

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

**OTIMIZAÇÃO ROBUSTA
MULTIOBJETIVO DO FRESAMENTO
HELICOIDAL DO AÇO INOXIDÁVEL
SUPER DUPLEX UNS S32760**

Thais Fernanda Sousa Pires

Itajubá

Mai de 2023

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

Thais Fernanda Sousa Pires

**OTIMIZAÇÃO ROBUSTA
MULTIOBJETIVO DO FRESAMENTO
HELICOIDAL DO AÇO INOXIDÁVEL
SUPER DUPLEX UNS S32760**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção como requisito à obtenção do título de *Mestre em Ciências em Engenharia de Produção*

Área de concentração: Engenharia de Produção

Orientadores:

Prof. João Roberto Ferreira, Dr. (Universidade Federal de Itajubá)

Prof. Robson Bruno Dutra Pereira, Dr. (Universidade Federal de São João del Rei)

Itajubá

Mai de 2023

DEDICATÓRIA

À Deus pela oportunidade de realização deste trabalho e a minha avó Marly (em memória) pelo apoio incondicional enquanto esteve presente nesta etapa.

AGRADECIMENTOS

Agradeço à Deus por me abençoar em mais uma realização da minha vida, por toda proteção e bençãos concedidas durante essa etapa. E agradeço à Nossa Senhora Aparecida por sempre me abençoar e guiar junto aos planos de Deus.

À minha mãe Marilene por estar sempre ao meu lado. Agradeço por todo amor, cuidado, por ser minha maior inspiração e por sempre acreditar em mim. Sem você nada disso seria possível.

Ao meu pai Ademir (em memória) por todo nosso tempo juntos, e por junto à minha mãe sempre ter acreditado e incentivado a minha vida nos estudos e serem meus maiores exemplos.

Ao meu irmão Felipe pelo apoio de sempre, amizade, compreensão e cuidado. Obrigada por estar ao meu lado em todas as minhas decisões.

À toda minha família pelo amor, orações e apoio.

À UFSJ pela disponibilização do laboratório para realização das medições de circularidade necessárias para o desenvolvimento do trabalho e por todo aprendizado adquirido ao longo da graduação.

Ao meu orientador João Roberto pela oportunidade e apoio para realização deste trabalho.

Ao meu coorientador Robson pelo apoio e parceria desde a graduação e disponibilidade e contribuição fundamentais ao longo do desenvolvimento deste trabalho.

Ao técnico e aluno José Veríssimo pela parceria e contribuição nas etapas do trabalho.

Aos amigos que fiz durante o mestrado, em especial ao Anderson Lino e Jéssica Tito que estiveram comigo desde o início dessa jornada.

À UNIFEI pela oportunidade e à FAPEMIG pelo auxílio financeiro.

RESUMO

Os aços inoxidáveis super duplex apresentam uma microestrutura mista constituída de fases de ferrita e austenita e são aplicados especialmente na indústria de óleo e gás, porém apresentam baixa usinabilidade. O processo de fresamento helicoidal consiste no movimento de rotação da fresa em torno do seu próprio eixo combinado com o avanço helicoidal, apresenta maior eficiência de processamento e menor custo comparado ao processo de furação convencional e permite à obtenção de furos de diferentes diâmetros. O processo apresenta ainda vantagens como altos níveis de qualidade de superfície, maior precisão dimensional e qualidade de forma, menores níveis de esforços de corte e maior suavidade na operação de usinagem. Este trabalho consiste na otimização robusta multiobjetivo do fresamento helicoidal do aço inoxidável super duplex UNS S32760. O planejamento experimental foi realizado com um arranjo composto central, considerando como fatores de controle o avanço axial por dente, o avanço tangencial por dente e a velocidade de corte. Foram avaliadas as respostas de rugosidade média, força axial de avanço e desvios de circularidade. Foi utilizado o projeto de parâmetro robusto, a metodologia de superfície de resposta para a condução dos experimentos, análise e modelagem das respostas de interesse, e realizada a otimização multiobjetivo através dos métodos de otimização de enxame de partículas multiobjetivo com distância de aglomeração (MOPSO-CD) e algoritmo genético de ordenação não-dominado (NSGA-II). Para o projeto de parâmetro robusto foram consideradas como variáveis de ruído o comprimento em balanço da fresa (l_{to}), a altura medida do furo usinado (l_b) e a vazão do fluido de corte (Q). A otimização evolutiva robusta multiobjetivo permite a avaliação dos níveis de fatores de processo. Diferentes atribuições para as funções objetivos foram analisadas com o intuito de obter soluções ótimas na Fronteira de Pareto para as respostas avaliadas.

