



JOAO VIVAS CISALPINO



**CONTROLE DE VELOCIDADE EM IMPRESSORAS 3D
UTILIZANDO O PRINCÍPIO DE FEEDFORWARD PARA
OTIMIZAR A VELOCIDADE E QUALIDADE DE IMPRESSÃO.**

LAVRAS - MG

2023

JOAO VIVAS CISALPINO

**CONTROLE DE VELOCIDADE EM IMPRESSORAS 3D UTILIZANDO O
PRINCÍPIO DE FEEDFORWARD PARA OTIMIZAR A VELOCIDADE E
QUALIDADE DE IMPRESSÃO.**

Monografia apresentada à Universidade Federal
de Lavras, como parte das exigências Curso de
Engenharia Mecânica, para a obtenção do título
de Bacharel.

Prof. Dr. Wander Gustavo Rocha Vieira
Orientador

LAVRAS - MG

2023

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos
da Biblioteca Universitária da UFLA**

Cisalpino, Joao Vivas Cisalpino

CONTROLE DE VELOCIDADE EM IMPRESSORAS
3D UTILIZANDO O PRINCÍPIO DE FEEDFORWARD
PARA OTIMIZAR A VELOCIDADE E QUALIDADE DE
IMPRESSÃO. / Joao Vivas Cisalpino. 1^a ed. rev., atual. e
ampl. – Lavras : UFLA, 2023.

32 p. : il.

Trabalho de conclusão de curso(Graduação)–Universidade
Federal de Lavras, 2023.

Orientador: Prof. Dr. Wander Gustavo Rocha Vieira.
Bibliografia.

1. TCC

CDD-808.066

JOAO VIVAS CISALPINO

**CONTROLE DE VELOCIDADE EM IMPRESSORAS 3D UTILIZANDO O
PRINCÍPIO DE FEEDFORWARD PARA OTIMIZAR A VELOCIDADE E
QUALIDADE DE IMPRESSÃO.**

Monografia apresentada à Universidade Federal
de Lavras, como parte das exigências Curso de
Engenharia Mecânica, para a obtenção do título
de Bacharel.

APROVADO em Data da defesa 2023.

Prof. Dr. Henrique UFLA
Prof. Dr. Belisario UFLA

Prof. Dr. Wander Gustavo Rocha Vieira
Orientador

**LAVRAS - MG
2023**

Dedicatoria

AGRADECIMENTOS

Agradecimentos

RESUMO

Palavras-chave: Palavras chave

ABSTRACT

Keywords: Keywords

LISTA DE FIGURAS

Figura 2.1 – Distribuição de uso de MA nas indústrias	13
Figura 2.2 – Número de artigos publicados sobre FDM ao longo do tempo	14
Figura 2.3 – Distribuição das pesquisas sobre FDM	14
Figura 2.4 – Logo Klipper	15
Figura 2.5 – Comparação da resposta ao degrau e da resposta a escada	16
Figura 2.6 – Fluxograma FBF	17
Figura 2.7 – Resultados práticos da LPFBF	17
Figura 2.8 – Fluxograma RFBF	18
Figura 2.9 – Look ahead em uma grande mudança de direção	19
Figura 2.10 – Look ahead em uma pequena mudança de direção	19
Figura 2.11 – Curva de velocidades trapezoidal em uma sequência de movimentos	20
Figura 3.1 – Visualização das posições do sistema	26

LISTA DE TABELAS

Tabela 3.1 – Tabela de parâmetros opcionais da FMINCON	28
Tabela 3.2 – Resultados experimentais dos coeficientes de rigidez e amortecimento de correias dentadas utilizadas em impressoras 3D	28

SUMÁRIO

1	INTRODUÇÃO	11
2	REFERENCIAL TEÓRICO	12
2.1	Manufatura Aditiva	12
2.2	Fused Deposition Modeling	13
2.2.1	Feedforward	15
2.2.1.1	Input Shaper	16
2.2.1.2	filtered basis function (FBF)	16
2.2.1.3	limited-preview filtered B-splines	17
2.2.1.4	Robust Filtered Basis Functions (RFBF)	18
2.3	Geração de comando	18
2.3.1	Look ahead	18
2.3.2	Curvas de velocidade trapezoidal	19
2.3.3	Espaço de Estados	20
2.3.3.1	Runge-Kutta	20
2.3.4	Objective Function Optimization	20
3	METODOLOGIA	22
3.1	Matlab	22
3.1.1	fmincon	22
3.2	Geração de Comando	22
3.2.1	Leitura Gcode	22
3.3	lookahead	23
3.3.1	Curva rapezoidal de velocidade	23
3.3.2	Interpolação	25
3.4	Modelagem dinâmica de uma impressora 3D	25
3.4.1	Espaço de estados	26
3.5	Otimização FMINCON	27
3.5.1	Restrições lineares e limites de borda	27
3.5.2	Restrições não lineares	27
3.5.3	Função objetivo (Objective Function)	27
3.5.4	Cofiguração da Fmincon	28
3.6	Dados base e configurações	28

4	RESULTADOS E DISCUSSÃO	29
4.1	Geração de Comando	29
4.2	Simulação Base	29
4.3	Otimização Fmincon	29
4.4	Performance Computacional	29
5	CONCLUSÃO	30
	REFERÊNCIAS	31

1 INTRODUÇÃO

texto

Entretanto, uma das grandes limitações da impressão 3D, principalmente do tipo **Fused Deposition Modeling**, é o tempo de impressão, que ainda limita muito o tamanho de peças impressas em um tempo razoável, geralmente sendo necessário reduzir muito a resolução da impressão.

Existe hoje, dentro da academia e das comunidades **maker**, uma busca por impressoras capazes de imprimir cada vez mais rápido mantendo a qualidade de impressão. Além da possível diminuição do tempo de impressão, além disso a capacidade de imprimir rapidamente acaba proporcionando uma capacidade de aumentar a qualidade de impressão proporcional à diferença entre a velocidade máxima e a velocidade de impressão.

Portanto, vê-se relevante à procura por técnicas que permitam capacidades superiores de qualidade e velocidade de impressão, que flexibilizam a tecnologia e aumentam a capacidade da utilização comercial viável da tecnologia.

