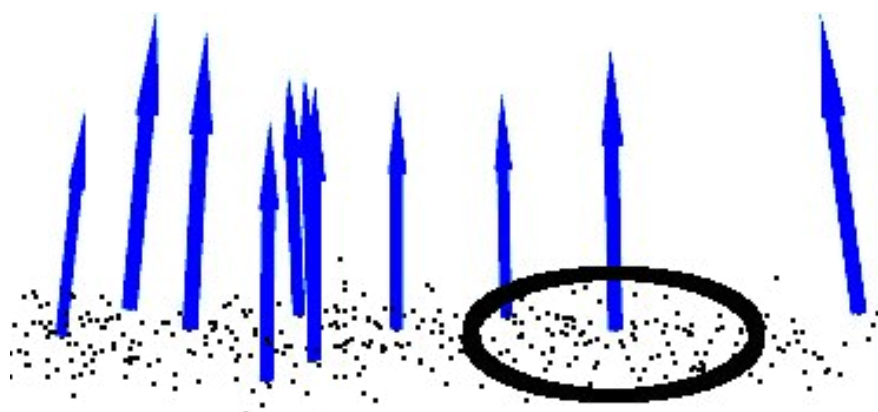


Figura 2.69 - Representação da normal (a azul) a uma superfície de pontos contidos numa certa área (rodeada por circunferência a preto).

A nuvem de partida é percorrida através de um algoritmo de pesquisa que utiliza uma estrutura de dados em árvore descrita anteriormente no capítulo 1.4.5. Da pesquisa na direção normal à área selecionada do molde são encontradas 10 correspondências entre as nuvens, das quais a melhor é devolvida como sendo o ponto pesquisado mais próximo nessa direção. Este valor de 10 correspondências foi usado por ser um valor por defeito sugerido no código fonte da função que calcula as correspondências entre nuvens na direção normal¹¹.

¹¹ <https://searchcode.com/codesearch/view/30555415/>

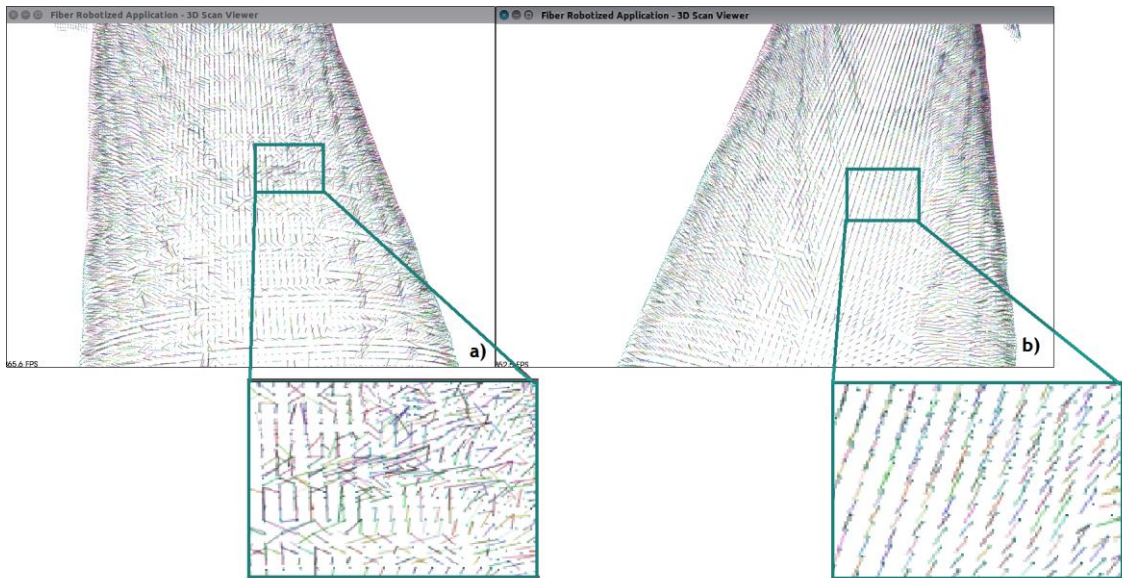


Figura 2.70 - Comparação das correspondências encontradas entre as duas nuvens usando uma área de 5 pontos para calcular as normais a) e usando uma área de 60 pontos b).

Foi utilizado o algoritmo 2 pois o primeiro não considerava, em situações como as da Figura 2.71 a distância na direção normal mas sim a menor distância. Sendo a linha preta o molde, a linha azul a camada de compósito, a linha verde a área de pontos dos quais se vai calcular a normal para se pesquisar nessa direção o ponto mais próximo da nuvem da camada de compósito, a linha amarela a distância da área a (que envolve pontos do molde) ao ponto c (ponto na camada de compósito) e a linha vermelha a distância da área a (que envolve pontos do molde) ao ponto b (ponto na camada de compósito), verificou-se que o algoritmo 1 assumia que a distância entre o molde e a camada depositada era a da linha amarela pois constata-se que é o vizinho mais próximo de a na superfície azul, enquanto que o algoritmo 2 considerava a superfície da camada de compósito na direção normal ao ponto correspondente no molde, ou seja, a distância representada pela linha vermelha.

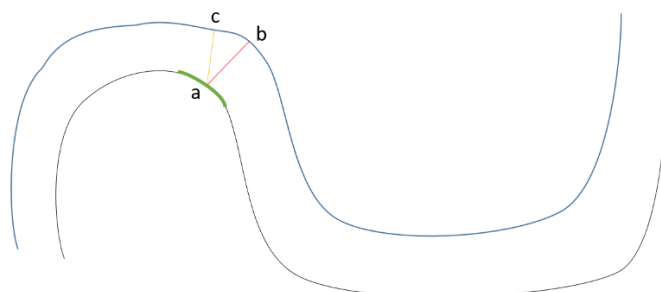


Figura 2.71 - Representação de situação de deposição sendo o molde a linha preta e a camada de compósito a linha azul e as linhas amarela e vermelha, duas distâncias.

A função que devolve as melhores correspondências ao longo de toda a nuvem permite o acesso à distância entre as superfícies, o que equivale ao valor das espessuras de material depositado. Através desses valores, calcula-se o valor de espessura mínimo, máximo e médio para cada partição das nuvens. Cada partição é percorrida e para cada distância calculada testa-se se o valor mínimo encontrado é menor que o valor atual calculado, caso não seja atualiza-se o valor mínimo. O mesmo se testa para o valor máximo encontrado, se este não for maior do que o valor atualmente calculado, atualiza-se esse valor máximo. No final, é calculada a média de todas as distâncias encontradas em cada parte de divisão da nuvem.

Para se garantir que as espessuras mínimas e máximas encontradas em cada parte de divisão das nuvens, não são falsos mínimos nem falsos máximos testa-se se existem mais de 60 pontos com correspondências com valores de distâncias num intervalo que contém o valor de espessura mínimo com uma tolerância de 1mm e o mesmo se testa para o valor máximo de espessura. Definiu-se 60 pontos por ser o valor que permite obter melhores resultados em vários testes feitos com valores diferentes. Valores maiores iriam descartar um número de pontos significativos da peça com espessura mínima criando uma inflação no valor de espessura mínima e valores menores iriam detetar pontos que erradamente foram associados a espessuras mínimas ou que não é relevante serem detetados para esta aplicação. Como se pode ver na Figura 2.72 b), o valor da espessura mínima foi alterado para a área [0], ou seja, detetou-se mais de 5 correspondências com o valor da espessura mínima mas sabe-se que menos de 60 correspondências, comparando com a Figura 2.72 a). O valor de espessura mínima é então um falso mínimo porque não tem um valor relevante de correspondências com esse valor. Pode-se verificar na realidade que a zona pertencente a esta área da peça que tem um valor de 0.12mm de espessura não é suficiente para ser considerado.

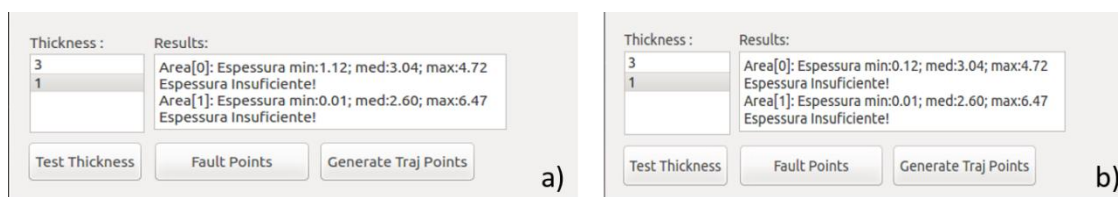


Figura 2.72 - Comparação dos resultados obtidos utilizando um mínimo de 60 pontos a) e 5 pontos b) para teste do intervalo de valores das distâncias calculadas.

Esta rotina foi necessária para diminuir a inflação no valor das espessuras devido ao facto de as duas nuvens em comparação não terem o mesmo número de pontos e, por vezes, as correspondências na camada de compósito se afastarem um pouco das normais à superfície do molde como se pode ver na Figura 2.73.

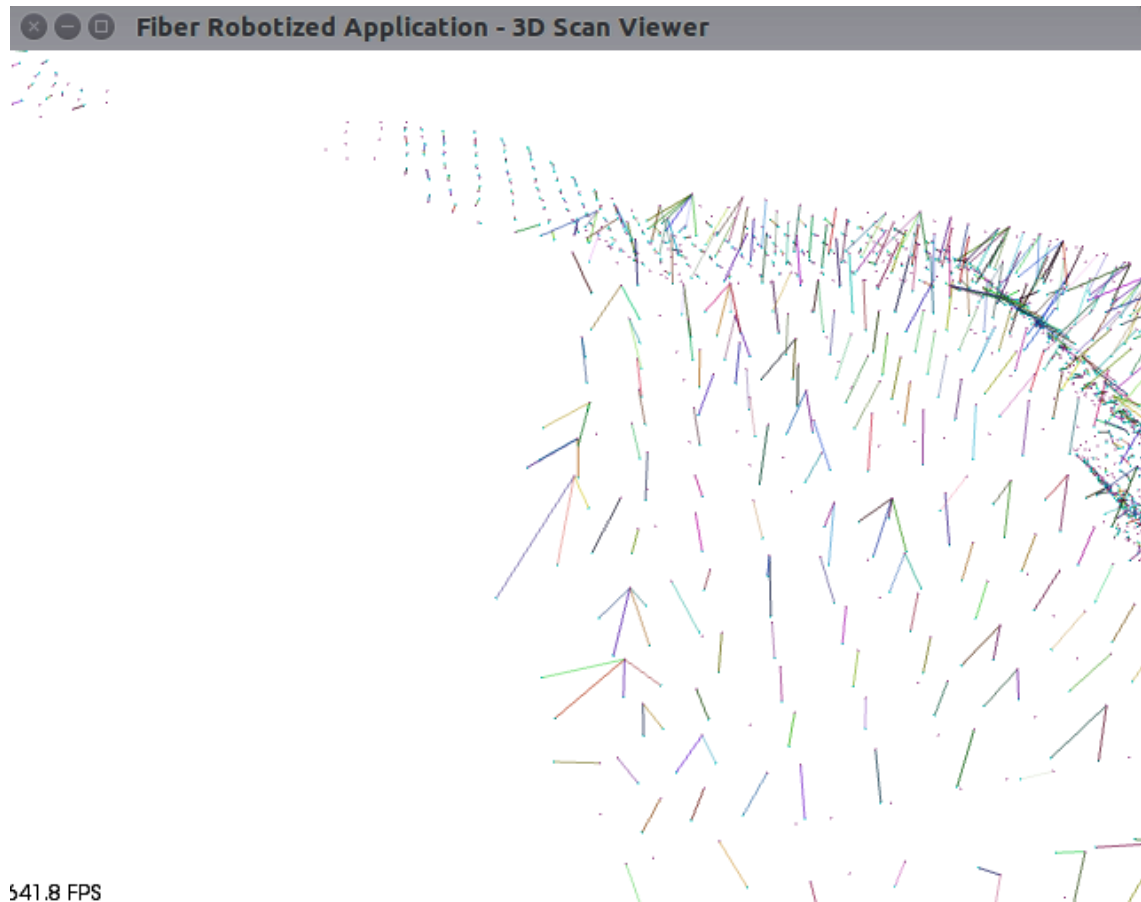


Figura 2.73 - Pormenor de região em que as correspondências são feitas entre pontos que se afastam da direção normal.