Palavras-chave: Fresamento helicoidal, aço inoxidável super duplex, metodologia de superfície de resposta, projeto de parâmetro robusto e otimização multiobjetivo

ABSTRACT

Super duplex stainless steels have a mixed microstructure consisting of ferrite and austenite phases and are used especially in the oil and gas industry, but have low machinability. The helical milling process consists of rotating the cutter around its own axis combined with the helical feed, it presents greater processing efficiency and lower cost compared to the conventional drilling process and allows obtaining holes of different diameters. The process also has advantages such as high levels of surface quality, greater dimensional accuracy and shape quality, lower levels of cutting efforts and greater smoothness in the machining operation. This work consists of a robust multi-objective optimization of helical milling of the super duplex stainless steel UNS S32760. The experimental design was carried out with a central composite arrangement, considering as control factors the axial feed per tooth, the tangential feed per tooth and the cutting speed. Mean roughness responses, axial thrust force and circularity deviations were evaluated. The robust parameter design was used, the response surface methodology for conducting the experiments, analysis and modeling of the responses of interest, and multi-objective optimization was performed through the methods of optimization of multi-objective particle swarm with agglomeration distance (MOPSO-CD) and non-dominated sorting genetic algorithm (NSGA-II). For the robust parameter design, the overhang length of the cutter (l_{to}), the measured height of the machined hole (l_b) and the cutting fluid flow rate (Q) were considered as noise variables. Robust multi-objective evolutionary optimization allows the evaluation of process factor levels. Different attributions for the objective functions were analyzed in order to obtain optimal solutions on the Pareto Frontier for the evaluated responses.

Keywords: *Helical milling, super duplex stainless steel, response surface methodology, robust parameter design and multi-objective optimization*

LISTA DE FIGURAS

Figura 2.1. Características do processo de fresamento helicoidal

Figura 2.2. Cinemática do fresamento helicoidal

Figura 2.3. Parâmetros relevantes no processo de fresamento helicoidal

Figura 2.4. Forças de corte no: a) sistema de coordenadas da peça; b) sistema de coordenadas da ferramenta

Figura 2.5. Planejamento Composto Central (a) 2 Fatores e (b) 3 Fatores

Figura 2.6. Cálculo da distância de aglomeração

Figura 2.7. Procedimento NSGA II

Figura 4.1. *Setup* experimental

Figura 4.2. Fresa *Coromill Plura* código ISO/ANSI R215.H4-10050DAC0P 1620 da *Sandvik*®

Figura 4.3. (a) Dispositivo de fixação montado com o corpo de prova; (b) corpo de prova e (c) corpo de prova usinado do aço inoxidável super duplex UNS S32760

Figura 4.4. *Setup* de medição da rugosidade

Figura 4.5. Circularímetro *Taylor Hobson*®

Figura 5.1. Gráficos de efeitos principais das respostas da rugosidade R_a

Figura 5.2. Gráficos de interações entre as variáveis de controle e ruído

Figura 5.3. Gráficos de Contorno para média (E) e Desvio-padrão (S) de R_a

Figura 5.4. Gráfico de efeitos principais das respostas da força axial F_a

Figura 5.5. Gráficos de interações entre as variáveis de controle e ruído

Figura 5.6. Gráficos de Contorno para média (E) e Desvio-padrão (S) de F_a

Figura 5.7. Gráfico de efeitos principais das respostas da circularidade total Ron_t

