



JOAO VIVAS CISALPINO

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

LAVRAS - MG

2023

JOAO VIVAS CISALPINO

**CONTROLE DE TRAJETÓRIA DE IMPRESSORAS 3D UTILIZANDO
ALGORITMO 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.

46 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
ALGORITMO 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



Este trabalho aborda a geração de comandos em impressoras 3D utilizando o método de manufatura aditiva conhecido como *Fused Deposition Modeling* (FDM). O objetivo principal é investigar e desenvolver metodologias 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 referencial teórico apresenta os fundamentos da manufatura aditiva, o FDM e técnicas de controle, trajetória e modelagem dinâmica do sistema. ~~A metodologia envolve o uso do software Matlab e a função fmincon para a geração de comandos.~~ Os resultados mostram correlações entre as variáveis de entrada, influências do modelo dinâmico e a performance computacional. ~~Dificuldades como a função objetivo, restrições não lineares da base e variáveis principais são discutidas.~~ Possíveis abordagens, como divisão de etapas de solução, são sugeridas. Conclui-se que a geração de comandos é crucial para a qualidade e eficiência da manufatura aditiva por FDM, e pesquisas futuras podem explorar soluções mais robustas. Este estudo contribui para o aprimoramento da manufatura aditiva, incentivando pesquisas adicionais e impulsionando sua aplicação em diversos setores industriais.

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

ABSTRACT

This work addresses the command generation in 3D printers using the additive manufacturing method known as Fused Deposition Modeling (FDM). The main objective is to investigate and develop control methodologies for command generation in 3D printers, enabling higher speeds while ensuring dimensional accuracy of the produced parts. The theoretical framework presents the fundamentals of additive manufacturing, FDM, and control techniques, trajectory analysis, and dynamic modeling of the system. The methodology involves the use of Matlab software and the `fmincon` function for command generation. The results demonstrate correlations between input variables, influences of the dynamic model, and computational performance. Difficulties such as the objective function, nonlinear constraints of the base, and main variables are discussed. Possible approaches, such as stage division for problem solving, are suggested. It is concluded that command generation is crucial for the quality and efficiency of FDM additive manufacturing, and future research can explore more robust solutions. This study contributes to the improvement of additive manufacturing, encouraging further research and promoting its application in various industrial sectors.

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

LISTA DE FIGURAS

Figura 2.1 – Distribuição de uso de MA nas indústrias	13
Figura 2.2 – Princípio e processo de impressão para FDM	14
Figura 2.3 – Número de artigos publicados sobre FDM ao longo do tempo	14
Figura 2.4 – Distribuição das pesquisas sobre FDM	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 – Ilustração integração implícita	21
Figura 3.1 – Curva de velocidade trapezoidal	22
Figura 3.2 – Curva de velocidade triangular	23
Figura 3.3 – Visualização das posições do sistema	25
Figura 4.1 – Caso referência - Comportamento no tempo das velocidades em x e y da ponta e da referência	30
Figura 4.2 – Caso referência - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	31
Figura 4.3 – Caso referência - Caminho percorrido x vs y da ponta e da referência . . .	31
Figura 4.4 – Caso referência - Num de fun x Viabilidade	32
Figura 4.5 – Caso 1A - Comportamento no tempo das velocidades em x e y da ponta e da referência	32
Figura 4.6 – Caso 1B - Comportamento no tempo das velocidades em x e y da ponta e da referência	32
Figura 4.7 – Caso 1C - Comportamento no tempo das velocidades em x e y da ponta e da referência	33
Figura 4.8 – Caso 1A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	33
Figura 4.9 – Caso 1C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	34
Figura 4.10 – Caso 1A - Caminho percorrido x vs y da ponta e da referência	34
Figura 4.11 – Caso 1C - Caminho percorrido x vs y da ponta e da referência	35
Figura 4.12 – Caso 1 - Num de fun x Viabilidade	35

Figura 4.13 – Caso 2B - Caminho percorrido x vs y da ponta e da referência	36
Figura 4.14 – Caso 2C - Caminho percorrido x vs y da ponta e da referência	36
Figura 4.15 – Caso 2 - Num de fun x Viabilidade	37
Figura 4.16 – Caso 3A - Comportamento no tempo das velocidades em x e y da ponta e da referência	37
Figura 4.17 – Caso 3 - Num de fun x Viabilidade	38
Figura 4.18 – Caso 4B - Comportamento no tempo das velocidades em x e y da ponta e da referência	38
Figura 4.19 – Caso 4B - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	39
Figura 4.20 – Caso 4 - Num de fun x Viabilidade	39
Figura 4.21 – Caso 5A - Comportamento no tempo das velocidades em x e y da ponta e da referência	40
Figura 4.22 – Caso 5B - Comportamento no tempo das velocidades em x e y da ponta e da referência	40
Figura 4.23 – Caso 5C - Comportamento no tempo das velocidades em x e y da ponta e da referência	41
Figura 4.24 – Caso 5A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	41
Figura 4.25 – Caso 5B - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	42
Figura 4.26 – Caso 5C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência	42
Figura 4.27 – Caso 5 - Num de fun x Viabilidade	43

LISTA DE TABELAS

Tabela 3.1 – Tabela de parâmetros opcionais da FMINCON	26
Tabela 3.2 – Tabela de parâmetros referência dos Estudos de Caso	28
Tabela 3.3 – Tabela dos Estudos de Caso	28
Tabela 3.4 – Tabela de especificações do computador	29

SUMÁRIO

1	INTRODUÇÃO	9
1.1	Impressora 3d	10
2	REFERENCIAL TEÓRICO	12
2.1	Manufatura Aditiva	12
2.2	<i>Fused Deposition Modeling</i>	12
2.2.1	<i>Feedforward</i>	15
2.2.1.1	<i>Input Shaper</i>	16
2.2.1.2	<i>Filtered basis function (FBF)</i>	16
2.2.2	Geração de comando	18
2.2.3	Gcode	18
2.2.4	Curvas de velocidade trapezoidal	19
2.3	Espaço de Estados	20
2.4	Integração implícita utilizando programação não linear	20
3	METODOLOGIA	22
3.1	Geração de Comando	22
3.1.1	Curva trapezoidal de velocidade	22
3.1.2	Interpolação	24
3.2	Modelagem dinâmica de uma impressora 3D	24
3.2.1	Espaço de estados	25
3.3	FMINCON	26
3.3.1	Configurações da função	26
3.3.2	Restrições lineares e limites de borda	26
3.3.3	Restrições não lineares	27
3.3.4	Função objetivo	27
3.4	Solução da trajetória da base	27
3.5	Estudos de caso	28
4	RESULTADOS E DISCUSSÃO	30
4.1	Resultados	30
4.1.1	Caso Padrão	30
4.1.2	Caso 1 - Variação da frequência	31
4.1.3	Caso 2 - Variação do coeficiente de amortecimento	34

4.1.4	Caso 3 - Variação na aceleração	35
4.1.5	Caso 4 - Variação da velocidade	37
4.1.6	Caso 5 - Variação do passo de tempo	38
4.2	Considerações futuras	40
4.2.1	Combinação com outros algoritmos	40
4.2.2	<i>Tuning</i>	41
5	CONCLUSÃO	44
	REFERÊNCIAS	46

1 INTRODUÇÃO

A manufatura aditiva tem se mostrado uma tecnologia promissora para a fabricação de peças e componentes em diversas áreas, como engenharia, medicina e indústria aeroespacial.

Com a crise global logística causada pela pandemia, a manufatura aditiva ganhou destaque devido à sua capacidade de produção flexível e descentralizada. Nesse contexto, a impressão 3D, em particular o método *Fused Deposition Modeling* (FDM), tem se tornado cada vez mais relevante, encontrando aplicações variadas em setores como aeroespacial, automobilístico e prototipagem rápida, tornando-se mais acessível.

A modelagem digital, por exemplo *Computer Aided Design* (CAD), anda lado a lado ao processo de impressão 3D e desempenha um papel fundamental no processo de criação e distribuição, permitindo a concepção de modelos tridimensionais precisos que podem ser compartilhados e reproduzidos de maneira descentralizada. Para se imprimir um desses modelos é feita uma preparação por meio de um *software* de fatiamento (*slicer*), que divide o modelo em camadas e gera os comandos necessários para a impressora 3D. A partir destes comandos a impressora interpreta os passos que deverão ser tomados e quando cada um deles deve ser realizado. Entre a interpretação e execução destes comandos existe uma diversidade de processos intermediários que tem impacto direto na qualidade e na velocidade de impressão.


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 metodologias para atuação de controle na geração de comandos em impressoras 3D utilizando o método FDM. Será abordado o referencial teórico, contemplando os princípios básicos da manufatura aditiva, os conceitos específicos do *Fused Deposition Modeling*, os métodos de geração de comando, como *look-ahead*, curvas de velocidade trapezoidal e o estado da arte quanto ao desenvolvimento de métodos de controle que buscam aprimorar a qualidade e diminuir o tempo necessário para imprimir uma peça, como *InputShaping*.