2 REFERENCIAL TEÓRICO

2.1 Manufatura Aditiva

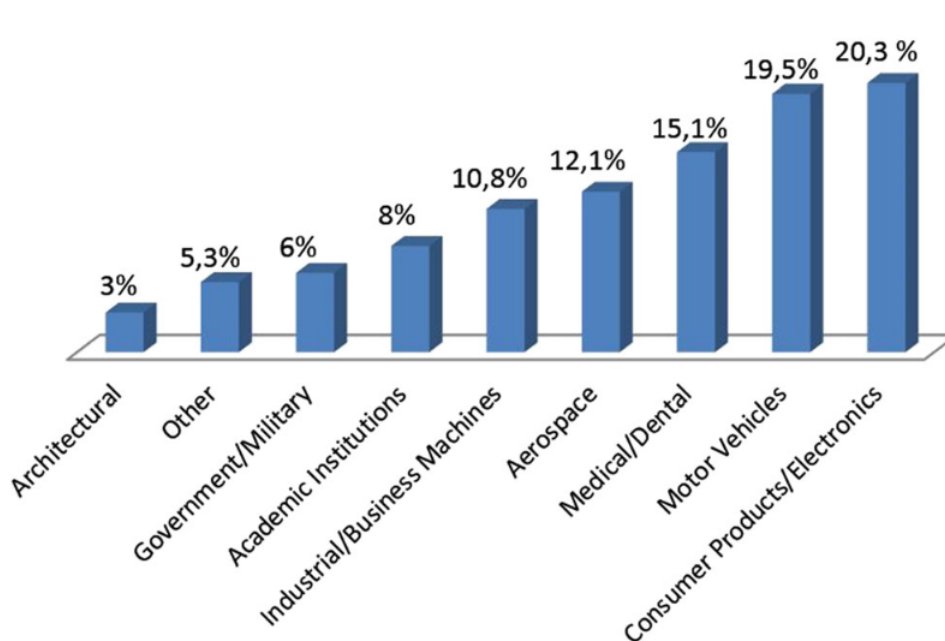
O princípio básico **por trás** da manufatura aditiva (MA) é a capacidade de fabricar um modelo tridimensional **diretamente**, sem a necessidade de um **planejamento** do processo, a partir de um modelo tridimensional digital normalmente criado a partir de Computer Aided Design (CAD). Uma das características principais da MA é a rapidez na qual é possível criar protótipo diretamente de modelos digitais, por conta disso, em um contexto de desenvolvimento de produto, o termo prototipagem rápida era utilizado. Entretanto, conforme a MA foi se aperfeiçoando era perceptível a capacidade dessas tecnologias não só se aterem à produção de protótipos, mas também de peças utilizadas em produtos finais. Além disso, o termo não considerava o princípio básico que unia essas tecnologias e assim o termo manufatura aditiva foi apresentado e adotado pela American Society for Testing and Materials (ASTM) GIBSON *et al.* (2015).

Apesar da manufatura aditiva ter sido criada a mais de 30 anos, apenas a partir de 2009, quando a última patente mais relevante de ***Fused Deposition Modeling*** (FDM) expirou. Com isso, vários entusiastas começaram a desenvolver essa tecnologia de uma maneira "caseira", com o forte movimento **RepRap**. Por conta dessa característica "caseira" e um senso de comunidade, os desenvolvimentos em sua maioria eram de caráter ***Open Source*** e com uma mentalidade de **acessibilizar** essa tecnologia para as pessoas. Com os avanços tecnológicos feitos pela comunidade, empresas, pessoas e a mídia começaram a se interessar cada vez mais, aumentando a popularidade das impressoras 3D e por consequência trazendo muita atenção para a manufatura aditiva, que a partir daí, mais pesquisas, mais empresas se interessavam em desenvolver esse tipo de tecnologia, não somente FDM (ATTARAN, 2017).

Atualmente, existe uma grande **variedade** de tecnologias e processos de manufatura aditiva. Estes variam na maneira com que depositam o material, nos princípios físicos que utilizam e nos materiais que podem ser utilizados. Como mencionado anteriormente, um dos métodos de manufatura aditiva mais populares é a tecnologia FDM, entretanto existem diversas outras tecnologias que tem crescido muito em popularidade como as tecnologias baseadas na cura seletiva de resinas, ***stereolithography*** (SLA) e ***Masked stereolithography Apparatus*** (MSLA), além de outras tecnologias menos acessíveis, mas com aplicações em diversas indústrias, como por exemplo ***selective laser melting*** (SLM) e ***selective laser sintering*** (SLS) (BIKAS; STA-

VROPOULOS; CHRYSSOLOURIS, 2016). Na figura 2.1 podemos observar a distribuição do uso de tecnologias MA por tipo de indústria.