Figura 5.8. Gráficos de interações entre as variáveis de controle e ruído

Figura 5.9. Gráficos de Contorno para média (E) e Desvio-padrão (S) de Ron_t

Figura 5.10. Fronteira de Pareto para a otimização de quatro objetivos

Figura 5.11. Fronteira de Pareto para a otimização de quatro objetivos

LISTA DE TABELAS

Tabela 1.1. Análise bibliométrica

Tabela 4.1. Composição química (% peso) do aço inoxidável super duplex UNS S32760

Tabela 4.2 Variáveis de controle e ruído adotados e seus níveis

Tabela 5.1 Planejamento experimental com respostas e resultados

Tabela 5.2. Teste t para os coeficientes do modelo de rugosidade

Tabela 5.3. Teste t para os coeficientes do modelo de força

Tabela 5.4. Teste t para os coeficientes de Ron_t

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

CAD – *Computer Aided Design* (Projeto Auxiliado por Computador)

CAM – *Computer Aided Manufacturing* (Manufatura Auxiliada por Computador)

CCD – *Central Composite Design* (Planejamento Composto Central)

CNC – *Computer Numeric Control* (Comando Numérico Computadorizado)

CFRP – *Carbon Fiber Reinforced Plastic* (Plástico Reforçado por Fibra de Carbono)

DOE – *Design of Experiments* (Planejamento de Experimentos)

EDM – *Electrical Discharge Machining* (Usinagem Eletroquímica)

DEMEP – Departamento de Engenharia Mecânica e Produção

LMAUT – Laboratório de Manufatura e Automação

MOPSO – *Multi-objective Particle Swarm Optimization* (Otimização de enxame de partículas multiobjetivo)

MOPSO-CD – *Multi-objective Particle Swarm Optimization with Crowding Distance* (Otimização de enxame de partículas multiobjetivo com distância de aglomeração)

MQL – Mínima quantidade de fluido

MRR – *Material Removal Rate* (Taxa de Remoção de Material)

MSE – *Mean Square Error* (Erro Quadrático Médio)

NSGA-II – *Non-dominated Sorting Genetic Algorithm* (Algoritmo genético de ordenação não-dominado)

PRE – *Pitting Resistance Equivalent* (Resistência Equivalente à Corrosão por Pites)

RPD – *Robust Parameter Design* (Projeto de Parâmetro Robusto)

RSM – *Response Surface Methodology* (Metodologia de Superfície de Resposta)