Será descrita a metodologia utilizada, descrevendo as etapas para a geração de comandos e a modelagem dinâmica da impressora 3D. Os resultados obtidos serão apresentados e discutidos, analisando as correlações entre as variáveis de entrada, as influências do modelo dinâmico e a performance computacional. ~~Serão abordadas as dificuldades encontradas durante o desenvolvimento do trabalho e possíveis abordagens para superá-las, como a combinação com outros algoritmos.~~

Por fim, serão apresentadas as conclusões, destacando as contribuições principais e apontando possíveis direções para pesquisas futuras. ~~A proposta é fornecer subsídios para aprimorar a geração de comandos em impressoras 3D, visando melhorar a qualidade e eficiência do processo de manufatura aditiva.~~

1.1 Impressora 3d

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 e geração de comando: 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.

2 REFERENCIAL TEÓRICO

2.1 Manufatura Aditiva

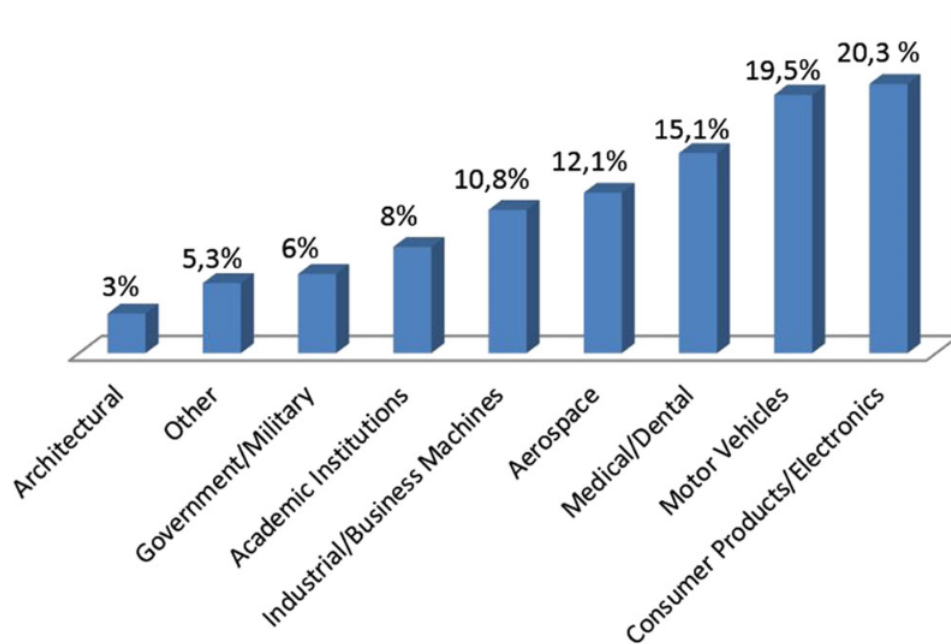
O princípio fundamental da manufatura aditiva (MA) consiste em fabricar um modelo tridimensional de forma integrada, dispensando a necessidade de planejar as operações de maneira individual e priorizando a consideração das configurações globais. O processo é calculado pelo fatiador com base em um modelo tridimensional digital, geralmente criado a partir de *Computer Aided Design* (CAD) e nas configurações do mesmo e resulta nas intruções necessárias para a máquina de manufatura aditiva construir o modelo físico. 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 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; CHRYSSOLOURIS, 2016). Na figura 2.1 podemos observar a distribuição do uso de tecnologias MA por tipo de indústria.

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

Figura 2.1 – Distribuição de uso de MA nas indústrias

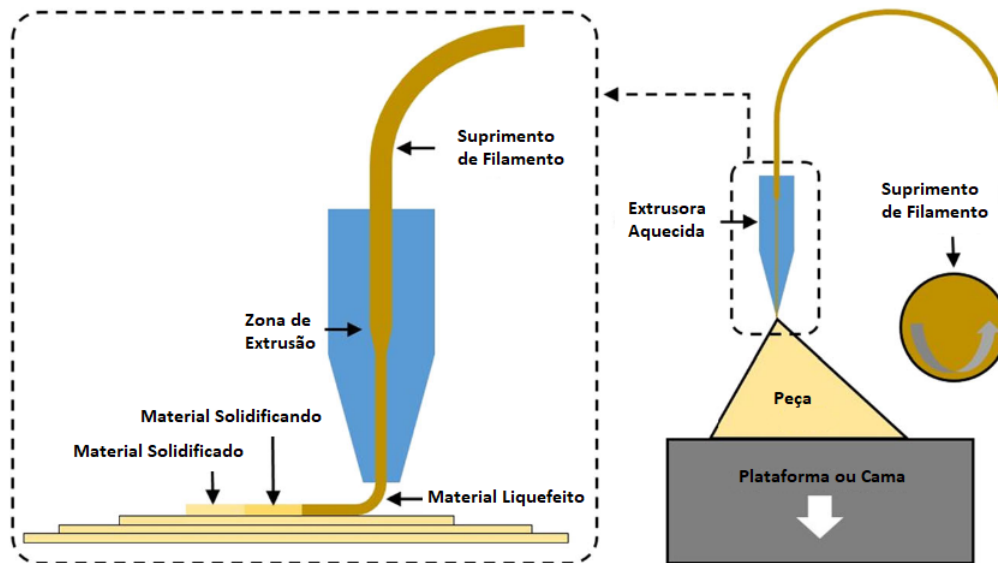


Fonte: BIKAS; STAVROPOULOS; CHRYSSOLOURIS, 2016

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). Podemos observar a disposição desses componentes na figura 2.2.

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 2.3), 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 27% são relacionados ao restante dos temas, incluindo avanços tecnológicos relacionados «««« HEAD a melhorias de *hardware* e *software*

Figura 2.2 – Princípio e processo de impressão para FDM



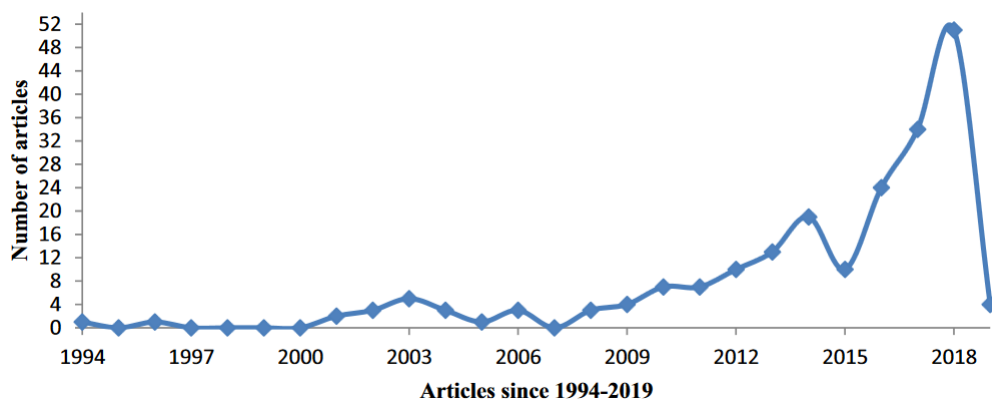
Fonte: Adaptado de BIKAS; STAVROPOULOS; CHRYSSOLOURIS, 2016

desses dispositivos (2.4). ===== a melhorias de *hardware* e *software* desses dispositivos.

(Figura 2). »»»» 37d3a707e6fb92d6614b2bbf36d28687a89c476d



Figura 2.3 – Número de artigos publicados sobre FDM ao longo do tempo

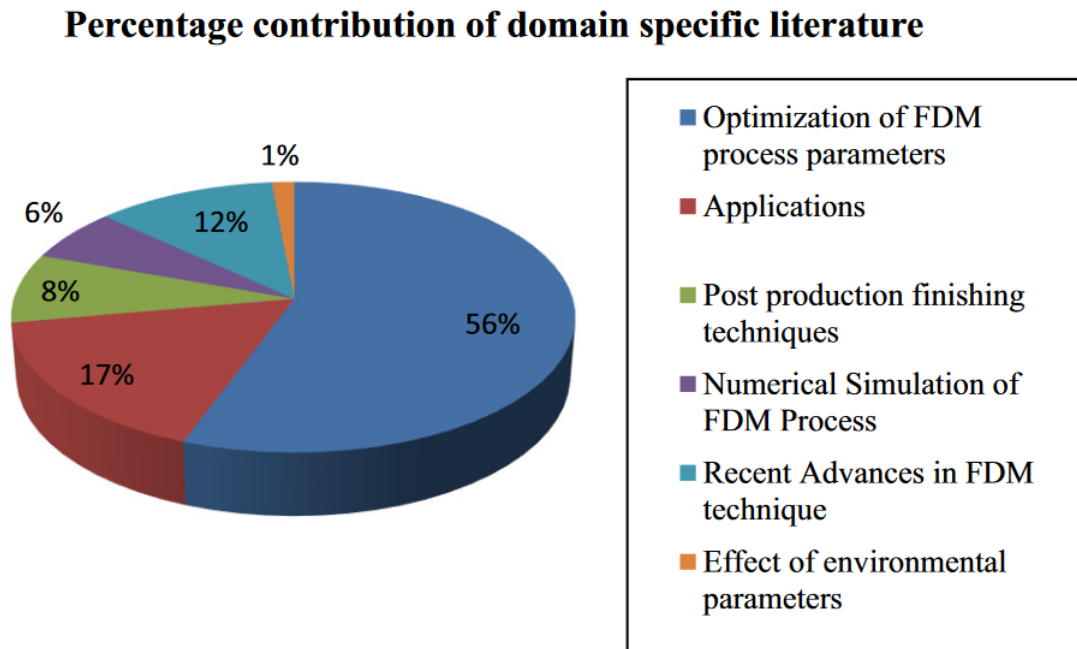


Fonte: VYAVAHARE *et al.*, 2020

Podemos separar, de maneira simplificada, o *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 digitalizado da peça a ser produzida, é 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 que ditarão a estratégia de trajetória para se realizar os diferentes deslocamentos na impressão. Esses



Figura 2.4 – Distribuição das pesquisas sobre FDM



Fonte: VYAVAHARE *et al.*, 2020

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, que permitiu velocidades de impressão mais altas, sem a necessidade de alteração física de impressoras antigas (KLIPPER, 2017a).