Figura 2.1 – Distribuição de uso de MA nas indústrias



Fonte: BIKAS; STAVROPOULOS; CHRYSSOLOURIS, 2016

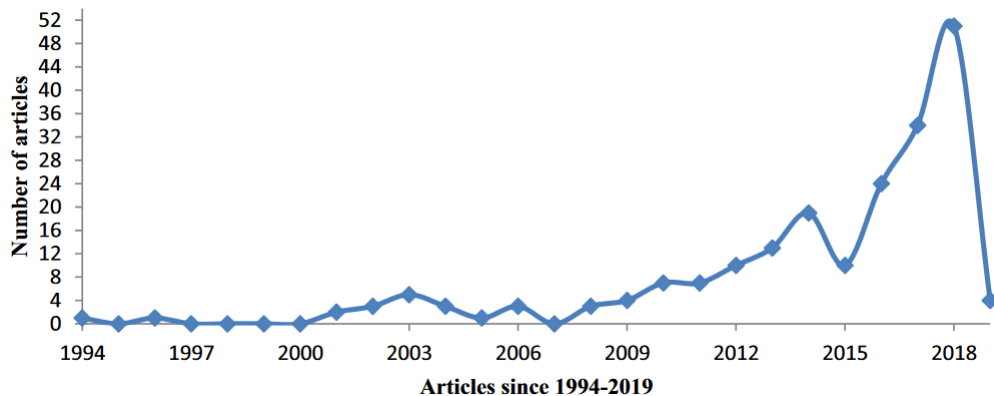
2.2 Fused Deposition Modeling

Fused Deposition Modeling (FDM) ou Fused Filament Fabrication (FFF) é uma das tecnologias MA mais populares como mencionado anteriormente. Ela se consiste por depositar material através de um processo onde um filamento de material é forçado dentro de uma câmara através, geralmente, de rolos dentados onde em uma região específica esse material é liquefeito. Por conta da pressão criada pelo filamento adentrando a câmara, ainda no estado sólido como um pistão, o material liquefeito é extrudado através de um bocal, comumente fabricado de bronze. Então, o filamento liquefeito é depositado em uma plataforma de forma a percorrer a trajetória desejada utilizando mecanismos movidos de forma controlada, geralmente por motores de passos. O processo é repetido camada por camada, de forma que elas estejam apoiadas por camadas anteriores e a primeira camada continue fixa na plataforma ou cama, até que o processo finalize (TURNER et al., 2014) (TURNER; STRONG; GOLD, 2014).

O trabalho de (VYAVAHARE *et al.*, 2020) apresenta algumas características sobre o desenvolvimento científico sobre FDM ao longo dos anos, tendo como base 211 artigos diferentes

de 1994 a 2020. É apresentado um grande salto no número de artigos publicados no tema em anos recentes (2015 a 2018) (figura 2.2), com 56% dos temas trabalhados em torno da otimização de parâmetros de impressão, acompanhado de 17% de trabalhos relacionados a aplicações utilizando o processo FDM, enquanto apenas 12% são relacionados a avanços tecnológicos relacionados como observado na figura 2.3.

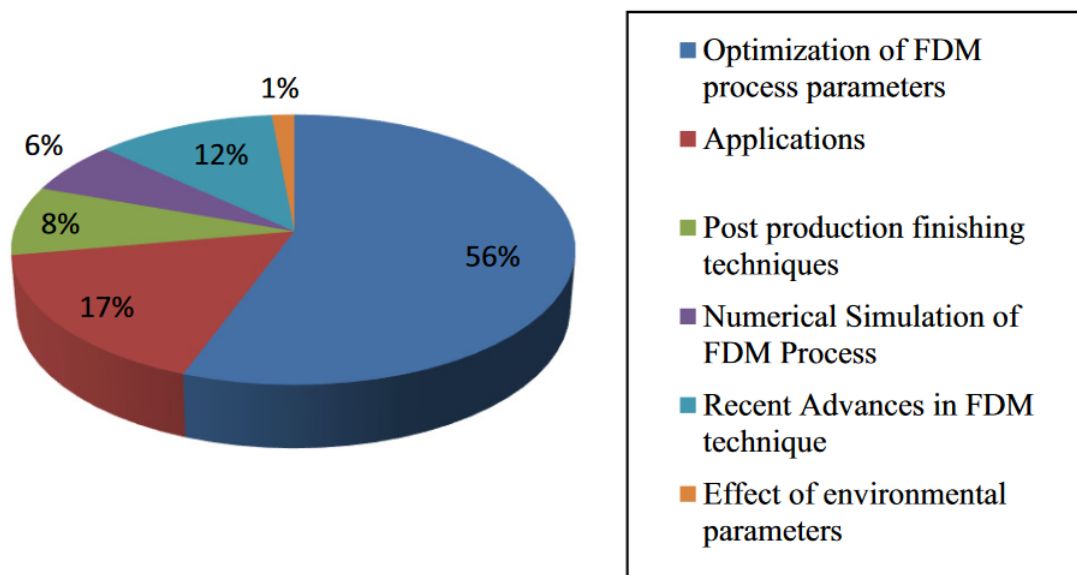
Figura 2.2 – Número de artigos publicados sobre FDM ao longo do tempo



Fonte: VYAVAHARE *et al.*, 2020

Figura 2.3 – Distribuição das pesquisas sobre FDM

Percentage contribution of domain specific literature



Fonte: VYAVAHARE *et al.*, 2020

Podemos separar, de maneira simplificada, a porção de software de impressoras 3D FDM em três principais etapas: fatiamento (slicing), geração de comando e controle. A etapa de fatiamento envolve a topologia e a criação de instruções a partir do modelo nessa fase onde

se decide a sequência de movimentos e outros eventos. Já na etapa de geração de comando, as instruções criadas pelo fatiador (slicer) na etapa anterior são interpretadas e os comandos detalhados são gerados, por exemplo as curvas de velocidade. Esses comandos são utilizados para movimentar os motores e outros equipamentos da impressora. Na etapa de controle, uma etapa relativamente nova nas impressoras 3D mais acessíveis, técnicas de controle são utilizadas para se diminuir vibrações e variações indesejadas em quaisquer parâmetros controlados, como a temperatura do bico. Um dos grandes avanços nessa etapa foi a implementação da técnica de Input Shaping por um firmware Open Source de impressora 3D chamado Klipper (figura 2.4). Após a inclusão dessa etapa, principalmente no controle dinâmico da impressora, as capacidades de velocidades e qualidade chegaram a outro patamar se comparados a impressoras que não implementam essa etapa (KLIPPER, 2017a).



Figura 2.4 – Logo Klipper



Fonte: KLIPPER, 2017a

2.2.1 Feedforward

Dentre os métodos de controle em aplicações FDM o Feedforward é o mais eficiente dada as limitações de custo em impressoras 3D comuns e é capaz de ter um impacto maior em sistemas conhecidos e sensíveis ao erro, onde buscam corrigir o erro antes que ele aconteça. O método de feedback é mais eficiente em diminuir o impacto de excitações externas ou desconhecidas, entretanto não consegue prever os efeitos do sistema, se encaixando melhor em aplicações de usinagem utilizando CNCs (Computer Numerical Control), onde o valor dos equipamentos é maior e as forças envolvidas no corte influenciam mais do que as vibrações do próprio sistema. Já no caso das impressoras, quase 100% dos efeitos é causado pelo próprio sistema. As principais limitações da aplicação de técnicas Feedforward em impressoras 3D são a dificuldade de montar um modelo representativo, a exigência computacional elevada e

