



JOAO VIVAS CISALPINO

**CONTROLE DE TRAJETÓRIA DE IMPRESSORAS 3D
UTILIZANDO ALGORITIMO ITERATIVO E
PROGRAMAÇÃO NÃO LINEAR**

LAVRAS - MG

2023

JOAO VIVAS CISALPINO

**CONTROLE DE TRAJETÓRIA DE IMPRESSORAS 3D UTILIZANDO
ALGORITIMO ITERATIVO E PROGRAMAÇÃO NÃO LINEAR**

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 TRAJETÓRIA DE IMPRESSORAS 3D
UTILIZANDO ALGORITIMO ITERATIVO E PROGRAMA-
ÇÃO NÃO LINEAR / Joao Vivas Cisalpino. 1^a ed. rev., atual.
e ampl. – Lavras : UFLA, 2023.

51 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 TRAJETÓRIA DE IMPRESSORAS 3D UTILIZANDO
ALGORITIMO ITERATIVO E PROGRAMAÇÃO NÃO LINEAR**

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

APROVADO em 21 de Julho de 2023.

Prof. Dr. Henrique UFLA

Prof. Dr. Belisario UFLA

Prof. Dr. Wander Gustavo Rocha Vieira
Orientador

**LAVRAS - MG
2023**

RESUMO

A manufatura aditiva, com ênfase na impressão 3D e no método "Fused Deposition Modeling"(FDM), destaca-se como uma tecnologia altamente promissora para a produção de peças complexas em quantidades reduzidas. Ela impulsiona a iterabilidade e a produção descentralizada sob demanda, encontrando aplicação em diversos setores, como engenharia, medicina e a indústria aeroespacial. O objetivo principal do trabalho é investigar e desenvolver uma metodologia para atuação de controle na geração de comandos em impressoras 3D de forma a possibilitar maiores velocidades e garantindo a precisão dimensional das peças produzidas. O desenvolvimento desse controle incorpora um algoritmo iterativo que minimiza os desvios entre o trajeto desejado para a impressora e o trajeto efetivamente percorrido, levando em consideração a dinâmica da impressora. Isso resulta em peças impressas de maior qualidade, permitindo aos usuários selecionar velocidades de impressão mais elevadas sem comprometer a qualidade do produto final, em comparação com a ausência desse método. O algoritmo foi submetido a testes de sensibilidade de parâmetros para avaliar seu desempenho sob diferentes condições. Os resultados desses testes foram satisfatórios e indicaram a capacidade do método de minimizar os desvios da impressora, com base em uma modelagem precisa. Além disso, o método apresenta potencial para expansão, possibilitando a consideração de nuances do modelo em várias regiões da impressora, aprimorando ainda mais sua eficácia.

Palavras-chave: Manufatura aditiva Modelagem por Fusão e Deposição (FDM) Geração de comandos Impressão 3D Algoritmos de controle Modelagem dinâmica Matlab fmincon

ABSTRACT

Additive manufacturing, with an emphasis on 3D printing and the "Fused Deposition Modeling"(FDM) method, stands out as a highly promising technology for the production of complex parts in reduced quantities. It drives iterability and decentralized on-demand production, finding applications in various sectors, such as engineering, medicine, and the aerospace industry. The primary goal of this work is to investigate and develop a methodology for control action in generating commands in 3D printers to enable higher speeds while ensuring the dimensional accuracy of the produced parts. The development of this control incorporates an iterative algorithm that minimizes deviations between the desired path for the printer and the path actually traveled, taking into account the printer's dynamics. This results in higher quality printed parts, allowing users to select higher printing speeds without compromising the quality of the final product compared to the absence of this method. The algorithm underwent parameter sensitivity tests to assess its performance under different conditions. The results of these tests were satisfactory and indicated the method's capability to minimize printer deviations based on accurate modeling. Furthermore, the method has potential for expansion, allowing for the consideration of nuances in the model in various printer regions, further enhancing its effectiveness.

Keywords: Additive manufacturing Fused Deposition Modeling (FDM) Command generation 3D printing Control algorithms Dynamic modeling Matlab fmincon

LISTA DE FIGURAS

Figura 1.1 – Fluxograma etapas Impressora 3D FDM	10
Figura 2.1 – Distribuição de uso de MA nas industrias	12
Figura 2.2 – Princípio e processo de impressão para FDM	13
Figura 2.3 – Indicação dos componentes de uma impressora 3D	15
Figura 2.4 – Interface do fatiador PrusaSlicer	15
Figura 2.5 – Exemplo de um arquivo Gcode	17
Figura 2.6 – Perfil de velocidade - Curva trapezoidal de velocidade	18
Figura 2.7 – Comparação da resposta ao degrau e da resposta a escada	20
Figura 2.8 – Fluxograma FBF	21
Figura 2.9 – Fluxograma RFBF	22
Figura 2.10 – Ilustração integração implícita	23
Figura 3.1 – Curva de velocidade trapezoidal	27
Figura 3.2 – Curva de velocidade triangular	27
Figura 4.1 – Caso referência - Comportamento no tempo das velocidades em x e y da ponta e da referência	33
Figura 4.2 – Caso referência - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	34
Figura 4.3 – Caso referência - Caminho percorrido x vs y da ponta e da referência	34
Figura 4.4 – Caso referência - Num de fun x Viabilidade	35
Figura 4.5 – Caso 1A - Comportamento no tempo das velocidades em x e y da ponta e da referência	36
Figura 4.6 – Caso 1B - Comportamento no tempo das velocidades em x e y da ponta e da referência	36
Figura 4.7 – Caso 1C - Comportamento no tempo das velocidades em x e y da ponta e da referência	36
Figura 4.8 – Caso 1A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	37
Figura 4.9 – Caso 1C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	37
Figura 4.10 – Caso 1A - Caminho percorrido x vs y da ponta e da referência	38
Figura 4.11 – Caso 1C - Caminho percorrido x vs y da ponta e da referência	38

Figura 4.12 – Caso 1 - Num de fun x Viabilidade	39
Figura 4.13 – Caso 2B - Caminho percorrido x vs y da ponta e da referência	40
Figura 4.14 – Caso 2C - Caminho percorrido x vs y da ponta e da referência	40
Figura 4.15 – Caso 2 - Num de fun x Viabilidade	41
Figura 4.16 – Caso 3A - Comportamento no tempo das velocidades em x e y da ponta e da referência	42
Figura 4.17 – Caso 3 - Num de fun x Viabilidade	42
Figura 4.18 – Caso 4B - Comportamento no tempo das velocidades em x e y da ponta e da referência	43
Figura 4.19 – Caso 4B - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	43
Figura 4.20 – Caso 4 - Num de fun x Viabilidade	44
Figura 4.21 – Caso 5A - Comportamento no tempo das velocidades em x e y da ponta e da referência	44
Figura 4.22 – Caso 5B - Comportamento no tempo das velocidades em x e y da ponta e da referência	45
Figura 4.23 – Caso 5C - Comportamento no tempo das velocidades em x e y da ponta e da referência	45
Figura 4.24 – Caso 5A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	46
Figura 4.25 – Caso 5B - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	46
Figura 4.26 – Caso 5C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	47
Figura 4.27 – Caso 5 - Num de fun x Viabilidade	47

LISTA DE TABELAS

Tabela 3.1 – Parâmetros modificados na função FMINCON.	29
Tabela 3.2 – Parâmetros referência dos Estudos de Caso.	31
Tabela 3.3 – Estudos de Caso	31
Tabela 3.4 – Especificações do computador	32

SUMÁRIO

1	INTRODUÇÃO	9
2	REFERENCIAL TEÓRICO	11
2.1	Manufatura Aditiva	11
2.2	<i>Fused Deposition Modeling</i>	12
2.2.1	Geração de Comando	16
2.2.2	Geração de trajetória	17
2.2.3	Curvas de velocidade trapezoidal	18
2.2.4	Técnicas de Controle	19
2.2.5	<i>Feedforward</i>	19
2.2.5.1	<i>Input Shaper</i>	19
2.2.5.2	<i>Filtered basis function (FBF)</i>	20
2.3	Espaço de Estados	22
2.4	Integração implícita utilizando programação não linear	22
3	METODOLOGIA	25
3.1	Modelagem dinâmica de uma impressora 3D	25
3.1.1	Espaço de estados	25
3.2	Geração de Comando	26
3.2.1	Curva trapezoidal de velocidade	26
3.2.2	Interpolação	28
3.3	FMINCON	29
3.3.1	Configurações da função	29
3.3.2	Restrições lineares e limites de borda	29
3.3.3	Restrições não lineares	29
3.4	Solução da trajetória da base	30
3.5	Simulação de controle de impressora 3D	30
3.5.1	Caso base	30
4	RESULTADOS E DISCUSSÃO	33
4.1	Resultados	33
4.1.1	Caso Base	33
4.1.2	Caso 1 - Variação da frequência	35
4.1.3	Caso 2 - Variação do coeficiente de amortecimento	39

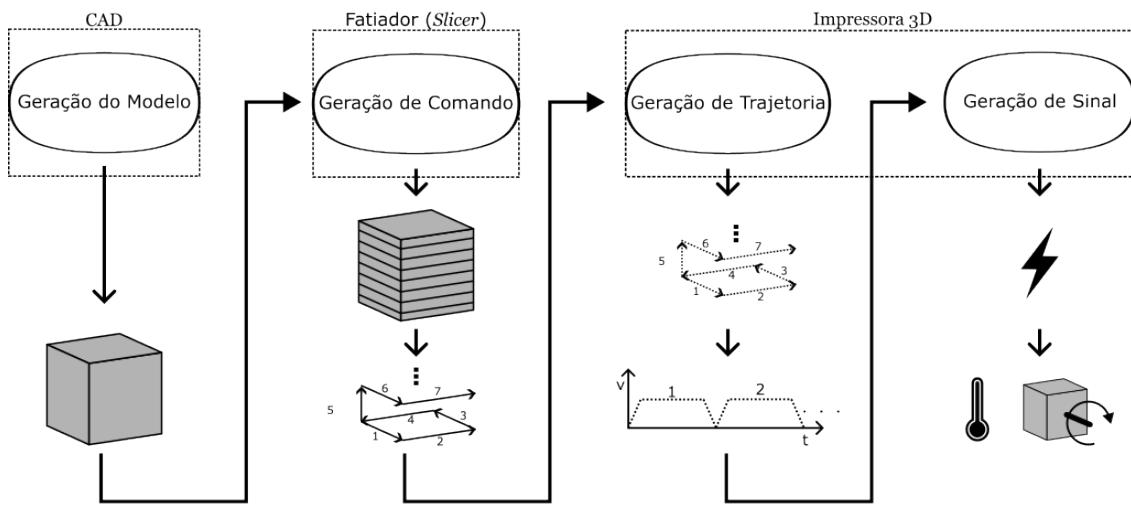
4.1.4	Caso 3 - Variação na aceleração	41
4.1.5	Caso 4 - Variação da velocidade	42
4.1.6	Caso 5 - Variação do passo de tempo	44
4.2	Considerações futuras	48
4.2.1	Combinação com outros algoritmos	48
4.2.2	Tuning	48
5	CONCLUSÃO	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

A manufatura aditiva emerge como uma tecnologia altamente promissora para a produção de peças e componentes em diversas áreas, incluindo engenharia, medicina e a indústria aeroespacial. Suas características distintivas viabilizam a fabricação de peças complexas em pequenas quantidades, promovendo uma notável iterabilidade, bem como suportando a produção descentralizada sob demanda. Nesse contexto, a impressão 3D, em particular o método de *Fused Deposition Modeling* (FDM), ganha destaque crescente, encontrando aplicações variadas nos setores aeroespacial, automobilístico e prototipagem rápida, ao mesmo tempo que se torna mais acessível e disseminada.

A modelagem digital desempenha um papel essencial no processo de impressão 3D, trabalhando em conjunto com ferramentas como o *Computer Aided Design* (CAD). Essa parceria possibilita a criação de modelos tridimensionais altamente precisos, que podem ser compartilhados e reproduzidos de forma descentralizada. Quando se trata de imprimir um desses modelos, a preparação é realizada através de um software de fatiamento, conhecido como *slicer*. O *slicer* divide o modelo em camadas e gera os comandos necessários para a impressora 3D, em geral esses comandos são representados pelo Gcode nas impressoras FDM. A impressora, por sua vez, interpreta esses comandos para determinar como proceder e quando executar cada ação, por fim atuando os motores e outros dispositivos, como resistências elétricas para o aquecimento. Na Figura 1.1 esse processo é exemplificado para uma impressora FDM através de um fluxograma. É importante notar que entre a interpretação e a execução desses comandos existem diversos processos intermediários que exercem influência direta sobre a qualidade e a velocidade da impressão, sendo um deles a geração de trajetória, onde é construído o comportamento ao longo do tempo a ser executado pelos atuadores e dispositivos, por exemplo movimentos e mudanças de temperatura.

Figura 1.1 – Fluxograma etapas Impressora 3D FDM



No entanto, uma das limitações significativas da impressão 3D, especialmente do tipo FDM, é o tempo de impressão, que ainda restringe o tamanho das peças produzidas em um período razoável. Frequentemente, é necessário utilizar camadas e linhas mais grossas para compensar esse aspecto, diminuindo a habilidade de se reproduzir detalhes menores. Diante disso, existe uma procura por maneiras de se imprimir mais rapidamente, sem comprometer a qualidade.

