

Calibração *Single Plane Hand-Eye* de Perfilômetro 2D para um Sistema Robótico de Manufatura Aditiva^{*}

Gilmar P. Cruz Júnior^{*} Nicolas Lizarralde^{**} Fernando Coutinho^{**}
Gustavo M. Freitas^{*} Fernando Lizarralde^{**}

^{*} Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil (e-mail: (gilmarpcjunior, gustavomfreitas)@ufmg.br).

^{**} Programa de Engenharia Elétrica - COPPE - Universidade Federal do Rio de Janeiro, RJ, Brasil (e-mail: (nicolas.lizarralde, fernando.coutinho, f.lizarralde)@coppe.ufrj.br).

Abstract: This paper presents a method of calibration of a 2D laser sensor (Profilometer) used in a robotic system of Additive Manufacturing. The proposed iterative method is based on *Hand-Eye* calibration and uses only information from data obtained with a single plane, without information of the plane localization. The calibration of the location of the profilometer with respect to the manipulator end-effector to which it is coupled is essential to extract geometric characteristics from parts manufactured by Additive Manufacturing. Simulations and experiments in a robotic Additive Manufacturing system show the effectiveness of the proposed method to determine the sensor pose with respect to the end-effector coordinate system.

Resumo: Neste artigo é apresentado um método de calibração de um sensor laser 2D (Perfilômetro) utilizada em um sistema robótico de Manufatura Aditiva. O método iterativo proposto é baseado na calibração *Hand-Eye* e utiliza somente informação dos dados obtidos de um único plano, sem a necessidade de saber a localização do plano. A calibração da localização do perfilômetro com relação ao efetuador do manipulador ao qual o mesmo é acoplado é fundamental para poder extrair características geométricas de peças fabricadas por Manufatura Aditiva. Simulações e experimentos em um sistema robótico de Manufatura Aditiva evidenciam a eficácia do método proposto para determinar a pose do sensor com respeito ao sistema de coordenadas do efetuador.

Keywords: Hand-Eye Calibration, Profilemeter, Additive Manufacturing, Robotic Manipulators, WAAM Processes.

Palavras-chaves: Calibração Hand-Eye; Perfilômetro; Manufatura Aditiva; Manipuladores Robóticos; Processos WAAM.

1. INTRODUÇÃO

A manufatura aditiva a arco elétrico (*Wire Arc Additive Manufacturing* - WAAM) é um processo de manufatura baseado na fusão de um arame metálico consumível. A peça é produzida após a deposição de diversas camadas de metal fundido. Uma das maiores vantagens do processo WAAM é a possibilidade de fabricação de peças de larga escala utilizando somente equipamentos de solda robotizada, já comumente utilizados na indústria. O processo de fabricação é feito camada após camada e, geralmente, a deposição de cada conjunto de camadas é monitorada por um operador, que planeja a próxima etapa de deposição baseado na altura da parte já depositada. Além disso, para que as especificações da peça, como porosidade, resistência e

^{*} Este trabalho foi financiado em parte pela Shell Brasil Petróleo Ltda., Embrapii, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e 88887.136349/2017-00, FAPERJ, FAPEMIG e Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (CNPq).

tensões internas sejam atendidas, é essencial que os cordões depositados mantenham uma geometria consistente e que a superfície de cada camada não possua descontinuidade em forma de cavidades, como ilustra a Figura 1.



Figura 1. Cordões produzidos por um processo WAAM e um defeito destacado na camada.

Por meio da construção de um modelo 3D de cada camada, com o auxílio de sensores exteroceptivos, a trajetória a ser executada pelo robô pode ser replanejada para a camada

seguinte, mantendo constante a distância entre o bico da tocha de solda e a superfície em que ocorrerá a deposição (Ščetinec et al., 2021). As falhas na deposição também podem ser corrigidas por uma nova deposição para o preenchimento das cavidades, antes de executar a deposição da camada seguinte (Zhang et al., 2021).

Com o objetivo criar um modelo 3D das camadas depositadas, sensores a *laser* são comumente utilizados devido a medições mais precisas, como sensores LiDAR's ou perfilômetros ópticos. Contudo, dado que os modelos são gerados por varreduras das peças ou camadas depositadas, a qualidade da representatividade está diretamente ligada à calibração extrínsecas dos parâmetros da pose do sensor com respeito a um referencial inercial. Para sistemas em que o sensor é acoplado a um robô manipulador, a calibração é denominada *Calibração Hand-Eye*.

Um exemplo de *Calibração Hand-Eye* pode ser encontrado em (Xu et al., 2022), onde é apresentado um algoritmo baseado no escaneamento 2D de vários objetos com geometria comuns (e.g. barra, esfera e cilindro achatado) comparando-os com as formas reais. Da mesma forma, em (Li et al., 2021) foi proposto um método de calibração de um sensor LiDAR 2D baseado na reconstrução 3D. Em (Chen et al., 2016) foi proposto um método com duas etapas: uma coleta manual de dados, onde os autores mediram um disco; e a calibração automática da transformação homogênea do sensor a um sistema de coordenadas conhecido. Um sistema de calibração que usa três planos ortogonais é apresentado em (Carlson et al., 2015), em que os autores utilizam uma estratégia linear e iterativa. Em (Sharifzadeh et al., 2020), a calibração proposta, denominada *calibração Single Plane Hand-Eye*, utiliza um único plano desconhecido.

Neste artigo é apresentado um método de calibração *Single Plane Hand-Eye* (Sharifzadeh et al., 2020) para estimar os parâmetros da pose de um perfilômetro com respeito ao sistema de coordenadas do efetuador de um manipulador robótico industrial. Este sistema de calibração faz parte de um sistema de manufatura WAAM (Figura 2) desenvolvido pelo COPPE/UFRJ em parceria com a Shell Brasil Petróleo Ltda.. Uma das vantagens da técnica proposta é a simplicidade por utilizar somente um plano desconhecido e realizar a aquisição dos dados do perfilômetro seguindo trajetórias livres. A implementação irá compor um sistema de calibração semi-automático para um sistema WAAM. O algoritmo foi desenvolvido inicialmente usando o *software* Matlab, onde foram geradas as medições do sensor utilizadas como base de dados para testar a implementação. De modo a avaliar os efeitos de uma calibração pobre/imprecisa em um ambiente mais próximo ao real, foi utilizado o simulador CoppeliaSim v4.2.0 com o ROS¹, utilizando os modelos virtualizados do robô e sensor. Experimentos reais foram realizados usando um robô KUKA KR-90 escaneando uma chapa metálica plana. Os dados coletados foram utilizados pelo algoritmo de calibração e os resultados obtidos são apresentados e discutidos.