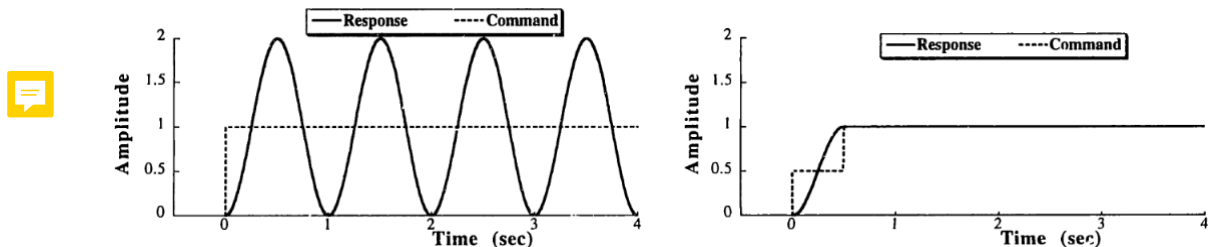


por fim a necessidade da simulação se estender do início ao fim, pela dependência de se basear no estado inicial da impressão (RAMANI; EDOIMIOYA; OKWUDIRE, 2020; DUAN; YOON; OKWUDIRE, 2018).

2.2.1.1 Input Shaper

Conhecendo a trajetória desejada e conhecendo características do sistema é possível computar os comandos fornecidos para calcular uma série de comandos, levando em consideração as características do sistema para que o comando de referência seja modificado de forma à trajetória final ser o mais próximo possível do comando de referência. Entretanto, ao invés de computar todo o comando de referência, é possível obter um comando modificado em tempo real através de um filtro. Uma das abordagens desse tipo de filtro de comando é o Input Shaper, onde variados Shapers são construídos levando em consideração diferentes objetivos e restrições (SINGHOSE, 1997). Podemos ver na figura 2.5 um exemplo comparativo das respostas ao degrau e da função escada aplicada pelo shaper.

Figura 2.5 – Comparação da resposta ao degrau e da resposta a escada



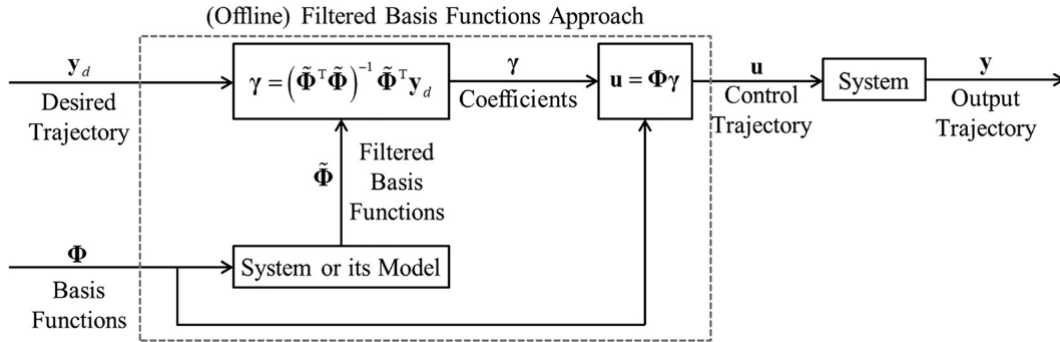
Fonte: SINGHOSE, 1997

Essa abordagem vem sendo explorada na comunidade Maker a partir de 2017 quando a última patente desse método expirou, e tem aprimorado a área como um todo, empurrando os limites anteriores de velocidade e precisão, sendo popularizada pelo Klipper (KLIPPER, 2017b).

2.2.1.2 filtered basis function (FBF)

O método FBF necessita que a trajetória a ser rastreada seja totalmente conhecida e que a trajetória controlada possa ser expressa como uma combinação linear de funções base possuindo coeficientes desconhecidos. As funções base são utilizadas em um controle feedforward utilizando o modelo dinâmico do sistema e selecionando os coeficientes de maneira a minimizar os erros dada uma trajetória desejada (figura 2.6). (RAMANI *et al.*, 2017)

Figura 2.6 – Fluxograma FBF

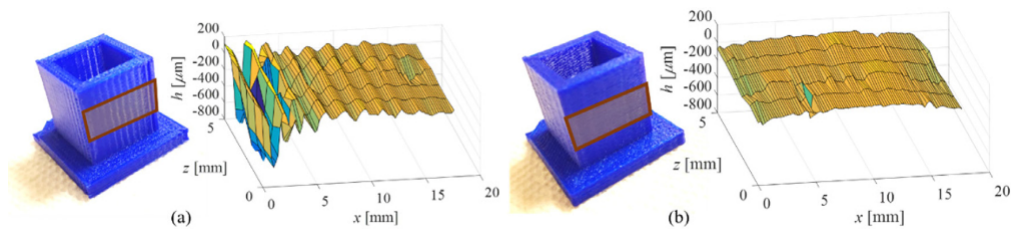
Fonte: RAMANI *et al.*, 2017

2.2.1.3 limited-preview filtered B-splines

Uma das maiores dificuldades que os métodos avançados para o controle feedforward de trajetórias é a necessidade de se conhecer completamente a trajetória desejada, o que implica em um grande custo computacional, principalmente em situações onde são necessárias uma grande quantidade de amostras da trajetória, por exemplo em casos de alta resolução e casos de longa duração. O limited-preview filtered B-splines (LPFBF) divide a trajetória desejada em subgrupos com um número menor de amostras e utiliza um algoritmo de *receding horizon* para calcular recursivamente os coeficientes da função B-spline que minimizam os erros de trajetória (DUAN; YOON; OKWUDIRE, 2018).

A partir dessa otimização da divisão da trajetória em subgrupos, esse método conseguiu ser testado utilizando uma impressora 3D de verdade com modelos simples. Apresentando resultados promissores apresentados na figura 2.7.