Assim, é relevante explorar técnicas que permitam alcançar capacidades superiores de qualidade e velocidade de impressão, flexibilizando a tecnologia e ampliando sua aplicação comercial viável.

Este trabalho tem como objetivo investigar e desenvolver uma metodologia para atuação de controle na geração de comandos em impressoras 3D utilizando o método FDM de forma a possibilitar maiores velocidades e garantindo a precisão dimensional das peças produzidas.



2 REFERENCIAL TEÓRICO

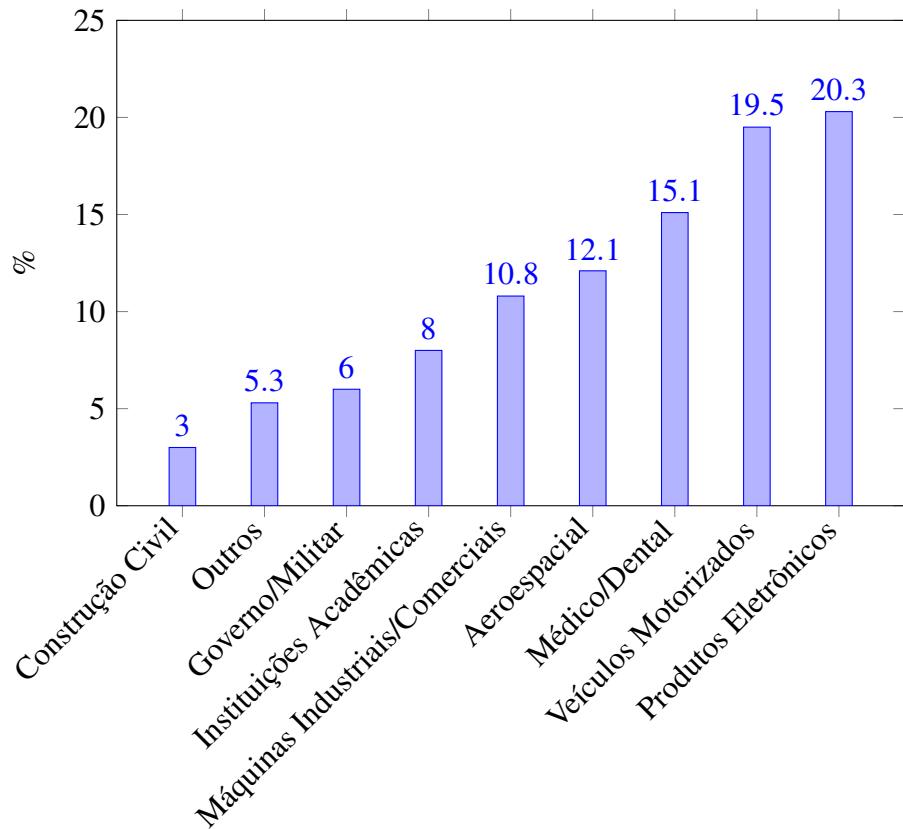
2.1 Manufatura Aditiva

O princípio fundamental da manufatura aditiva (MA) consiste em fabricar um modelo tridimensional de forma integrada, dispensando a necessidade de planejar as operações de maneira individual. O processo é calculado pelo fatiador com base em um modelo tridimensional digital, geralmente criado a partir de *Computer Aided Design* (CAD) e nas configurações dos parâmetros do fatiador e resulta nas instruções necessárias para a máquina de manufatura aditiva construir o modelo físico, as instruções e a maneira com que são transmitidas para a impressora varia de acordo com a tecnologia e a implementação específica dos modelos de impressora. Uma maneira comum dessas instruções serem transmitidas é através do Gcode, podendo ser complementado com outras informações como por exemplo arquivos de imagens.

Uma das características principais da MA é a rapidez na qual é possível criar protótipos 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 prototipagem rápida 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).

Atualmente, existe uma grande variedade de tecnologias e processos de manufatura aditiva. Os métodos de impressão 3D variam na maneira como depositam o material, como extrusão, sinterização a laser e estereolitografia. Eles também diferem nos princípios físicos que utilizam, como fusão, cura por luz e aglutinação. Além disso, os materiais que podem ser utilizados incluem plásticos, resinas, metais e cerâmicas. 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; STAVROPOULOS; CHRYSSOLOUDIS, 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 industrias

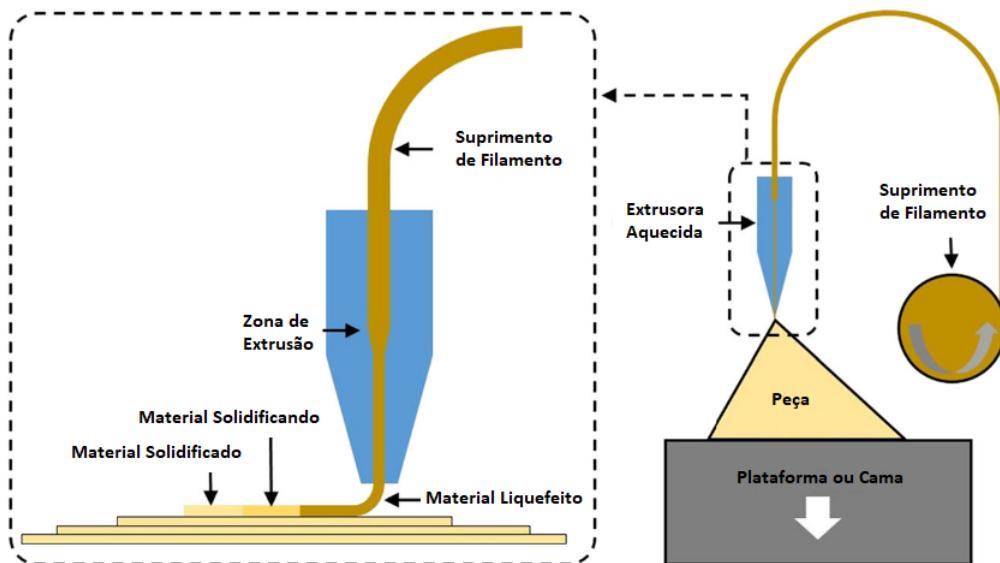


Fonte: Adaptado de BIKAS; STAVROPOULOS; CHRYSSOLOUDURIS, 2016

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 dentro do extrusor, ainda no estado sólido, 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; STRONG; GOLD, 2014). A Figura 2.2 ilustra este processo.

Figura 2.2 – Princípio e processo de impressão para FDM



Fonte: Adaptado de BIKAS; STAVROPOULOS; CHRYSSOLOOURIS, 2016

Este procedimento meticuloso exige a colaboração de vários componentes essenciais da impressora 3D. Estes componentes incluem:

Extrusora: A extrusora é responsável por derreter o filamento de material termoplástico e extrudá-lo em forma de filamento derretido. Ela consiste em um bico aquecido (*hotend*) que funde o material e um motor que empurra o filamento através do bico. Alguns modelos mais avançados podem ter extrusoras duplas para imprimir com materiais diferentes ou suportes solúveis.

Mesa de impressão: A mesa de impressão é a superfície onde o objeto está sendo construído. Ela é aquecida em muitas impressoras FDM para ajudar a aderência do material à superfície. Além disso, algumas mesas de impressão têm características especiais, como superfícies texturizadas ou magnéticas, para facilitar a aderência e a remoção do objeto após a conclusão.

Plataforma de construção: A plataforma de construção é o suporte físico onde a mesa de impressão é montada. Ela pode ser ajustada em altura para nivelar a superfície de impressão e garantir que a primeira camada do objeto seja depositada com precisão.

Motor de movimento: Impressoras 3D FDM possuem motores de movimento que controlam a posição da extrusora e da mesa de impressão ao longo dos eixos X, Y e Z. Geralmente são motores de passo e seus movimentos de rotação são geralmente convertidos em movimentos lineares através de correias ou parafusos de rosca trapezoidal.

Filamento: O filamento é o material de alimentação para a impressora 3D. Ele é um longo fio de plástico termoplástico que é inserido na extrusora e derretido durante o processo de impressão. Os filamentos vêm em várias cores e tipos de material, dependendo do objeto a ser impresso.

Software de fatiamento: Para imprimir um objeto em uma impressora 3D FDM, você precisa de um software de fatiamento. Este software converte modelos 3D em camadas finas onde é definido um percurso preenchendo essas camadas junto a outros comandos e configurações, como a temperatura do bico injetor, as velocidades máximas desejadas de cada movimento, a unidade utilizada, entre outras configurações e comandos. Estes comandos descritos através de Gcode e enviado para a impressora.

Sistema de controle: A eletrônica de controle inclui a placa-mãe da impressora, que recebe comandos do software, em geral no formato Gcode, e os traduz em movimentos dos motores, controle de temperatura da extrusora e da mesa de impressão, velocidade dos ventiladores entre outros acessórios. Ela também pode ter uma tela de exibição e controles para operação manual.

A Figura 2.3 indica alguns destes componentes e a Figura 2.4 mostra a interface de um fatiador.

Figura 2.3 – Indicação dos componentes de uma impressora 3D

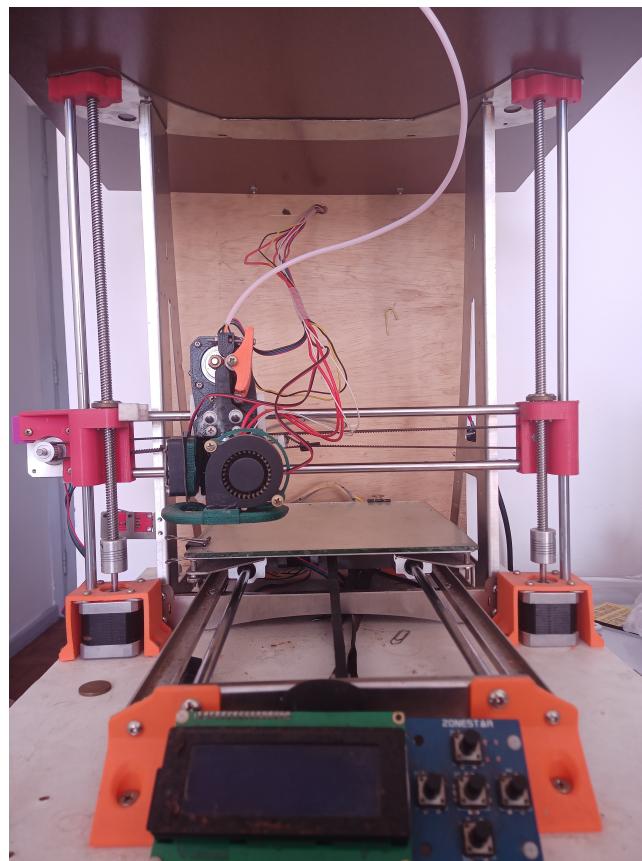
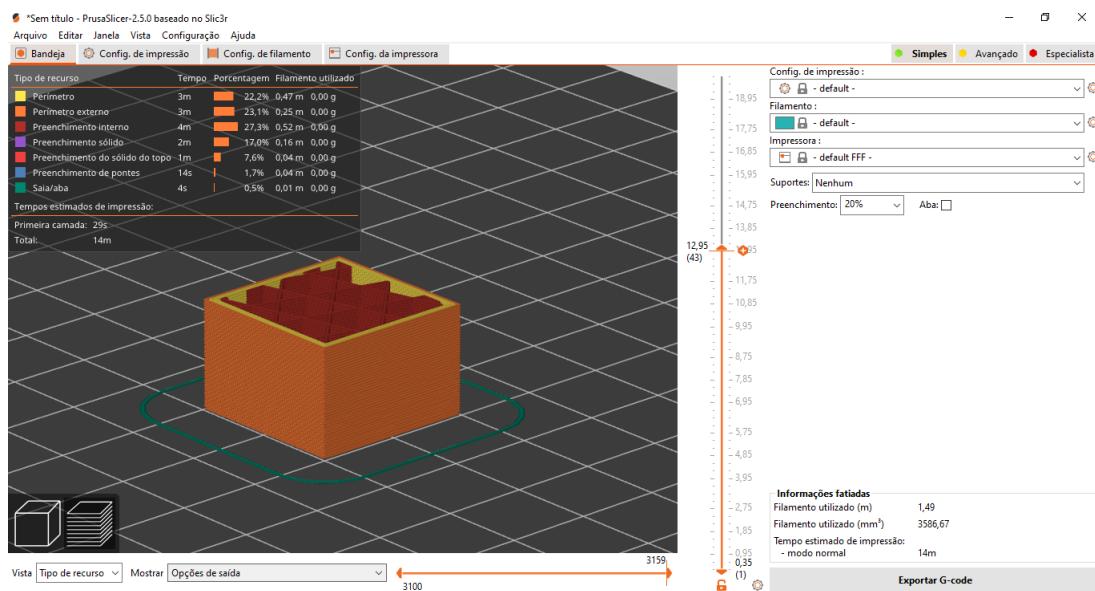


Figura 2.4 – Interface do fatiador PrusaSlicer



Podemos separar, de maneira simplificada, o *software* de impressoras 3D FDM em algumas etapas: fatiamento (*slicing*), geração de comando e controle. A etapa de geração de comando, realizado pelo fatiador, onde o volume do modelo é dividido em camadas, definindo um

percurso através de uma sequência de comandos. Esses comandos controlam os movimentos a serem realizados, definição de configurações, o controle da temperatura e outros dispositivos. Já na etapa de geração de trajetória, os comandos criados pelo fatiador (*slicer*) na etapa anterior são interpretadas, entretanto esses comandos não descrevem em detalhes. Portanto, é necessário a definição do comportamento que a impressora deve tomar para cumprir o comando interpretado, no caso de um comando de movimento por exemplo precisamos construir a trajetória dos eixos para poder realizar o comando com sucesso, assim chamamos esta etapa de geração de trajetória. Podemos, antes de prosseguir para a próxima etapa, utilizar de técnicas de controle para melhorar a performance da execução dos comandos, por exemplo utilizar os sensores de temperatura para modificar a taxa de aquecimento do bico extrusor, ou utilizar essas técnicas de controle para alterar a trajetória construída na etapa anterior de maneira a contribuir para a qualidade da impressão. Por fim, temos a etapa de geração de sinais, onde a impressora precisa converter as abstrações definidas sobre seus dispositivos em sinais que são responsáveis em controlá-los. Por exemplo, converter as características do movimento de um eixo em uma série de pulsos elétricos que controlam os motores de passo.