Dentre as principais contribuições deste artigo estão a implementação da técnica em um sistema real de manufatura aditiva, a utilização do simulador CoppeliaSim para avaliação do sistema, a implementação de algoritmos

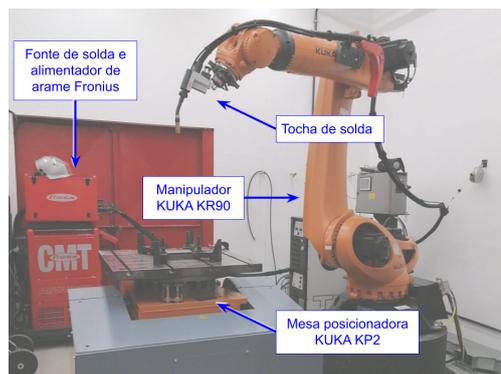


Figura 2. Célula de Manufatura Aditiva a Arco Elétrico.

encapsulados em pacotes do ROS, a definição de uma trajetória de calibração para a execução dos algoritmos de forma automatizada no processo de calibração, e a validação da calibração utilizando um objeto de forma conhecida.

O restante deste artigo está dividido em mais quatro seções. Na segunda Seção é apresentado detalhadamente o sistema de *Calibração Single Plane Hand-Eye*. Na terceira Seção é descrito o arcabouço experimental desenvolvido. Já na quarta Seção são apresentados os experimentos e simulações realizadas para validar o algoritmo implementado. Ao final, na quinta Seção, são apresentados a conclusão sobre os resultados obtidos e os trabalhos futuros.

2. CALIBRAÇÃO SINGLE PLANE HAND-EYE

Para descrever o sistema de calibração *Single Plane Hand-Eye*, são primeiramente definidas as nomenclaturas utilizadas para representar os sistemas de coordenadas e transformações usadas no restante deste artigo. O sistema de coordenadas do sensor é descrito pela letra S , o do efetuador pela letra E e da base do manipulador pela letra R , como ilustra a Figura 3. A transformação homogênea entre dois sistemas de coordenadas é definida como $\mathbf{T}_a^b \in SE(3)$, onde o índice inferior a descreve o sistema de coordenadas local e o índice superior o sistema de referência. Da mesma forma, $\mathbf{R}_a^b \in SO(3)$ denota a matriz de rotação de a com respeito a b e $\mathbf{p}_a^b \in \mathbb{R}^3$ é o vetor de posição de a com respeito a b .

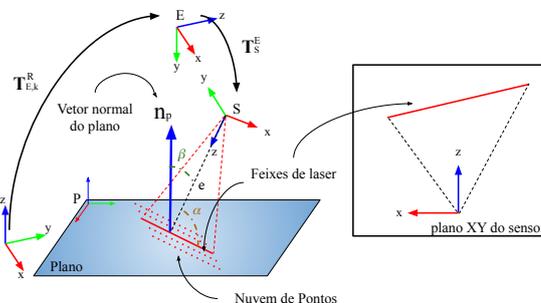


Figura 3. Representação do sistema de coordenadas do sensor, do efetuador e do robô, adaptado de (Sharifzadeh et al., 2020). O vetor \mathbf{n}_p é a normal ao plano e (d, α, β) são os parâmetros da medida do sensor em relação ao plano. Na sub-figura do lado direito é mostrado o plano XZ do sensor apresentando os dados coletados.

¹ <https://www.ros.org/>

O objetivo do algoritmo de calibração é estimar a transformação \mathbf{T}_S^E que representa a transformação do sensor S com respeito ao efetuador E . O sistema de coordenadas de referência utilizado pode ser qualquer um que esteja acoplado à árvore de transformações do robô manipulador, por exemplo, uma ferramenta ou um elo específico. No processo de calibração, os dados do sensor 2D, $\mathbf{p}_s = [p_{s,x}, p_{s,y}, p_{s,z}]^T$, são medidos em relação ao sistema de coordenadas local, onde o componente $p_{s,y}$ é igual a zero, devido aos dados serem representado no plano XZ do sensor, conforme apresentado no lado direito da Figura 3. Os pontos \mathbf{p}_s do sensor podem ser representados no sistema de coordenadas da base do robô usando a Equação (1):

$$\hat{\mathbf{p}}_R = \mathbf{T}_{E,k}^R \mathbf{T}_S^E \hat{\mathbf{p}}_S, \quad (1)$$

onde $\hat{\mathbf{p}}_R = [p_{R,x}, p_{R,y}, p_{R,z}, 1]^T$ é o vetor homogêneo da posição de uma medida do sensor com respeito ao robô, $\hat{\mathbf{p}}_S = [p_{S,x}, p_{S,y}, p_{S,z}, 1]^T$ é o vetor homogêneo da medida do sensor com respeito ao sistema local, e $\mathbf{T}_{E,k}^R$ é a transformação homogênea do efetuador para a configuração das juntas $\mathbf{q}[k] \in \mathbb{R}^6$ no instante de tempo k .

No processo de calibração considera-se que a transformação $\mathbf{T}_{E,k}^R$ é conhecida e dada pela cinemática do robô. Sendo assim, a transformação \mathbf{T}_S^E do sensor para o efetuador pode ser calculada usando os valores dos pontos $\hat{\mathbf{p}}_R$ aplicando um método iterativo de duas etapas, começando com uma estimativa inicial determinada, por exemplo, por inspeção visual. O princípio do algoritmo de calibração é conhecer a equação do plano formado pelos pontos medidos pelo sensor e minimizar a distância de cada ponto a este plano. O processo é iterativo e possui duas etapas: na primeira, o plano de calibração é calculado e, na segunda, encontra-se a transformação \mathbf{T}_S^E .

2.1 Primeira Etapa do Sistema de Calibração

Na primeira etapa do algoritmo de calibração, o Componente de Análise Principal (PCA) (Hastie et al., 2009) é aplicado para calcular o vetor unitário normal do plano \mathbf{n}_p usando o conjunto de pontos do sensor $\mathbf{p}_{R,i}$, para $i = 1, 2, \dots, N_p$, coletados em mais de uma varredura. Então, a equação do plano 3D é determinada pelo vetor normal \mathbf{n}_p e o centro de massa do conjunto de pontos \mathbf{p}_{CM} , conforme explicado na Figura 4.