Figura 2.7 – Resultados práticos da LPFBF

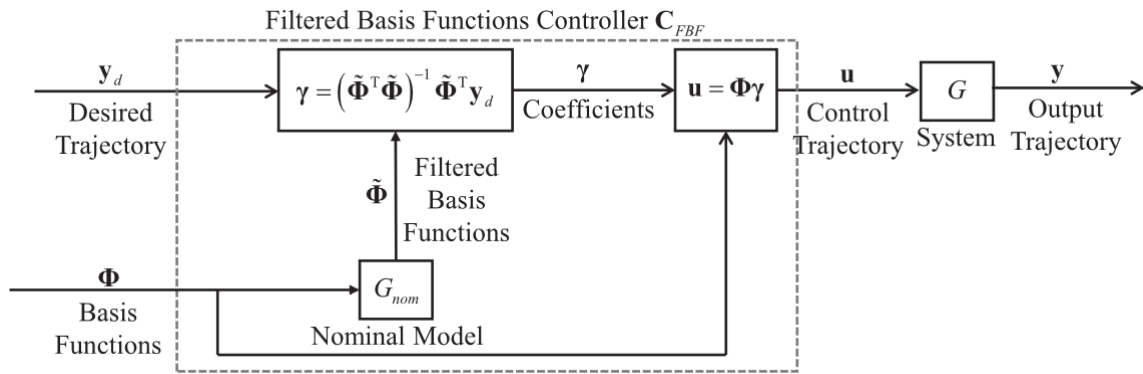


Fonte: DUAN; YOON; OKWUDIRE, 2018

2.2.1.4 Robust Filtered Basis Functions (RFBF)

Uma vulnerabilidade de algoritmos feedforward, especialmente a abordagem FBF e LPFBF é degradação da precisão a partir de imprecisões no modelo e perturbações na dinâmica da impressora. Levando isso em consideração RAMANI; EDOIMIOYA; OKWUDIRE (2020) propôs um algoritmo construído com base no LPFBF. Nesse método é utilizado uma função de custo a ser **minimizads** de maneira a alterar o modelo dinâmico dentro do fluxograma (figura 2.8), ao invés do modelo dinâmico original como era feito na LPFBF, assim se tornando mais robusta e resiliente a imprecisões e perturbações.

Figura 2.8 – Fluxograma RFBF



Fonte: RAMANI; EDOIMIOYA; OKWUDIRE, 2020

2.3 Geração de comando

A geração de comando é o processo que coordena a ativação dos atuadores, motores, dentre outros componentes de uma impressora. Ele recebe como base uma série de comandos que precisam ser interpretados e interpolados. Esse processo é responsável pelo controle de velocidade, aceleração dentre outras atividades que variam no tempo (YU *et al.*, 2020).

2.3.1 Look ahead

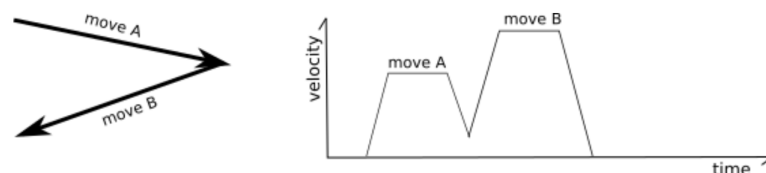
No processo de impressão 3D são fornecidos para a impressora uma sequência de pontos no espaço e limitações de velocidade entre os mesmos. A velocidade nos pontos é compartilhada entre trajetos em sequência, o que torna considerá-los independentemente ineficiente, introduzindo aceleração e desaceleração desnecessária impactando negativamente no tempo de impressão e na qualidade da peça impressa. O algoritmo Look Ahead procura manter o máximo de velocidade possível entre movimentos distintos, evitando acelerações e desacelerações

desnecessárias, como podemos observar nas figuras 2.10 e 2.9, apesar de ser necessário um pré-processamento desses pontos que introduzem um custo computacional maior (YU et al. 2020) (YU *et al.*, 2020; KLIPPER, 2017b).

2.3.2 Curvas de velocidade trapezoidal

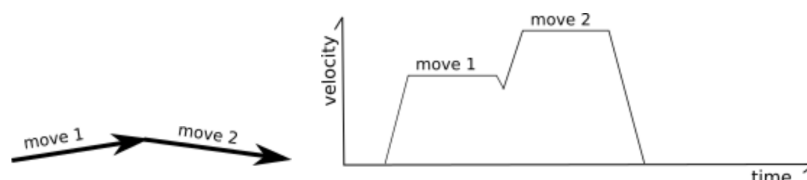
As impressoras 3D entre outros equipamentos, como máquinas CNC, necessitam de um planejamento de velocidade, pois o **Gcode** fornece apenas as velocidades desejadas de cada movimento, enquanto o algoritmo de lookahead calcula as velocidades de junção entre os movimentos, portanto ainda se faz necessário planejar o comportamento da velocidade ao longo do tempo do trajeto entre a velocidade inicial e final do movimento (figura 2.11). Uma das maneiras mais simples para a criação dessa curva de velocidade é a criação de uma curva trapezoidal, onde podemos separar o setor em 3 segmentos de aceleração constante. Em um primeiro momento uma crescente de velocidade até a velocidade desejada, seguido de um segmento de velocidade constante e por fim um segmento de desaceleração constante até a velocidade final. Alguns ajustes são feitos para as diferentes condições de velocidade inicial, final e velocidade máxima atingida com uma determinada aceleração máxima, que pode fazer com que se diminua a quantidade de segmentos (YU *et al.*, 2020; KLIPPER, 2017b).

Figura 2.9 – Look ahead em uma grande mudança de direção



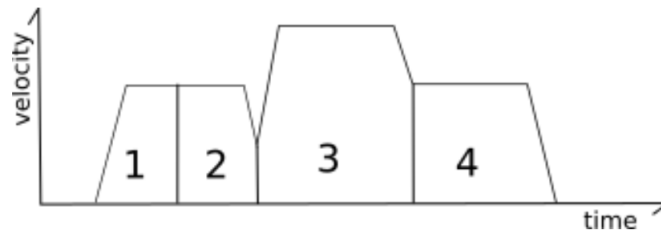
Fonte: KLIPPER, 2017b

Figura 2.10 – Look ahead em uma pequena mudança de direção



Fonte: KLIPPER, 2017b

Figura 2.11 – Curva de velocidades trapezoidal em uma sequência de movimentos



Fonte: KLIPPER, 2017b

2.3.3 Espaço de Estados

A maioria dos sistemas dinâmicos pode ser escritos através de uma formulação chamada de espaço de estados, que tem como objetivo expressar modelos de equações diferenciais parciais (EDP) ou ordinárias (EDO) de ordem superior como um conjunto de EDPs ou EDOs de primeira ordem. Essa formulação é construída a partir de um princípio de autoregressão das equações. Na equação 2.1 podemos observar uma EDO de segunda ordem representando um sistema massa mola simples, logo abaixo (2.2) a mesma equação representada na formulação de espaço de estados (HAMILTON, 1994).

$$m\ddot{x} + c\dot{x} + kx = f(t) \quad (2.1)$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ k/m & c/m \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} f(t) \quad (2.2)$$

2.3.3.1 Runge-Kutta

Runge Kutta é um algoritmo de integração numérica baseada na série de Taylor onde, nas formulações explícitas são calculados 4 coeficientes que são utilizados para estimar o próximo ponto, assim caracterizando um algoritmo iterativo de valor inicial (BETTIS, 1979; DORMAND; PRINCE, 1980).