2.2.1 Geração de Comando

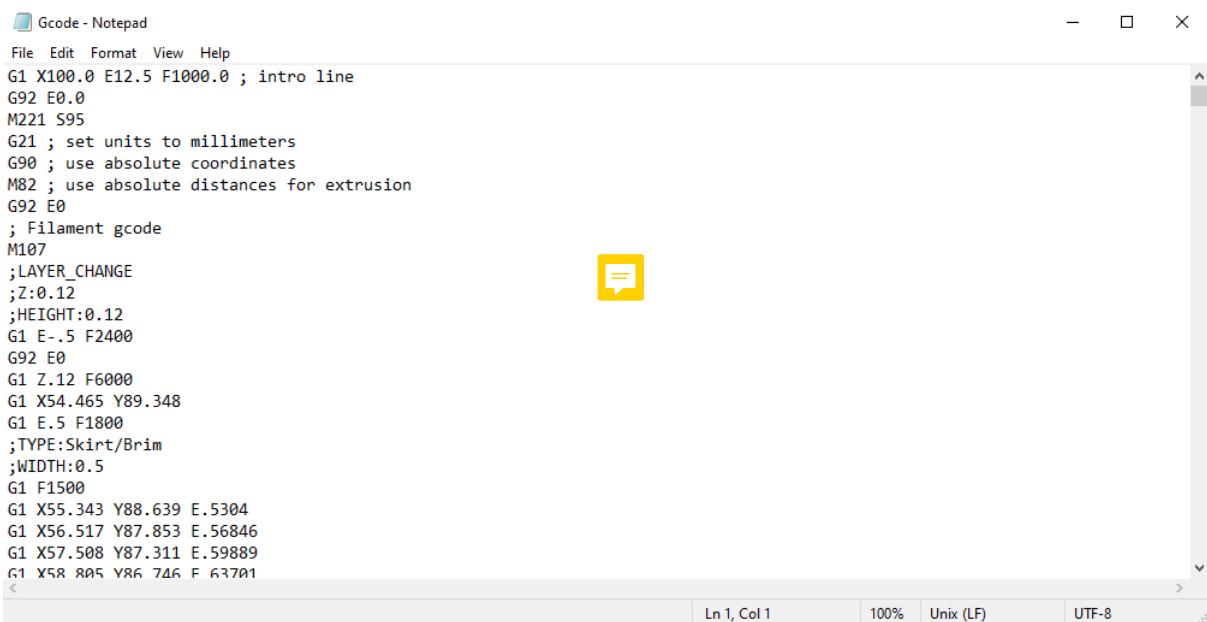
O Gcode (Código G) é uma linguagem de programação usada em impressoras 3D e máquinas CNC para controlar o movimento e as ações da máquina durante o processo de fabricação. Ele é muito utilizado pelos fatiadores para representar os comandos para a impressora 3D. O Gcode é composto por uma série de comandos textuais, cada um com um formato específico. Aqui estão alguns elementos-chave da estrutura típica de um comando G-code:

- Prefixo (Código G): Todo comando G-code começa com a letra 'G', que indica que é um comando de movimento ou função.
- Número do Comando: Após o 'G', segue um número que identifica o tipo específico de comando. Por exemplo, 'G0' é frequentemente usado para mover rapidamente a cabeça de impressão para uma posição, enquanto 'G1' é usado para movimentos de impressão lineares.
- Parâmetros: Após o número do comando, podem seguir-se parâmetros adicionais. Esses parâmetros variam dependendo do comando, mas podem incluir coordenadas de posi-

cionamento (X, Y, Z), velocidades de movimento, taxas de alimentação, temperaturas e outros valores relevantes.

- Comentários: O G-code também pode incluir comentários precedidos por um ponto e vírgula (;) ou entre parênteses (). Os comentários não afetam a execução do programa, mas ajudam a documentar o código para facilitar a compreensão.
- Fim de Linha: Cada comando G-code é normalmente concluído com um caractere de fim de linha, como o retorno de carro ('\n') ou a combinação de retorno de carro e nova linha ('\r\n'), dependendo do sistema.

Figura 2.5 – Exemplo de um arquivo Gcode



```

Gcode - Notepad
File Edit Format View Help
G1 X100.0 E12.5 F1000.0 ; intro line
G92 E0.0
M221 S95
G21 ; set units to millimeters
G90 ; use absolute coordinates
M82 ; use absolute distances for extrusion
G92 E0
; Filament gcode
M107
;LAYER_CHANGE
;Z:0.12
;HEIGHT:0.12
G1 E-.5 F2400
G92 E0
G1 Z.12 F6000
G1 X54.465 Y89.348
G1 E.5 F1800
;TYPE:Skirt/Brim
;WIDTH:0.5
G1 F1500
G1 X55.343 Y88.639 E.5304
G1 X56.517 Y87.853 E.56846
G1 X57.508 Y87.311 E.59889
G1 X58.805 Y86.746 E.63701
<
Ln 1, Col 1      100%    Unix (LF)    UTF-8

```

2.2.2 Geração de trajetória

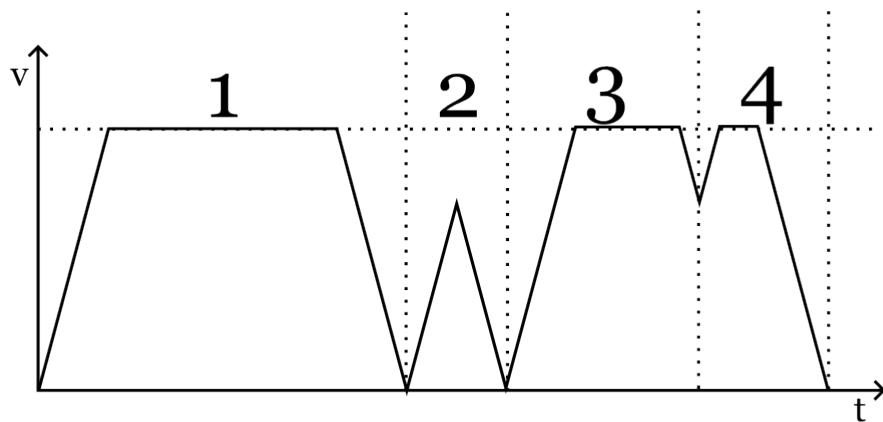
A geração de trajetória é o processo que define o comportamento dos eixos da impressora ao longo do tempo com base nos comandos interpretados, incluindo além dos eixos de movimento, eixos de temperatura, potência dos ventiladores e de taxa de extrusão por exemplo. Esses diferentes componentes precisam coordenados e seu comportamento necessita de ser definido para que seja possível converter essas abstrações no mundo físico posteriormente. No caso dos eixos de movimento, uma estratégia para se construir esse comportamento são as curvas de velocidade trapezoidal (YU *et al.*, 2020).

2.2.3 Curvas de velocidade trapezoidal

A construção das curvas de velocidade trapezoidal se dá no objetivo de se deslocar de uma posição e velocidade inicial para uma posição e velocidade final, considerando uma velocidade máxima. Assim, se utiliza uma configuração de aceleração armazenada na impressora a fim de definir o comportamento das variações de velocidade. Vale lembrar que os comandos de movimento podem possuir atuação de múltiplos eixos e dado o objetivo das condições iniciais e finais de cada eixo se darem ao mesmo tempo, é utilizado o módulo do vetor de velocidade, composto pelas velocidades de cada eixo.

A partir dessas características, ao construir o perfil de velocidade em uma situação onde se tenha tempo suficiente para atingir a velocidade máxima, com base na aceleração definida pela impressora, o perfil de velocidade tem uma aparência trapezoidal, justificando o nome a estratégia. Entretanto, caso não seja possível alcançar a velocidade máxima dentro do tempo e deslocamento possível para o movimento, o perfil de velocidade passa a possuir uma aparência triangular, causada pelo fato de não existir mais uma seção onde é mantida a velocidade máxima (YU *et al.*, 2020; KLIPPER, 2017). A Figura 2.6 mostra um perfil de velocidade de uma sequência de quatro movimentos, apresentando seções trapezoidais e triangulares.

Figura 2.6 – Perfil de velocidade - Curva trapezoidal de velocidade



2.2.4 Técnicas de Controle

2.2.5 *Feedforward*

Nos métodos de controle para impressoras 3D *Fused Deposition Modeling* (FDM), o controle *feedforward* se consolida como a estratégia mais eficaz para os eixos de movimento, especialmente quando se pondera as restrições orçamentárias típicas de impressoras 3D padrão.

O controle *feedforward* é uma abordagem utilizada em sistemas automáticos, destinada a antecipar e corrigir possíveis perturbações que possam interferir em um sistema. Ele atua proativamente, antes que tais perturbações comprometam a saída prevista. Essa técnica é especialmente útil em sistemas onde é possível prever com exatidão as perturbações.

No universo das impressoras 3D, que são tipicamente menos susceptíveis a interferências externas, é factível prever o comportamento dinâmico da máquina com precisão, baseando-se em um modelo apropriado, e sem a necessidade de sensores de alto custo. Assim, o controle *feedforward* emerge como uma solução prática para otimizar o desempenho de impressoras já em operação, exigindo alterações físicas mínimas e simples ajustes no *software*.

Contudo, a implementação do controle *feedforward* em impressoras 3D apresenta desafios. Entre eles, a complexidade em estabelecer um modelo fidedigno, a demanda por capacidade computacional significativa e a obrigação de simular o processo de impressão do começo ao fim, considerando a influência do estado inicial (RAMANI; EDOIMIOYA; OKWUDIRE, 2020; DUAN; YOON; OKWUDIRE, 2018)

2.2.5.1 *Input Shaper*

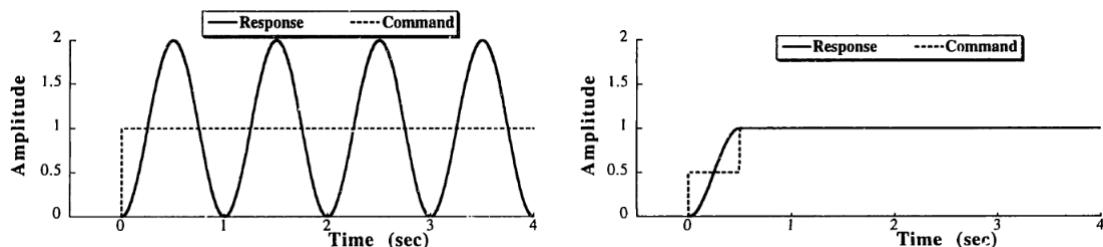
Ao entender a trajetória pretendida e as peculiaridades do sistema, podemos deduzir uma sequência de comandos que ajustam o comando de referência com base nessas características. O objetivo é alinhar a trajetória final o máximo possível ao comando de referência. No entanto, em vez de processar todo o comando inicial, podemos adaptá-lo em tempo real usando um filtro especializado. O *Input Shaper* é uma abordagem exemplar desse tipo de filtro, onde diversos *Shapers* são formulados considerando variados propósitos e limitações.

O *Input Shaping* é uma metodologia de controle criada para atenuar ou anular vibrações indesejadas em sistemas mecânicos ou estruturas manipuladas por atuadores, como em robôs, veículos e equipamentos industriais. Essa estratégia aprimora a sequência de pulsos de entrada, concebendo-os de modo a suprimir a ativação das frequências naturais de ressonância do

sistema. Assim, o Input Shaping desempenha um papel crucial na diminuição da resposta vibratória, conferindo maior precisão e estabilidade ao sistema durante a realização de movimentos ou operações.

O valor do *Input Shaping* é evidente em sistemas de controle de alta exatidão, como na robótica, onde a supressão de vibrações é fundamental para a excelência operacional e o posicionamento acurado. Utilizando o entendimento das especificidades do sistema e da trajetória almejada, essa técnica ajusta os comandos de entrada para evitar a ativação das frequências vibratórias intrínsecas, originando sistemas de controle mais robustos e eficientes. (SINGHOSE, 1997). A Figura 2.7 ilustra uma comparação entre as respostas ao estímulo de degrau e a função escalonada aplicada pelo *shaper*.