Na interface, o utilizador obtém a informação sobre quais as espessuras mínimas, médias e máximas em cada região da peça no campo “Results”. No exemplo ilustrado na Figura 2.68 foi introduzido pelo utilizador uma espessura de 2 mm para a primeira área definida anteriormente e 3mm para a segunda. Posto isto, o utilizador deve testar a espessura através de “Test Thickness” que despoleta o algoritmo que compara a espessura introduzida com a espessura mínima encontrada em cada partição. Esta comparação é feita entre a espessura introduzida pelo utilizador e a espessura mínima verificada pois é importante que toda a peça seja coberta pelo menos com a espessura desejada pelo utilizador. O facto da espessura depositada ultrapassar o valor desejado não é relevante.

Os resultados obtidos depois de ser executado o algoritmo associado a “Test Thickness” foram que, na primeira área, a espessura média é de 3.06 mm e na segunda área de 2.74 mm tal como seria expectável visto que a fita depositada tinha aproximadamente 3 mm de espessura. Mostra também que a espessura mínima é de 1.36 mm devido a pontos de emenda onde a fita não cobriu totalmente o molde. Na segunda área, a espessura mínima é de 0 mm aproximadamente pois como se pode ver na Figura 2.63 há uma parte do molde que não foi

coberto. As espessuras introduzidas pelo utilizador foram comparadas com as espessuras mínimas em cada área e culminou num resultado de “Espessura insuficiente” para as duas.

Mediante as correspondências com valores de espessura insuficiente avalia-se se existem regiões com pontos suficientes para se considerar a necessidade de um reforço nessa zona recorrendo ao algoritmo descrito na Figura 2.77. Faz-se uma pesquisa num raio de 5mm a cada ponto e testa-se se nessa área têm 5 vizinhos. Isto significa que existem à sua volta pelo menos 5 pontos o que garante que existe ali uma área que justifica reforço de deposição de compósito (Figura 2.75). O facto de se ter estabelecido 5 pontos deve-se a testes feitos que revelaram que valores acima deste não detetariam áreas significativas e valores abaixo detetariam áreas que não se justificam serem reforçadas para esta aplicação. Evita também que se faça reforço em pontos isolados. O facto de se fazer a pesquisa num raio de 5 mm é precisamente porque há necessidade da pesquisa ser feita num raio pequeno, pois com um valor maior corre-se o risco de pesquisar em regiões de espessura insuficiente vizinhas que não estariam necessariamente agrupadas. Como se pode ver na Figura 2.74 em a) considerou-se que se mantém pontos com 15 vizinhos também com espessuras insuficientes e perdeu-se uma área da primeira partição da nuvem que necessita de reforço, comparando com a Figura 2.75 e em b) considerou-se que se mantém pontos com pelo menos 1 vizinho também com espessura insuficiente e apareceram pequenas áreas na zona azul da nuvem que não é relevante que haja reforço.

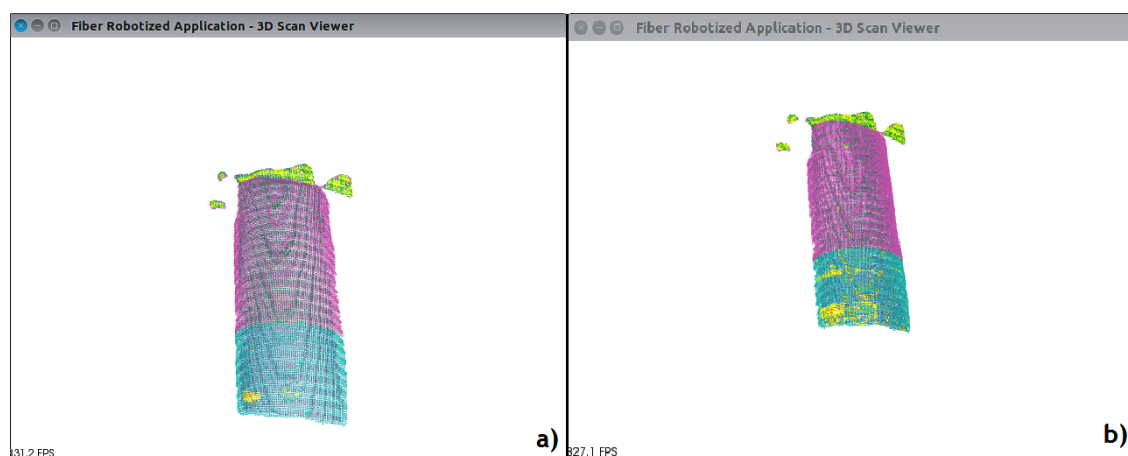


Figura 2.74 - Comparação da seleção de pontos com espessuras insuficientes considerando 15 vizinhos a) e considerando 1 vizinho b).

As áreas que necessitam de reforço aparecem destacadas assim que se seleciona o botão “Fault Points” tal como se pode ver na Figura 2.75.

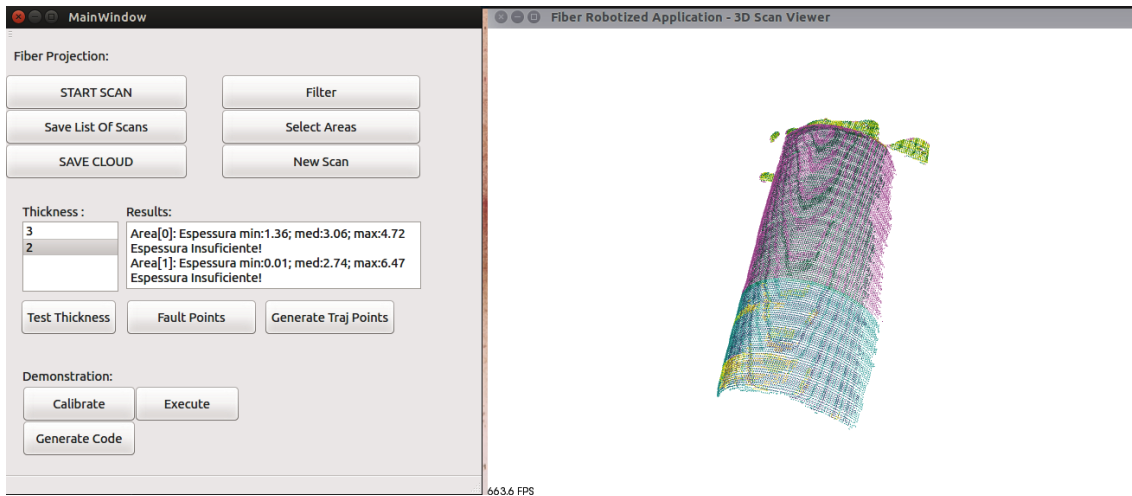


Figura 2.75 - Visualização da nuvem com as áreas de espessura insuficiente destacadas a amarelo.

São depois selecionados através de “Generate Traj Points” os pontos da trajetória do robô que estão mais próximos destas áreas para que este se possa deslocar aos pontos que necessitam de reforço. Esta seleção é feita através de uma pesquisa na nuvem que contém pontos que correspondem a regiões com espessura em falta em que para cada ponto dessa nuvem é detetado o vizinho mais próximo no conjunto de pontos da trajetória de deposição do robô através do método utilizado anteriormente para determinar correspondências entre as superfícies (Figura 2.78). Obteve-se, assim, o resultado apresentado na Figura 2.76 em que se verifica que os pontos destacados a azul no espaço por cima da superfície da peça estão localizados na direção das zonas que foram destacadas a amarelo anteriormente na Figura 2.75. Esses pontos correspondem a pontos selecionados da trajetória de deposição do robô.

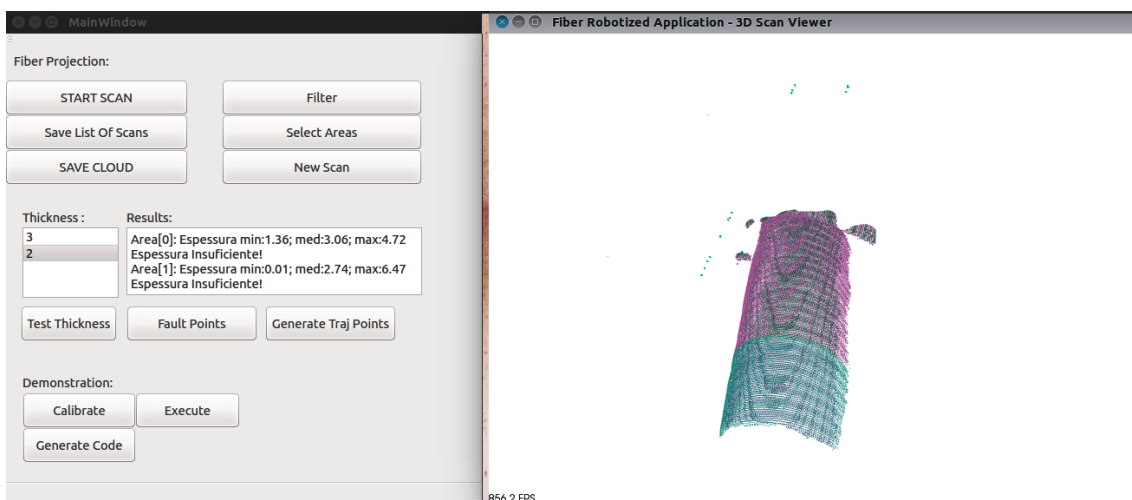


Figura 2.76 - Visualização dos pontos a azul da trajetória do robô para reforço em zonas com espessura insuficiente.

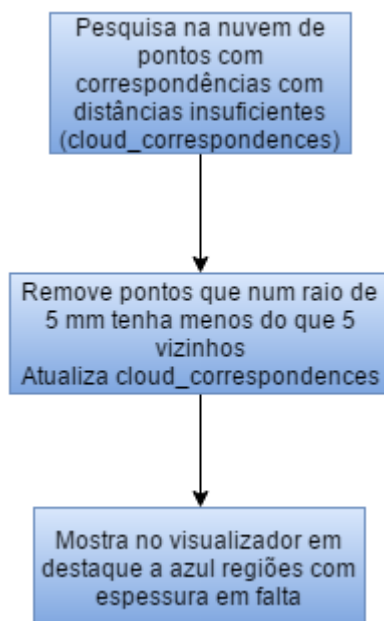


Figura 2.77 - Diagrama que representa a detecção de regiões de pontos com espessura insuficiente.

Os pontos da trajetória do robô selecionados por estarem próximos de zonas que necessitam de reforço, são guardados e é gerado um programa para o robô com uma trajetória que contenha apenas esses pontos (Este algoritmo está preparado para gerar programas para o robô ABB e Motoman).

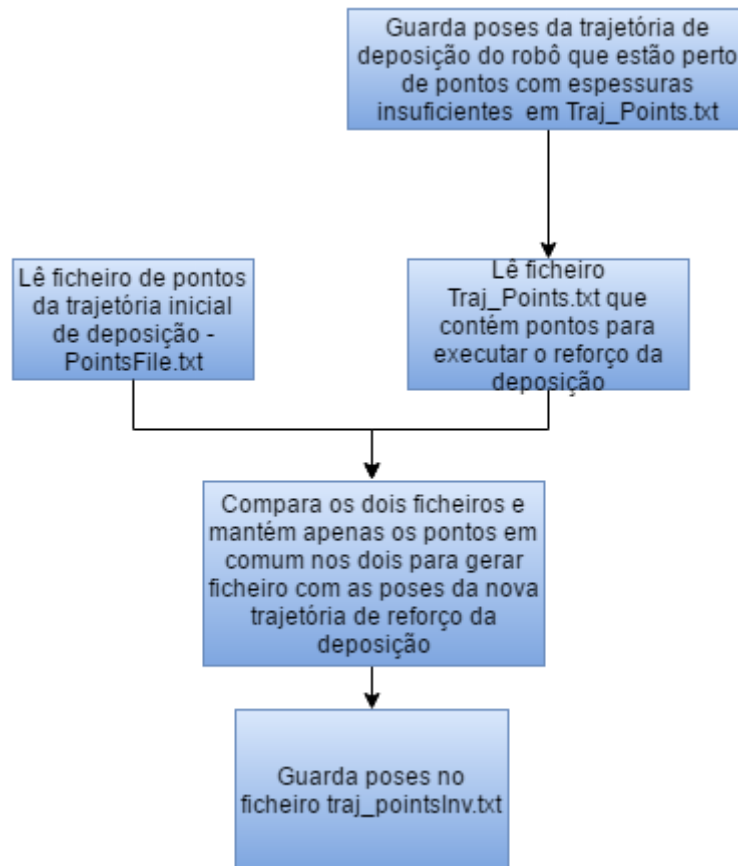


Figura 2.78 - Diagrama que representa seleção de pontos da trajetória do robô a serem novamente reproduzidos para reforço em regiões de espessura insuficiente.