2.3.4 Objective Function Optimization

Funções objetivo de otimização são elementos importantes na execução de algoritmos de otimização. Alguns estudos apontam que a qualidade e o número de variáveis de projeto cruciais para o resultado da otimização, além disso outros sub-parâmetros também são relevantes

como restrições, limites e valores iniciais. A coerência entre estes fatores depende habilidade e de uma visão abrangente de seu criador (ALBAGHDADI; BAHAROM; SULAIMAN, 2021).



3 METODOLOGIA



3.1 Matlab

MATLAB é um software interativo para computação numérica que possui uma série de ferramentas, funções, visualisadores, ferramentas para debugging, estrutura de dados, entre outros auxílios que facilitam o desenvolvimento e o estudo de atividades que utilizam a computação numérica. Por conta desses facilitadores, o MATLAB é amplamente utilizado na indústria e no meio acadêmico. (HIGHAM; HIGHAM, 2016) Dada essa característica e a existência da função FMINCON a disposição no ambiente MATLAB, foi feita a escolha de se utilizá-lo para a construção do código presente neste trabalho.

3.1.1 fmincon

Como o modelo matemático a ser otimizado é multivariável e possuindo restrições não-lineares, a função FMINCON do ambiente do MATLAB é utilizada para otimizar as variações de velocidade de forma a diminuir o erro de trajetória associado às flexibilidades do sistema que causam perturbações e vibrações indesejadas. É uma função baseada em gradientes que busca por todos os mínimos locais de uma região que satisfaz outras restrições estipuladas (ALBAGHDADI; BAHAROM; SULAIMAN, 2021). Ela utiliza um conjunto de restrições superiores e inferiores para cada ponto e otimiza a função considerando as restrições estabelecidas pela função não linear, utilizando as equações de movimento para encontrar a solução da EDO de maneira a otimizar os parâmetros. Também permite uma série de configurações incluindo a escolha de diferentes algoritmos de otimização entre outros parâmetros sobre a função.

3.2 Geração de Comando

3.2.1 Leitura Gcode

Foi considerado no mapeamento do Gcode apenas comandos G1, extraindo as informações dos eixos X, Y e do *feedrate* (F). Com base nesses valores uma matriz 3 por n é criada, n sendo o número de comandos lidos do arquivo Gcode.



3.3 lookahead

Para a construção da curva de velocidade trapezoidal a partir da matriz de posições e *feedrate* é necessário o cálculo das direções dos movimentos a serem realizados. As direções são representadas por vetores unitários calculados a partir do vetor deslocamento dividido pelo mesmo vetor normalizado (3.1). Sendo o vetor deslocamento obtido pela diferença entre dois vetores de posição sequenciais.

$$dir_{vector} = \frac{v}{norm(v)} \quad (3.1)$$

Essas direções são utilizadas para o cálculo das velocidades pelo *look ahead*, considerando os parâmetros de desvio máximo de junção (δ_{junc}), que corresponde a distância máxima entre o vertice da curva e a extremidade do raio e a aceleração máxima (acc_{max}) (figura ??). Importante ressaltar que se trata de um desvio virtual do caminho, utilizado para calcular a velocidade na curva (v_{junc}), que é calculada a partir da equação 3.2, enquanto o raio (R) é calculado a partir da equação 3.3, considerando α como o ângulo entre os vetores de velocidade, neste caso calculados a partir dos vetores de direção unitária (dir_k) (equação 3.4) calculado na etapa anterior.

$$v_{junc} = \sqrt{acc_{max} * R} \quad (3.2)$$

$$R = \delta_{junc} \frac{\sin\left(\frac{\alpha}{2}\right)}{1 - \sin\left(\frac{\alpha}{2}\right)} \quad (3.3)$$

$$\sin\left(\frac{\alpha}{2}\right) = \left(\frac{norm(dir_1 + dir_2)}{2}\right) \quad (3.4)$$

Por fim, é necessário empregar uma lógica onde a velocidade de junta nunca seja maior que a velocidade de entrada ou velocidade de saída.

3.3.1 Curva trapezoidal de velocidade

A partir dessa matriz de posições e, agora também velocidades, utilizamos a função responsável por gerar a curva trapezoidal de velocidades.

Essa função separa o deslocamento total do movimento em 3 fases de aceleração constante. Na primeira fase a velocidade trazida da velocidade inicial até a velocidade desejada a

aceleração constante, na segunda fase a velocidade é mantida constante na velocidade desejada e por fim na terceira fase a velocidade é levada da velocidade desejada até a velocidade final. Entretanto, em algumas situações pode não ser possível alcançar a velocidade desejada e o perfil se limitar a duas fases, outras condições onde alguma das velocidades é igual a velocidade desejada faz com que a quantidade de fases visíveis seja reduzida.