UFSJ – Universidade Federal de São João del Rei

UNIFEI – Universidade Federal de Itajubá

LISTA DE SÍMBOLOS

Símbolos relacionados ao processo de fresamento helicoidal

a_e - profundidade radial

a_ρ^* - máxima profundidade de corte axial

D_b - diâmetro do furo

D_h - diâmetro da hélice

f_{za} - avanço axial por dente

f_{zt} - avanço tangencial por dente

n - rotação do *spindle*

v_f - velocidade de avanço helicoidal da hélice

v_{ft} - velocidade tangencial

v_{fha} - velocidade de avanço axial da hélice

v_{fht} - velocidade de avanço tangencial da hélice

z - número de dentes da fresa

Símbolos da RSM e CCD

β_0 - uma constante

ε - erro experimental

x_i - variáveis de controle

y - resposta de interesse

n_a - número de pontos axiais

n_c - número de pontos centrais

n_f - número de pontos fatoriais

Símbolos do RPD e MSE

β_0 - uma constante

B - matriz com termos quadráticos

ε - erro experimental

Δ - matriz diagonal de autovalores

k - número de variáveis de controle

x_i - variáveis de controle

y - resposta de interesse

\hat{y} - valor predito pelo modelo para a resposta y

\mathbf{y} - vetor de respostas

SUMÁRIO

1. INTRODUÇÃO	1
1.1 Contextualização da pesquisa.....	1
1.2 Objetivos.....	4
1.3 Justificativa.....	5
1.4 Estrutura da pesquisa.....	7
2. FUNDAMENTAÇÃO TEÓRICA	8
2.1 Fresamento helicoidal.....	8
2.1.1 Cinemática do fresamento helicoidal.....	9
2.1.2 Vantagens do processo de Fresamento Helicoidal.....	12
2.1.3 Aplicações.....	13
2.2 Rugosidade.....	14
2.2.1 Forças de corte.....	15
2.2.2 Circularidade.....	16
2.2.3 Produtividade.....	17
2.3 Planejamento de Experimentos.....	17
2.3.1 Metodologia de Superfície de Resposta e Planejamento Composto Central.....	18
2.3.2 Projeto de Parâmetro Robusto e Erro Quadrático Médio.....	20
2.4 Otimização MultiObjetivo.....	23
2.4.1 Otimização de enxame de partículas multiobjetivo.....	23
2.4.2 Algoritmo genético de ordenação não-dominado.....	26
3. MÉTODO DE PESQUISA	27
4. PROCEDIMENTO EXPERIMENTAL	27
4.1 Equipamentos e materiais.....	27
4.2 Planejamento experimental e métodos de otimização.....	31
5. RESULTADOS E DISCUSSÕES	33
5.1 Resultados experimentais.....	33
5.2 Rugosidade média R_a	34
5.3 Força axial de avanço.....	39

5.4 Desvios de Circularidade.....	44
5.5 Otimização evolutiva robusta multiobjetivo através do MOPSO-CD e NSGA-II.....	49
6. CONCLUSÕES.....	52
6.1 Sugestões para trabalhos futuros.....	53
REFERÊNCIAS.....	54
APÊNDICES.....	62

1. INTRODUÇÃO

1.1 Contextualização da pesquisa

O desenvolvimento dos aços inoxidáveis representa um avanço significativo na fabricação de componentes resistentes à corrosão (DAMIN; GENEROSO; NETO, 2017). Os aços inoxidáveis possuem como composição base ferro, carbono, cromo e níquel, o que permite que se forme uma camada passiva de óxidos na superfície, quando em contato com o oxigênio, impedindo o contato do metal base com a atmosfera agressiva (CARBÓ, 2008). Com o aumento da demanda das indústrias por materiais de alta resistência à corrosão e alta resistência mecânica em relação ao custo, a aplicação destes materiais é uma opção, especialmente em ambientes severos, como indústria petroquímica, química, aeroespacial e de energia, plataformas de óleo e gás e plantas de dessalinização (OLIVEIRA, 2013; SMIDERLE *et al.*, 2014; TAVARES *et al.*, 2018).

A combinação de elementos como cromo, molibdênio e nitrogênio confere aos aços inoxidáveis duplex boa estabilidade química e possibilidade de aplicação em substituição ao bronze e outras ligas. Os aços inoxidáveis super duplex apresentam propriedades mais nobres, maior resistência mecânica e valor de resistência à corrosão por pites superior em relação aos inoxidáveis duplex (GAMARRA, 2018). Como forma de classificação das ligas resistentes à corrosão, para que o aço inoxidável seja considerado duplex, o material deve apresentar resistência equivalente à corrosão por pites, geralmente conhecida pela sigla PRE (*Pitting Resistance Equivalent*) maior que 20, enquanto os aços inoxidáveis super duplex devem apresentar PRE superior a 40. O PRE é obtido conforme a composição química do material, considerando a variação percentual dos elementos (NILSSON, 1992), conforme apresentado na Equação 1.1.

$$PRE = \%Cr + 3,3(\%Mo + 0,5(\%W)) + 16(\%N) \quad (1.1)$$

Os aços inoxidáveis super duplex apresentam uma microestrutura mista constituída de fases de ferrita e austenita. Quando os aços inoxidáveis duplex apresentam o equilíbrio de fase

ideal, com proporções aproximadamente iguais de fases de ferrita e austenita, eles apresentam maior resistência à corrosão e maior resistência mecânica. Com o aumento de investimentos em tecnologia em diversos segmentos na indústria, a aplicação destes materiais é interessante para diversos setores, como ambientes altamente corrosivos, com aplicações crescentes na indústria de óleo e gás (KROLCZYK *et al.*, 2013; POLICENA *et al.*, 2018; TAVARES *et al.*, 2018).