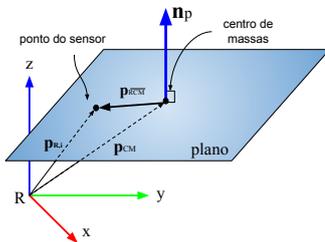


Figura 4. Diagrama de um plano com vetor unitário normal \mathbf{n}_p e o ponto do centro de massa \mathbf{p}_{CM} e qualquer ponto medido $\mathbf{p}_{R,i}$.

Considerando que os pontos \mathbf{p}_{CM} e $\mathbf{p}_{R,i}$ pertencem ao plano, o vetor \mathbf{n}_p e \mathbf{p}_{RCM} são ortogonais, onde $\mathbf{p}_{RCM} = \mathbf{p}_R - \mathbf{p}_{CM}$. Portanto, o produto escalar dos dois vetores é zero ($\mathbf{n}_p^T \mathbf{p}_{RCM} = 0$). Então, a equação do plano ($n_x x + n_y y + n_z z - d = 0$) é calculada como:

$$\mathbf{n}_p^T \mathbf{p}_{RCM} = 0, \quad (2)$$

$$\mathbf{n}_p^T (\mathbf{p}_R - \mathbf{p}_{CM}) = 0, \quad (3)$$

$$\mathbf{n}_p^T \mathbf{p}_R - \mathbf{n}_p^T \mathbf{p}_{CM} = 0, \quad (4)$$

$$\mathbf{n}_p^T \mathbf{p}_R = \mathbf{n}_p^T \mathbf{p}_{CM}, \quad (5)$$

$$\mathbf{n}_p^T \mathbf{p}_R = d, \quad (6)$$

onde $\mathbf{n}_p^T \mathbf{p}_{CM} = d$.

Expressando os pontos $\mathbf{p}_{R,i}$ e o vetor normal \mathbf{n}_p usando a forma homogênea $\hat{\mathbf{p}}_{R,i}$ e $\hat{\mathbf{n}}_p$, respectivamente, e substituindo $\hat{\mathbf{p}}_{R,i}$ pela Equação (1), a equação do plano pode ser reescrita como:

$$\hat{\mathbf{n}}_p^T \mathbf{T}_{E,k}^R \mathbf{T}_S^E \hat{\mathbf{p}}_S = d. \quad (7)$$

2.2 Segunda Etapa do Sistema de Calibração

Na segunda etapa do algoritmo de calibração, a matriz \mathbf{T}_S^E , na Equação (7) pode ser resolvida usando a solução de mínimos quadrados para um sistema na forma $\mathbf{A}\mathbf{w} = \mathbf{Y}$, se a pseudo-inversa da matriz \mathbf{A} tiver posto completo. Visto que os dados do sensor são adquiridos em perfil 2D, os pontos medidos estão no plano XZ do sensor. Como resultado, a segunda coluna da matriz \mathbf{T}_S^E não pode ser resolvida diretamente. No entanto, usando as restrições de ortogonalidade para a matriz de rotação, a segunda coluna do \mathbf{R}_S^E pode ser calculada pelo produto vetorial da primeira e da última coluna.

Reescrivendo a Equação (7) para empilhar cada coluna da matriz \mathbf{T}_S^E no vetor $\mathbf{w} \in \mathbb{R}^9$ sem a última linha e a segunda coluna, cria-se um sistema de equações dado por:

$$\begin{bmatrix} \mathbf{n}_p^T \mathbf{R}_{E,k}^R p_{S,x} & \mathbf{n}_p^T \mathbf{R}_{E,k}^R p_{S,z} & \mathbf{n}_p^T \mathbf{R}_{E,k}^R \end{bmatrix} \mathbf{w} = d - \mathbf{n}_p^T \mathbf{p}_{E,k}^R, \quad (8)$$

$$\mathbf{A}_k \mathbf{w} = Y_k, \quad (9)$$

onde $\mathbf{R}_{E,k}^R$ é a matriz de rotação do efetuador com respeito à base do robô para a medida k , $\mathbf{p}_{E,k}^R$ é o vetor de posição do efetuador para a base do robô para a medida k , $\mathbf{A} \in \mathbb{R}^{1 \times 9}$ é um vetor linha, e $Y_k \in \mathbb{R}$ é uma escalar.

Para que a Equação (9) tenha uma solução única pelo vetor \mathbf{w} , é necessário um conjunto de N_p pontos coletados em k diferentes poses. No entanto, isso pode ser alcançado empilhando k 's entradas na Eq. (9) de acordo com:

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_{N_p} \end{bmatrix} \mathbf{w} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{N_p} \end{bmatrix}, \quad (10)$$

$$\mathbf{A}\mathbf{w} = \mathbf{Y}. \quad (11)$$

Os parâmetros \mathbf{w}^* são encontradas pelo seguinte problema de minimização:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{w} - \mathbf{Y}\|_2. \quad (12)$$

Como \mathbf{w} contém apenas a primeira (R_1) e a última (R_3) colunas do \mathbf{R}_S^E , a segunda coluna (R_2) pode ser calculada pelo produto vetorial $R_2 = R_3 \times R_1$. A matriz \mathbf{R}_S^E resultante, em geral, não pertencerá ao $SO(3)$. Então, a matriz de rotação válida mais próxima pode ser encontrada

usando a matriz de raiz quadrada como apresentado em (Eggert et al., 1997):

$$\mathbf{R}_{closest} = \mathbf{R}_S^E ((\mathbf{R}_S^E)^T \mathbf{R}_S^E)^{-\frac{1}{2}}. \quad (13)$$

O procedimento de ortogonalização da matriz \mathbf{R}_S^E altera as entradas correspondentes em \mathbf{w}^* e os coeficientes resultantes não resolvem mais a Equação (11). Um segundo problema de otimização pode então ser formulado para re-estimar a parte translacional de \mathbf{w} , dada a parte rotacional ortogonalizada. Portanto, o vetor \mathbf{w} pode ser decomposto de acordo com:

$$\mathbf{w} = \begin{bmatrix} \mathbf{w}_R \\ \mathbf{w}_p \end{bmatrix}, \quad (14)$$

onde $\mathbf{w}_R \in \mathbb{R}^6$ é um vetor formado pelas colunas da matriz $\mathbf{R}_{closest}$ empilhadas e $\mathbf{w}_p \in \mathbb{R}^3$ é um vetor de posição.