Para identificar se a velocidade desejada será alcançada é calculado a velocidade pico (v_p) que é obtida extrapolando retas com as inclinações da aceleração na velocidade inicial e final. É possível obter a velocidade de pico através da equação 3.5.

$$v_p = \sqrt{\frac{(v_i^2 + v_f^2)}{2} + acc * des_{tot}} \quad (3.5)$$

A partir da comparação da velocidade pico com a velocidade desejada, indicada pelo *feedrate* disponibilizado no Gcode, é possível determinar o padrão da curva de velocidade deste movimento. Caso a velocidade de pico for maior do que a velocidade desejada, temos 3 fases de deslocamento que podem ser calculadas pelas equações 3.6 e 3.7. Caso a velocidade de pico seja igual ou menor do que a velocidade desejada, teremos 2 fases de deslocamento que são calculadas a partir da equação 3.6.

$$des_{segment} = \frac{(v_f^2 - v_i^2)}{(2 * acc_{segment})} \quad (3.6)$$

$$des_{middle} = des_{total} - (des_{up} + des_{down}) \quad (3.7)$$

É possível calcular também os intervalos de tempo dessas fases, através das equações 3.8 e 3.9.

$$dt_{segment} = \frac{(v_f - v_i)}{acc_{segment}} \quad (3.8)$$

$$dt_d = \frac{des_d}{v_d} \quad (3.9)$$

Além disso é calculado a variação de velocidade nos intervalos pela equação 3.10.

$$\Delta vel = v_f - v_i \quad (3.10)$$

Esses passos resultam em uma nova matriz contendo informações sobre a variação da posição, do tempo, da velocidade e sobre a aceleração e direção de deslocamento nos pontos iniciais e finais do Gcode e também nos pontos onde existe uma alteração na aceleração.

3.3.2 Interpolação

A partir dessa matriz, é utilizada uma função de interpolação para dividir cada intervalo dessa matriz em intervalos menores baseados em um passo de tempo definido para esta interpolação. Assim criando-se uma nova matriz dos dados interpolados. Para se dividir esses intervalos é possível utilizar a equação 3.11 calculando o número de passos neste intervalo, anexando à matriz os passos de tempo e por fim o restante do intervalo, calculado pela equação 3.12. Com base nestes passos de tempo, é possível calcular o deslocamento para cada um destes passos através da equação 3.13.

$$N_{steps} = \lceil \frac{\Delta t_i}{\Delta t_{step_size}} - 1 \rceil \quad (3.11)$$

$$\Delta t_{laststep} = \Delta t_i - \Delta t_{step_size} * N_{steps} \quad (3.12)$$

$$\Delta des_i = \Delta v_i * \Delta t_i + \frac{acc_{segment} * \Delta t_i^2}{2} \quad (3.13)$$

A partir dos vetores de direção e da função acumuladora que se consite em acumular os valores de um vetor. Obtemos uma matriz de posições, velocidades e tempo.

$$\begin{aligned} v_0 &= init_{value} + v_0 \\ v_k &= v_k + v_{k-1} \end{aligned} \quad (3.14)$$

3.4 Modelagem dinâmica de uma impressora 3D

Para a modelagem do sistema mecânico são consideradas as seguintes simplificações:

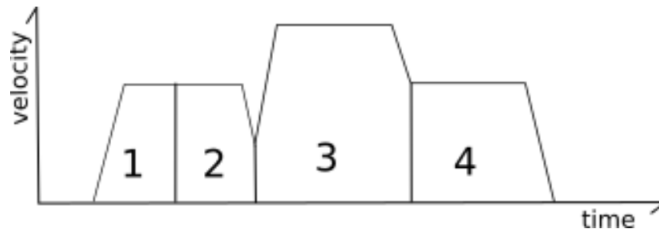
- Não existe escorregamento nem perda de potência na interação entre a polia e a correia
- A correia apresenta um comportamento equivalente à uma mola e um amortecedor em paralelo

- O bico injetor é um corpo rígido uniforme de geometria simples
- A correia está acoplada nos dois lados da peça que se movimenta nos trilhos, entretanto como a correia só permite o tensionamento ela será considerada como um conjunto mola amortecedor simples



Para a modelagem dinâmica dos eixos X e Y da impressora 3D, é considerado que os eixos são completamente independentes, a flexibilidade da correia é aproximada utilizando um conjunto mola amortecedor e a transmissão de movimento e torque dos motores é considerada como ideal e não será abordada. Assim duas posições de estudo surgem para cada eixo, uma delas representa a posição ideal, caso o sistema não possuísse nenhuma flexibilidade ou perda, que também é a posição desejada pelo usuário (px_b). A segunda posição considera as forças inerciais e a flexibilidade introduzida pela correia, ou seja, a posição real simulada pelo modelo e no caso empírico a posição real (px) como na figura 3.1.

Figura 3.1 – Visualização das posições do sistema



$$m\ddot{p}x + c(\dot{p}x - \dot{p}x_b) + k(px - px_b) = 0$$

$$\ddot{p}x = -\frac{c}{m}(\dot{p}x - \dot{p}x_b) - \frac{k}{m}(px - px_b)$$

$$\ddot{p}x = -\frac{c}{m}\dot{p}x + \frac{c}{m}\dot{p}x_b - \frac{k}{m}px + \frac{k}{m}px_b$$

$$\ddot{p}x = -\frac{c}{m}\dot{p}x - \frac{k}{m}px + \frac{c}{m}\dot{p}x_b + \frac{k}{m}px_b \quad (3.15)$$

3.4.1 Espaço de estados

A formulação de espaço de estados foi utilizada com intuito de facilitar as operações e a a solução do sistema, dado sua característica de dividir uma equação diferencial de ordem superior em um sistema de equações diferenciais de ordem 1. O modelo dinâmico do sistema

é apresentado na formulação de espaço de estados na equação 3.16, baseado na equação 3.15, utilizando a mesma equação para a base do eixo y.

$$\begin{bmatrix} \dot{p}x \\ \ddot{p}x \\ \dot{p}y \\ \ddot{p}y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_x}{m_x} & -\frac{c_x}{m_x} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{k_x}{m_x} & -\frac{c_x}{m_x} \end{bmatrix} \begin{bmatrix} px \\ \dot{p}x \\ py \\ \dot{p}y \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_x}{m_x} & \frac{c_x}{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{k_x}{m_x} & \frac{c_x}{m_x} \end{bmatrix} \begin{bmatrix} px_b \\ \dot{p}x_b \\ py_b \\ \dot{p}y_b \end{bmatrix} \quad (3.16)$$