Durante a operação de usinagem dos aços inoxidáveis podem ocorrer formações instáveis de cavacos, devido à baixa condutividade térmica e ao endurecimento por deformação. Modificações mecânicas são presentes com o aumento da velocidade de corte, uma vez que as ligas de aços inoxidáveis são caracterizadas pelas altas taxas de encruamento (KORKUT *et al.*, 2004; NOMANI *et al.*, 2017). Ao longo da operação de corte, a ferramenta alterna de um regime de corte macio para duro, nas fases de austenita e ferrita, respectivamente, podendo resultar em vibrações no sistema e diminuição do tempo de vida da ferramenta (KOYEE *et al.*, 2014).

A furação convencional apresenta algumas dificuldades como altos esforços na direção axial, altos níveis de força de avanço, quebra de material na saída do furo, formação de rebarbas e falha catastrófica da broca (EGUTI E TRABASSO, 2014; JI *et al.*, 2015). Entre os desafios técnicos relacionados à furação em comparação com outros processos de usinagem, na furação convencional a velocidade de corte no centro da broca é nula. Em virtude disto, a remoção de material nesta região é realizada por um processo semelhante à extrusão ao invés de cisalhamento, que pode resultar em alta força de empuxo e conseqüente limitação da produtividade do processo, além de quebra de material na saída do furo (IYER *et al.*, 2007). Devido ao espaço confinado nos canais helicoidais da broca e do atrito, também ocorrem dificuldades na evacuação dos cavacos e na dissipação do calor, ocasionando altos níveis de desgaste da ferramenta e comprometimento na qualidade do furo, especialmente para furos com elevados diâmetros e comprimento, além de baixa capacidade de processamento na obtenção de furos acabados (LI *et al.*, 2010; OLVERA *et al.*, 2012).

Apesar de ser uma das últimas etapas na linha de produção, os processos de usinagem demonstram alta demanda por confiabilidade e robustez, devido ao alto valor econômico agregado resultante também do processamento das operações anteriores, assegurando a necessidade de aperfeiçoamento das técnicas utilizadas nestes processos (TONSHOFF E KONIG, 1994. Segundo Wang *et al.* (2015) as operações para obtenção de furo correspondem a 50% do processamento total no escopo da usinagem.

O processo de fresamento helicoidal possibilita a obtenção de furos de diferentes diâmetros ajustando o diâmetro da trajetória helicoidal sem necessidade de troca de ferramenta durante o processo. Comparado à furação convencional, o fresamento helicoidal apresenta vantagens como alta precisão, menores níveis de força na direção axial e corte intermitente na aresta de corte periférica, com a remoção de material sendo realizada pelas arestas de corte frontais e periféricas da fresa de forma contínua (CHEN *et al.*, 2020). Em relação à furação convencional, as forças de corte e temperatura também são relativamente menores no fresamento helicoidal e alta qualidade da superfície e precisão dimensional podem ser geradas neste processo em uma única operação (WANG *et al.*, 2019). Devido a trajetória helicoidal realizada pela fresa há a possibilidade de realização de operações de desbaste e acabamento sem necessidade de troca de ferramenta, maior precisão geométrica, baixos níveis dos esforços de corte, facilidade de evacuação de cavaco, baixa formação de rebarba e podem ser avaliadas diferentes estratégias de lubri-refrigeração neste processo. Deste modo, a aplicação do processo de fresamento helicoidal se apresenta como uma opção sustentável na obtenção de furos e associada à aplicação de métodos estatísticos, matemáticos e computacionais no estudo do processo pode-se atingir a obtenção de furos com mais qualidade e maior produtividade, proporcionando benefícios econômicos, sociais e ambientais neste setor industrial (DENKENA *et al.*, 2008; IYER *et al.*, 2007; BRINSKMEIR *et al.*, 2008; PEREIRA *et al.*, 2017).