Denotadas \mathbf{A}_R e \mathbf{A}_p como as colunas das partes de rotação e posição da matriz \mathbf{A} , respectivamente, o vetor ótimo \mathbf{w}_p^* é calculado resolvendo o problema de otimização:

$$\mathbf{A}_p \mathbf{w}_p = \mathbf{Y} - \mathbf{A}_R \mathbf{w}_R, \quad (15)$$

$$\mathbf{A}_p \mathbf{w}_p = \mathbf{Y}_p, \quad (16)$$

$$\mathbf{w}_p^* = \underset{\mathbf{w}_p}{\operatorname{argmin}} \|\mathbf{A}_p \mathbf{w}_p - \mathbf{Y}_p\|_2. \quad (17)$$

Portanto, a transformação homogênea final \mathbf{T}_S^E é definida como:

$$\mathbf{T}_S^E = \begin{bmatrix} \mathbf{R}_{closest} & \mathbf{w}_p \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (18)$$

no qual $\mathbf{0}_{1 \times 3}$ é um vetor linha de zeros.

A Figura 5 ilustra o diagrama do algoritmo de calibração *Single Plane Hand-Eye*. As entradas do processo são os escaneamentos realizados sobre o plano, expressos com respeito ao sistema de coordenadas do sensor, e a pose inicial dada, por exemplo, por estimativa visual. Já a saída é a transformação homogênea do sensor para o efetuador. As duas etapas do processo iterativo são apresentadas e as variáveis do processo são destacadas entre cada subprocesso.

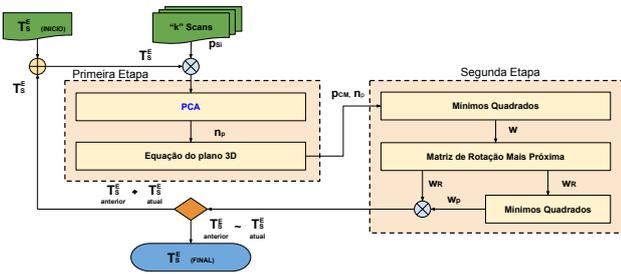


Figura 5. Diagrama do algoritmo de calibração *Single Plane Hand Eye*.

2.3 Convergência e Critério de Parada

O algoritmo *Single Plane Hand-Eye* é uma solução iterativa com duas etapas, conforme explicado anteriormente. Dada uma transformação homogênea \mathbf{T}_S^E obtida por inspeção visual, os pontos $\mathbf{p}_{R,i}$ serão calculados com erros e podem não pertencer ao mesmo plano. No entanto, o método PCA estima as componentes principais do conjunto de pontos,

onde a terceira componente corresponde ao vetor normal ao plano médio \mathbf{n}_p . A cada iteração do algoritmo de calibração, a transformação \mathbf{T}_S^E é calculada minimizando a distância de cada ponto $\mathbf{p}_{R,i}$ ao plano estimado \mathbf{n}_p . Dado que há variação suficiente na transformação homogênea \mathbf{T}_S^E e que a inclinação do plano real não é próxima de zero, de acordo com Sharifzadeh et al. (2020), o algoritmo irá convergir para a solução ótima.

Portanto, o critério de parada é definido entre duas transformações homogêneas consecutivas \mathbf{T}_S^E pela Norma de Frobenius (Böttcher and Wenzel, 2008), como:

$$F_n = \|\mathbf{T}_{S,t-1}^E - \mathbf{T}_{S,t}^E\|_{FN}, \quad (19)$$

$$F_n = \|\mathbf{B}\| = \left(\sum_{i,j} b_{i,j}^2 \right)^{\frac{1}{2}}, \quad (20)$$

onde $\mathbf{T}_{S,t-1}^E$ e $\mathbf{T}_{S,t}^E$ são definidas nas iterações $t-1$ e t .

Um segundo critério de parada que pode ser utilizado, como apresentado nos resultados é a distância média dos pontos $\mathbf{p}_{R,i}$ para o respectivo plano $(\mathbf{n}_{p,k}, \mathbf{p}_{CM,k})$, estimado pelo algoritmo de calibração a partir dos pontos $\mathbf{p}_{R,i}$. A média dos pontos é calculada a partir da métrica:

$$\mathcal{M}_{d_m} = \frac{1}{N_p} \sum_{i=1}^{N_p} \frac{|\mathbf{n}_p^T \mathbf{p}_{R,i} - \mathbf{n}_p^T \mathbf{p}_{CM}|}{\|\mathbf{n}_p\|}. \quad (21)$$

3. ARCABOUÇO EXPERIMENTAL

Esta seção descreve as ferramentas implementadas para testar e validar o algoritmo de calibração *Single Plane Hand-Eye*. Para isso, foram utilizados os softwares Matlab e CoppeliaSim v4.3.0². Nas simulações numéricas, os dados de medição do sensor foram gerados projetando os pontos em um plano escolhido. Estas medições são posteriormente utilizadas no algoritmo de calibração. No CoppeliaSim foi importado o modelo do manipulador KUKA KR-90, equipado com um sensor Hokuyo 2D LiDAR que foi ajustado com as mesmas configurações do perfilômetro scanCONTROL 2950-100/BL utilizado nos experimentos reais.

3.1 Simulador Numérico

Para a simulação numérica foi utilizado o software Matlab, onde os dados do laser foram gerados calculando a distância do sistema de coordenadas do sensor ao plano pre-determinado. Portanto, foram primeiramente geradas as poses do efetuador com respeito à base do robô. Para ter uma maior variação da pose do efetuador, descrito aqui como $\mathbf{p}_E^R = [x, y, z, \phi, \theta, \psi]^T$, foi gerado um caminho sobre o plano $(\mathbf{n}_p, \mathbf{p}_p)$. As componentes x e y da posição foram calculadas usando uma elipse paramétrica dada por:

$$x = r \cos(\alpha_f) + p_{p,x}, \quad (22)$$

$$y = 2r \sin(\alpha_f) + p_{p,y}, \quad (23)$$

onde α_f varia de $-\pi$ a π , $r = 0,25$ é o raio menor e $p_{p,x}$ e $p_{p,y}$ são as coordenadas da origem da elipse, determinadas

² <https://www.coppeliarobotics.com/>

pelos componentes x e y do vetor de posição do plano com respeito à base do robô.