2.2.1 *Feedforward*

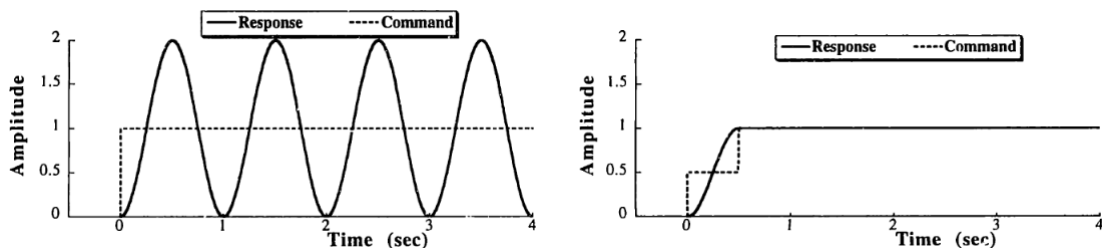
Dentre os métodos de controle em aplicações FDM o *Feedforward* é o mais eficiente dada as limitações de custo em impressoras 3D comuns e é capaz de ter um impacto maior em sistemas conhecidos e sensíveis ao erro, onde buscam corrigir o erro antes que ele aconteça. As principais limitações da aplicação 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



Ao conhecer a trajetória desejada e as características do sistema, é possível calcular uma série de comandos que levam em consideração essas características para modificar o comando de referência. Isso permite que a trajetória final seja a mais próxima 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

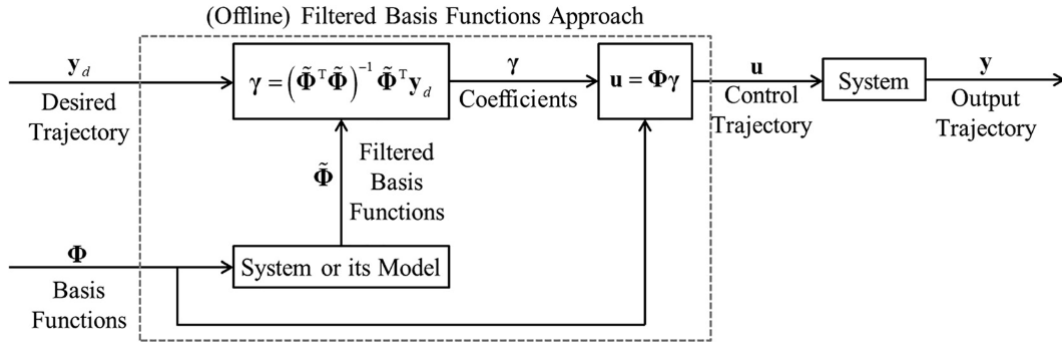
~~Essa abordagem vem sendo explorada na comunidade "faça você mesmo" a partir de 2017 quando a última patente desse método expirou, e tem aprimorado a área como um todo, empurrando os limites anteriores de velocidade e precisão, sendo popularizada pelo Klipper (KLIPPER, 2017b).~~

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)

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

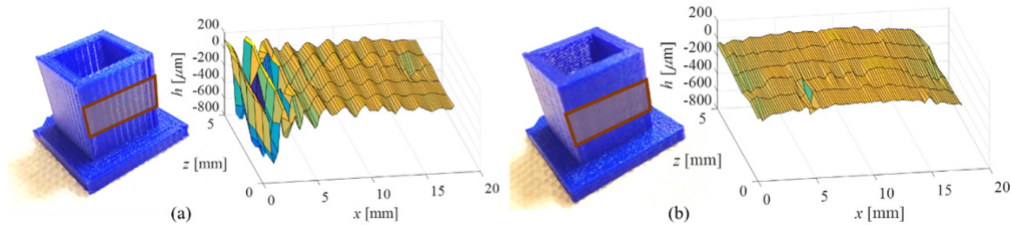
Figura 2.6 – Fluxograma FBF

Fonte: RAMANI *et al.*, 2017

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. ~~Apresentando resultados promissores apresentados na figura 2.7.~~

Figura 2.7 – Resultados práticos da LPFBF



Fonte: DUAN; YOON; OKWUDIRE, 2018

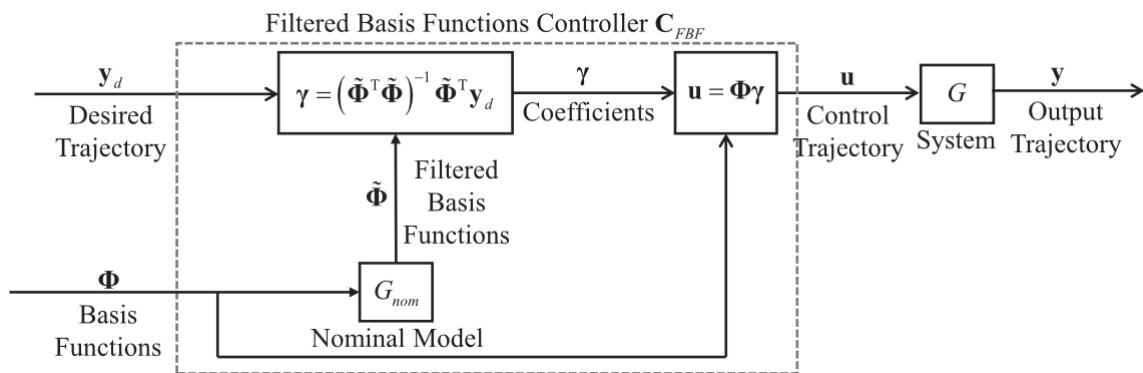
Com base nos desenvolvimentos nos trabalhos de (RAMANI *et al.*, 2017) e (DUAN; YOON; OKWUDIRE, 2018), comentados anteriormente, (RAMANI; EDOIMIOYA; OKWUDIRE, 2020) busca atacar um segundo desafio prático na implementação da FBF, sendo o primeiro desafio prático o custo computacional que foi endereçado pelo trabalho de (DUAN; YOON; OKWUDIRE, 2018) através da LFBFBF, 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 2.8.

Figura 2.8 – Fluxograma RFBF



Fonte: RAMANI; EDOIMIOYA; OKWUDIRE, 2020

2.2.2 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).

2.2.3 Gcode

O G-code (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 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 posicionamento (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.

2.2.4 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 máximas de cada movimento, assim é necessário planejar o comportamento da velocidade ao longo tempo dentro de cada comando.

Uma das maneiras mais simples para a criação dessa curva de velocidade é de considerar as transições de velocidade com aceleração constante, criando um perfil de velocidade trapezoidal, em geral com 3 segmentos. O primeiro segmento saindo da velocidade inicial para a velocidade máxima, o segundo segmento com velocidade constante e o terceiro segmento saindo da velocidade máxima alcançada para a velocidade final. Em alguns casos as condições do comando e as configurações de aceleração não possibilitam que a impressora alcance a velocidade máxima estabelecida no comando, nesses casos não existe o segundo segmento, ou seja, não existe no perfil um estágio de velocidade constante. (YU *et al.*, 2020; KLIPPER, 2017b). (figura ??).

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. 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).

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

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

2.4 Integração implícita utilizando programação não linear

HARGRAVES; PARIS (1987) descreve um algoritmo para a solução numérica direta de problemas de controle ótimo (HARGRAVES; PARIS, 1987). 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.

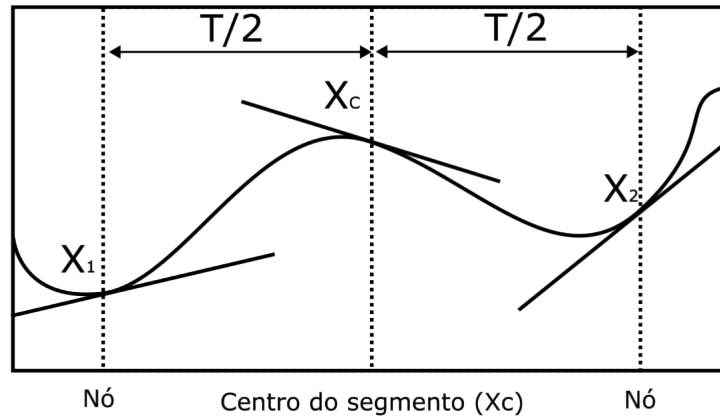
Uma das principais vantagens desse método é sua facilidade de implementação e sua capacidade de lidar com uma ampla gama de problemas de otimização de trajetória. Isso inclui a consideração de restrições de caminho, estados descontínuos e desigualdades de controle.

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 contínua, ou seja, o valor da função e de sua derivada precisa ser igual para ambos polinômios nas conexões como observado na figura 2.9

O procedimento base pode ser aplicado pelos seguintes passos.