Figura 2.7 – Comparaçāo da resposta ao degrau e da resposta a escada



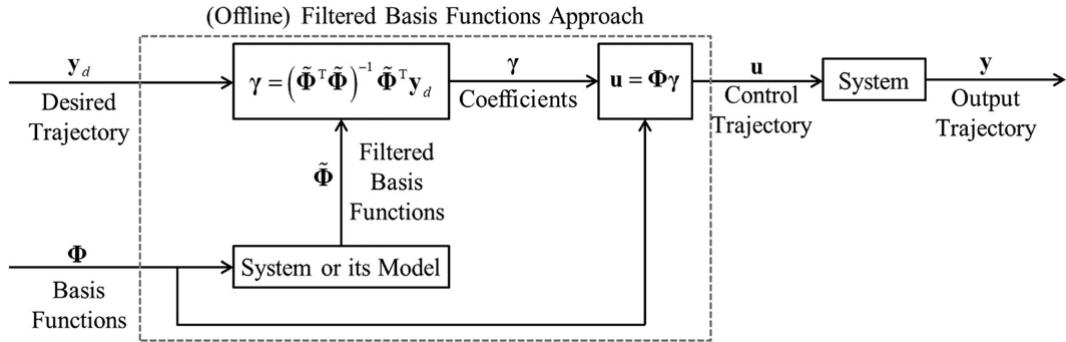
Fonte: SINGHOSE, 1997

2.2.5.2 *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 2.8). (RAMANI *et al.*, 2017)



Figura 2.8 – Fluxograma FBF

Fonte: RAMANI *et al.*, 2017

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* adapta a solução do FBF de forma a dividir 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.

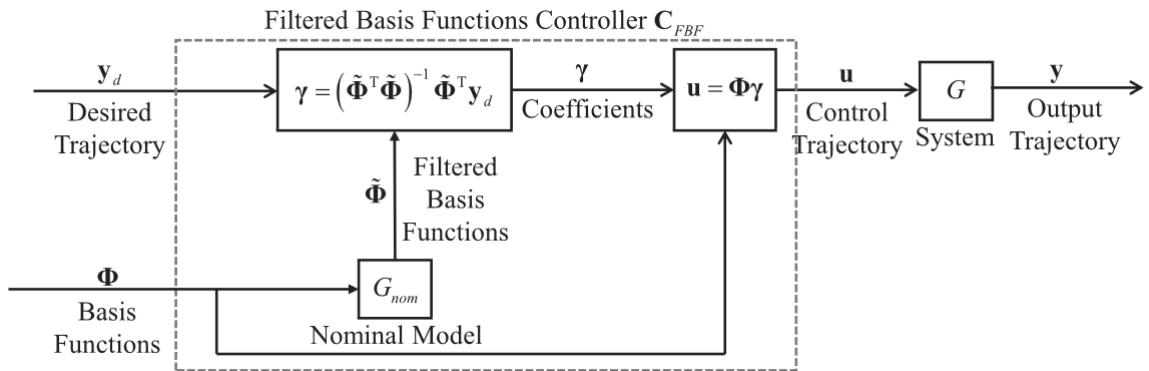
Com base nos desenvolvimentos nos trabalhos de (RAMANI *et al.*, 2017) e (DUAN; YOON; OKWUDIRE, 2018), comentados anteriormente, (RAMANI; EDOIMIOYA; OKWUDIRE, 2020) busca atacar um segundo desafio prático na implementação da FBF, sendo o primeiro desafio prático o custo computacional que foi endereçado pelo trabalho de (DUAN; YOON; OKWUDIRE, 2018) através da LBFBF, permitindo a aplicação do algoritmo no mundo real.

Este segundo desafio se trata da degradação de precisão de rastreamento da abordagem FBF, causada por imprecisões no modelo ou incertezas na dinâmica da planta atrelado a característica do método de se utilizar puramente uma abordagem de *feedforward*. A não participação de *feedbacks* sensoriais do mundo real abre espaço para uma crescente divergência entre o modelo e a realidade.

Considerando o requisito de manter a performance computacional alcançada com o método LPFBF, (RAMANI; EDOIMIOYA; OKWUDIRE, 2020) propõe, também, a utilização de

um filtro robusto em substituição à dinâmica nominal da planta para filtrar as funções de base. Esse filtro robusto é construído com base no inverso de um controlador *feedforward* ótimo, que minimiza uma função de custo de erro para lidar com a incerteza conhecida da planta como um filtro robusto. O esquema do método é apresentado na figura 2.9.

Figura 2.9 – Fluxograma RFBF



Fonte: RAMANI; EDOIMIOYA; OKWUDIRE, 2020

2.3 Espaço de Estados

Os sistemas dinâmicos podem ser descritos 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. 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).

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

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

2.4 Integração implícita utilizando programação não linear

HARGRAVES; PARIS (1987) descreve um algoritmo para a solução numérica direta de problemas de controle ótimo. Este algoritmo emprega uma abordagem que utiliza polinômios cúbicos para representar as variáveis de estado. Adicionalmente, recorre à interpolação linear

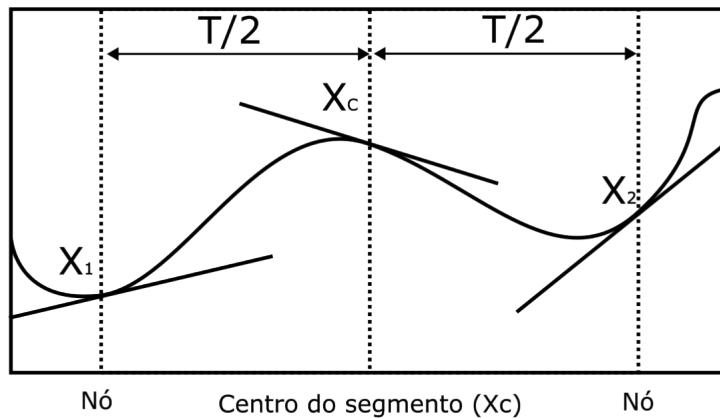
para tratar as variáveis de controle. Esse enfoque converte efetivamente o problema de controle ótimo em um problema de programação matemática.

Uma das principais vantagens desse método é sua facilidade de implementação e sua capacidade de lidar com uma ampla gama de problemas de otimização de trajetória. Isso inclui a consideração de restrições de caminho, estados descontínuos e desigualdades de controle.

O método alcança sua aproximação das soluções das equações diferenciais através da subdivisão de cada estado na matriz de espaço de estados em segmentos. Cada um desses segmentos é representado por polinômios de terceiro grau.

Os valores de estado são então selecionados de maneira a garantir que a curva resultante da concatenação desses polinômios seja continua, ou seja, o valor da função e de sua derivada precisa ser igual para ambos polinômios nas conexões como observado na figura 2.10

Figura 2.10 – Ilustração integração implícita



Fonte: HARGRAVES; PARIS, 1987

O procedimento base pode ser aplicado pelos seguintes passos.

A equação 2.3 avalia o estado no centro do segmento, onde x representa o estado, T representa o comprimento do segmento e f_i representa o valor da função avaliado em x_i . O subscrito c representa o centro do segmento.

$$x_c = \frac{x_1 + x_2}{2} + T \frac{f_1 - f_2}{8} \quad (2.3)$$

Da mesma maneira sua derivada é apresentada na equalção 2.4.

$$\dot{x}_c = -3 \frac{x_1 + x_2}{2T} + \frac{f_1 + f_2}{4} \quad (2.4)$$

A equação 2.5 define então o valor do defeito no centro do segmento.

$$\Delta = f_c - \dot{x}_c \quad (2.5)$$

Considerando também que a entrada do sistema pode ser avaliada de forma aproximada no centro do segmento através da equação 2.6.

$$u_c = \frac{u_1 + u_2}{2} \quad (2.6)$$

Os valores de estado agora podem ser alterados de maneira que o defeito tenda a zero.

3 METODOLOGIA

3.1 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

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 (x_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 (x) como na figura ??.

$$\begin{aligned} m\ddot{x}_p + c(\dot{x}_p - \dot{x}_b) + k(x_p - x_b) &= 0 \\ \ddot{x}_p &= -\frac{c}{m}\dot{x}_p - \frac{k}{m}x_p + \frac{c}{m}\dot{x}_b + \frac{k}{m}x_b \end{aligned} \tag{3.1}$$

(Figura - placeholder)

3.1.1 Espaço de estados

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

3.2, onde $\dot{x} = x * A + u * B$. Baseado na equação 3.1 expandimos as matrizes e vetores na equação 3.3. Como o modelo dos eixos x e y estão sendo considerados como iguais, também podemos utilizar a equação 3.3 para a base do eixo y.

$$\dot{x} = A * x + B * u \quad (3.2)$$

$$\begin{bmatrix} \dot{x}_p \\ \ddot{x}_p \\ \dot{y}_p \\ \ddot{y}_p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_x}{m_x} & -\frac{c_x}{m_x} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{k_x}{m_x} & -\frac{c_x}{m_x} \end{bmatrix} \begin{bmatrix} x_p \\ \dot{x}_p \\ y_p \\ \dot{y}_p \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_x}{m_x} & \frac{c_x}{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{k_x}{m_x} & \frac{c_x}{m_x} \end{bmatrix} \begin{bmatrix} x_b \\ \dot{x}_b \\ y_b \\ \dot{y}_b \end{bmatrix} \quad (3.3)$$

3.2 Geração de Comando

A geração de comando se inicia na leitura do Gcode para a obtenção dos movimentos e velocidades desejadas. Foi considerado no mapeamento do Gcode apenas comandos G1, extraíndo 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. Em geral a unidade de F em arquivos Gcode gerados pelos fatiadores se dão em milímetros por minuto, mas é convertida para milímetros por segundo na construção da matriz de entrada.

3.2.1 Curva trapezoidal de velocidade

A partir da matriz de entrada e das velocidades finais e iniciais dos movimentos estabelecido pelos comandos do Gcode, utilizamos a função responsável por gerar a curva trapezoidal de velocidades.

Essa função constrói a curva de velocidade trapezoidal a partir dos dados de entrada, sendo estes o deslocamento do movimento, as velocidades iniciais, velocidades finais e a velocidade desejada. Para identificar se a velocidade desejada será alcançada é calculada a velocidade pico (v_p) através do encontro das retas de velocidades. É possível obter a velocidade de pico através da Equação 3.4.

Figura 3.1 – Curva de velocidade trapezoidal

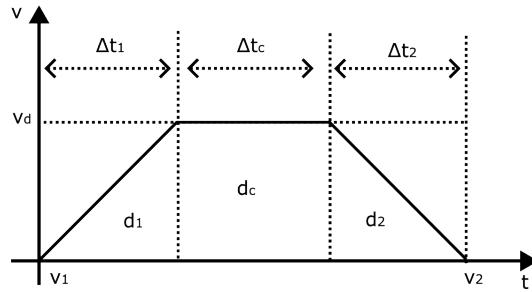
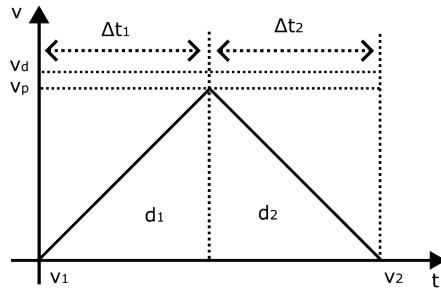


Figura 3.2 – Curva de velocidade triangular



$$v_p = \sqrt{\frac{(v_1^2 + v_2^2)}{2} + ad} \quad (3.4)$$

A partir da comparação da velocidade pico com a velocidade desejada, indicada pelo *feedrate* disponibilizado no Gcode, é possível determinar se a velocidade desejada será alcançada e assim determinar o perfil das curvas e como serão calculadas. Caso a velocidade de pico for maior do que a velocidade desejada, temos 3 fases de deslocamento que podem ser calculadas pelas Equações 3.8 e 3.9. 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 3.8.

$$d_1 = \frac{(v_d^2 - v_1^2)}{(2a)} \quad (3.5)$$

$$d_2 = \frac{(v_2^2 - v_d^2)}{(2a)} \quad (3.6)$$

$$d_1 = \frac{(v_p^2 - v_1^2)}{(2a)} \quad (3.7)$$

$$d_2 = \frac{(v_2^2 - v_p^2)}{(2a)} \quad (3.8)$$

$$d_c = d - (d_1 + d_2) \quad (3.9)$$

É possível calcular também os intervalos de tempo dessas fases, através das Equações 3.14 e 3.12.

$$t_1 = \frac{(v_d - v_1)}{a} \quad (3.10)$$

$$t_2 = \frac{(v_2 - v_d)}{a} \quad (3.11)$$

$$t_c = \frac{d_c}{v_d} \quad (3.12)$$

$$t_1 = \frac{(v_p - v_1)}{a} \quad (3.13)$$

$$t_2 = \frac{(v_2 - v_p)}{a} \quad (3.14)$$

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.