A componente z da posição do vetor \mathbf{p}_E^R , bem como as componentes de orientação ϕ , θ e ψ , são calculadas usando uma função seno, como descrito a seguir:

$$z = 0,0075 \sin(8\alpha_f) + p_{p,z} + h, \quad (24)$$

$$\phi = \frac{\pi}{8} \sin(8\alpha_f) \quad (25)$$

$$\theta = \frac{\pi}{18} \sin(8\alpha_f) + \pi \quad (26)$$

$$\psi = \frac{\pi}{4} \sin(8\alpha_f) \quad (27)$$

onde $p_{p,z}$ é a componente z do vetor de posição do plano e h é um *offset* no eixo z acima da posição do plano.

A Figura 6 mostra o gráfico das componentes geradas pelas Equações (22) a (27). Na Figura 6(a), o gráfico apresenta o caminho gerado para a posição (x, y, z) em preto. Na Figura 6(b) o primeiro gráfico mostra o componente roll (ϕ) em vermelho, o segundo gráfico mostra o componente pitch (θ) em verde e o terceiro mostra a componente yaw (ψ) em azul.

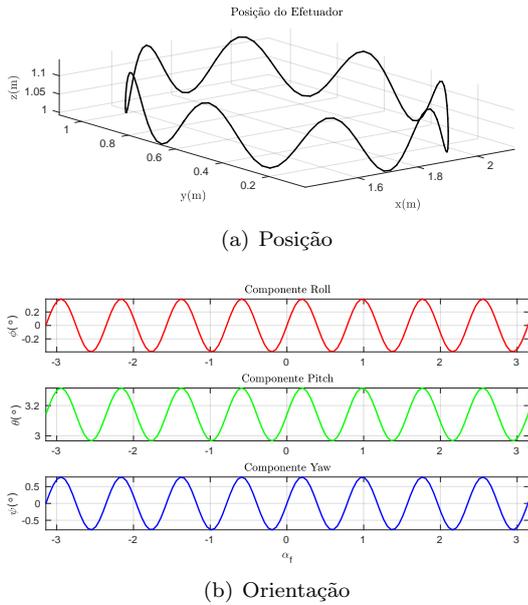


Figura 6. Caminhos gerados para a pose do efetuador. Em (a) é apresentado o gráfico do caminho formado pelas componentes da posição (x, y, z) e em (b) os gráficos das componentes da orientação (ϕ, θ, ψ) .

Após determinadas as poses do efetuador, dada a matriz de transformação homogênea \mathbf{T}_S^E verdadeira, os pontos referentes às medições do plano pelo sensor são calculados, conforme ilustra a Figura 7. O objetivo da projeção é definir os valores de cada feixe do *laser* com respeito ao sistema de coordenadas local S . Portanto, para uma sequência de pontos sobre o eixo x do sistema de coordenadas S ($\mathbf{p}_{S,i} = [p_{S,x}, 0, 0]^T$) são calculadas as distâncias $d_{p,i}$ para o plano a partir da Equação (7), como:

$$d_{p,i} = \frac{|\mathbf{n}_p^T \mathbf{p}_{R,i} - d|}{\|\mathbf{n}_p\|}. \quad (28)$$

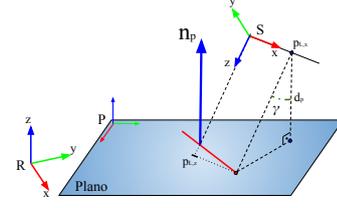


Figura 7. Diagrama utilizado como base para o cálculo das medições do feixe do *laser* projetado em um plano.

Para calcular a componente $p_{S,z}$ são utilizadas as propriedades geométricas do triângulo retângulo formado por d_p , $p_{z,z}$ e o plano normal, conforme mostra a Figura 7. Visto que $\cos(\gamma) = -\mathbf{n}_p^T \mathbf{z}_S$, em que \mathbf{z}_S é o vetor unitário na direção do eixo z do sistema S , a componente $p_{S,z}$ para cada ponto i é dada por:

$$p_{S,zi} = \frac{d_{p,i}}{-\mathbf{n}_p^T \mathbf{z}_S}. \quad (29)$$

Portanto, as medidas do sensor são geradas formando o vetor $\mathbf{p}_S = [p_{S,x}, 0, p_{S,z}]^T$, por meio das componentes $p_{S,x}$, definida por um conjunto de pontos sobre x , e $p_{S,z}$ conforme descrito nas Equações (28) e (29). A Figura 8 ilustra em verde os pontos projetados no plano aplicando a metodologia descrita.

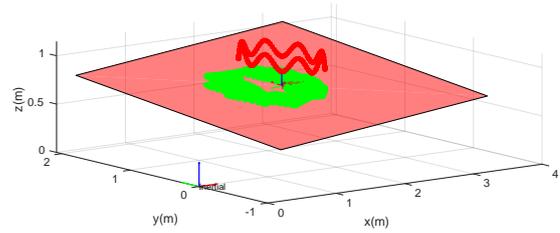


Figura 8. Nuvem de pontos e caminho do efetuador calculados com o auxílio do Matlab. O plano utilizado como base é apresentado em vermelho claro, os pontos medidos em verde e as posições do efetuador em vermelho.

3.2 Simulador CoppeliaSim

O simulador CoppeliaSim v4.3.0 é a outra ferramenta utilizada para verificar a calibração *Single Plane Hand-Eye*. Para substituir o perfilômetro, foi usado um Hokuyo 2D LiDAR com as mesmas configurações do sensor real acoplado em uma versão digital do KUKA KR-90. A Figura 9 mostra o braço robótico no ambiente simulado com o sensor. O braço do robô é controlado por meio do ângulo das articulações, o que permite que seja utilizado em paralelo com o robô real.