Figura 2.9 – Ilustração integração implícita



Fonte: HARGRAVES; PARIS, 1987

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 equação 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 Geração de Comando

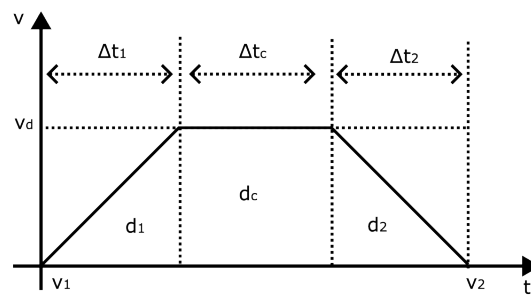
A geração de comando se inicia na leitura do Gcode para a obtenção dos movimentos e velocidades desejadas. Foi considerado no mapeamento do Gcode apenas comandos G1, extraindo as informações dos eixos X, Y e do *feedrate* (F). Com base nesses valores uma matriz 3 por n é criada, n sendo o número de comandos lidos do arquivo Gcode. 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**.

3.1.1 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 constroi 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 **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 3.1.

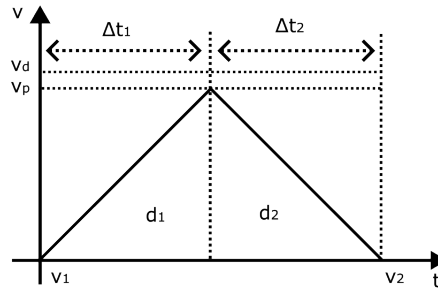
Figura 3.1 – Curva de velocidade trapezoidal



$$v_p = \sqrt{\frac{(v_1^2 + v_2^2)}{2} + a * d} \quad (3.1)$$

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

Figura 3.2 – Curva de velocidade triangular



maior do que a velocidade desejada, temos 3 fases de deslocamento que podem ser calculadas pelas equações 3.5 e 3.6. 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 3.5.

$$d_1 = \frac{(v_d^2 - v_1^2)}{(2 * a)} \quad \text{☞} \quad (3.2)$$

$$d_2 = \frac{(v_2^2 - v_d^2)}{(2 * a)} \quad (3.3)$$

$$d_1 = \frac{(v_p^2 - v_1^2)}{(2 * a)} \quad (3.4)$$

$$d_2 = \frac{(v_2^2 - v_p^2)}{(2 * a)} \quad (3.5)$$

$$d_c = d - (d_1 + d_2) \quad (3.6)$$

É possível calcular também os intervalos de tempo dessas fases, através das equações 3.11 e 3.9.

$$t_1 = \frac{(v_d - v_1)}{a} \quad (3.7)$$

$$t_2 = \frac{(v_2 - v_d)}{a} \quad (3.8)$$

$$t_c = \frac{d_c}{v_d} \quad (3.9)$$

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

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

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

Esses passos resultam em uma nova matriz contendo informações sobre o 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.

3.1.2 Interpolação

A partir dos pontos de posição, velocidade, aceleração e tempo de cada eixo, é utilizada uma função de interpolação esses intervalos em pontos intermediarios baseados em um passo de tempo definido para esta interpolação. Para se dividir esses intervalos é possível utilizar a equação 3.12 calculando o número de passos neste intervalo, anexando à matriz os passos de tempo e por fim o restante do intervalo, calculado pela equação 3.13. Com base nestes passos de tempo, é possível calcular o deslocamento para cada um destes passos através da equação 3.14.

$$N = \lceil \frac{\Delta t}{\Delta p} - 1 \rceil \quad (3.12)$$

$$\Delta t_f = \Delta t - \Delta p * N \quad (3.13)$$

$$\Delta d_i = \Delta v_i * \Delta t_i + \frac{a_s * \Delta t_i^2}{2} \quad (3.14)$$

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.



3.2 Modelagem dinâmica de uma impressora 3D

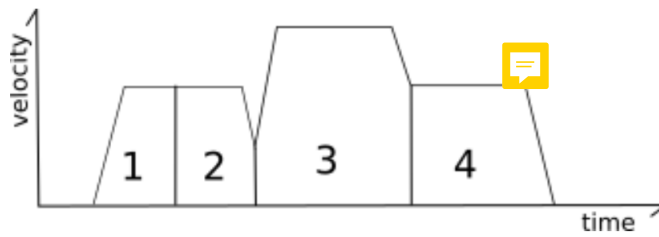
Para a modelagem do sistema mecânico são consideradas as seguintes simplificações:

- 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 3.3.

Figura 3.3 – Visualização das posições do sistema



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

(3.15)

3.2.1 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 com um número maior de equações. O modelo dinâmico do sistema é apresentado na formulação de espaço de estados na equação

3.16, onde $\dot{x} = A * x + B * u$. Baseado na equação 3.15 expandimos as matrizes e vetores na equação 3.17. Como o modelo dos eixos x e y estão sendo considerados como iguais, também podemos utilizar a equação 3.17 para a base do eixo y.

$$\dot{x} = A * x + B * u \quad (3.16)$$

$$\begin{bmatrix} \dot{x}_p \\ \ddot{x}_p \\ \dot{y}_p \\ \ddot{y}_p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_x}{m_x} & -\frac{c_x}{m_x} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{k_x}{m_x} & -\frac{c_x}{m_x} \end{bmatrix} \begin{bmatrix} x_p \\ \dot{x}_p \\ y_p \\ \dot{y}_p \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{k_x}{m_x} & \frac{c_x}{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{k_x}{m_x} & \frac{c_x}{m_x} \end{bmatrix} \begin{bmatrix} x_b \\ \dot{x}_b \\ y_b \\ \dot{y}_b \end{bmatrix} \quad (3.17)$$

3.3 FMINCON



3.3.1 Configurações da função

Foi utilizado o seguinte conjunto de configurações opcionais da função (tabela 3.1). As configurações não indicadas se mantem na configuração padrão, que pode ser encontrada nos manuais da mesma.

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

3.3.2 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 (0 a 200 milímetros) para as posições x_b e y_b .



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. Além disso, foi utilizado também como restrição de igualdade a arrancada, atrelado a um condicional para que se comporte como uma desigualdade. A variável de restrições de desigualdades não foi populada, os motivos são apresentados na seção de resultados e discussão.

3.3.4 Função objetivo

Foram realizados alguns testes com funções objetivo e seus resultados são apresentados na seção de resultados e discussão, entretanto a função objetiva foi definida como zero, ou seja, para qualquer valor ela retornara zero nas chamadas da FMINCON.

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 é fixado pela trajetória desejada presente na forma de variáveis globais.

Para conseguir realizar os cálculos necessários dentro da função de restrições não lineares é utilizado o seguinte conjunto de equações básicas (3.18) para se derivar a curva posição-tempo considerando aceleração constante, 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. Assim construindo o vetor de velocidade a partir da condição inicial de deslocamento e velocidade zero, para ambos os eixos (x e y).

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

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

3.5 Estudos de caso

Foi realizado uma série de simulações utilizando como base uma sequência de dois movimentos em 90 graus, 10 milímetros ao longo do eixo x e depois 10 milímetros ao longo do eixo y, através de um arquivo Gcode de teste. Em cada uma dessas simulações um dos parâmetros é alterado, enquanto o restante permanece em uma configuração base, apresentada na tabela 3.2. Além disso, em cada teste foram separados algumas variações do mesmo parâmetro em questão e as versões do teste são identificadas por uma letra de A a C. Assim, as configurações para as diferentes simulações adicionais segue a tabela 3.3.

Tabela 3.2 – Tabela de parâmetros referência dos Estudos de Caso

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

Tabela 3.3 – Tabela dos Estudos de Caso

Caso	Parâmetro	Valor A	Valor B	Valor C	Unidade
1	Frequência	50	200	500	rad/s
2	Coefficiente 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 tamanho 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 – Tabela de especificações do computador

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

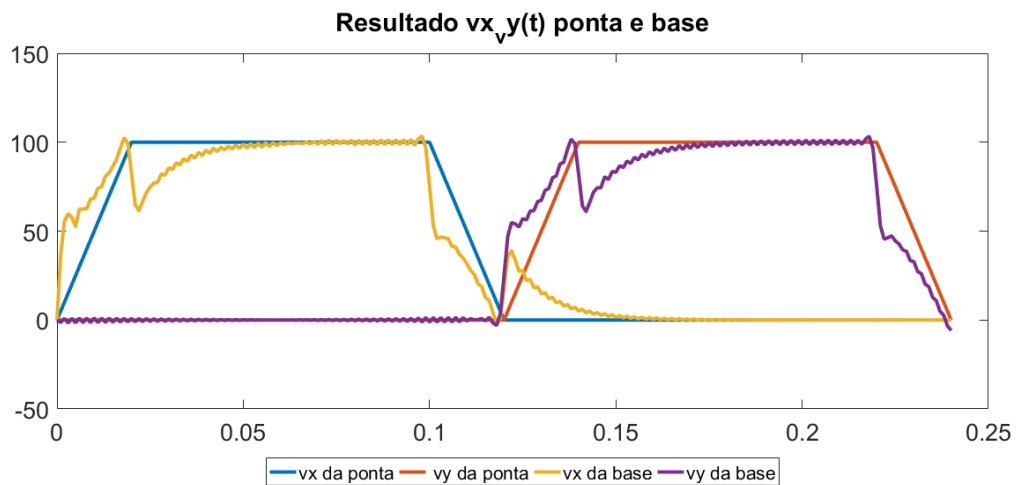
4 RESULTADOS E DISCUSSÃO

4.1 Resultados

4.1.1 Caso Padrão

Podemos observar os resultados das curvas geradas pelo Caso utilizando os parâmetros referência nas figuras 4.1, 4.2 e 4.3. Nelas temos o comportamento compensador, especialmente nas curvas que tem o tempo como abscissa. ~~Notamos também, principalmente na figura 4.1 uma oscilação de alta frequência que é um artefato da resposta da FMINCON, essa oscilação se encontrava em amplitude muito maior e foi controlada através de restrições quanto ao arranque, ou seja, a mudança da aceleração no tempo. Entretanto, essa restrição apresentou problemas de priorização de restrições que será comentado na seção de dificuldades.~~ Notou-se também que a amplitude dessas oscilações tende a ser menor conforme exista uma maior resolução na interpolação do tempo, ou seja, apresente dt menor.