2.2.5 Componente mecânica

Para adaptar a pistola de projeção de compósito tanto ao processo de demonstração como ao de reprodução da trajetória por parte do manipulador robótico, imprimiu-se uma peça (Figura 2.82). Esta permite integrar o marcador luminoso pertencente ao sistema 6DMimic na furação lateral (Figura 2.80), permite fixar ao manipulador na furação frontal (Figura 2.79) e permite fixar na ferramenta de projeção pela furação da parte de trás (Figura 2.81).



Figura 2.79 - Vista frontal do modelo da peça de adaptação da pistola em *Solidworks*.



Figura 2.80 - Vista lateral do modelo da peça de adaptação da pistola em *Solidworks*.

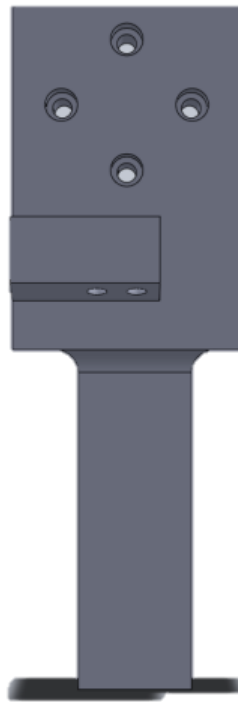


Figura 2.81 - Vista de trás do modelo da peça de adaptação da pistola em *Solidworks*.



Figura 2.82 - Vista lateral da peça de adaptação montada na pistola.

2.3 Enrolamento filamentar

2.3.1 Introdução

O enrolamento filamentar (Figura 2.83) caracteriza-se pelo processo de enrolamento de fitas constituídas por filamentos de fibras de carbono ou fibras de vidro, ou outro tipo de fibras, sobre um mandril em rotação.

O INEGI, no laboratório de materiais compósitos, tem uma máquina de enrolamento filamentar CNC de 6 eixos com controlo independente que inclui um mandril em torno do qual se colocam os moldes em rotação para o processo de enrolamento. No entanto, este sistema de enrolamento filamentar em cooperação com um sistema de visão foi implementado com recurso a uma estrutura que simula o mandril do qual se sabe a posição através de um *encoder* (Figura 2.86) e um sensor de proximidade (Figura 2.88) que dá um impulso em cada rotação do mandril, sendo este sinal utilizado para referenciar a posição angular do mesmo. O sistema de visão permite fazer varrimentos do molde para detetar a presença de orifícios em posições aleatórias com o intuito de um manipulador robótico ser deslocado a esse local para auxílio no desvio do material enrolado. Este processo é importante devido ao facto das fitas se moverem consoante a força nelas exercida e considerando que durante o processo de enrolamento estas estão a ser puxadas, a sua posição não se mantém de acordo com a trajetória da cabeça que a puxa mas sim de acordo com a posição atual da cabeça caso não haja obstáculos no caminho. A função desta cooperação com o robô seria precisamente a de criar obstáculos ou pontos de fixação do compósito durante o tempo necessário à sua consolidação ou simplesmente desviar a fita para que não passe por cima de orifícios.

O molde utilizado foi uma garrafa de gás cedida pela Amtrol Alfa ao INEGI onde foram acrescentados orifícios dispostos aleatoriamente nos tampos da garrafa para testes.



Figura 2.83 - Enrolamento Filamentar [10].

2.3.2 Arquitetura Funcional

A arquitetura funcional do sistema de enrolamento filamentar descrita na Figura 2.84 tem como entradas os sinais digitais dados pelo *encoder* e pelo sensor de proximidade que são adquiridos pelo Arduino e processados de forma a informar o programa no PC qual a posição do mandril paralelamente aos dados que são enviados do sensor VC Nano 3D sobre o perfil do molde. Como saída, obtém-se a presença/posição de orifícios no molde. A comunicação entre o Arduino e o computador é feita por porta série e entre o computador e o sensor laser é feita por Ethernet através do protocolo TCP/IP.

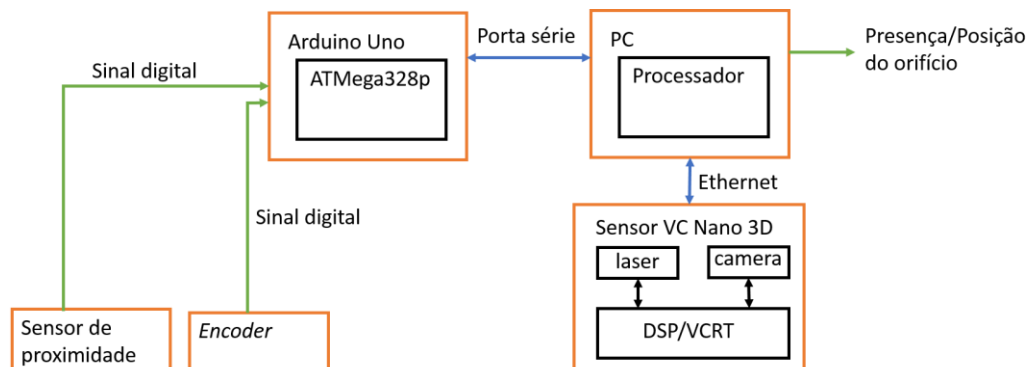


Figura 2.84 - Arquitetura funcional relativa ao sistema de enrolamento filamentar.

2.3.3 Componentes de *Hardware*

Laser VC Nano 3D

O laser utilizado para fazer um varrimento do molde para enrolamento filamentar foi um sensor laser de triangulação inteligente. Este equipamento integra um laser que projeta uma linha e uma camera inteligente capaz de captar dados 3D em tempo real. Inclui também um DSP para processamento dos dados obtidos por triangulação para o cálculo das distâncias ao objeto (coordenada Z) para cada ponto do conjunto ao longo da linha de laser sobre o objeto (coordenada X).

Especificações:

Sensor: 1/1.8" CMOS, *scan rate* 400 Hz.

Resolução: 1280(H) x 1024(V) pixel, WVGA.

Tamanho de pixel: 5.3(H) μm x 5.3(V) μm .

Obturador: Alta velocidade: 21 μs + *in steps of* 21 μs .

Rácio de aquisição: 50 fps.

Aquisição: Controlada por programa ou por *trigger* externo de alta velocidade, aquisição livre de *jitter*.

Processador: 700 MHz DSP.

RAM: 128 MB DDR2-SDRAM.

FLASH EPROM: MB *Flash memory* (nonvolatile).

DIG. I/Os: 2 Entradas, 4 Saídas.

Trigger I/Os: 1 entrada de *trigger* e uma saída *flash*.

Interfaces: Ethernet 10/100 Mbit.

Alimentação: 24 V +/-20 % DC, max. 300 mA.

Consumo: aprox. 2.6 W.

Dimensões: aprox. 140 mm x 70 mm x 35 mm, aprox. 400 g.

Laser: Laser classe 2, 100 mW, 660 nm de comprimento de onda.

Distância de trabalho: 164 mm.

Intervalo de medida horizontal: 60 mm - 125 mm.

Intervalo de medida vertical: +/- 70 mm.

Resolução lateral: 50 μm - 100 μm .

Resolução vertical: μm - 80 μm .

Comunicação com laser VC Nano 3D

A comunicação com este sensor laser é feita por Ethernet sob o protocolo TCP/IP. A ligação é estabelecida recorrendo a sockets em que o laser funciona como servidor e o PC como cliente.

IP sensor : 192.168.0.65

Porta : 1096

O pedido de dados bem como a mudança de parâmetros do laser é feita através do envio de comandos que podem ser consultados nas tabelas do Anexo A. Estabelecida a conexão entre o sensor e o PC, devem ser enviados comandos de parametrização caso necessário e o comando para ligar o laser (CMD_LASER_MODE), de seguida apenas será necessário definir o modo de funcionamento (CMD_MODE) para que o sensor envie as coordenadas dos pontos que adquiriu ao longo da linha do laser [39].

A comunicação e a troca de dados entre o laser e o PC efetuam-se nos seguintes passos:

- O cliente solicita a conexão ao servidor.
- O servidor responde ao pedido.

- O cliente confirma a conexão.
- O servidor fica a aguardar o envio de comandos.
- O cliente envia comando para ligar o laser constituído por: Cmd_ID + Host Counter + Action + Value (16 bytes).
- O servidor responde ao pedido com uma mensagem constituída por um Global Header : Response_ID + Cam_Counter + Meta Header and Data Size (12 bytes) e por um Meta Header : Cmd_ID + Error Nr. (8 bytes).
- O cliente envia comando NOWAIT para que o servidor espere até que os dados sejam enviados na totalidade sob a forma de uma mensagem constituída por: Cmd_ID + Host Counter + Action + Value (16 bytes). Este comando será útil no caso de o cliente demorar a receber os dados e evita a perda dos mesmos, no entanto pode causar atrasos na próxima varredura do laser.
- O servidor responde ao pedido com uma mensagem constituída por um Global Header : Response_ID + Cam_Counter + Meta Header and Data Size (12 bytes) e por um Meta Header : Cmd_ID + Error Nr. (8 bytes).
- O cliente envia comando de solicitação de pontos constituído por: Cmd_ID + Host Counter + Action + Value (16 bytes).
- O servidor responde ao pedido com uma mensagem constituída por um Global Header : Response_ID + Cam_Counter + Meta Header and Data Size (12 bytes) e por um Meta Header: Cmd_ID + Error Nr. (8 bytes).
- O servidor envia os pontos em diversos pacotes fazendo a divisão de acordo com o espaço livre no buffer em cada envio. Cada pacote tem um Global Header constituído por: Mode + Cam_Counter + Meta Header and Data Size (12 bytes), tem um Meta Header constituíd por: Number of Points + LineCounter (8 bytes) e de seguida envia os dados na forma x0y0x1y1x2y2 ... xnyn.
- O cliente solicita a desconexão
- O servidor aceita pedido e desconeta.

Máquina de enrolamento filamentar

O trabalho desenvolvido no âmbito desta dissertação dirigido à automatização do processo de enrolamento filamentar tem o intuito de ser implementado no laboratório de materiais compósitos do INEGI onde existe um equipamento de enrolamento filamentar CNC de 6 eixos (Figura 2.85) com controlo independente e que inclui também 3 módulos independentes de tensionamento unitário de filamentos de fibras. Embora, para este projeto se tenha testado o

conceito numa estrutura que simula o efeito desta máquina, foi estudado o modo de funcionamento da mesma em busca de uma solução de integração dos dois sistemas.



Figura 2.85 - Imagem da máquina de enrolamento instalada no INEGI.

O controlador desta máquina tem instalado o Sinumerik 840Di da Siemens.

Este *software* é dividido em: sistema operativo, *software* NC, *software* HMI (*Human Machine Interface*) e *software* PLC. O sistema operativo instalado é o *Windows* que corre num PC industrial. O *software* NC é executado no PC em tempo real paralelamente ao *Windows* e de onde se pode retirar informação sobre a posição dos eixos da máquina de enrolamento. O *software* HMI é a interface de utilização onde o operador pode introduzir comandos e programas. O PLC por *software* (*softPLC*) integra o programa do utilizador bem como as configurações para o PLC integrado na placa MCI e é desenvolvido no SIMATIC STEP 7 da *Siemens*.

Pensou-se na integração do sistema de enrolamento robotizado com o Sinumerik através da análise do programa instalado no PLC e utilizando os I/Os do mesmo para se detetar momentos do processo com interesse na cooperação do robô. A ideia seria a deteção de orifícios no molde por parte do sistema de visão e transmissão dessa informação ao PLC por uma entrada digital, fazendo com que o PLC atuasse no programa de enrolamento de modo a interrompe-lo até que o robô tivesse tempo de desviar ou fixar a fita de compósito. Da mesma forma, o robô emitiria um sinal ao PLC assim que terminasse a sua intervenção de modo a que o PLC retome o processo. No entanto, assim que se fez exportação do programa presente no PLC verificou-se que os blocos de estrutura do mesmo estavam bloqueados por parte do fabricante. Em contacto com a equipa de assistência da *Siemens* confirmou-se que esta situação é recorrente pois a *Siemens* fornece *templates* de cada parte integrante do sistema mas o projeto é feito pelo fornecedor que, por vezes, protege parte do funcionamento do equipamento. Recorreu-se ao contacto com a ENTEC, empresa que instalou o equipamento no INEGI, que sugeriu que se

fizessem pequenas alterações estritamente necessárias no programa do PLC. Assim, considerando a escassez de tempo disponível para a conclusão deste trabalho, optou-se por provar o conceito do sistema de enrolamento automático, construindo uma estrutura que simula a máquina de enrolamento e o mandril e recorrendo a um *encoder* e a um sensor de proximidade para se conhecer a posição do molde em torno do mandril.