Para o processamento dos dados do sensor, bem como as transformações de cada link do manipulador, foi criado um meta-pacote em ROS chamado de *hand_eye_calibration_system*, com três pacotes ROS para processar os dados: *transform_to_pointcloud2*, *profile_mapping* e *calibration_tool*. O primeiro pacote foi implementado para converter a mensagem gerada pelo CoppeliaSim para o formato PointCloud2 do ROS. O segundo pacote cria um perfil 3D com a digitalização usando a biblioteca PCL

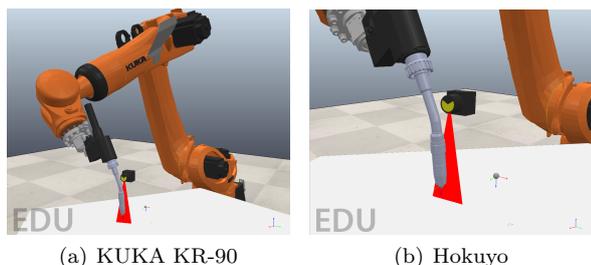


Figura 9. Digitalização do manipulador KUKA KR-90 em (a) e um sensor Hokuyo digitalizado e configurado com os parâmetros do sensor real.

fundindo as medições do *laser* em uma estrutura *KD-tree*. Por fim, o terceiro pacote é a primeira versão do algoritmo de calibração implementada usando as bibliotecas *openCv* e *PCL*. Todos os pacotes foram programados usando *C++*.

3.3 Ambiente real

A validação final dos experimentos ocorreu em uma célula de manufatura aditiva composta por um robô manipulador KUKA KR90, uma mesa posicionador KUKA KP2 e uma fonte de solda Fronius CMT. O perfilômetro foi acoplado à tocha de solda com o auxílio de um perfil de alumínio. Para compor o plano de calibração foi utilizado uma chapa de aço posicionada sobre a mesa KUKA KP2. De modo a reduzir os ruídos de na medição do plano pelo perfilômetro, foi aplicado à chapa uma camada de revelador branco fosco. A Figura 10 mostra o robô e os equipamentos utilizados na calibração.

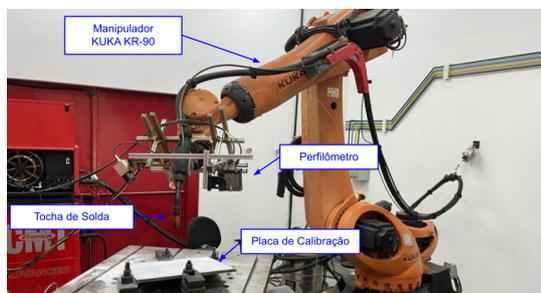


Figura 10. Manipulador KUKA KR-90 com o perfilômetro scanCONTROL 2950-100/BL.

O pacote ROS *kuka_experimental*³ foi utilizado para leitura da configuração do robô, enquanto o metapacote *scancontrol*⁴ foi utilizado para a realização de leituras do perfilômetro.

4. SIMULAÇÕES E EXPERIMENTOS

Nesta seção são apresentados os resultados obtidos por simulações usando os *softwares* Matlab e Coppeliasim, conforme descrito na Seção 3, e os resultados experimentais com o robô manipulador KUKA KR-90.

³ https://github.com/ros-industrial/kuka_experimental

⁴ <https://github.com/sam-x1/scancontrol>

4.1 Resultados de Simulação Numérica

Em uma primeira avaliação sobre os efeitos de uma calibração imprecisa do sensor, foi realizado o escaneamento de uma superfície plana com o auxílio do simulador Coppeliasim. A Figura 11 mostra as nuvens de pontos geradas pelo pacote *profile_mapping*. Na Figura 11(a), é apresentado o plano resultante calculado usando a transformação verdadeira fornecida pelo *software*. Na Figura 11(b) é apresentado o plano resultante calculado pela transformação errada. A escala de cores indica a altura em *z* dos pontos com respeito ao referencial inercial. O plano desformado (Figura 11(b)) possui cores diferentes em comparação com o plano da Figura 11(a). Com isso, foi verificado que para deslocamentos lineares, o erro de calibração não afeta a forma do perfil 3D. Contudo, quando o sensor realiza um movimento de rotação, os erros da calibração se acumulam e ficam mais evidentes.

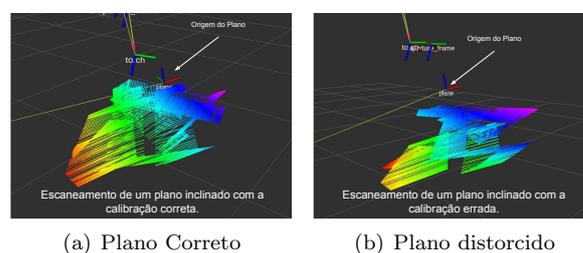


Figura 11. Escaneamento de um plano inclinado utilizando o simulador Coppeliasim. Em (a) é apresentado a nuvem de pontos com respeito à base do robô calculada pela pose correta do sensor e em (b) dada pela pose errada.

Utilizando os dados do sensor calculados com o caminho mostrado na Figura 6, o algoritmo de calibração *Single Plane Hand-Eye* foi executado usando duas transformações homogêneas T_S^E diferentes, conforme descrito na Tabela 1. A Figura 12 mostra a convergência de cada componente da pose do sensor com respeito ao efetuador a cada iteração.

Tabela 1. Resultados para a estimação das poses do sensor com respeito ao efetuador para as simulações no Matlab.

Pose	Ground-truth	Exp. 1		Exp. 2	
		Início	Final	Início	Final
$x(m)$	0,100	0,050	0,100	0,150	0,100
$y(m)$	0,100	0,110	0,099	0,050	0,099
$z(m)$	0,100	0,090	0,099	0,110	0,097
$\phi(o)$	5,000	2,700	5,001	6,000	5,003
$\theta(o)$	5,000	2,500	4,999	6,000	4,991
$\psi(o)$	5,000	3,000	5,004	6,000	5,026
Norma Frobenius		2,742e-02		6,945e-02	
Nº de Iterações		25		24	

Em ambos os experimentos, todas as componentes da pose do sensor com respeito ao efetuador convergiram para os valores *Ground-truth*, representado com uma linha traço-pontilhada, com um erro menor que 1 mm. O primeiro experimento, apresentado na Figura 12(a) convergiu em 25 iterações com um erro de $2,742e - 02$ calculado pela Norma de Frobenius (Böttcher and Wenzel, 2008). No segundo experimento (Figura 12(b)) a calibração convergiu em 24 iterações com um erro de $6,945e - 2$. A Figura 13 mostra a