3.2.2 Interpolação

A partir dos pontos de posição, velocidade, aceleração e tempo de cada eixo, é utilizada uma função de interpolação esses intervalos em pontos intermediários baseados em um passo de tempo definido para esta interpolação. Para se dividir esses intervalos é possível utilizar a Equação 3.15, que calcula o número de passos neste intervalo, anexando à matriz os passos de tempo e por fim o restante do intervalo, calculado pela Equação 3.16. Com base nesses passos de tempo, é possível calcular o deslocamento para cada um destes passos através da Equação 3.17.

$$N = \lceil \frac{\Delta t}{\Delta p} - 1 \rceil \quad (3.15)$$

$$\Delta t_f = \Delta t - \Delta p N \quad (3.16)$$

$$\Delta d_i = \Delta v_i \Delta t_i + \frac{a_s \Delta t_i^2}{2} \quad (3.17)$$

3.3 FMINCON

3.3.1 Configurações da função

A Tabela 3.1 apresenta o conjunto de configurações customizadas da função FMINCON. As configurações não explicitadas foram mantidas como os padrões disponibilizados pelo Matlab. A função objetivo foi considerada nula por hipótese.

Tabela 3.1 – Parâmetros modificados na função FMINCON.

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

3.3.2 Restrições lineares e limites de borda

Não foram utilizadas restrições lineares na otimização. Os limites superiores (*upper-bound*) e inferiores (*lowerbound*) foram definidos com base nos limites físicos da impressora para as posições x_b e y_b (ver Seção ??).

3.3.3 Restrições não lineares

A função para as restrições não lineares foi implementada com base nas equações 2.3, 2.4, 2.5, 2.6 de maneira a popular a variável de restrições de igualdades com os valores do defeito dos segmentos (Δ) e também com os valores iniciais.

A arrancada também foi utilizada como restrição de igualdade, atrelada ao condicional ?? para que se comporte como uma inequalidade. A variável de restrições de inequalidades não foi populada (ver Seção ?? Resultados e Discussão).

3.4 Solução da trajetória da base

É considerada como trajetória desejada a trajetória obtida através da função de geração de comando apresentada anteriormente e são utilizados os vetores de tempo e posição como variáveis globais, para serem acessados dentro da função das restrições não lineares, chamada pela FMINCON. Além disso, esses mesmos vetores de posição também são considerados como o chute inicial e variável principal na FMINCON, sendo adaptados em forma de matriz.

A variável principal inserida na FMINCON é a matriz contendo os vetores de posição da base (x_b e y_b), enquanto os vetores de posição da ponta são fixados pela trajetória desejada presente na forma de variáveis globais.

As Equações 3.18 e 3.19 foram utilizadas para reconstrução do vetor de velocidade a partir dos vetores deslocamento e tempo, sendo dt a variação do tempo no intervalo, des_n o deslocamento final do intervalo, des_i o deslocamento inicial no intervalo, a_n a aceleração do intervalo, v_n a velocidade final do intervalo e v_i a velocidade inicial do intervalo.

$$a(k) = \frac{2}{\Delta t(k)} \left(\frac{d(k) - d(k-1)}{\Delta t(k)} - v(k-1) \right) \quad (3.18)$$

$$v(k) = v(k-1) + a(k)\Delta t(k) \quad (3.19)$$

3.5 Simulação de controle de impressora 3D

3.5.1 Caso base

Com base no modelo proposto, foi simulada a trajetória do ponteiro em 10 milímetros ao longo do eixo x e depois 10 milímetros ao longo do eixo y. Para as análises do caso base, foram avaliados os cenários com controle e sem controle. Também foram analisados os perfis de deslocamento e velocidade.

A Tabela 3.2 apresenta os parâmetros de frequência, coeficiente de amortecimento, aceleração base, velocidade desejada e resolução de interpolação ("parâmetros-chave") considerados para este cenário base.

Tabela 3.2 – Parâmetros referência dos Estudos de Caso.

Parâmetro	Valor	Unidade
Frequência	100	rad/s
Coeficiente de amortecimento	0,5	-
Aceleração base	5000	mm/s^2
Velocidade desejada	100	mm/s
Resolução de interpolação	0,005	s

O tamanho dos vetores, o tempo de simulação e o índice viabilidade da FMINCON também foram analisados. O tamanho dos vetores é definido na fase de geração de comando e tem influência direta da resolução de interpolação definida e do tempo necessário para percorrer o caminho dada as definições de velocidades na geração de comando. O tempo de simulação começa a ser contado logo depois da geração de comando e para quando a função FMINCON termina de ser executada. A viabilidade é um dos resultados da função FMINCON e este representa o valor da maior restrição não cumprida.

O computador utilizado para a realização das simulações foi um notebook Acer com as configurações apresentadas na Tabela 3.4.

Tabela 3.3 – Estudos de Caso

Caso	Parâmetro	Valor A	Valor B	Valor C	Unidade
1	Frequência	50	200	500	rad/s
2	Coeficiente de amortecimento	0	1	2	-
3	Aceleração base	1000	10000	-	mm/s^2
4	Velocidade desejada	50	200	-	mm/s
5	Resolução de interpolação	0,01	0,001	0,0002	s

Além das curvas de posição, deslocamento e velocidade, algumas variáveis também foram explicitadas para análise. São elas o tamnho dos vetores, o tempo de simulação e a viabilidade ao longo . O tamanho dos vetores é definido na fase de geração de comando e tem influência direta da resolução de interpolação definida e do tempo necessário para percorrer o caminho dada as definições de velocidades na geração de comando. O tempo de simulação começa a ser contado logo depois da geração de comando e para quando a função FMINCON termina de ser executada. A viabilidade é um dos resultados da função FMINCON e este representa o valor da maior restrição não cumprida.

A máquina utilizada para a realização das simulações foi um notebook acer com as configurações apresentadas na tabela 3.4.

Tabela 3.4 – Especificações do computador

Processador	Intel I7-5500U 2.40GHz
Memoria	8,00 GB
Placa de vídeo	Nvidia Geforce 920M
Sistema	64 bits

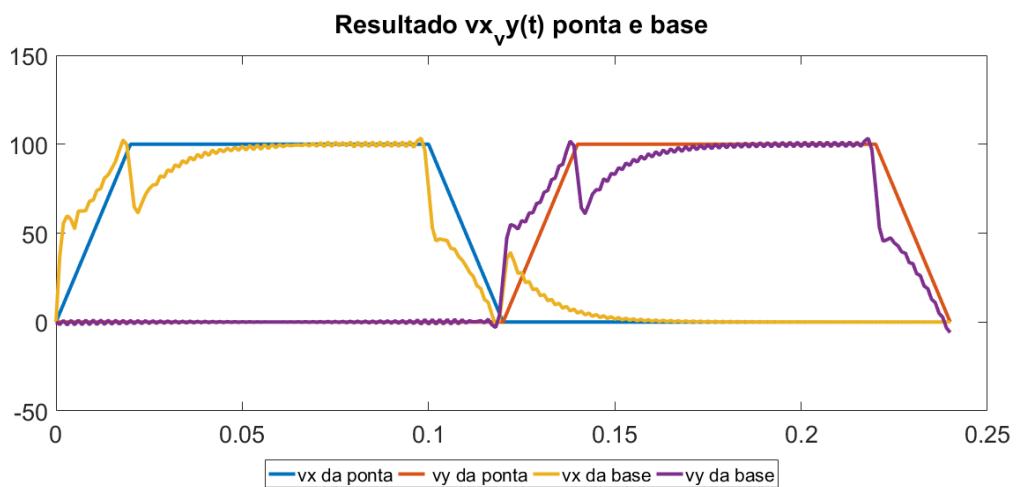
4 RESULTADOS E DISCUSSÃO

4.1 Resultados

4.1.1 Caso Base

A Figura 4.1 apresenta o perfil de velocidades para o caso base, no qual evidencia-se o comportamento compensador do sistema e uma oscilação de alta frequência, artefato da resposta da FMINCON. A amplitude das oscilações tende a ser menor conforme exista uma maior resolução na interpolação do tempo, ou seja, apresente dt menor. Ademais, conforme mencionado na seção anterior, o arranque foi restringido afim de controlar a amplitude dessas oscilações.

Figura 4.1 – Caso referência - Comportamento no tempo das velocidades em x e y da ponta e da referência



A Figura 4.2 ...

Figura 4.2 – Caso referência - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência

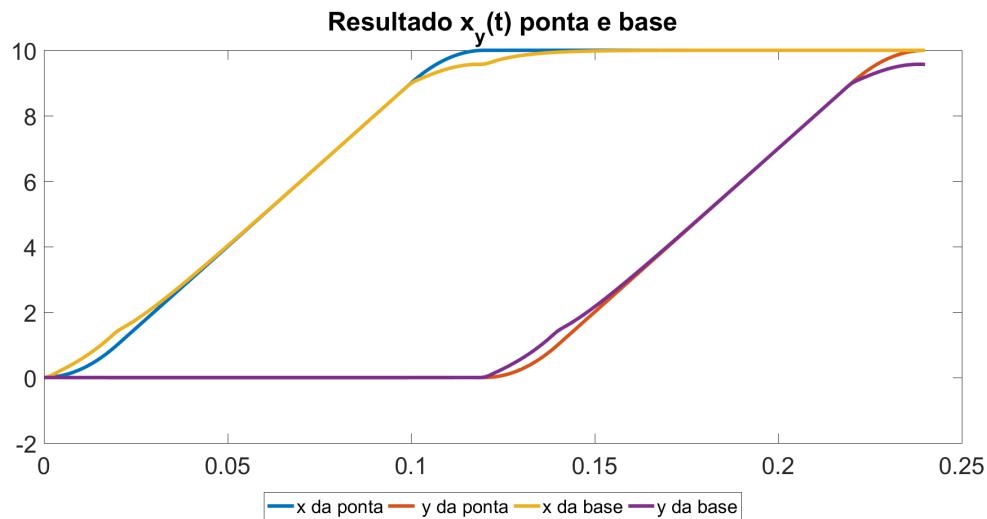


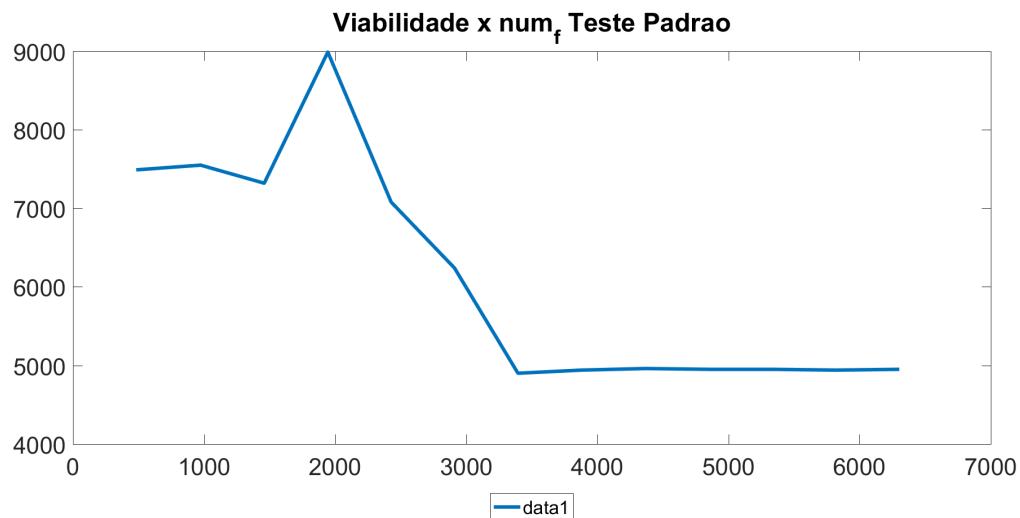
Figura 4.3 – Caso referência - Caminho percorrido x vs y da ponta e da referência



No gráfico da figura 4.4 nos indica a progressão das iterações da FMINCON, sendo cada ponto uma iteração diferente, possuindo um valor correspondente ao número de vezes em que a função objetivo e as restrições foi avaliada no eixo x e o valor da viabilidade, que como comentado na seção de metodologia, indica o maior valor de restrição não cumprido. Podemos perceber o processo de convergência no decorrer das iterações, o que costuma indicar uma rodada de sucessos. Percebe-se também que os níveis iniciais da viabilidade ficam próximos de

$7,5 \times 10^3$, que foram niveis próximos na maioria dos Casos. Por fim, temos que o tempo elapsado de simulação para este Caso foi de 89,8 segundos, com os vetores de posição possuindo um tamanho de 241 elementos.

Figura 4.4 – Caso referência - Num de fun x Viabilidade



4.1.2 Caso 1 - Variação da frequência

Os resultados da variação da variável de entrada frequência natural, que varia basicamente o comportamento da planta do modelo dinâmico, exemplifica as diferenças nas amplitudes dos desvios entre a ponta e a referência e uma menor necessidade de compensação, assim como previsto de um sistema de rigidez maior. As diferenças podem ser observadas de forma clara se comparadas as figuras 4.5, 4.6 e 4.7, comparando também as diferenças entre o comportamento do deslocamento e do caminho tomado entre os dois extremos do Caso (A e C) nas figuras 4.8, 4.9, 4.10 e 4.11.