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



No gráfico da figura 4.4 nos indica a progressção 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 níveis próximos na maioria dos Casos. Por fim, temos que o tempo elapsado

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

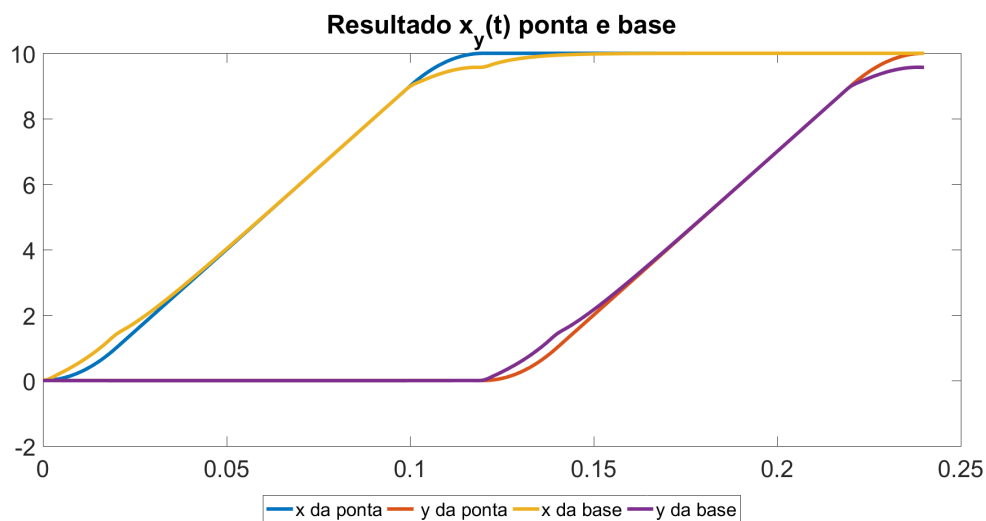


Figura 4.3 – Caso referência - Caminho percorrido x vs y da ponta e da referência

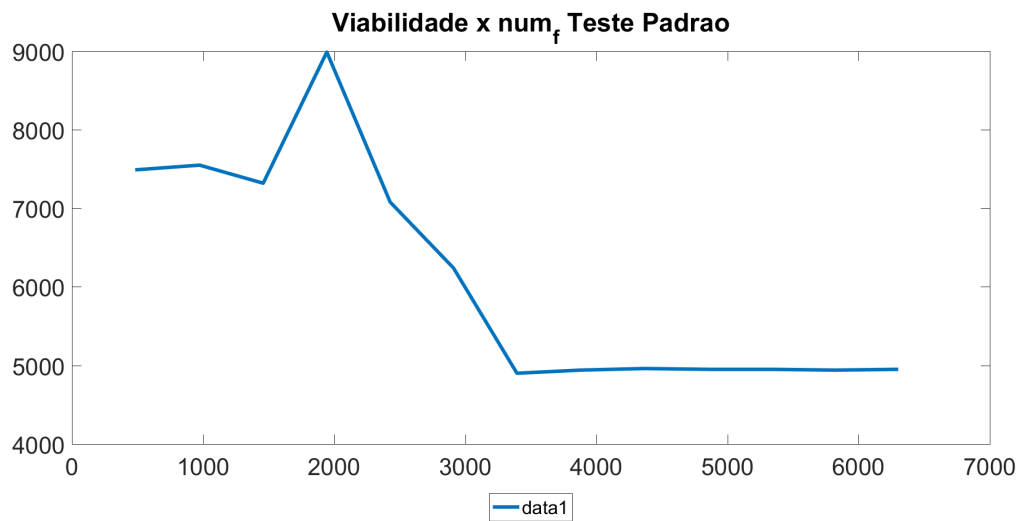


de simulação para este Caso foi de 89,8 segundos, com os vetores de posição possuindo um tamanho de 241 elementos.

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

Figura 4.4 – Caso referência - Num de fun x Viabilidade



como previsto de um sistema de rigidez maior. As diferenças podem ser observadas de forma clara se compadas 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

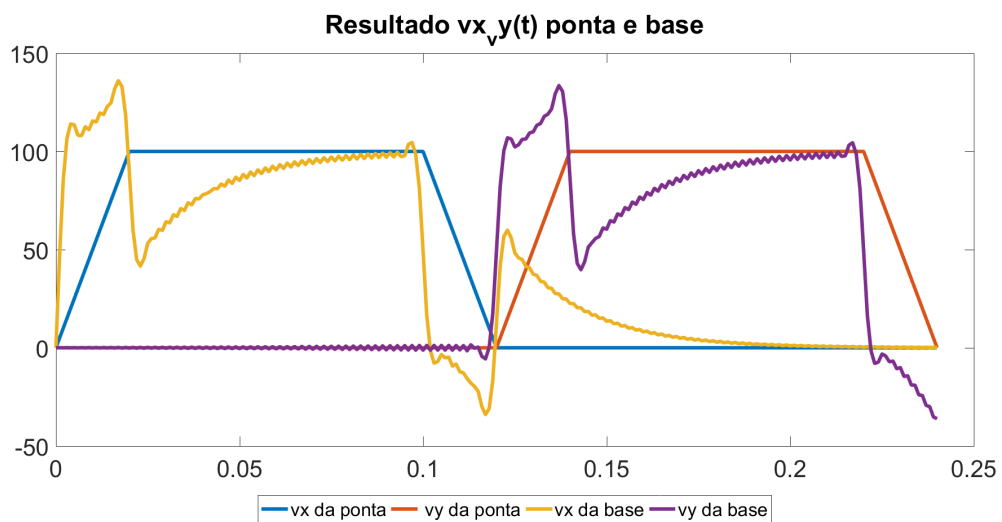


Figura 4.6 – Caso 1B - Comportamento no tempo das velocidades em x e y da ponta e da referência

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

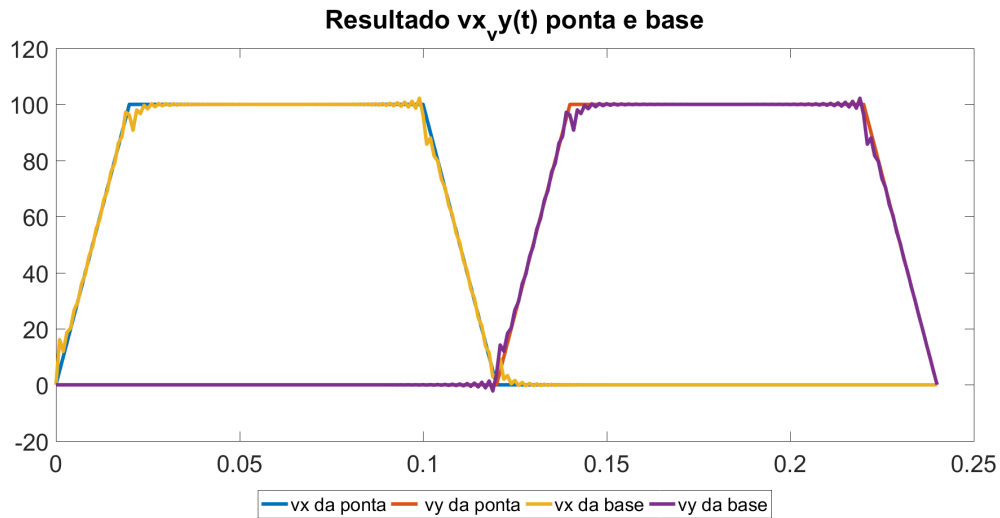
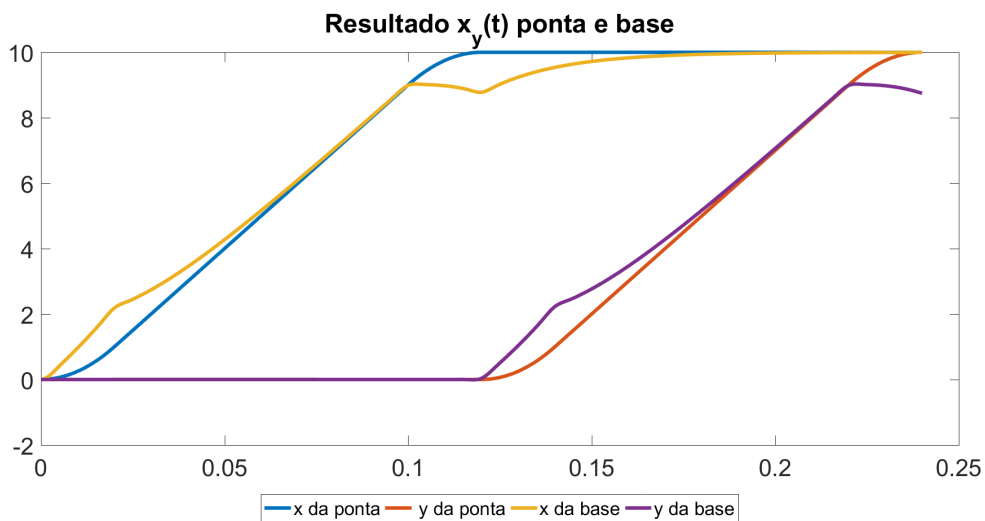