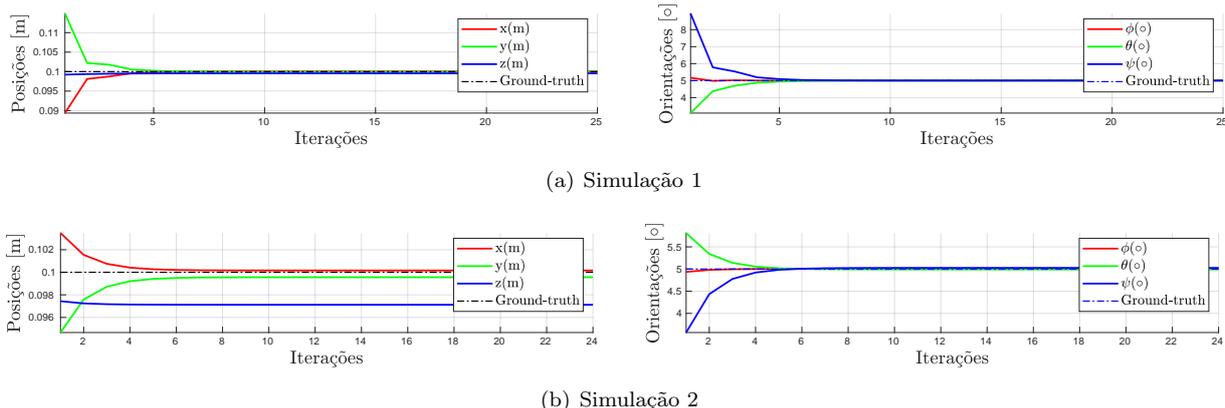


Figura 12. Componentes da pose do sensor com respeito ao efetuator a cada iteração para a primeira simulação em Matlab em (a) e a para a segunda em (b).

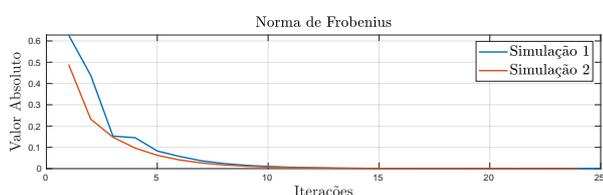


Figura 13. Norma de Frobenius para ambos os experimentos em Matlab.

Norma de Frobenius por iteração para os dois experimentos descritos.

4.2 Resultados Experimentais

Para testar e avaliar o funcionamento do algoritmo de calibração *Single Plane Hand-Eye* em ambiente real de operação foi realizado um experimento em uma célula de manufatura com o robô KUKA KR-90 equipado com um perfilômetro scanCONTROL 2950-100/BL. Para isso, foi realizado um escaneamento de uma chapa metálica plana apoiada sobre a mesa posicionadora KUKA KP2, como mostra a Figura 10. Para validar os resultados da calibração, foram coletados quatro pontos de contato da ferramenta com os quatro cantos da chapa metálica.

A Figura 14 mostra o resultado da calibração do sensor apresentando cada uma das componentes da pose do sensor com respeito ao efetuator, onde é observado que em 6 iterações o algoritmo converge para um valor que minimiza o erro $\|\mathbf{Aw} - \mathbf{Y}\|$. Já a Figura 15 mostra a convergência da Norma de Frobenius entre as transformações homogêneas de duas iterações consecutivas e a distância médias dos pontos escaneados com respeito ao plano normal estimado pelo método PCA na respectiva iteração.

As Tabelas 2 e 3 apresentam os resultados quantitativos do experimento de calibração do perfilômetro. Na Tabela 2, é possível notar que os resultados do algoritmo de calibração ficaram próximos da estimativa inicial realizada por medições espaciais e inspeção visual. As diferenças foram em torno de 2 a 3 mm para as componentes da pose e de 2 a 5 graus para as componentes de orientação.

A Tabela 3 mostra a média e desvio padrão das distâncias dos pontos medidos pelo sensor em relação ao plano *Ground-truth*, gerado pelo método PCA com os 4 pontos de contatos coletados sobre o plano. Os resultados da estimativa visual

Tabela 2. Resultado da calibração para o perfilômetro montado no robô manipulador KUKA KR-90.

Pose	$x(m)$	$y(m)$	$z(m)$	$\phi(°)$	$\theta(°)$	$\psi(°)$
Inicial	0,055	0,183	-0,183	177,620	0,000	-2,864
Final	0,078	0,161	-0,162	-178,450	-3,485	-8,112

foram de $5,33 \pm 4,80$ mm e os da calibração resultante do algoritmo *Single Plane Hand-Eye* foi de $2,76 \pm 2,21$ mm. A redução dos valores indica que os pontos do sensor transformados para o sistema de coordenadas inercial com a pose calibrada estão mais próximos e alinhados ao plano *Ground-truth* do que os pontos transformados com a estimativa visual.

Tabela 3. Distância média dos pontos medidos até o plano *ground-truth*.

Pose	Média (m)	Desvio Padrão (m)
Estimativa Visual	5,336e-03	4,798e-03
Calibração Final	2,763e-03	2,214e-03

Outra forma de validar a calibração utilizada foi o escaneamento de um tubo de aço de raio médio 109,25 mm a altura de 32,0 cm, ilustrado em azul-claro na Figura 16. As medições foram realizadas com o manipulador KUKA KR-90. A Figura 16 apresenta os escaneamentos do tubo usando a pose calibrada (em verde) e a não calibrada (em vermelho).

O tubo escaneado usando pose não calibrada apresentou um raio médio de 101,90 mm com um erro de 6,73%. Já a pose calibrada apresentou um raio médio de 107,19 com um erro de 1,88%. Note que a calibração *Single Plane Hand-Eye* apresentou uma considerável redução do erro, permitindo uma medição mais próxima do raio médio real. Além de representar geométrica e espacialmente o tubo escaneado, como pode ser observado pelo deslocamento no eixo z da Figura 16(a).