Figura 4.5 – Caso 1A - Comportamento no tempo das velocidades em x e y da ponta e da referência

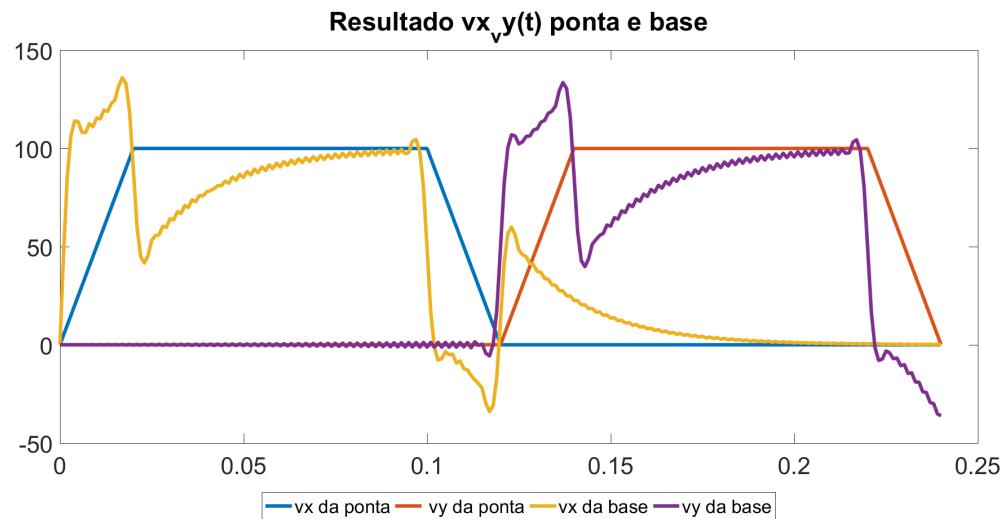


Figura 4.6 – Caso 1B - Comportamento no tempo das velocidades em x e y da ponta e da referência

Figura 4.7 – Caso 1C - Comportamento no tempo das velocidades em x e y da ponta e da referência

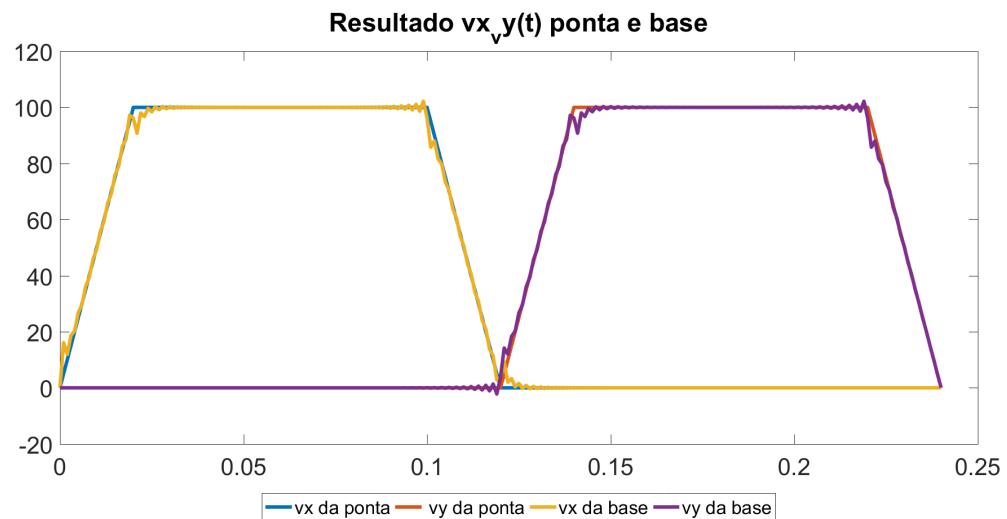


Figura 4.8 – Caso 1A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência

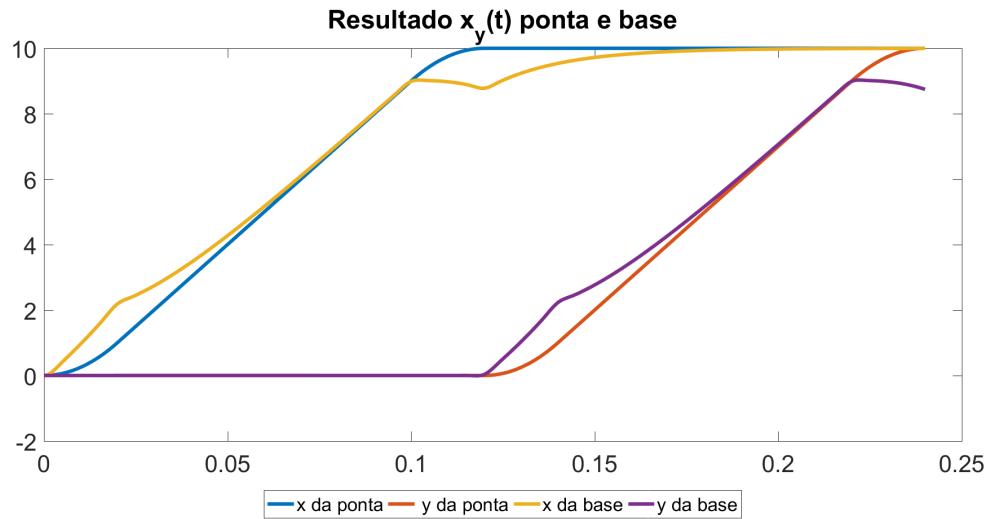


Figura 4.9 – Caso 1C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência

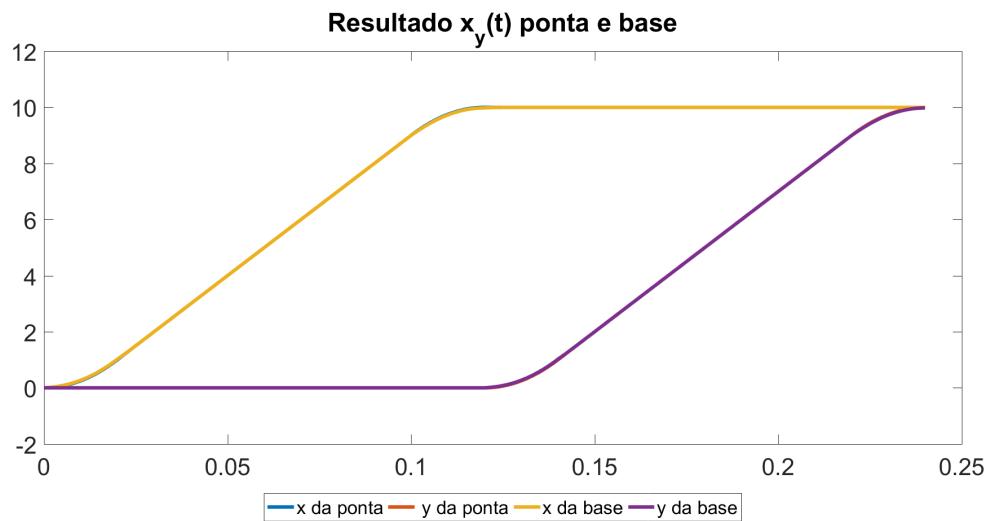


Figura 4.10 – Caso 1A - Caminho percorrido x vs y da ponta e da referência

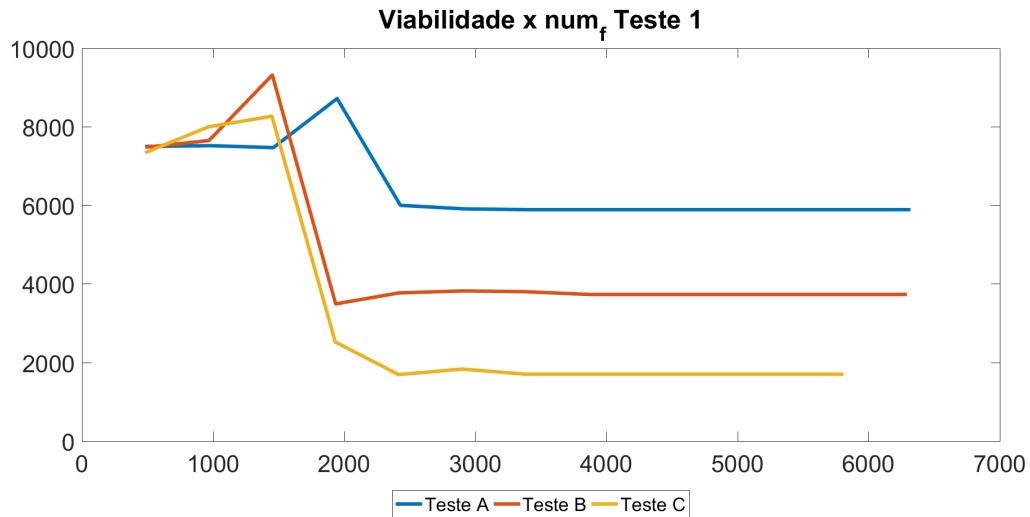


Figura 4.11 – Caso 1C - Caminho percorrido x vs y da ponta e da referência



Já na figura 4.12 é possível notar um comportamento semelhante de convergência, mas nota-se que nos Casos de maior rigidez a viabilidade alcançada foi menor, ou seja, menores violações das restrições impostas, dados primordialmente pelos menores desvios e menor necessidade de compensação. Essa característica pode ser observada também nos tempos de simulação que foram reduzindo de 96,76s (Caso A) até 80,77s (Caso C), com o Caso B ficando entre os dois com 89,04s, todos esses com vetores de posição de 241 elementos.

Figura 4.12 – Caso 1 - Num de fun x Viabilidade



4.1.3 Caso 2 - Variação do coeficiente de amortecimento

No Caso 2, que varia o coeficiente de amortecimento, podemos avaliar de forma mais clara as diferenças através das figuras 4.13 e 4.14 que mostram as curvas dos caminhos do Caso b e c respectivamente, podemos comparar também com a figura 4.3 que possui um coeficiente intermediário entre os Casos A e B. Sendo o Caso A não apresentado, por não ter convergido, apresentando praticamente os mesmos valores de entrada, esse comportamento é identificado na figura 4.15, onde podemos observar também um padrão convergência para valores de viabilidade menor conforme coeficiente de amortecimento também aumenta, ou seja, uma maior facilidade de compensar os desvios. Todos se mantiveram com 241 elementos nos vetores de posição e tempos de simulação de $69,89e52s$ para os Casos 2A, 2B e 2C respectivamente.

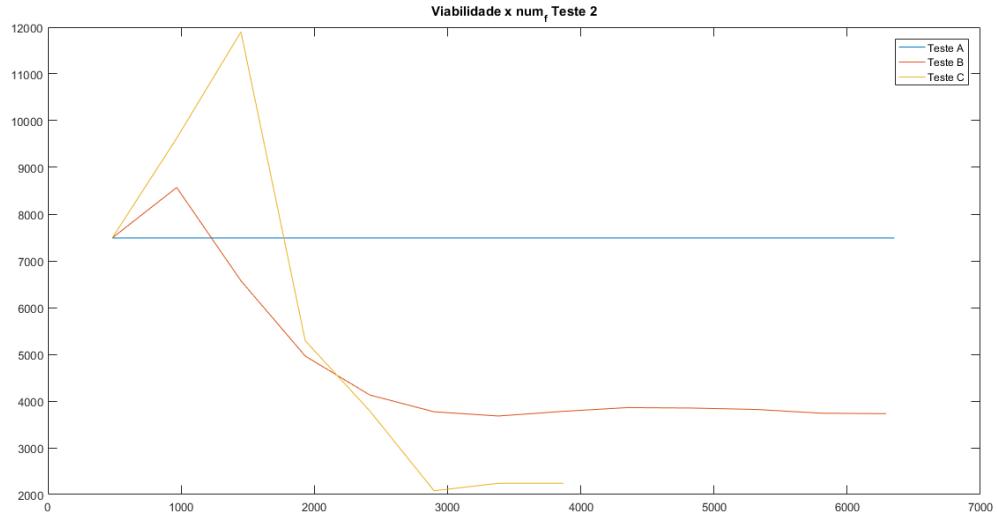
Figura 4.13 – Caso 2B - Caminho percorrido x vs y da ponta e da referência



Figura 4.14 – Caso 2C - Caminho percorrido x vs y da ponta e da referência



Figura 4.15 – Caso 2 - Num de fun x Viabilidade



4.1.4 Caso 3 - Variação na aceleração

Podemos observar na figura 4.16 uma curva de velocidade diferente, dado que reduzimos a aceleração na geração de comando, a velocidade não conseguiu alcançar o nível de velocidade desejada e assim toma-se a forma de um triângulo. Já pela figura 4.17, podemos observar que a função teve dificuldades para solucionar o Caso A, enquanto para o Caso B teve um comportamento semelhante aos Casos anteriores, com um aumento nos valores de viabilidade. Um dos motivos possíveis para essa dificuldade se da no tamanho dos vetores trabalhados, no Caso A o vetor de posições possuía 401 elementos, e levou 195s para realizar a simulação, enquanto para o Caso B o vetor de posição possuia 221 elementos e levou 78s, valores bem mais próximos aos Casos referência assim como suas curvas.