Figura 4.8 – Caso 1A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência



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

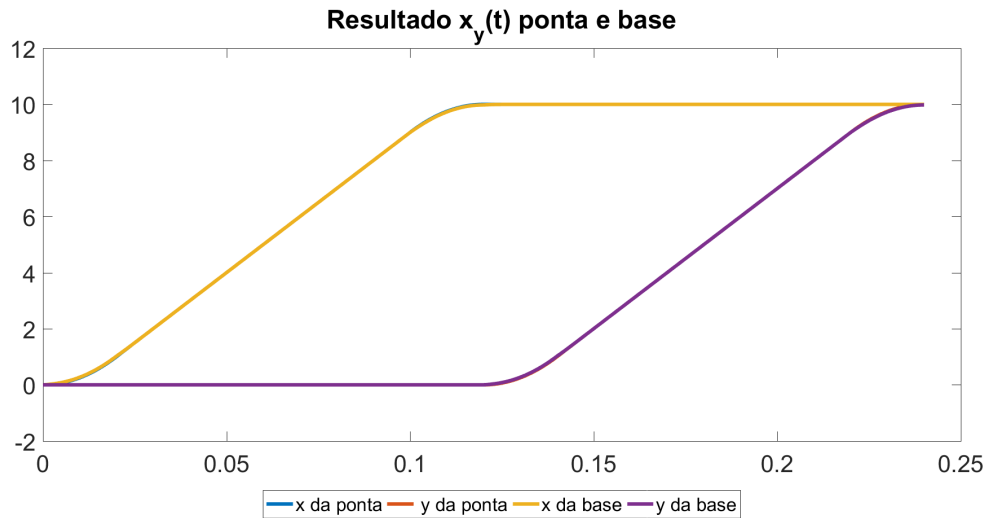


Figura 4.10 – Caso 1A - Caminho percorrido x vs y da ponta e da referência



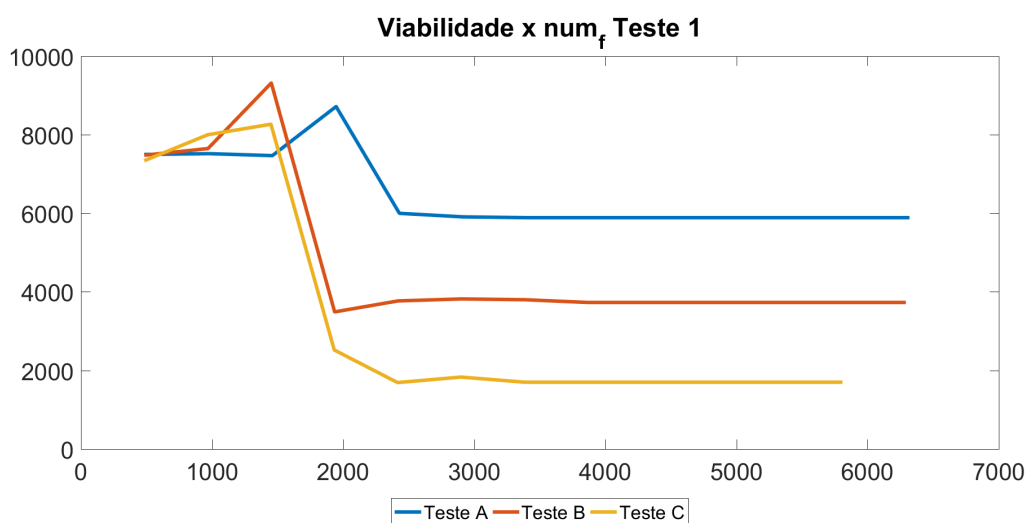
4.1.3 Caso 2 - Variação do coeficiente de amortecimento

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 o mesmos valores de entrada, esse comportamento é identificado na figura 4.15, onde podemos observar também um padrão convergência para valores de via-

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



Figura 4.12 – Caso 1 - Num de fun x Viabilidade



bilidade 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.

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

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



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 possuía 221 elementos e levou 78s, valores bem mais próximos aos Casos referência assim como suas curvas.

Figura 4.15 – Caso 2 - Num de fun x Viabilidade

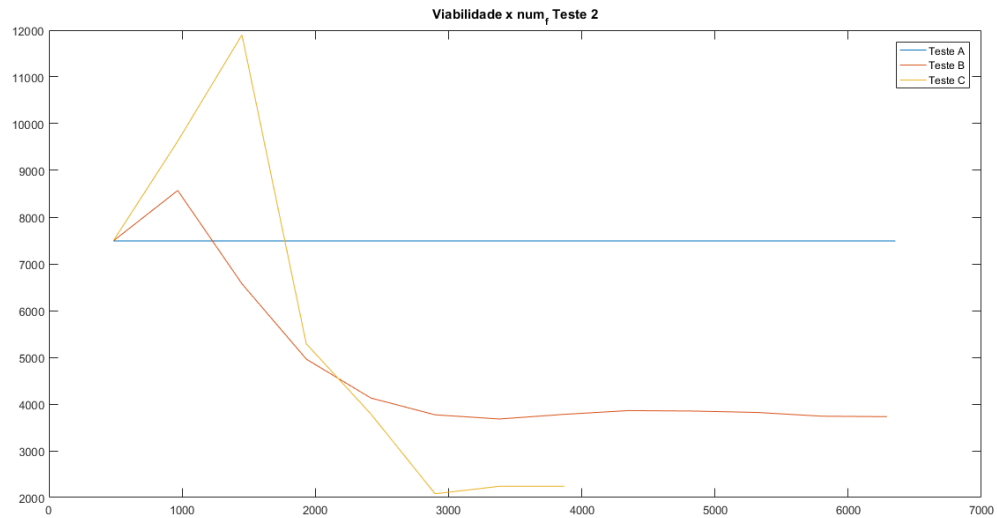
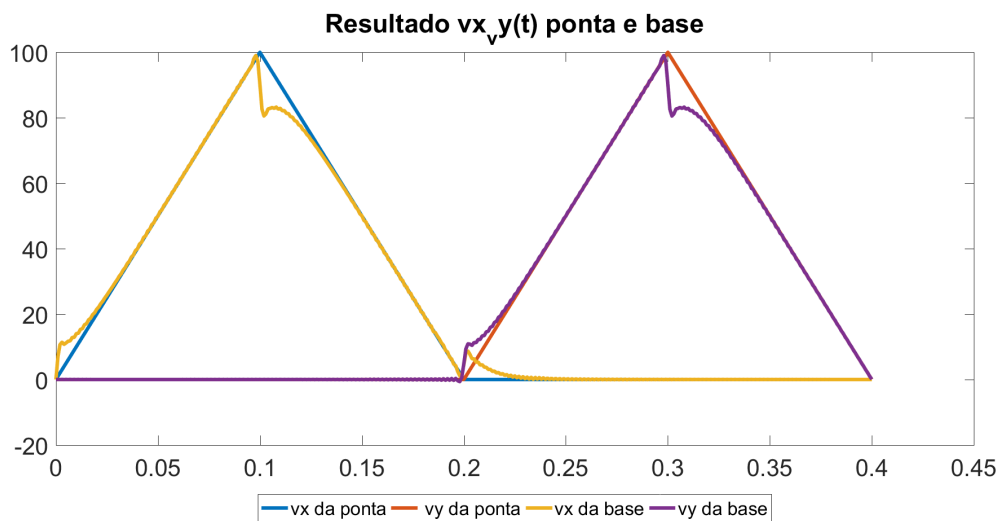


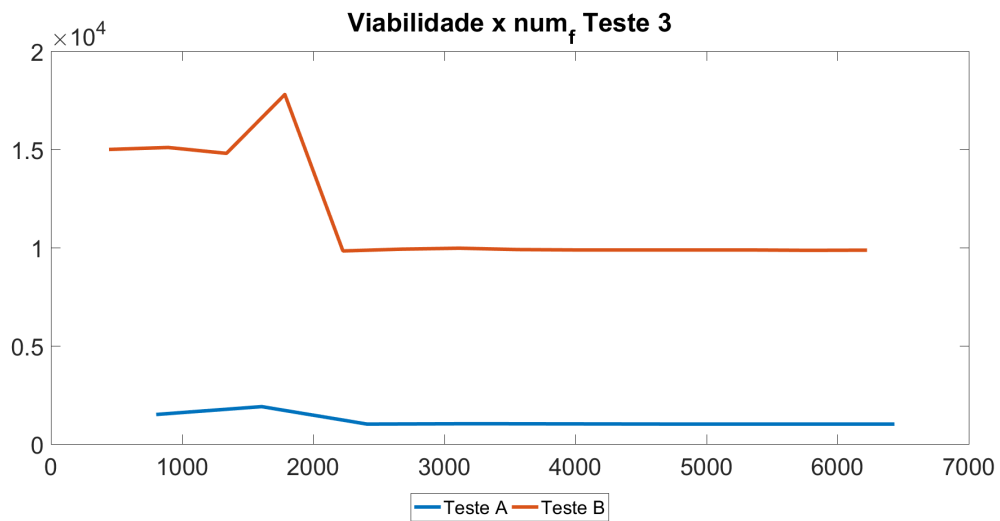
Figura 4.16 – Caso 3A - Comportamento no tempo das velocidades em x e y da ponta e da referência