5. CONCLUSÕES E TRABALHOS FUTUROS

Neste artigo foi apresentado um método de calibração, denominado *Single Plane Hand-Eye*, para estimar os parâmetros da pose de um perfilômetro com respeito ao sistema de coordenadas do efetuator de um manipulador robótico industrial. As vantagens desta técnica consistem

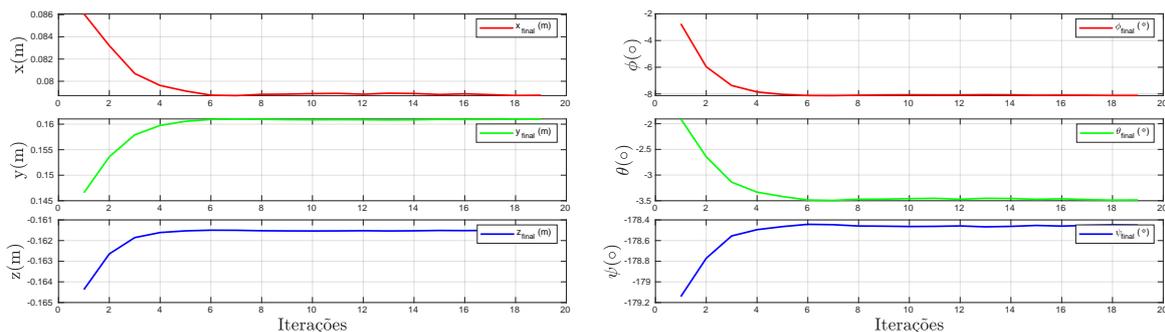
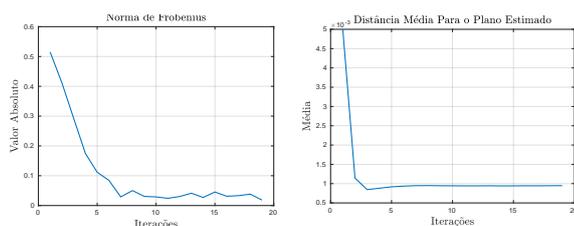
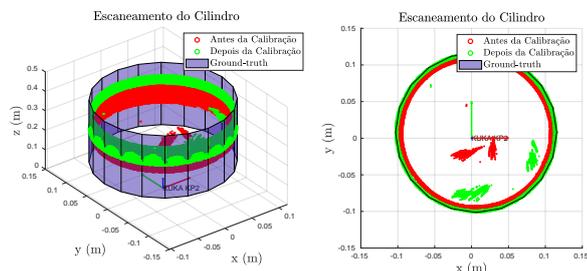


Figura 14. Componentes da pose do sensor com respeito ao efetuador a cada iteração para o experimento realizado com o robô Kuka KR-90.



(a) Norma de Frobenius (b) Distância Média

Figura 15. Em (a) é apresentado norma de Frobenius para experimento real com Kuka KR-90 em uma célula de manufatura e em (b) a distância média dos pontos medidos pelo sensor em relação ao plano estimado pelo PCA para o experimento real a cada iteração.



(a) Vista em Perspectiva (b) Vista Superior

Figura 16. Escaneamento de um tubo de aço, onde em (a) é apresentado a vista em perspectiva e em (b) a vista superior.

na simplicidade por usar somente um plano desconhecido e na utilização de varreduras livres como base. Porém, é necessário que os escaneamentos possuam variação suficiente para que o método convirja. O algoritmo de calibração descrito foi implementado usando o *software* Matlab. Os dados do sensor foram gerados projetando pontos do sistema de coordenadas do sensor em um plano pré-determinado com respeito à base do robô. A metodologia apresentou eficácia nos resultados numéricos.

Um segundo simulador foi usado para analisar os efeitos de uma calibração imprecisa do sensor com modelos digitalizados do robô e sensor. Experimentos reais com robô KUKA KR-90 foram realizados para testar e avaliar o algoritmo de calibração, validando os resultados com um plano *ground-truth* gerado por pontos de contato medidos sobre o plano real. Os resultados mostram que o método convergiu para um valor que minimiza o erro de calibração e que o plano estimado está alinhado ao plano real. Uma segunda validação foi realizada por meio do escaneamento

de um tubo de aço, em que a calibração apresentou uma melhor representatividade geométrica e espacial do objeto.

Como trabalhos futuros, uma nova versão do algoritmo será implementada em código aberto encapsulado como um pacote do ROS e irá compor um sistema de calibração semi-automático. Novos caminhos serão gerados e novas técnicas de otimização serão implementadas. O desempenho do método será avaliado e comparado com outras metodologias, e uma análise matemática do sistema será estudada verificando sua região de convergência.

Agradecimentos: Os autores gostariam de agradecer a equipe do LNTSOLD/UFRJ pela sua colaboração no tema WAAM.

REFERÊNCIAS

- Böttcher, A. and Wenzel, D. (2008). The Frobenius norm and the commutator. *Linear Algebra and its Applications*, 429(8-9), 1864–1885.
- Carlson, F.B., Johansson, R., and Robertsson, A. (2015). Six DOF eye-to-hand calibration from 2D measurements using planar constraints. In *2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 3628–3632.
- Chen, W., Du, J., Xiong, W., Wang, Y., Chia, S., Liu, B., Cheng, J., and Gu, Y. (2016). A noise-tolerant algorithm for robot-sensor calibration using a planar disk of arbitrary 3-D orientation. *IEEE Trans. on Automation Science and Engineering*, 15(1), 251–263.
- Eggert, D.W., Lorusso, A., and Fisher, R.B. (1997). Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine vision and applications*, 9(5), 272–290.
- Hastie, T., Tibshirani, R., Friedman, J.H., and Friedman, J.H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.
- Li, M., Du, Z., Ma, X., Dong, W., and Gao, Y. (2021). A robot hand-eye calibration method of line laser sensor based on 3D reconstruction. *Robotics and Computer-Integrated Manufacturing*, 71, 102136.
- Sharifzadeh, S., Biro, I., and Kinnell, P. (2020). Robust hand-eye calibration of 2D laser sensors using a single-plane calibration artefact. *Robotics and Computer-Integrated Manufacturing*, 61, 101823.
- Xu, J., Hoo, J.L., Dritsas, S., and Fernandez, J.G. (2022). Hand-eye calibration for 2D laser profile scanners using straight edges of common objects. *Robotics and Computer-Integrated Manufacturing*, 73, 102–221.
- Zhang, J., Wang, Q., Xiao, G., and Zhou, J. (2021). Filling path planning and polygon operations for wire arc additive manufacturing process. *Mathematical Problems in Engineering*, 2021, 6683319.
- Ščetinec, A., Klobčar, D., and Bračun, D. (2021). In-process path replanning and online layer height control through deposition arc current for gas metal arc based additive manufacturing. *Journal of Manufacturing Processes*, 64, 1169–1179.