Figura 4.16 – Caso 3A - Comportamento no tempo das velocidades em x e y da ponta e da referência

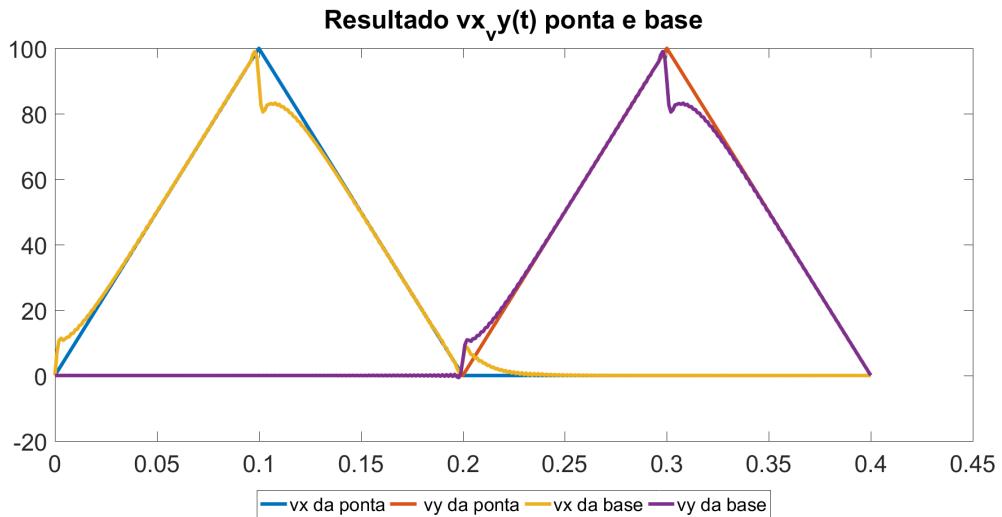
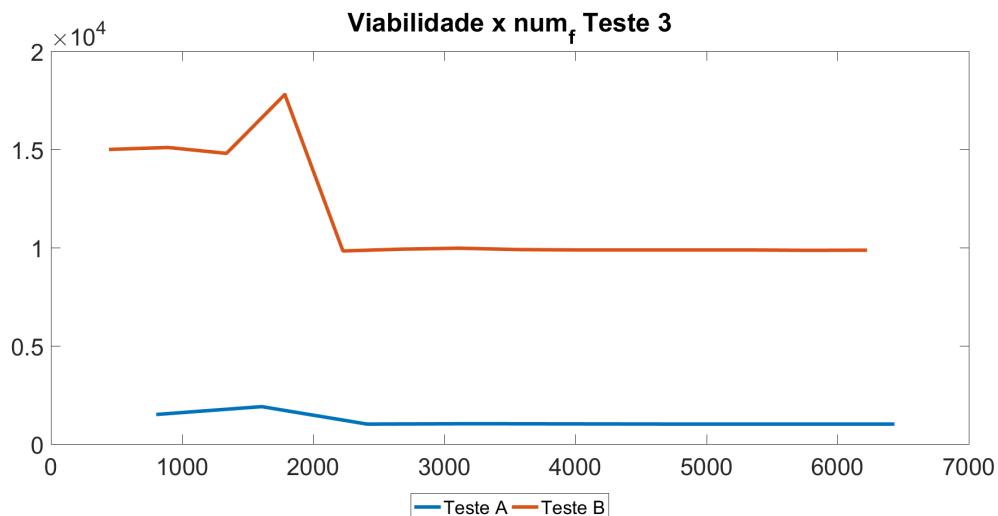


Figura 4.17 – Caso 3 - Num de fun x Viabilidade



4.1.5 Caso 4 - Variação da velocidade

Variando a velocidade desejada, através do Gcode de entrada, analisamos os impactos de velocidades desejadas mais baixas e mais altas. As mais baixas (Caso A) tem seu maior impacto no tamanho dos vetores e no tempo de simulação, já que percorre o percurso em mais tempo, gerando uma malha com mais pontos, dada a resolução de dt como fixa. Ficando assim com um tempo de simulação de 219s e 421 elementos nos vetores de posição. Enquanto para o Caso B, o

tempo de simulação ficou em 60s e com 181 elementos, mas dada a velocidade desejada maior e uma mesma aceleração máxima na geração de comando, a curva de velocidade se assemelha a do Caso 3A exposta na figura 4.18 e com a curva de deslocamentos apresentada na figura 4.19. Podemos comparar também o padrão de convergência na figura 4.20

Figura 4.18 – Caso 4B - Comportamento no tempo das velocidades em x e y da ponta e da referência

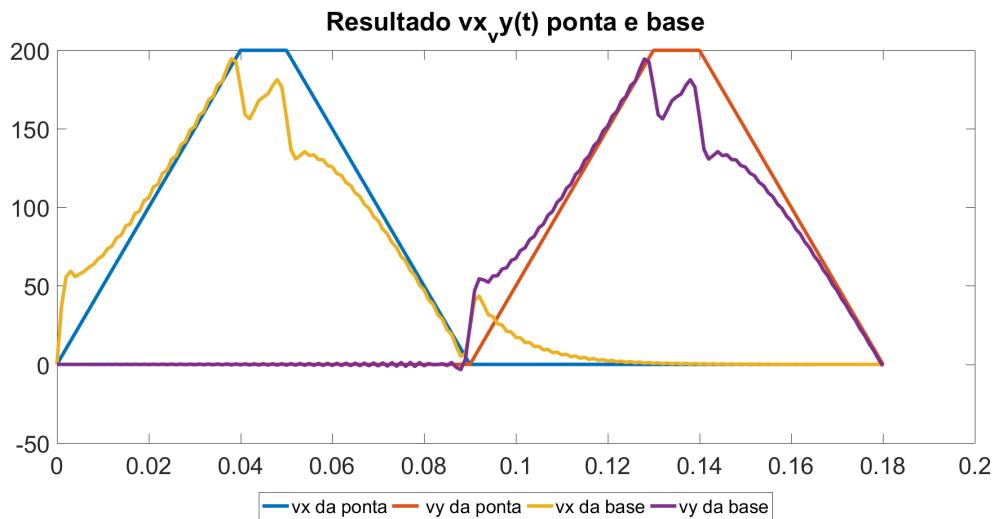


Figura 4.19 – Caso 4B - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência

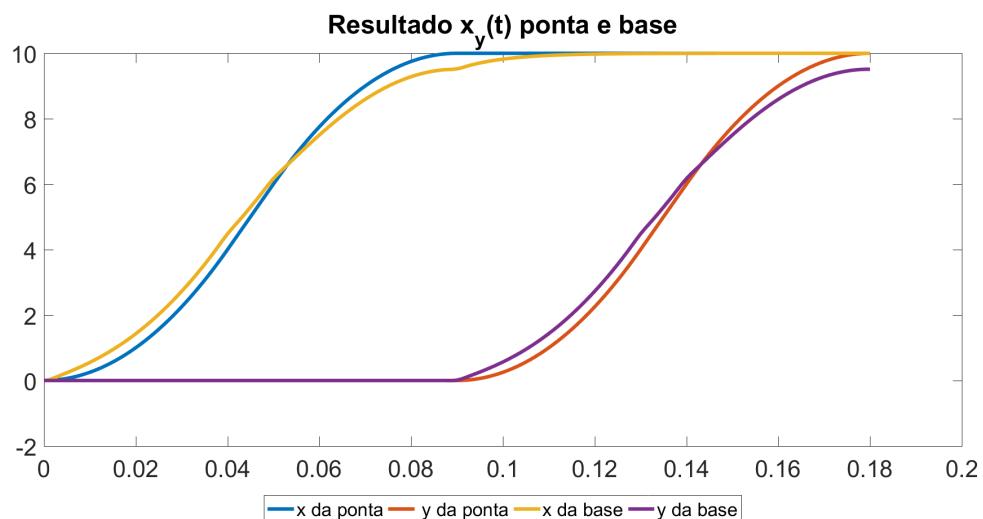
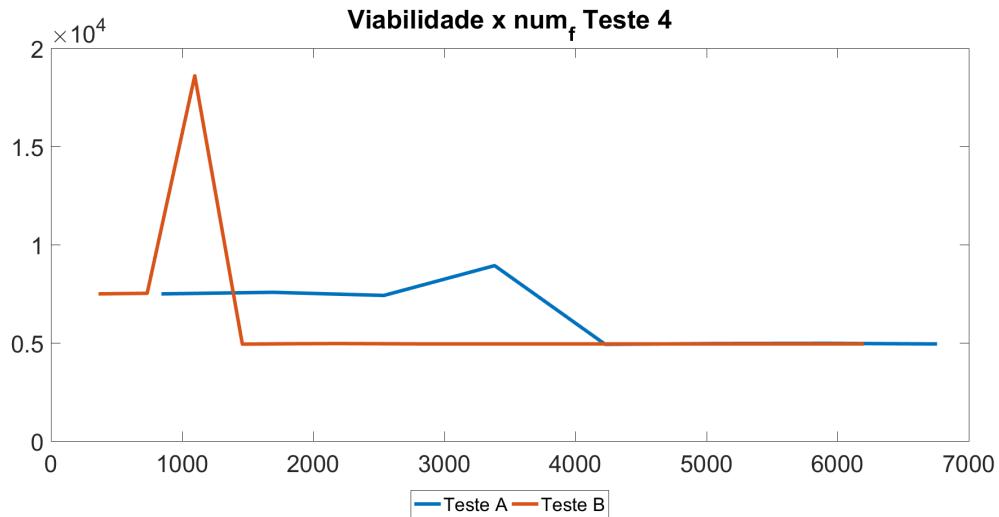


Figura 4.20 – Caso 4 - Num de fun x Viabilidade



4.1.6 Caso 5 - Variação do passo de tempo

Por fim, evidenciamos as diferenças nos resultados com resoluções diferentes para a malha de tempo e seu impacto nas respostas. As figuras 4.21, 4.22 e 4.23 revelam principalmente a influência desse parâmetro nas oscilações nos gráficos de velocidade, entretanto a diferença é menos nítida se avaliados os gráficos de deslocamento das figuras 4.24, 4.25 e 4.26.

Figura 4.21 – Caso 5A - Comportamento no tempo das velocidades em x e y da ponta e da referência

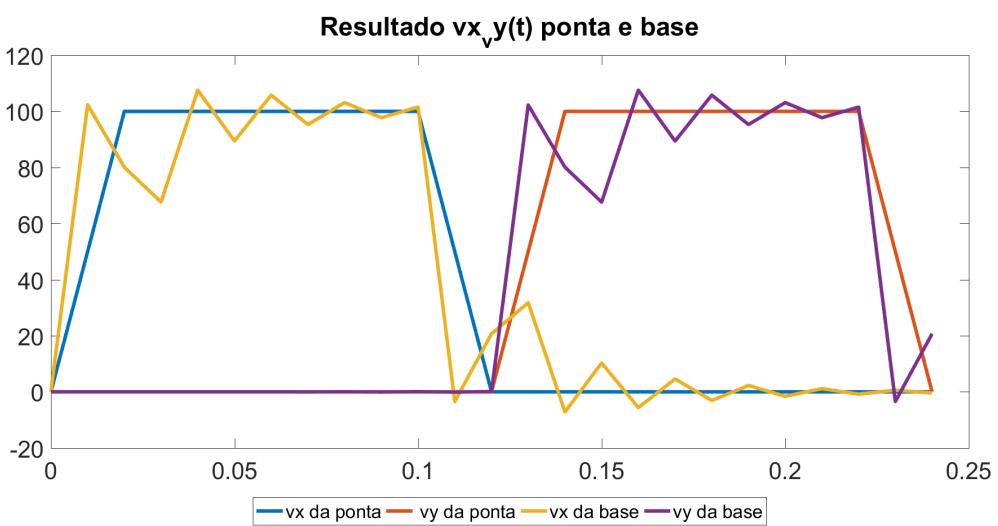


Figura 4.22 – Caso 5B - Comportamento no tempo das velocidades em x e y da ponta e da referência

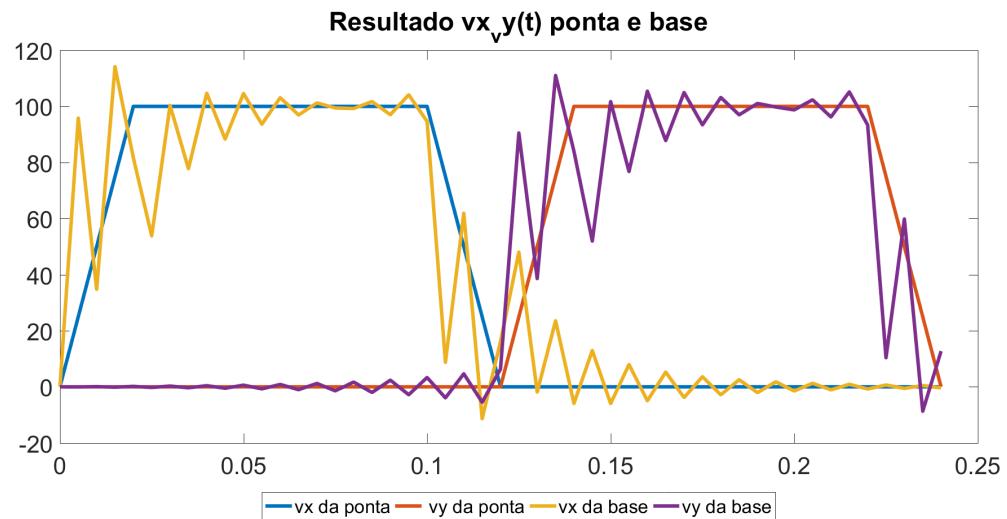


Figura 4.23 – Caso 5C - Comportamento no tempo das velocidades em x e y da ponta e da referência