4.1.5 Caso 4 - Variação da velocidade

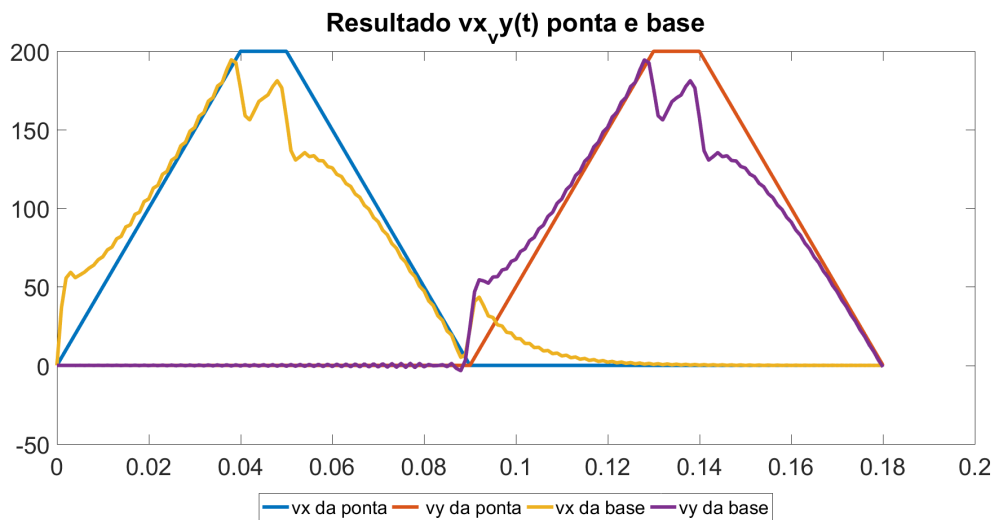
Variando a velocidade desejada, através do Gcode de entrada, analisamos os impactos de velocidades desajadas mais baixas e mais altas. As mais baixas (Caso A) tem seu maior impacto no tamanho dos vetores e no tempo de simulação, já que percorre o percurso em mais tempo, gerando uma malha com mais pontos, 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

Figura 4.17 – Caso 3 - Num de fun x Viabilidade



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



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

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

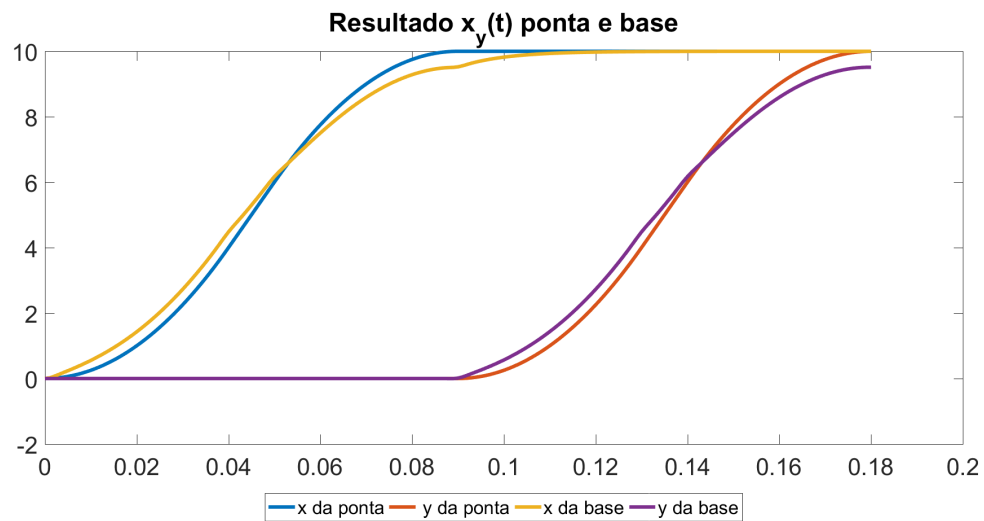
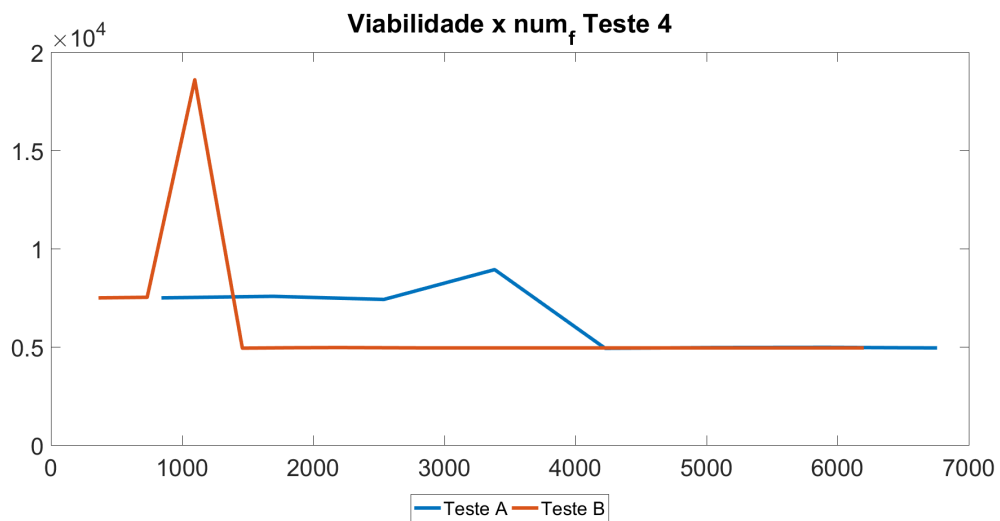


Figura 4.20 – Caso 4 - Num de fun x Viabilidade



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.

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

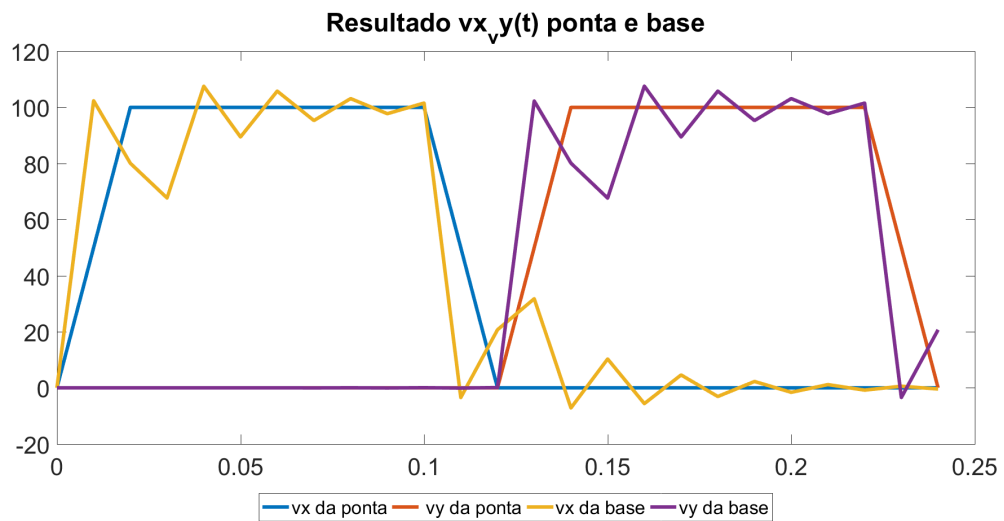
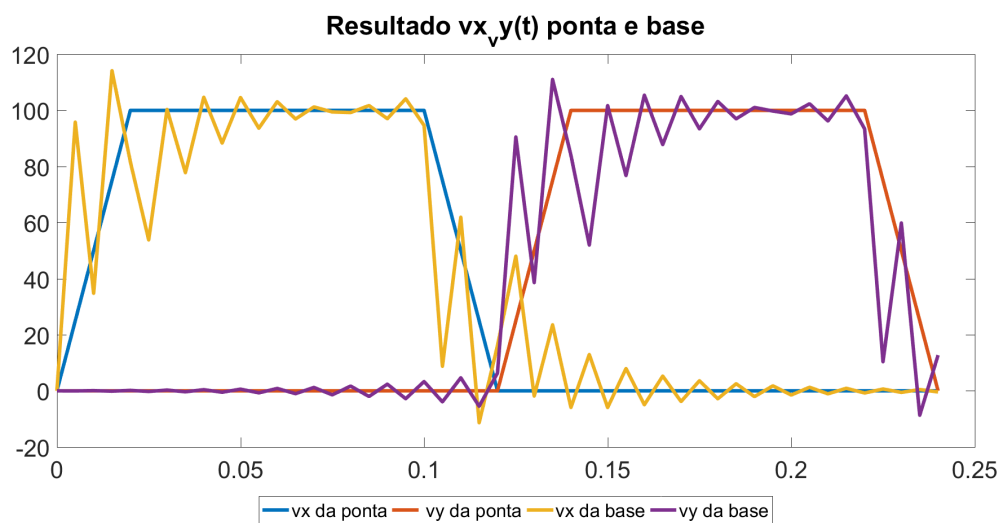


Figura 4.22 – Caso 5B - Comportamento no tempo das velocidades em x e y da ponta e da referência



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.