Em uma notação simplificada temos a equação 3.17

$$\dot{x} = A * x + B * u \quad (3.17)$$

3.5 Otimização FMINCON

3.5.1 Restrições lineares e limites de borda

Não foi utilizado nenhuma restrição linear nessa otimização. Os limites superiores (upperbound) e inferiores (lowerbound) foram definidos baseados nos limites físicos da impressora para as posições x, xb e y,yb, enquanto os limites para o vetor de variação de tempo foi estipulado uma margem configuravel baseada no próprio vetor de variação do tempo. Estas definições estão representadas pelo conjunto de equações 3.18.

$$\dot{x} = A * x + B * u \quad (3.18)$$

3.5.2 Restrições não lineares

***** Blocked *****

3.5.3 Função objetivo (Objective Function)

A função objetivo (FO) foi escolhida visando minimizar a diferença entre as posições reais simuladas e a posição desejada pelo usuário. Para isso, foi utilizada a equação 3.19, onde seu resultado é um escalar que quantifica o resultado do vetor a ser otimizado, permitindo que o algoritmo procure alterar os valores do vetor, sem quebrar suas restrições e limites, de maneira a minizar o resultado da FO.

$$Ob = (x - x_{desejado}) * (x - x_{desejado})' + (y - y_{desejado}) * (y - y_{desejado})' \quad (3.19)$$

3.5.4 Configuração da Fmincon

Foi utilizado as seguintes configurações da função fmincon:

****configurações****

Tabela 3.1 – Tabela de parâmetros opcionais da FMINCON

Opção	Valor
TolFun	0.000000001
MaxIter	100000
Display	iter
DiffMinChange	0.0001
Algorithm	interior-point
StepTolerance	1e-12
MaxFunEvals	700000

3.6 Dados base e configurações

aceleração máxima desvio de junção delta tempo kx ky bx by mx my

Levando em conta que modelo teste possui um comprimento de 20 cm e que os valores de rigidez e de amortecimento podem variar com a tensão aplicada, marca e tempo de uso, serão utilizados os valores aproximados para duas configurações (Tabela 2). A massa é estimada como sendo de 250 g que é uma estimativa com base nos conjuntos extrusora e hot end mais comuns.

Tabela 3.2 – Resultados experimentais dos coeficientes de rigidez e amortecimento de correias dentadas utilizadas em impressoras 3D

teste	teste2	teste3
1	2	3
4	5	x+1

4 RESULTADOS E DISCUSSÃO

4.1 Geração de Comando

4.2 Simulação Base

4.3 Otimização Fmincon

4.4 Performance Computacional

5 CONCLUSÃO

O presente trabalho não conseguiu alcançar seus objetivos de maneira completa dada a limitação da implementação nos limites de aceleração. Entretanto, os resultados obtidos mostram o potencial da técnica iterativa de otimização, necessitando de uma execução melhor para ser capaz de atuar de forma completa. Outra dificuldade da técnica é o tempo de processamento, onde não é possível realizar o processamento em real-time sendo necessário utilizá-lo em conjunto com um pré processamento do Gcode.

REFERÊNCIAS

- ALBAGHDADI, A. M.; BAHAROM, M. B.; SULAIMAN, S. A. bin. Parameter design optimization of the crank-rocker engine using the fmincon function in matlab. In: IOP PUBLISHING. **IOP Conference Series: Materials Science and Engineering**. [S.l.], 2021. v. 1088, n. 1, p. 012072.
- ATTARAN, M. The rise of 3-d printing: The advantages of additive manufacturing over traditional manufacturing. **Business horizons**, Elsevier, v. 60, n. 5, p. 677–688, 2017.
- BETTIS, D. Runge-kutta algorithms for oscillatory problems. **Zeitschrift für angewandte Mathematik und Physik ZAMP**, Springer, v. 30, n. 4, p. 699–704, 1979.
- BIKAS, H.; STAVROPOULOS, P.; CHRYSSOLOURIS, G. Additive manufacturing methods and modelling approaches: a critical review. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 83, p. 389–405, 2016.
- DORMAND, J. R.; PRINCE, P. J. A family of embedded runge-kutta formulae. **Journal of computational and applied mathematics**, Elsevier, v. 6, n. 1, p. 19–26, 1980.
- DUAN, M.; YOON, D.; OKWUDIRE, C. E. A limited-preview filtered b-spline approach to tracking control—with application to vibration-induced error compensation of a 3d printer. **Mechatronics**, Elsevier, v. 56, p. 287–296, 2018.
- GIBSON, I. *et al.* Applications for additive manufacture. **Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing**, Springer, p. 451–474, 2015.
- HAMILTON, J. D. State-space models. **Handbook of econometrics**, Elsevier, v. 4, p. 3039–3080, 1994.
- HIGHAM, D. J.; HIGHAM, N. J. **MATLAB guide**. [S.l.]: SIAM, 2016.
- KLIPPER. **Klipper Documentation**. 2017. Disponível em: <<https://www.klipper3d.org/>>.
- KLIPPER. **Klipper Kinematics Documentation**. 2017. Disponível em: <<https://www.klipper3d.org/Kinematics.html>>.
- RAMANI, K. S. *et al.* Tracking control of linear time-invariant nonminimum phase systems using filtered basis functions. **Journal of Dynamic Systems, Measurement, and Control**, American Society of Mechanical Engineers Digital Collection, v. 139, n. 1, 2017.
- RAMANI, K. S.; EDOIMIOYA, N.; OKWUDIRE, C. E. A robust filtered basis functions approach for feedforward tracking control—with application to a vibration-prone 3-d printer. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 25, n. 5, p. 2556–2564, 2020.
- SINGHOSE, W. E. **Command generation for flexible systems**. Tese (Doutorado) — Massachusetts Institute of Technology, 1997.
- TURNER, B. N.; STRONG, R.; GOLD, S. A. A review of melt extrusion additive manufacturing processes: I. process design and modeling. **Rapid prototyping journal**, Emerald Group Publishing Limited, v. 20, n. 3, p. 192–204, 2014.
- VYAVAHARE, S. *et al.* Fused deposition modelling: a review. **Rapid Prototyping Journal**, Emerald Publishing Limited, v. 26, n. 1, p. 176–201, 2020.

YU, K. *et al.* Application of the five-phase s-curve velocity model on fdm three-dimensional printer. In: IEEE. **2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)**. [S.l.], 2020. p. 1365–1371.