A modelagem de processos realizada por experimentos planejados tem como objetivo identificar uma aproximação para uma determinada resposta ou variável independente em relação às variáveis controláveis deste processo. O método de projeto experimental é considerado uma abordagem efetiva para otimizar os vários parâmetros de usinagem (KURT e BAGCI, 2007). Considerando o planejamento e análise de processos, o Planejamento de Experimentos (*Design of Experiments – DOE*) pode contribuir de maneira efetiva na melhoria das atividades desenvolvidas na maioria das organizações, permitindo a geração de dados apropriados para uma análise estatística eficaz (MONTGOMERY, 2013). Uma das metodologias de projeto de experimentos usualmente utilizadas consiste na Metodologia de Superfície de Resposta (*Response Surface Methodology – RSM*) que abrange um conjunto de técnicas de construção e modelos empíricos (BOX e DRAPER, 2007). Para alcançar o objetivo do método, através da modelagem da resposta em função das variáveis de processo em uma determinada região de interesse e encontrar níveis ótimos operacionais, experimentos planejados devem ser conduzidos. O Projeto de Parâmetro Robusto (*Robust Parameter Design – RPD*) busca o delineamento de sistemas insensíveis a fatores ambientais que podem

influenciar seu desempenho e aliado à metodologia de superfície de resposta possibilita economia na experimentação do estudo da interação entre as variáveis de ruído e processo, e também na obtenção dos modelos de média e variância para a resposta de interesse a partir da propagação do erro em relação às variáveis de ruído (MYERS *et al.*, 1992; PEREIRA, 2017). Segundo Taguchi (1986), o RPD permite determinar a função perda do produto com a proposta da sua otimização empregando técnicas estatísticas, permitindo a partir desta análise a identificação dos parâmetros ótimos de projeto que minimizam ou mesmo eliminam as influências dos fatores ruído no desempenho do produto ou processo. Deste modo permite integrar os estudos e verificações das condições de usinagem, como tempos e custos de processamento, com a finalidade de modelar as variáveis de interesse, como também verificar a influência dos fatores de processo nas especificações das respostas analisadas. A otimização multiobjetivo do processo permite selecionar os níveis ideais de condições de corte para melhor alcance da estabilidade do processo.

O presente trabalho tem por finalidade a análise, modelagem e otimização robusta multiobjetivo do processo de fresamento helicoidal do aço inoxidável super duplex UNS S32760. No processo de fresamento helicoidal, as variáveis de controle consideradas estão relacionadas à cinemática do processo. Com a aplicação de métodos estatísticos/matemáticos no processo de fresamento helicoidal espera-se obter superfícies acabadas com alta qualidade e precisão.

1.2 Objetivos

O objetivo principal deste trabalho é a otimização robusta multiobjetivo do processo de fresamento helicoidal do aço inoxidável super duplex UNS S32760. Para alcançar este objetivo foram utilizadas as metodologias de planejamento de experimentos, metodologia de superfície de resposta, projeto de parâmetro robusto, erro quadrático médio e otimização multiobjetivo. Os objetivos específicos são listados abaixo:

- Avaliação da influência das variáveis de controle, avanço axial, avanço tangencial e velocidade de corte na rugosidade média, na força axial de avanço e na circularidade dos furos;

- Avaliação da influência do comprimento em balanço da fresa, altura medida do furo e vazão do fluido de corte nas respostas de interesse;
- Análise e otimização das condições de usinagem do processo de fresamento por interpolação helicoidal do aço inoxidável super duplex UNS S32760;
- Obtenção dos modelos de resposta para as respostas analisadas, relacionando as variáveis de controle e ruído a partir da metodologia de superfície de resposta;
- Estudo da robustez do processo a partir da avaliação das interações das variáveis de processo e ruído;
- Análise e modelagem das superfícies geradas para as respostas de interesse no processo de fresamento helicoidal do aço inoxidável super duplex UNS S32760;
- Obtenção dos modelos de média, variância e erro quadrático médio para cada uma das respostas avaliadas;
- Realização da otimização multiobjetivo evolutiva dos modelos obtidos utilizando os algoritmos evolutivos MOPSO e NSGA-II.