Encoder rotativo

O *encoder* rotativo utilizado na estrutura que simula a máquina de enrolamento filamentar foi o E6A2-CW5 da Omron.



Figura 2.86 - Encoder E6A2-CW5 da Omron.

Este *encoder* permite extrair informação da posição do mandril para ser utilizada na reconstrução do perfil do molde e/ou obter posição do orifício e gera um impulso por cada incremento de 3.6° no deslocamento angular do eixo do mandril.

Especificações:

Alimentação: 12 a 24 VDC.

Corrente: $\leq 20\text{mA}$.

Saída: Coletor aberto.

Tempo de resposta: $1.0 \mu\text{s}$ max.

O *encoder* rotativo tem uma topologia de saída de coletor aberto, o que significa que o sinal de saída é aplicado à base de um transistor NPN interno, cujo coletor é aberto, ou seja, exteriorizado num dos fios de saída. Assim, necessita de uma resistência de pull-up entre a saída e o VCC.

Tem duas saídas A e B (Figura 2.87) com ondas quadradas em quadratura permitindo conhecer-se a direção da rotação mediante a onda que é detetada.

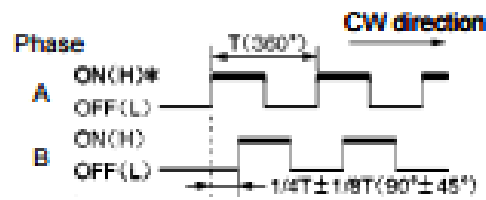


Figura 2.87 - Ondas de saída do *encoder* E6A2 - CW5C.

Sensor de proximidade

O sensor utilizado para detetar a rotação de uma volta completa do mandril foi o S51-pa nk da Datalogic.



Figura 2.88 - Sensor S51-pa nk da Datalogic.

O mandril tem junto uma patilha que fica ao alcance do sensor de proximidade sempre que completa uma volta.

Especificações:

Alimentação: 10 a 30 VDC.

Corrente: ≤ 35 mA.

Saída: NPN.

Tempo de resposta: 1.0 ms max.

ATMega328p

O microcontrolador utilizado para receber os sinais do *encoder* e do sensor de proximidade foi o ATmega 328p. Estes componentes estão conectados ao ATmega328p, de acordo com a Figura 2.89, nos pins de interrupção externa INT 0 e INT 1. O microcontrolador é alimentado pela placa de Arduino que está ligada ao computador por USB.

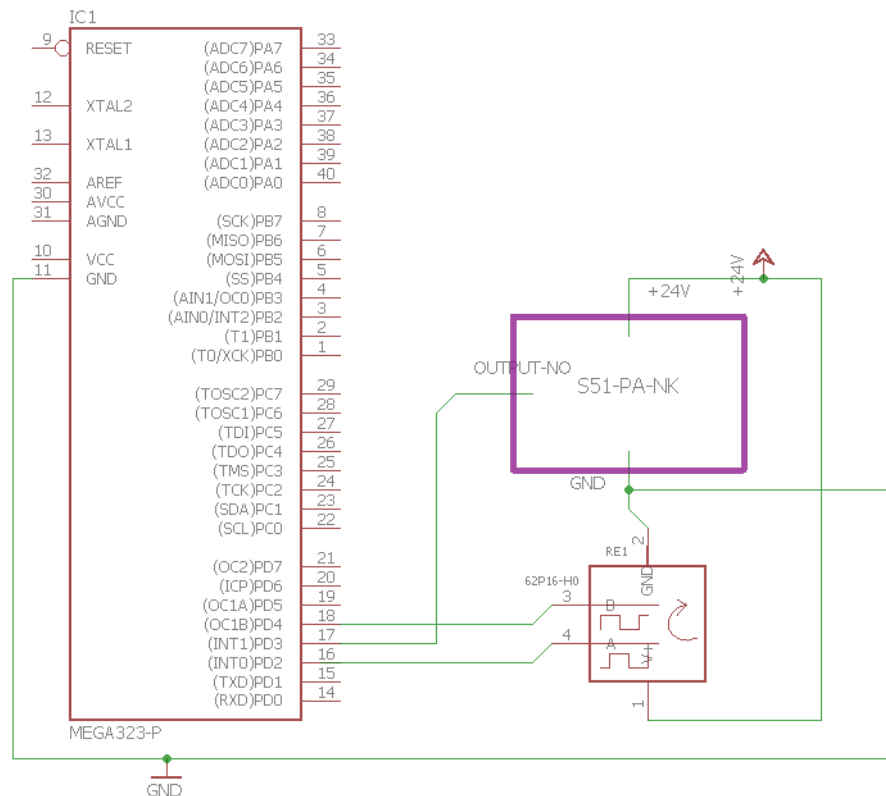


Figura 2.89 - Esquema elétrico da ligação do *encoder* e do sensor de proximidade ao ATmega328p.

Estrutura de simulação da máquina de enrolamento filamentar

A estrutura utilizada para simulação da máquina de enrolamento filamentar é constituída por um mandril com rotação auxiliada por rolamentos, um suporte para o sensor de proximidade e para o *encoder* e um suporte para o sensor laser (Figura 2.90).

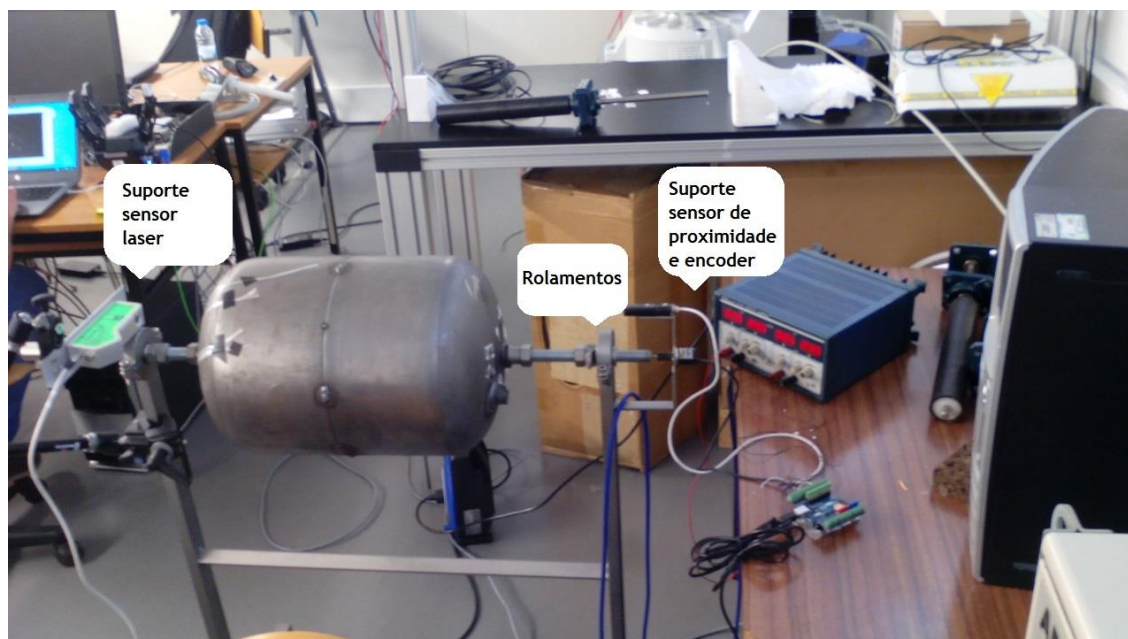


Figura 2.90 - Estrutura para enrolamento filamentar.

2.3.4 Software

Com o objetivo de se detectar automaticamente a presença e a localização de orifícios num molde de um depósito de gás, instalou-se na estrutura de simulação da máquina de enrolamento, um mandril em torno do qual se faz girar o molde manualmente, um *encoder* e um sensor de proximidade, bem como o sensor laser que, medindo distâncias, permite obter o perfil do molde. Os sinais do *encoder* e do sensor de proximidade estão conectados a um ATmega 328p e despoletam uma interrupção externa.

Então, sempre que o *encoder* envia um impulso, no ATmega328p corre uma rotina onde é incrementado um contador que retorna a zero sempre que o mandril completa uma volta. Isto porque, sempre que o sensor de proximidade envia um impulso, o contador é posto a zero. O contador incrementa sempre que o *encoder* é girado num sentido e decrementa sempre que é girado no sentido contrário.

De seguida, o valor do contador a cada dois incrementos é enviado pela porta série para o computador. No computador está a ser executada uma aplicação que comunica com o sensor laser e que recebe valores da porta série. Sempre que um valor é recebido na porta série é executada uma rotina que envia ao sensor laser um comando para que este envie uma trama de pontos que corresponde às distâncias que o sensor está a medir. Para que todos os pontos sejam enviados, tem que ser feita uma espera em que a rotina está a ler no buffer os valores até terminar a trama. Como se pode ver através da Figura 2.91 e Figura 2.92, os pontos adquiridos pelo sensor podem ser visualizados em gráficos através de um programa fornecido juntamente com o equipamento. Estes pontos foram adquiridos da superfície do reservatório de gás numa

zona do tampo onde não tem orifícios (Figura 2.91). E depois, foram adquiridos pontos numa zona com orifício (Figura 2.92). A partir da aplicação desenvolvida para este projeto, com a implementação do protocolo de comunicação com o sensor foram obtidos os pontos que posteriormente foram guardados num ficheiro para futuro processamento.

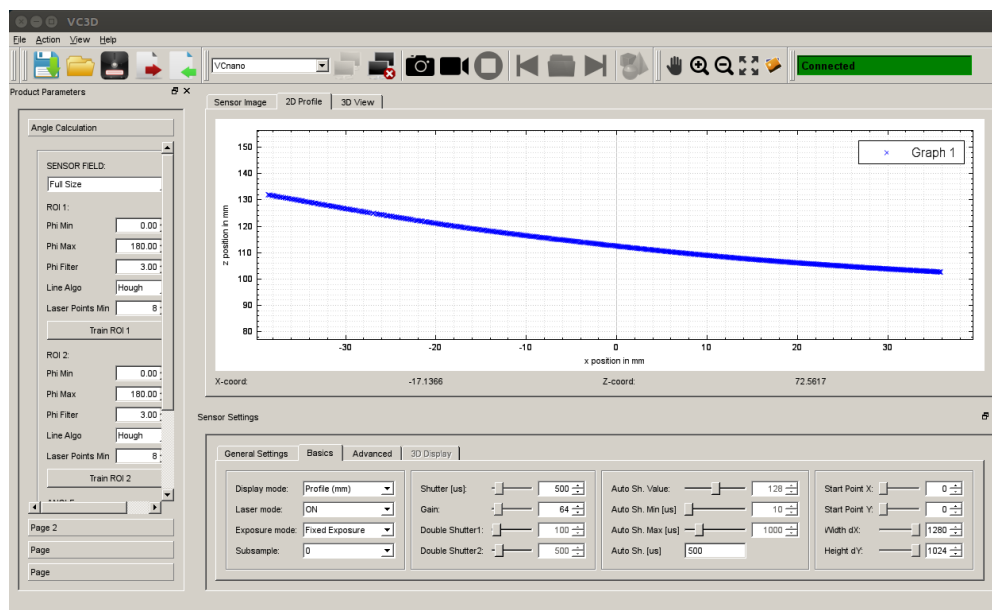


Figura 2.91 - Aplicação que permite visualizar os pontos adquiridos pelo sensor numa zona sem orifícios.