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

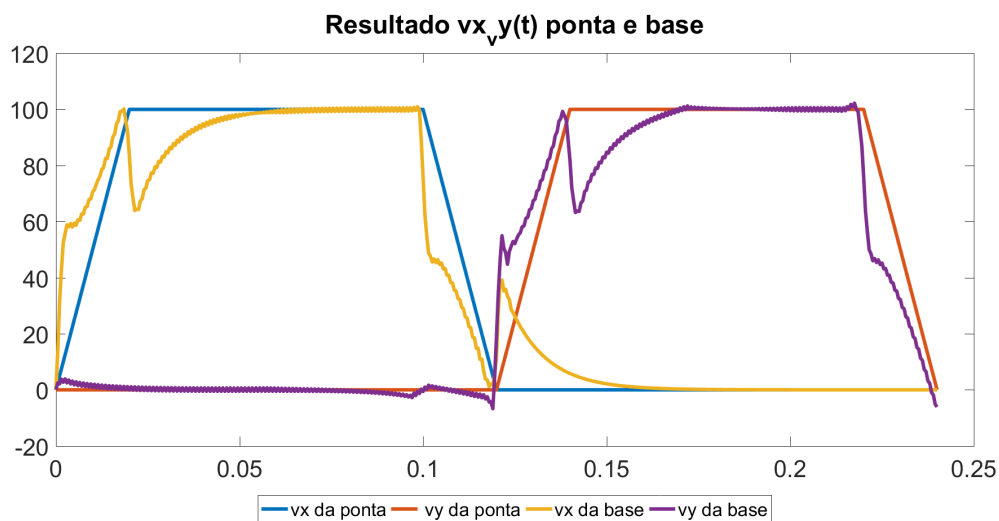
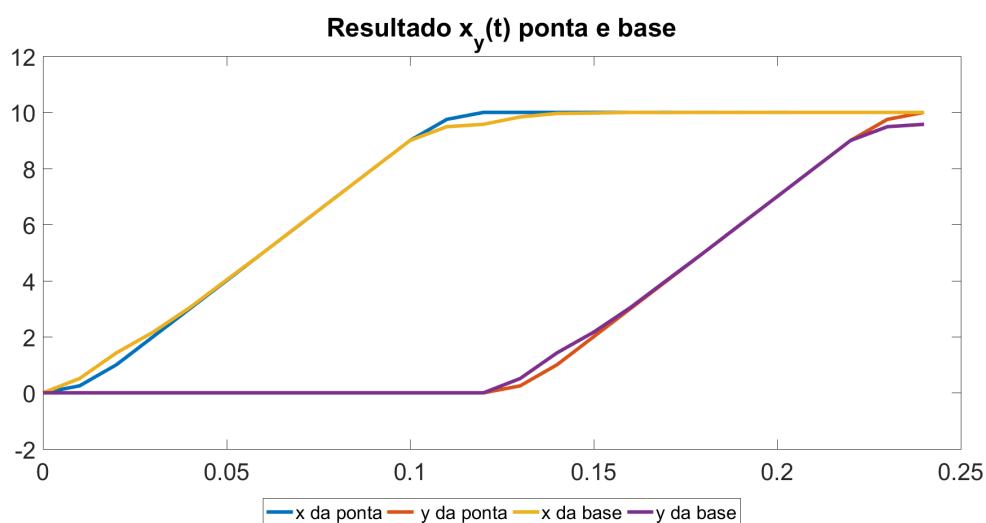


Figura 4.24 – Caso 5A - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência



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

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

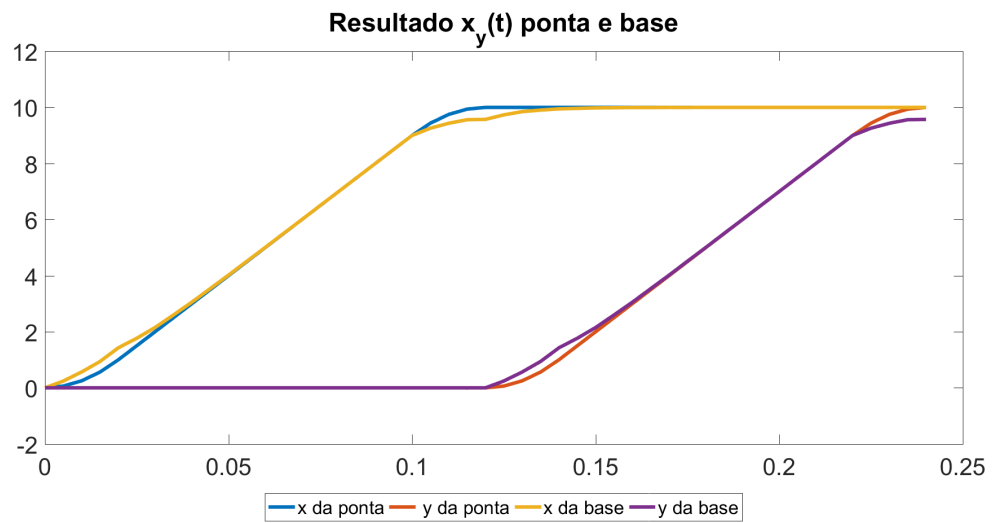
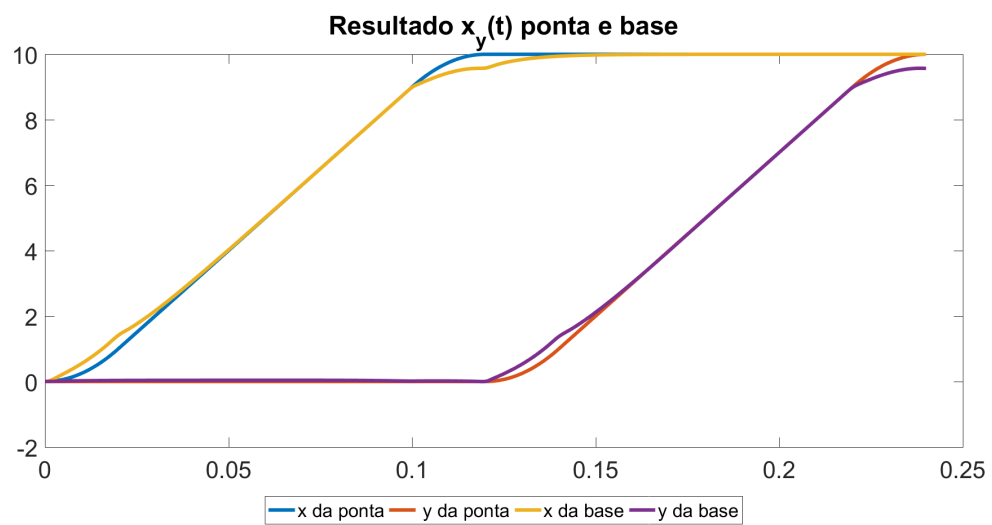
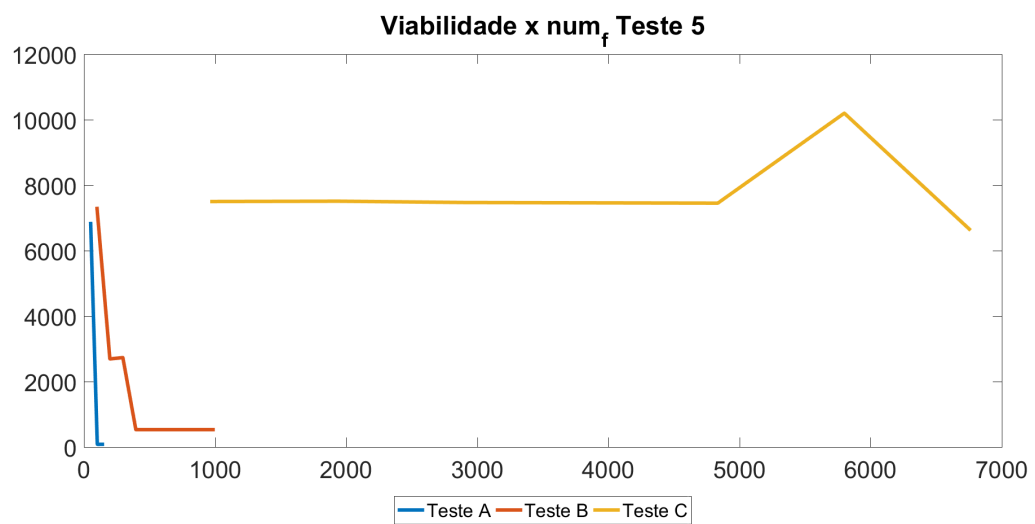


Figura 4.26 – Caso 5C - Comportamento no tempo dos deslocamentos em x e y da ponta e da referência



~~em frequência no ponto central da impressora. Considerando também que a utilização desse tipo de malha, teria pouco impacto computacional.~~

Figura 4.27 – Caso 5 - Num de fun x Viabilidade



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 a 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.; 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.
- 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 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.