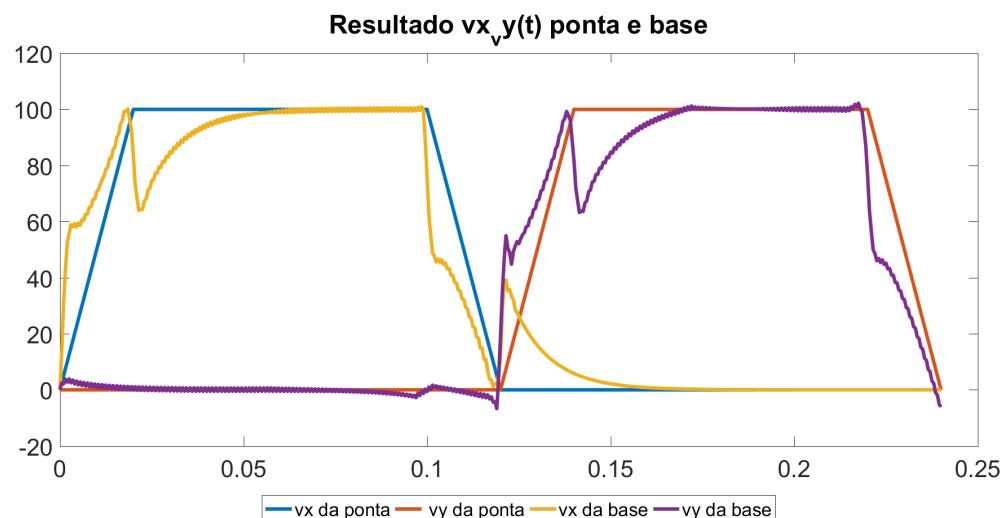


Figura 4.24 – Caso 5A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência

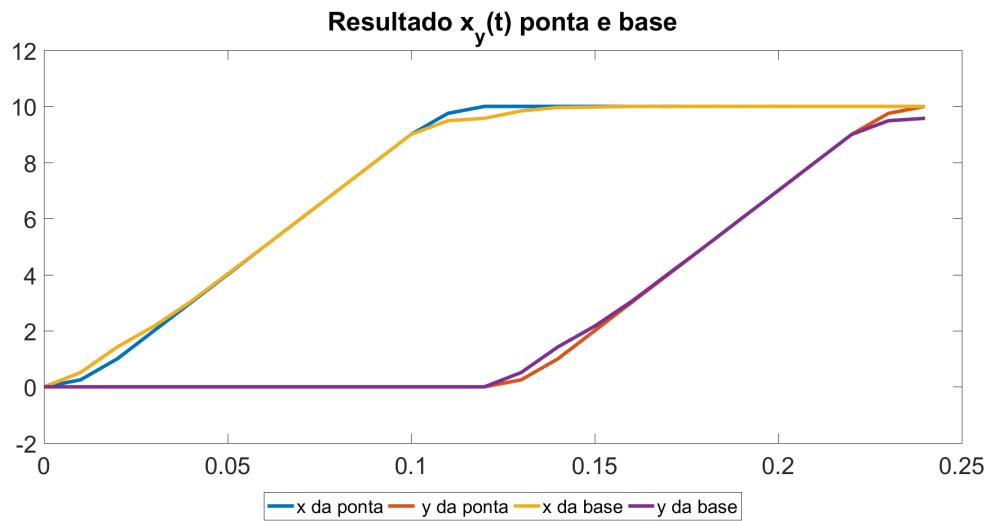


Figura 4.25 – Caso 5B - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência

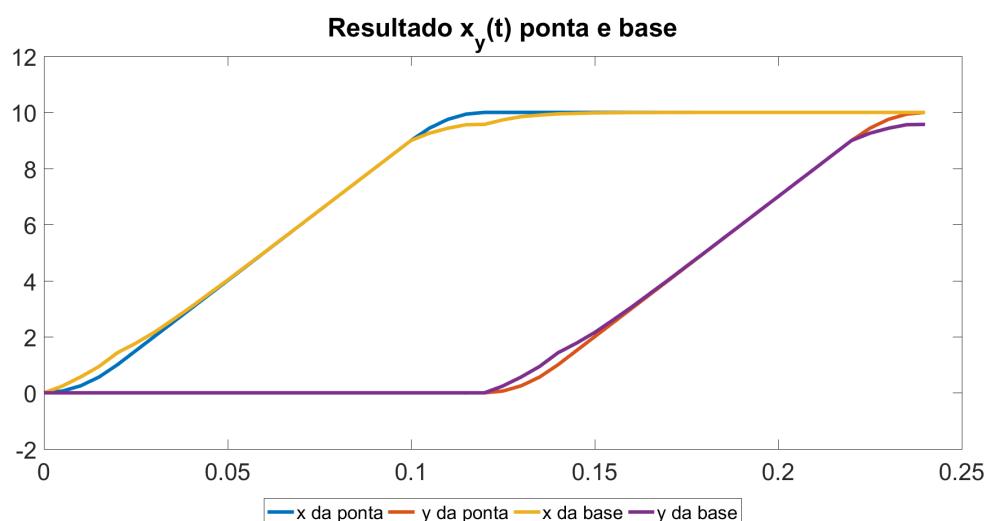
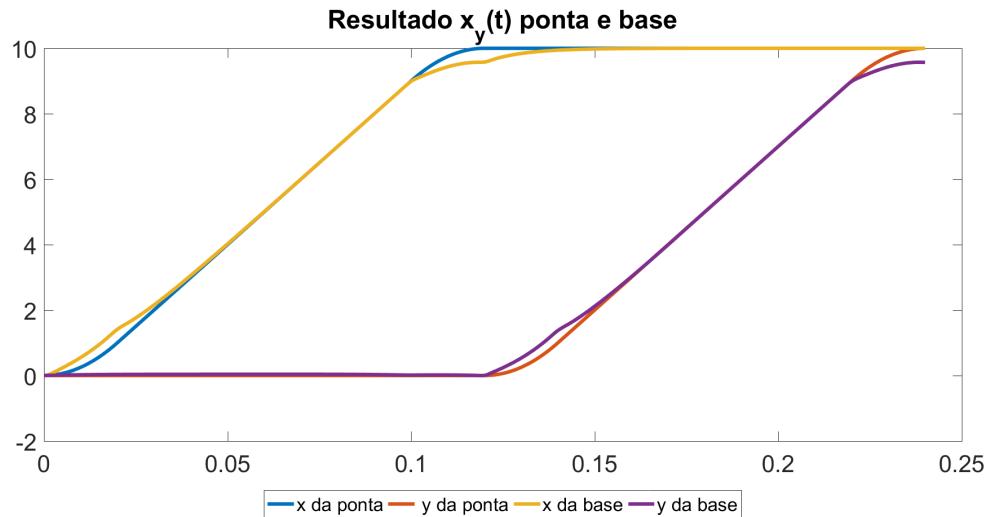
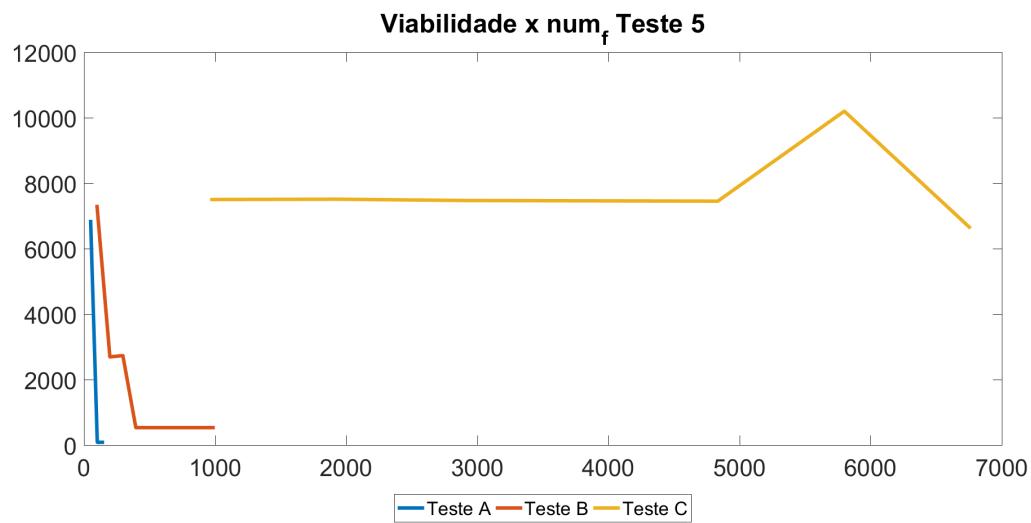


Figura 4.26 – Caso 5C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência



Podemos notar também o impacto na velocidade e facilidade de se convergir, podendo ser observado na figura 4.27, assim como os tempos de simulação 0, 7s, 3s e 279s para os Casos A, B e C respectivamente. Além do tamanho dos vetores que está diretamente relacionado, respectivamente em 25, 49 e 481 para os Casos A, B e C.

Figura 4.27 – Caso 5 - Num de fun x Viabilidade



4.2 Considerações futuras

4.2.1 Combinação com outros algoritmos

Uma possível abordagem a ser explorada utilizando a ideia do método deste trabalho é a sobreposição de algoritmos, onde um método referenciado em uma planta do sistema poderia buscar remover uma parcela das vibrações, atuando de forma estagiada, com a participação de um método como *InputShaping* para atacar as vibrações remanescentes.

4.2.2 Tuning

Uma possibilidade que o tipo de método abordado neste trabalho oferece é a capacidade de otimizar, de maneira semelhante aos mapas de injeção eletrônica para carros, os parâmetros da planta para uma determinada posição. Assim, oferecendo a capacidade de se ajustar em grande nível de detalhe as peculiaridades do sistema, podendo até construir a malha utilizando sensores, semelhantemente a rotinas de configuração de *InputShaping* que amostram o comportamento em frequência no ponto central da impressora. Considerando também que a utilização desse tipo de malha, teria pouco impacto computacional.

5 CONCLUSÃO

Conclusão

Nesta monografia, foi abordada a geração de comandos em impressoras 3D utilizando o método de manufatura aditiva conhecido como *Fused Deposition Modeling* (FDM). O objetivo principal do trabalho foi investigar e desenvolver metodologias para possibilitar velocidades de movimentação maiores e garantir a precisão dimensional das peças produzidas.

No referencial teórico (Capítulo 2), foram apresentados os conceitos fundamentais da manufatura aditiva e do FDM, incluindo o uso de algoritmos de controle, análise de trajetórias e modelagem dinâmica do sistema.

A metodologia utilizada (Capítulo 3) envolveu o uso do *software Matlab*, em particular a função fmincon, para a geração de comandos. Foram descritas as etapas para a leitura do *Gcode*, a geração de comandos base, incluindo as curvas trapezoidais de velocidade, a interpolação. Além disso, foram descritas as etapas de implementação do método desenvolvido neste trabalho, incluindo a modelagem dinâmica necessária, as funções de suporte como a integração implícita não linear e o uso da função FMINCON do Matlab.

No Capítulo 4, foram apresentados os resultados e discussões. Correlações entre as variáveis de entrada foram analisadas, bem como as influências do modelo dinâmico nas simulações e seus impactos na performance computacional. As dificuldades encontradas foram discutidas de maneira explicar a abordagem trilhada pelo presente trabalho, bem como discussões referentes a possíveis maneiras de lidar com os mesmos. Por fim, foi discutido abordagens para trabalhos futuros, como a implementação de uma modelagem do sistema dinâmico para regiões específicas da impressora.

Com base nos resultados obtidos e nas discussões realizadas, conclui-se que a geração de comandos é um aspecto fundamental para a qualidade e eficiência da manufatura aditiva por FDM. As metodologias e técnicas abordadas neste trabalho fornecem uma base sólida para a melhoria contínua do processo, possibilitando um avanço nas técnicas disponíveis para serem exploradas com objetivo de permitir uma produção de peças com maior precisão dimensional e redução de tempo de fabricação.

Em suma, este estudo representa um passo importante para o aprimoramento da manufatura aditiva por FDM, fornecendo subsídios teóricos e metodológicos para a geração de comandos mais eficientes e precisos. Espera-se que as contribuições deste trabalho incentivem

pesquisas adicionais e promovam avanços na área, impulsionando a utilização da manufatura aditiva em diversos setores industriais.

REFERÊNCIAS

- BIKAS, H.; STAVROPOULOS, P.; CHRYSSOLOUDIS, G. Additive manufacturing methods and modelling approaches: a critical review. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 83, p. 389–405, 2016.
- 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.
- HARGRAVES, C. R.; PARIS, S. W. Direct trajectory optimization using nonlinear programming and collocation. **Journal of guidance, control, and dynamics**, v. 10, n. 4, p. 338–342, 1987.
- 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.
- 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.I.], 2020. p. 1365–1371.