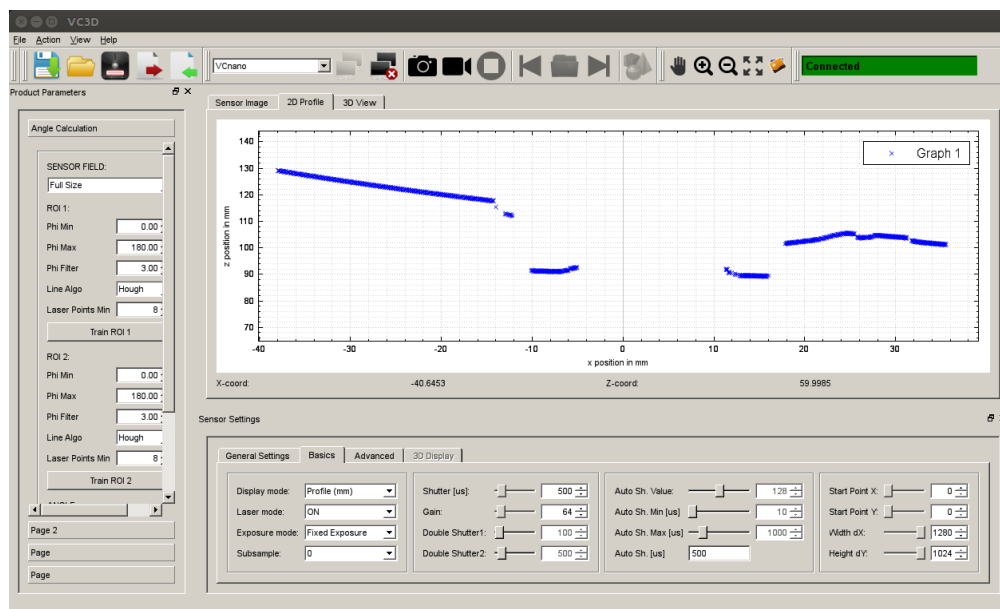


Figura 2.92 - Aplicação que permite visualizar os pontos adquiridos pelo sensor numa zona com orifícios.

Tendo os valores adquiridos pelo sensor laser correspondentes a uma volta completa do mandril, é possível fazer uma reconstrução da superfície do molde de onde se pode localizar

orifícios. Localizados os orifícios, é possível enviar ao manipulador um sinal para que este execute uma rotina pré-programada de auxílio no desvio das fitas nessas zonas. Comparando sucessivas digitalizações ao longo do processo de enrolamento pode-se calcular a espessura de material enrolado e avaliar a posição das fitas de fibra.

2.3.5 Componente mecânica

Foi estudada (Figura 2.93) e imprimida (Figura 2.94) uma peça de adaptação do sensor laser VC Nano 3D ao manipulador. A peça não foi imprimida na totalidade por falta de material mas optou-se por não se repetir a sua impressão porque era suficiente para os testes que foram feitos. Inicialmente seria este o sensor laser a utilizar para monitorizar o processo de projeção, no entanto, como a sua entrega atrasou foi apenas utilizado para o processo de enrolamento e neste, ele é colocado num suporte fixo.

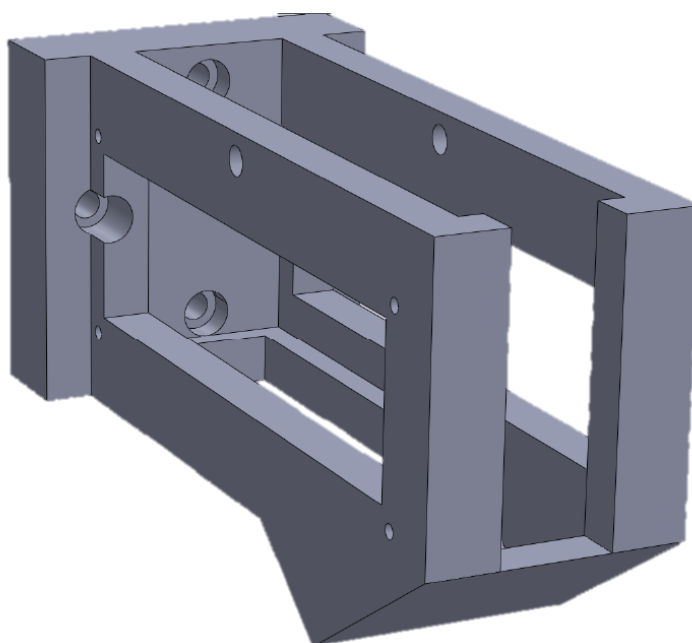


Figura 2.93 - Modelo *Solidworks* da peça de adaptação do sensor laser VC Nano 3D ao manipulador.



Figura 2.94 - Imagem real da peça de suporte para o sensor.

Capítulo 3

3.1 Testes ao material

No decorrer dos testes feitos ao processo robotizado de enrolamento filamental, simulou-se a aplicação de fibra de carbono e fibra de vidro em fita num molde de garrafa de gás cedida pela Amtrol Alfa ao INEGI (Figura 3.95).



Figura 3.95 - Simulação da aplicação de fibra de carbono em fita no molde.

Posteriormente, foi feito um scan do molde com o Laser RF625 obtendo-se a nuvem de pontos representada na Figura 3.96, observando-se lacunas na silhueta do molde precisamente na posição onde estavam dispostas as fitas.

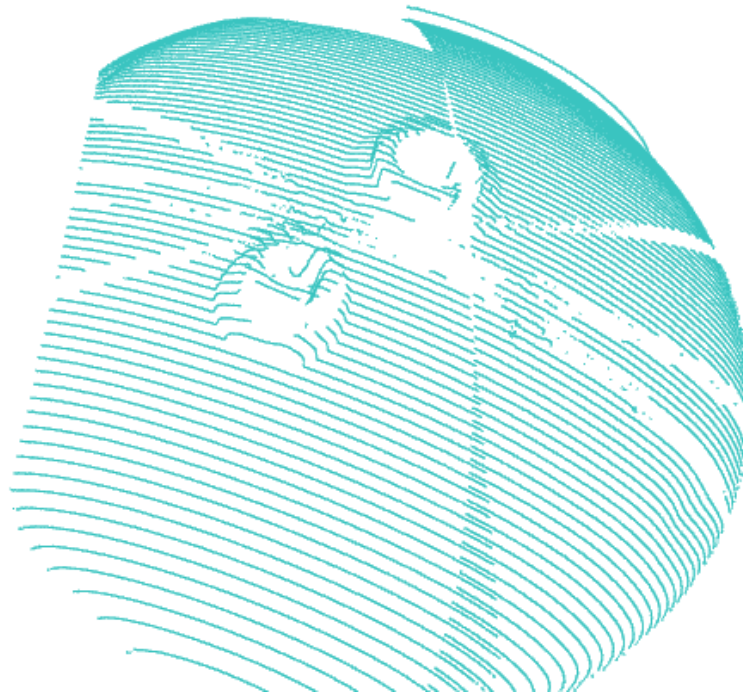


Figura 3.96 - Nuvem de pontos resultante da aplicação de fibra de carbono em fita no molde.

Este resultado levou a um teste posterior em que se incidiu a radiação do laser perpendicularmente à fita de fibra e se adquiriram imagens em duas direções: na direção perpendicular à fita (Figura 3.97), ou seja, na direção de incidência do laser e na direção de aquisição da camera do sensor (Figura 3.98). Tendo-se verificado que a radiação continua visível na primeira situação e torna-se praticamente invisível na segunda situação. Neste caso, a camera fica incapaz de ver a radiação do laser pois, na direção em que se encontra, a radiação está invisível. Este facto origina uma falta de pontos nas zonas onde está disposta a fita. Consequentemente, torna-se impossível determinar distâncias do sensor ao material e torna-se impossível determinar a quantidade de material disposto.

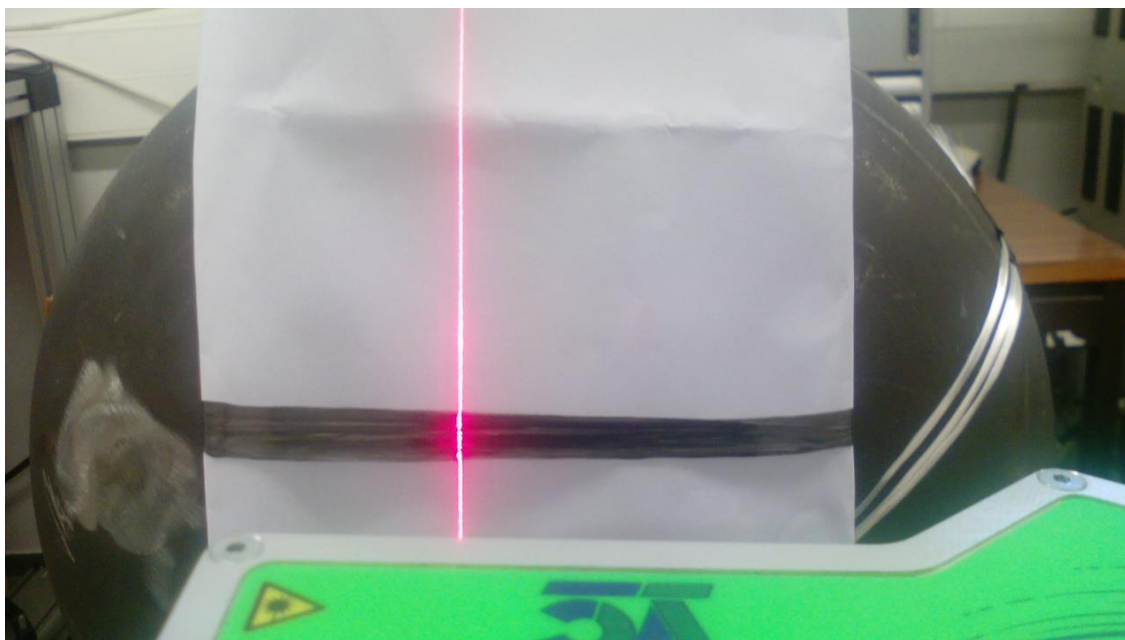


Figura 3.97 - Imagem adquirida num ângulo de 90° com radiação laser a incidir perpendicularmente na fita de fibra de carbono.

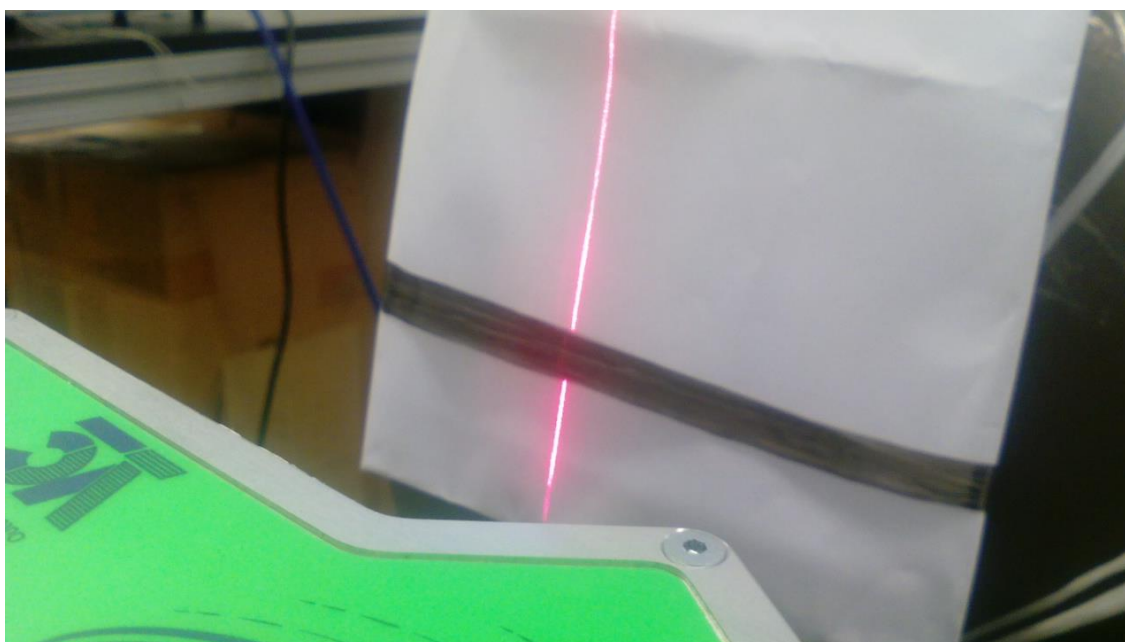


Figura 3.98 - Imagem adquirida na direção de aquisição da camera do sensor com radiação laser a incidir perpendicularmente na fita de fibra de carbono.

Testou-se, também, a situação em que a radiação laser incide no material não perpendicularmente mas sim num ângulo menor, tipicamente 45° e capturou-se imagens tanto na direção em que o laser estava a incidir (Figura 3.99) como na direção de aquisição da camera do sensor (Figura 3.100). Verificou-se que, nestas direções, a radiação não é visível, no entanto, existe um ângulo (Figura 3.101) superior ao angulo a que se encontra a camera, em que a

radiação é visível. De qualquer forma este fenómeno inviabiliza a utilização deste sensor para se medir distâncias a superfícies de fibras de carbono.

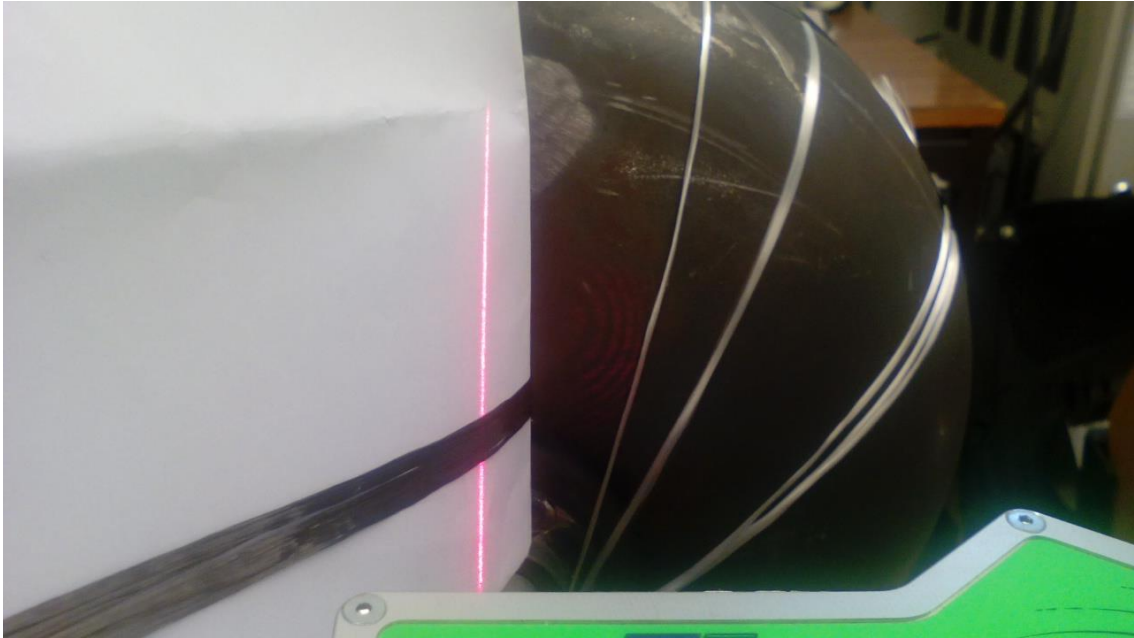


Figura 3.99 - Imagem adquirida na direção de emissão da radiação laser que está a incidir na fita de fibra de carbono com um ângulo menor do que 90° .

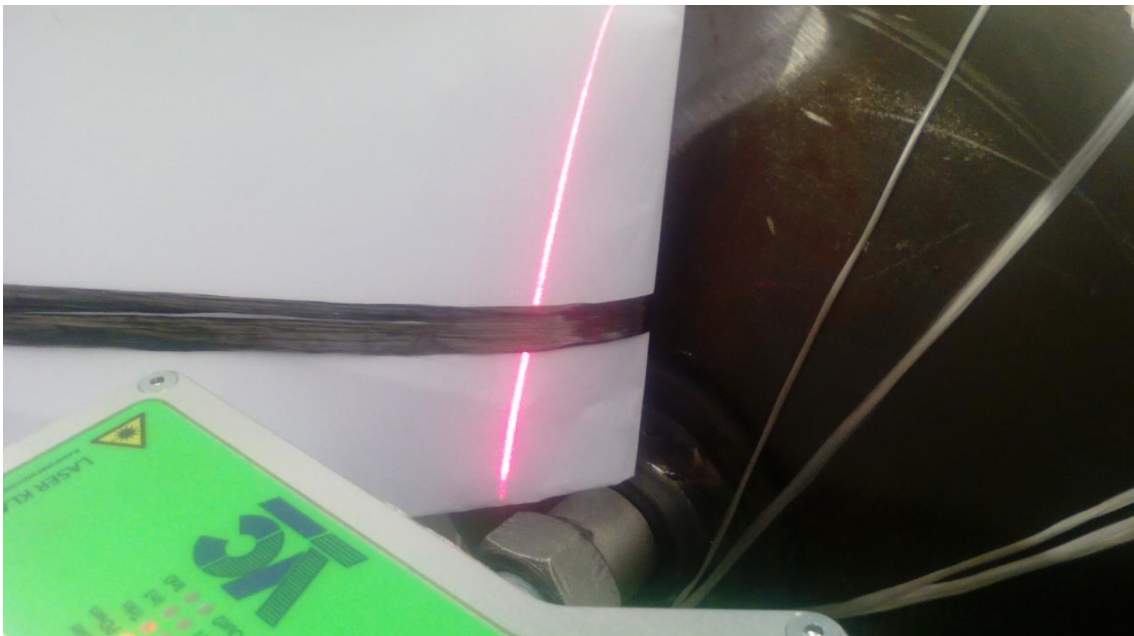


Figura 3.100 - Imagem adquirida na direção de aquisição da camera do sensor com a radiação laser a incidir na fita de fibra de carbono com um ângulo menor do que 90° .

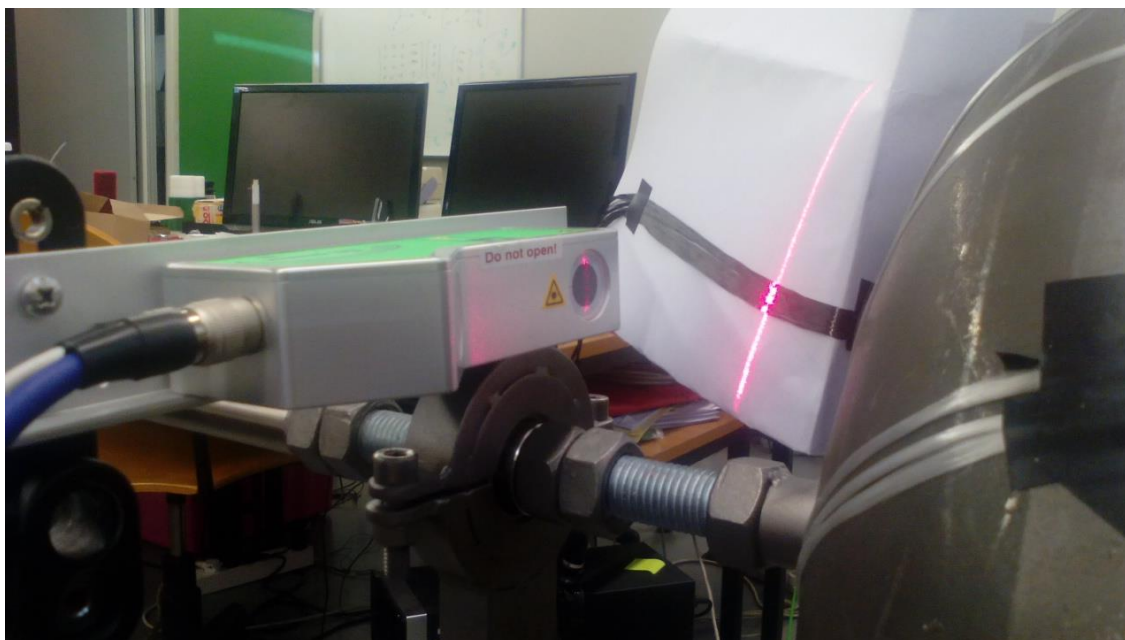


Figura 3.101 - Imagem adquirida na direção capaz de captar a radiação laser refletida que está a incidir na fita de fibra de carbono com um ângulo menor do que 90°.

Os mesmos testes foram feitos utilizando-se agora fibra de vidro em fita (Figura 3.102) que, ao contrário do material anterior, mantém a radiação visível permitindo que a camera a capte. Como se pode ver na Figura 3.102 a), a digitalização da superfície apresenta relevo nas zonas onde a fibra de vidro em fita está colocada (Figura 3.102 b)).

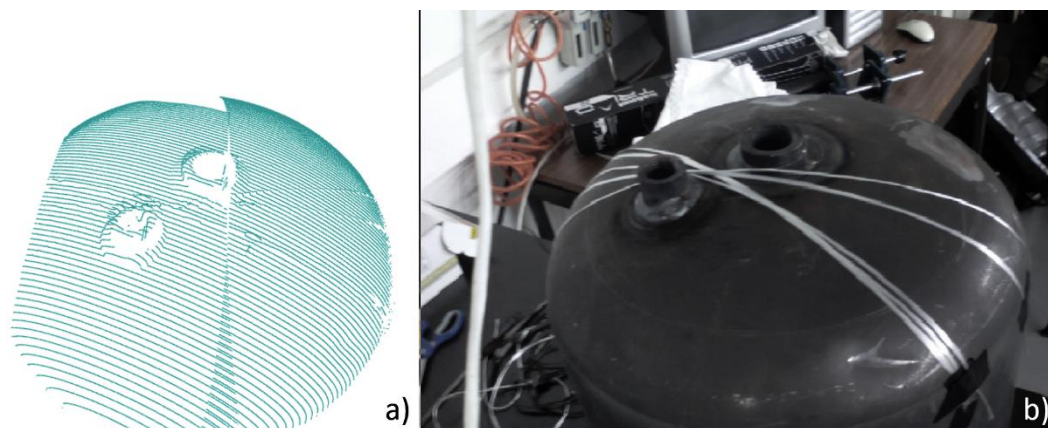


Figura 3.102 - Comparação de nuvem de pontos resultante da aplicação de fibra de vidro em fita no molde a) com imagem real b).

Na Figura 3.103 pode ver-se que a radiação é refletida e visível na direção de incidência mas também na direção de aquisição da camera do sensor (Figura 3.104).

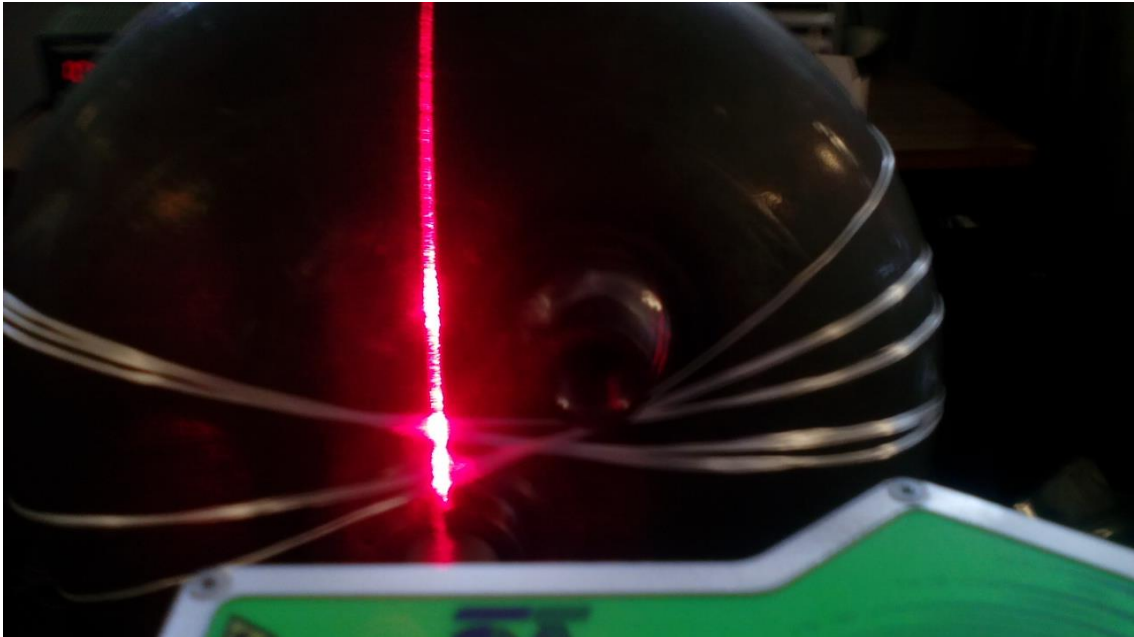


Figura 3.103 - Imagem adquirida num ângulo de 90° com radiação laser a incidir perpendicularmente na fita de fibra de vidro.

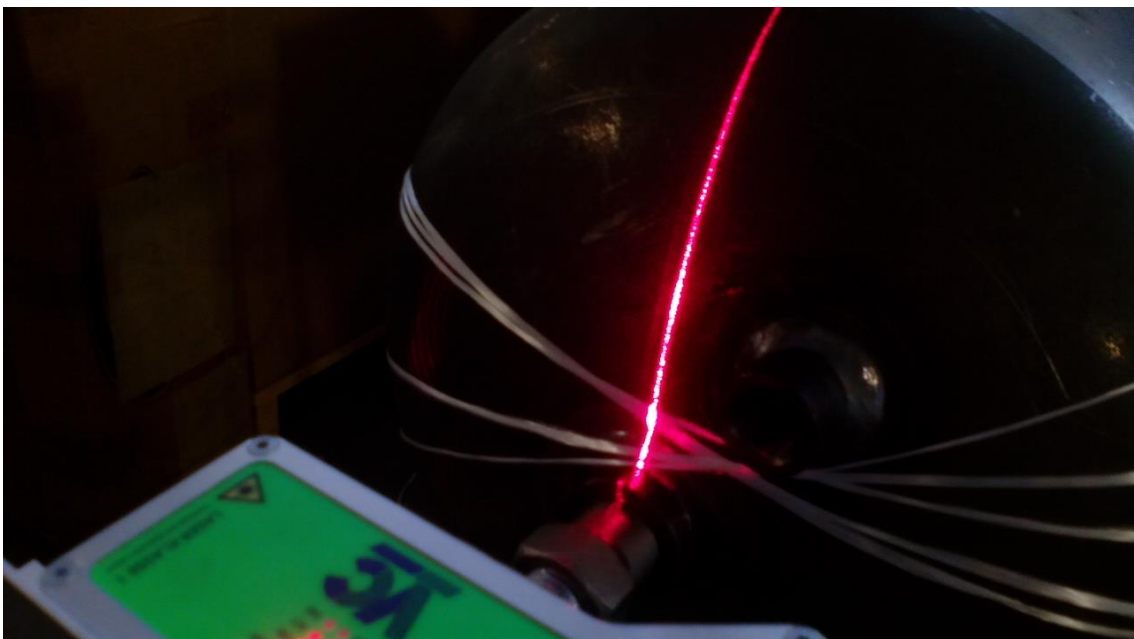


Figura 3.104 - Imagem adquirida na direção de aquisição da camera do sensor com radiação laser a incidir perpendicularmente na fita de fibra de vidro.

Capítulo 4

4.1 Conclusões e Trabalho Futuro

De uma forma geral e sob o ponto de vista funcional a aplicação desenvolvida para deposição robotizada de fibra de vidro e fibra de carbono compreende:

- Interface gráfica
- Seleção de áreas e introdução de valores de espessuras desejados
- Informação sobre o estado do processo, nomeadamente espessura do material já depositado
- Reforço automático de locais com espessura insuficiente
- Otimização do processo de reforço de forma a que não sejam reforçadas áreas que não são significativas para esta aplicação e garante que áreas significativas são reforçadas
- Algoritmo de minimização do erro no cálculo da espessura
- O armazenamento de informação sobre as nuvens (espessuras, trajetórias de deposição, trajetórias de reforço)
- Processo de demonstração de trajetórias
- Permite programação de diferentes manipuladores

A aplicação desenvolvida para monitorização do processo de enrolamento filamentar compreende:

- Aquisição dos pontos correspondentes a medidas da superfície do molde sincronizados com um *encoder* e um sensor de posição que permitirá monitorizar e mediar a superfície onde está aplicada a fibra em fita.
- Deteção automática dos orifícios presentes no molde e medição das suas dimensões.

Desenvolveu-se, testou-se e validou-se o módulo de monitorização da aplicação de fibras por projecção. Criou-se uma interface gráfica de contacto com o utilizador e um algoritmo para processamento dos dados adquiridos do sensor nomeadamente para cálculo de espessuras de

material depositado e geração de trajetórias para reforço localizado da aplicação de fibras. Testou-se o sistema de ensinamento por demonstração através da aplicação 6DMimic, desenvolvido no âmbito de outro projeto.

Por fim, desenvolveu-se e testou-se o módulo de detecção de orifícios para enrolamento filamentar.

Embora não se tenha um sistema completo com todos os processos automatizados, desenvolveu-se, testou-se e validou-se módulos que funcionam independentemente que serão úteis em evoluções futuras no sentido da total robotização e automatização dos processos de enrolamento e projeção.

A escolha do sensor laser para a aplicação que automatiza o processo de projeção de fibras baseou-se no facto de ser o mais indicado e mais preciso na medição de distâncias que permite criar um perfil da superfície digitalizada.

Sobre os testes feitos ao material de interesse para este projeto, conclui-se que este sistema de visão que utiliza a técnica de triangulação não é apropriado para fazer monitorização do material depositado em aplicações que utilizem a fibra de carbono.

Outro equipamento a considerar seria um sensor com a tecnologia TOF - *Time of Flight*. No entanto, esta depende também da reflexão no material e, pelos testes feitos ao material, esta tecnologia só resultaria se o laser incidisse sempre num ângulo de 90° com a superfície de fibra de carbono. Assim, ter-se-ia um ambiente controlado de aquisição. No entanto não é exequível dada a irregularidade das superfícies resultantes da deposição deste tipo de material.

Relativamente ao laser usado para monitorização do processo de enrolamento este tem instalado um programa que funciona como servidor que adquire as medições e as processa para posteriormente as enviar. É necessário um programa que funcione como cliente que solicite o envio de dados. Esse programa foi desenvolvido para este projeto, mas explorado para esta aplicação específica. Poderia ser criada uma *library* ou uma DLL de comunicação com o sensor para facilitar o desenvolvimento de futuras aplicações para o sensor.

Em complemento à detecção de orifícios no processo de enrolamento sugere-se a utilização do sistema de visão também para monitorizar o posicionamento da fibra em fita e a espessura da peça. Desta forma, é possível controlar a qualidade do enrolamento e tomar conhecimento de casos em que as fibras não fixam no sítio correto e isso condiciona a qualidade do produto final.

Também, de acordo com a pesquisa feita em relação ao funcionamento da máquina de enrolamento filamentar instalada no INEGI, é possível a integração do Sinumerik com o sistema robotizado de enrolamento filamentar por meio do PLC os seus I/Os para detetar e atuar em

momentos do processo com interesse na cooperação do robô. Ou então analisar diretamente a informação contida no módulo NC de modo a obter a posição dos eixos da máquina.

No seguimento de todo o trabalho desenvolvido sugere-se também como trabalho futuro a detecção da orientação das peças no espaço, utilizando o sistema de visão. As nuvens de pontos resultantes das sucessivas digitalizações ao longo do processo de projeção de fibras devem ser comparadas para se achar uma correspondência entre elas e as reorientar. Este procedimento permite, independentemente da orientação do molde na bancada de trabalho, calcular a espessura do compósito depositado. Por outro lado, um sistema que organiza o armazenamento da informação sobre as peças já digitalizadas, poderá permitir a associação automática de uma trajetória de deposição ou de digitalização com um tipo de peça. Isto permitiria automatizar a parte do processo em que, consoante o molde que se pretende cobrir, se tem que seleccionar a trajetória de deposição e a trajetória de digitalização.

Capítulo 5

5.1 Anexos

Anexo A

Tabela 5.1– Lista de comandos para comunicação com o sensor laser VC Nano 3D.

Command name	Nr.	Effect
CMD_RESET	0	Stop image acquisition, reset host and camera counter and set NBR_LINES to 0
CMD_LASER_MODE	1	1: set laser on 0: set laser off
CMD_ROI_X	2	Set the ROI x starting point at X
CMD_ROI_Y	3	Set the ROI y starting point at X
CMD_ROI_DX	4	Set the horizontal ROI width received at X
CMD_ROI_DY	5	Set the vertical ROI width received at X
CMD_GAIN	6	Set the gain at X
CMD_SHUTTER	7	Set the shutter at X
CMD_TRIGGER	8	0: set the line trigger off 1 (TRIG_RAIS): set the line trigger on raising value 2 (TRIG_FALL): set the line trigger on falling value -1: Auto trigger mode (camera trigger, see CMD_AUTO_TRIG_PERIODE) -2: Auto trigger mode (like -1, starting at a raising edge) -3: Auto trigger mode (like -1, starting at a falling edge)
CMD_MODE	9	-1: sensor activ, no result transfer via TCP/IP 0: MODE_STANDBY, no sensor activity 1: MODE_TAKE_PIC 2: MODE_2D_POINTLIST 3: MODE_2D_POINTLIST_AND_LASER_IMG 4: MODE_2D_POINTLIST_MM (order x0 z0 / x1 z1 / x2 z2 / ...) 5: MODE_2D_POINTLIST_MM (order x0 x1 x2 ... / z0 z1 z2 ...) 6: MODE_BINARY_PIC 7: Not used any more 8: MODE_PRODUCT_RESULTS if sensor works in product mode Look at the description below for more info about its meta Header
CMD_NBR_LINES	10	Set the number of frames that will be taken, if X equals -1 there will be endless capturing until CMD_RESET has been sent.
CMD_METHOD	11	Set the line extraction method: 0 center detection at pixel accuracy 1 BaryCenter detection 2 BaryCenter detection with speckle filter (recommended)
CMD_ALGORITHM	12	Set the Algorithm parameter 0 Fixed point algorithm (multiplied by 16 for sub pixel accuracy) 1 Floating point algorithm
CMD_THRESH	13	Set threshold for laser line detection
CMD_MINWIDTH	14	Set min. Laser Line Width
CMD_MAXWIDTH	15	Set max. Laser Line Width
CMD_EXPMODE	16	Set exposure mode: 0: FIX_EXPOSURE (use CMD_SHUTTER) 1: AUTO_EXPOSURE (overwrites CMD_SHUTTER [MINSH..MAXSH]) 2: DOUBLE_EXPOSURE (use CMD_DÓUBLE_SH1 and 2)
CMD_LASERVAL	17	AUTO_EXPOSURE mode: Select Shutter time for an average laser brightness of laser value
CMD_MINSH	18	AUTO_EXPOSURE mode: Min. allowed Shutter time [µs]

CMD_MAXSH	19	AUTO_EXPOSURE mode: Max. allowed Shutter time [µs]
CMD_DOUBLE_SH1	20	DOUBLE_EXPOSURE mode: Shutter time 1 [µs] – main image shutter
CMD_DOUBLE_SH2	21	DOUBLE_EXPOSURE mode: Shutter time 2 [µs] – second image, if no laser detection in main image
CMD_FILTER	22	Laser Line Detection Filter Mode 0: No Filter >0 : skip n reflections from top (if more lines are in the image) <0 : skip n reflections from bottom (if more lines are in the image)
CMD_ROI_X_MM	23	Read the ROI x starting point [mm]
CMD_ROI_Y_MM	24	Read the ROI y starting point at y
CMD_ROI_DX_MM	25	Set the horizontal ROI width received at X
CMD_ROI_DY_MM	26	Set the vertical ROI width received at X
CMD_LIB_VERSION	27	Get 3D Scanner Library Version
CMD_PLC_IN	28	Get all PLC inputs (Bit0 ... Bit3)
CMD_PLC_OUT	29	Set all PLC outputs (Bit0 ... Bit3)
CMD_ETHERNET_PACKAGE	30	Add n scan lines together in one Ethernet package. 0: write as much scan lines into camera memory and send the results at the end or if memory is full (about 11000 scans). This method is the fastest scan mode, as full processor power is available for scanning. But it will take more time at the end to transfer all data at once. Don't use this mode, if you scan permanently. 1: send scan line results directly after every scan (recommended) n: put n scan lines together in one Ethernet package. Recommended is max. of n=4 scan lines! Please have a look at CMD_SEND_NOWAIT !
CMD_AUTO_TRIG_PERIODE	31	Camera auto trigger mode in 1/10 frames per second 500 means 50 fps. See also CMD_AUTO_TRIG_ERR
CMD_AUTO_TRIG_ERR	32	nr scans which are not in time during Autotrigger (allowed jitter < 50µs) >0 means, camera is not fast enough for selected scan speed, n lines are out of tolerance. Customer have to reset the value.
CMD_SAVE_SETTINGS	33	Saves latest settings in camera flash. Camera will start with this parameters
CMD_SEND_NOWAIT	34	0: send will wait until all data are transferred If host is not ready, it will lead to a delay of the next scan line 1: send as much data as possible (recommended) If host is not ready, it will temporally fill camera memory. If memory is full, it stops scanning, until all data are transferred
CMD_SUBSAMPLE	35	0: full resolution for laser line detection 1: combine two lines for laser detection (fast laser line detection) 2: combine four lines for laser detection (fast laser line detection)
CMD_WORLD_ACCURACY	36	New sensors always have Kanatani calibration, which is faster and more accurate. This parameter is only used at Tsai calibration. 0: accurate float algorithm but slow 1: fast I40 algorithm with typical 0.1% inaccuracy but four times faster
CMD_SEND_INFO_PAR	37	Sends the parameters for lens calibration
CMD_READ_PRODUCT_FILE	38	reads the product parameter from the camera file "fd:/ProdPar.001"
CMD_CAMERA_SNR	39	Gets the serial number of the camera ID.
CMD_SET_PRODUCT_DATA	40	Set product parameter Host -> Cam. Use set mode (Action = 0) as Command Header and as X value the number of Bytes of the production file followed by the data itself.
CMD_SEND_PRODUCT_DATA	41	Send product parameter Cam -> Host. Use read mode (Action = 1) as Command Header. After the header size of 16 Bytes the product file data follows. The size of header and data are defined in the
		Command Header.
CMD_READ_SENSOR_FILE	42	reads the product parameter from the camera file "fd:/VC3DPar.001"
CMD_SET_SENSOR_DATA	43	Set sensor parameter Host -> Cam. Use set mode (Action = 0) as Command Header and as X value the number of Bytes of the production file followed by the data itself.
CMD_SEND_SENSOR_DATA	44	Send sensor parameter Cam -> Host. Use read mode (Action = 1) as Command Header. After the header size of 16 Bytes the product file data follows. The size of header and data are defined in the Command Header.

Anexo B

Tabela 5.2 - Código para robô ABB, em RAPID, para digitalização de peça.

```

MODULE CL_TRAP

VAR iodev serialCom;
PERS num S_TIME:=1000;
VAR intnum timeint;
VAR robtarget Robotpos{3000};
VAR robtarget p1;
VAR num count:=0;
VAR num stop_scan:=1;

LOCAL PROC OpenSerialCom(string comPort)
  Open comPort+":",serialCom\Bin;
  ClearIOBuff serialCom;
  TPWrite "Serial communication open";
ENDPROC

LOCAL PROC CloseSerialCom()
  Close serialCom;
  TPWrite "Serial communication closed";
ENDPROC

PROC main()
  Confl\Off;
  OpenSerialCom "COM1";
  ClearIOBuff serialCom;

  CONNECT timeint WITH trigger;
  ITimer 0.35, timeint;
  IEnable;
  stop_scan:=1;
  MoveJ [[170.34,552.7,672.9],[0.489206,-0.492047,0.495588,0.522456],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,fine,tool0;
  MoveL [[190.56,-80.75,317.79],[0.0985995,0.00291943,-0.995103,0.00633643],[0,-1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,fine,
laser\WObj:=wobjprojmap;
  stop_scan:=0;
  MoveL [[190.56,328.01,310.79],[0.0985916,0.00291596,-0.995104,0.00633656],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v10,fine,
laser\WObj:=wobjprojmap;
  stop_scan:=1;
  MoveL [[120.67,-80.35,310.79],[0.0985452,0.00293156,-0.995108,0.00634216],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,fine,
laser\WObj:=wobjprojmap;
  stop_scan:=0;
  MoveL [[120.69,328.07,310.79],[0.0986153,0.00288707,-0.995101,0.00633269],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v10,fine,
laser\WObj:=wobjprojmap;
  stop_scan:=1;
  MoveL [[48.73,-80.16,310.79],[0.0986306,0.00290425,-0.9951,0.00633396],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,fine,
laser\WObj:=wobjprojmap;

```

```

    stop_scan:=0;
    MoveL [[48.73,328.91,310.79],[0.098587,0.00291109,-
0.995104,0.00633351],[0,-1,-1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v10, fine,
laser\WObj:=wobjprojmap;
    stop_scan:=1;
    MoveL [[-48.26,-80.95,310.79],[0.10978,-0.00431,0.99392,-0.00654],[1,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, fine, laser\WObj:=wobjprojmap;
    stop_scan:=0;
    MoveL [[-48.26,228.78,310.80],[0.10978,-0.00431,0.99392,-0.00654],[1,0,-
1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]], v10, fine, laser\WObj:=wobjprojmap;
    stop_scan:=1;
    MoveJ [[170.34,552.7,672.9],[0.489206,-0.492047,0.495588,0.522456],[0,-
1,0,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,fine,tool0;
    write_to_file;
    CloseSerialCom;
ENDPROC

TRAP trigger
IF stop_scan<1 THEN
    SetDO D010_5,1;
    SetDO \SDelay:=0.1, D010_5, 0;
    p1 := Crobt(\Tool:=laser \WObj:=wobjprojmap);
    count:=count+1;
    Robotpos{count}:=p1;
ENDIF
ENDTRAP

PROC write_to_file()
    VAR iodev file;
    Open "HOME:/points.txt", file\write;
    FOR i FROM 1 TO COUNT DO
        Write file, "" \Pos:=Robotpos{i}.trans\nonewline;
        Write file, "," \Orient:=Robotpos{i}.rot;
    ENDFOR
    Close file;
ENDPROC

ENDMODULE

```

Referências

- [1] N. G. Akkus, G. Girgin, C. , "Control of the pretension in filament winding process," *Acta Mechanica et Automatica*, vol. 2, no. 3, pp. 5-10, 2008.
- [2] D. Heider, M. J. Piovoso, and J. W. Gillespie Jr, "Application of a neural network to improve an automated thermoplastic tow-placement process," *Journal of Process Control*, vol. 12, pp. 101-111, January 2002, 2002.
- [3] P. Mertiny and F. Ellyin, "Influence of the filament winding tension on physical and mechanical properties of reinforced composites," *Composites Part A: Applied Science and Manufacturing*, vol. 33, pp. 1615-1622, 12/2002, 2002.
- [4] M. C. Rezende, M. L. Costa, and E. C. Botelho, "Materiais Compósitos Poliméricos," in *Compósitos Estruturais*, ed São Paulo, Brasil: Artliber Editora Ltda, 2011, pp. 21-41.
- [5] W. Zhong, H. Yang, H. Li, and J. Xu, "Control system design of robotized filament winding for elbow pipe," in *Measurement, Information and Control (ICMIC), 2013 International Conference on*, 2013, pp. 1081-1085.
- [6] M. Araújo, R. Fangueiro, and H. Hong, *Têxteis Técnicos: Materiais do Novo Milênio* vol. 1: Litografia A.C., 2000.
- [7] L. M. Silva, F. L. Alves, and A. T. Marques, *Materiais de Construção*. Porto: Publindústria, 2013.
- [8] K. A. Friedrich, Abdulhakim A., "Manufacturing Aspects of Advanced Polymer Composites for Automotive Applications," *Applied Composite Materials*, 10 February 2012, 2012.
- [9] M. F. S. F. Moura, A. B. Morais, and A. G. Magalhães, *Materiais Compósitos - Materiais, Fabrico e Comportamento Mecânico*. Porto: Publindústria, 2005.
- [10] G. F. Pereira, *Monitorização de reservatórios produzidos por enrolamento filamantar usando sensores ópticos embebidos no compósito*. Porto, 2011.
- [11] J. S. MACK, R., "Filament winding process in thermoplastics," in *Manufacturing Techniques for Polymer Matrix Composites (PMCs)*, K.-T. H. Suresh G Advani, Ed., ed 80 High Street, Sawston, Cambridge CB22 3HJ, UK: Woodhead Publishing Limited, 2012.
- [12] R. E. R. Lobo, "Estudo e desenvolvimento do comando e controlo para um equipamento industrial automatizado de enrolamento filamantar," Mestrado Integrado em Engenharia Mecânica, Departamento de Engenharia Mecânica, Universidade do Minho, Braga, 2012.
- [13] H. L. Giacoletto. (2002) The art of filament winding. *Reinforced plastics*.
- [14] L. Carrino, W. Polini, and L. Sorrentino, "Modular structure of a new feed-deposition head for a robotized filament winding cell," *Composites Science and Technology*, vol. 63, pp. 2255-2263, 11/2003, 2003.
- [15] A. Anglani, A. Manta, and F. Nucci, "Simulation for a Sound Analysis of Robot Part Program in Filament Winding Process," in *Proceedings of 15th European Simulation Multiconference*, 2001, pp. 6-9.
- [16] L. Carrino, W. Polini, and L. Sorrentino, "Influence of winding speed on winding tension in robotized filament winding of complex-shape parts," *Composites Science and Technology*, vol. 65, pp. 1574-1581, 2005.
- [17] L. Carrino, W. Polini, and L. Sorrentino, "Method to evaluate winding trajectories in robotized filament winding," *Journal of Composite Materials*, vol. 38, pp. 41-56, 2004.
- [18] W. Polini and L. Sorrentino, "Influence of winding speed and winding trajectory on tension in robotized filament winding of full section parts," *Composites Science and Technology*, vol. 65, pp. 1574-1581, 8/2005, 2005.

- [19] W. Polini and L. Sorrentino, "Winding trajectory and winding time in robotized filament winding of asymmetric shape parts," *Journal of Composite Materials*, vol. 39, pp. 1391-411, 2005.
- [20] W. Polini and L. Sorrentino, "Actual safety distance and winding tension to manufacture full section parts by robotized filament winding," *Journal of Engineering Materials and Technology, Transactions of the ASME*, vol. 128, pp. 393-400, 2006.
- [21] G. F. Rossano, C. Martinez, M. Hedelind, S. Murphy, and T. A. Fuhlbrigge, "Easy robot path programming concepts: an industrial perspective on path creation," in *2013 44th International Symposium on Robotics (ISR 2013), 24-26 Oct. 2013*, Piscataway, NJ, USA, 2013, p. 6
- [22] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, pp. 87-94, 4/2012, 2012.
- [23] G. Jing-Guo, "Programming by demonstration by optical tracking system for dual arm robot," in *2013 44th International Symposium on Robotics (ISR)*, Seoul, 2013, pp. 1-7.
- [24] T.-W. Kim, H.-M. Lee, and J.-B. Kim, "A user interface for vision sensor based indirect teaching of a robotic manipulator," *Journal of Institute of Control, Robotics and Systems*, vol. 19, pp. 921-927, 2013.
- [25] M. R. Pedersen and V. Kruger, "Gesture-Based Extraction of Robot Skill Parameters for Intuitive Robot Programming," *Journal of Intelligent & Robotic Systems*, vol. 80, pp. 149-163, 2015.
- [26] Y. Maeda and R. Aburata, "Teaching and reinforcement learning of robotic view-based manipulation," in *2013 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 26-29 Aug. 2013*, Piscataway, NJ, USA, 2013, pp. 87-92.
- [27] P. Neto, J. N. Pires, and A. P. Moreira, "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition," *Industrial Robot*, vol. 37, pp. 137-47, 2010.
- [28] P. Neto, J. N. Pires, and A. P. Moreira, "CAD-based off-line robot programming," in *2010 IEEE Conference on Robotics, Automation and Mechatronics (RAM 2010), 28-30 June 2010*, Piscataway, NJ, USA, 2010, pp. 516-21.
- [29] P. Neto and N. Mendes, "Direct off-line robot programming via a common CAD package," *Robotics and Autonomous Systems*, vol. 61, pp. 896-910, 8/2013, 2013.
- [30] L. Shi, X. Tian, and C. Zhang, "Automatic programming for industrial robot to weld intersecting pipes," *International Journal of Advanced Manufacturing Technology*, vol. 81, pp. 2099-2107, 2015.
- [31] C. Schou, J. S. Damgaard, S. Bogh, and O. Madsen, "Human-robot interface for instructing industrial tasks using kinesthetic teaching," in *2013 44th IEEE International Symposium on Robotics, ISR 2013, October 24, 2013 - October 26, 2013*, Seoul, Korea, Republic of, 2013.
- [32] M. A. M. Ferreira "High Level Programmable and Flexible Industrial Robotized Cells," PhD, DEEC, University of Porto, Porto: FEUP, 2014.
- [33] Y. Finnestad, "Analysis of Kinect motion capture capabilities for automated robotic painting systems," Mater Degree, Stavanger, 2015.
- [34] A. Robotics. (2015, accessed: 29 Janeiro 2016). Simplified Robot Programming Available: https://library.e.abb.com/public/8e8168587cb1ef4cc1257ddc0033de9f/SRP_Final_external.pdf
- [35] F. Tahriri, M. Mousavi, H. J. Yap, M. D. Siti Zawiah, and Z. Taha, "Optimizing the Robot Arm Movement Time Using Virtual Reality Robotic Teaching System," *International Journal of Simulation Modelling*, vol. 14, pp. 28-38, 03/2015, 2015.
- [36] PCL. (2016). *pcl (3.0 ed.)*. Available: <http://pointclouds.org/>
- [37] Infaimon, "Tecnología TOF - Sensores CMOS paa obtención d imágenes 3D en tiempo real," ed, 2010.
- [38] R. S. Instruments. (2016, 30-05-2016). *Laser Scanners Series RF625 User's manual*.
- [39] V. Components. (2016, 02-06-2016). *VC 3D Laser Scanner Protocol & Parameters*.