1.3 Justificativa

A principal justificativa para a realização deste estudo é a escassez de trabalhos científicos relacionados ao processo de fresamento helicoidal do aço inoxidável super duplex na literatura atual. Com a finalidade de comprovar a relevância do trabalho, assim como sua contribuição científica para o tema de pesquisa foi realizada uma análise bibliométrica considerando o processo de fresamento helicoidal e o aço inoxidável super duplex. A análise foi feita em fevereiro de 2022 e as bases de dados utilizadas para esta busca foram *Scopus* e *Web of Science*. Inicialmente, foi elaborada a busca pelos termos de forma individual no título, resumo e palavras chave no intuito de verificar a quantidade de artigos relacionados ao tema na literatura. Posteriormente, foi realizada a busca nas mesmas bases de dados combinando ambos os termos nos títulos, resumos e palavras chave. Os resultados encontrados são apresentados na Tabela 1.1.

Tabela 1.1. Análise bibliométrica

Base de dados	<i>Helical milling</i>	<i>Super Duplex Stainless Steel</i>	<i>Helical milling + Super Duplex Stainless Steel</i>
<i>Scopus</i>	216	575	0
<i>Web of Science</i>	81	137	0

Os resultados encontrados na análise bibliométrica certificam que há espaço para investigação do processo de fresamento helicoidal e sua utilização nos aços inoxidáveis super duplex.

Através da análise bibliométrica foi possível verificar a carência de trabalhos relacionados ao tema, comprovando a necessidade do estudo. Uma maior discussão e aprofundamento do tema, no que se refere a utilização do fresamento no setor de petróleo e gás, apresenta a possibilidade de inserção de novos processos na indústria na busca de ganhos em produtividade e garantia de sustentabilidade nos processos. A possibilidade de obtenção de furos com maior qualidade e precisão nos aços inoxidáveis super duplex podem garantir minimização dos custos e maior vantagem competitiva no cenário industrial, relacionado à tecnologia e inovação.

A associação do estudo dos efeitos das variáveis de processo nas respostas de interesse é fundamental para a confirmação da escolha correta dos níveis dos parâmetros. Através da análise e modelagem da previsão dos níveis de qualidade e esforços de corte no processo, é possível a antecipação das características do processo e um melhor atendimento das especificações de qualidade do produto usinado.

A utilização do planejamento de experimentos no desenvolvimento de pesquisas é uma estratégia útil para redução do número de experimentos e conseqüente redução dos custos com experimentação. O projeto de parâmetro robusto é empregado para obter modelos robustos de rugosidade, força e circularidade em relação as variáveis de ruído. A aplicação de métodos de otimização é realizada com a finalidade de solucionar as respostas de interesse simultaneamente. Assim, as vantagens obtidas na escolha do processo e a investigação dos parâmetros permitem a possibilidade de desenvolvimento das pesquisas na área e aplicação no contexto industrial.

1.4 Estrutura da pesquisa

Este trabalho está estruturado em 5 capítulos. O capítulo introdutório descreve a composição e aplicação dos aços inoxidáveis e necessidade de busca de alternativas na usinagem destes materiais, devido à baixa usinabilidade do material e limitações do processo de furação convencional. O processo de fresamento helicoidal é proposto, associado ao uso de métodos estatísticos com o intuito de redução da variabilidade e otimização deste processo. Assim, os métodos de modelagem e otimização são apresentados, os objetivos e justificativa são abordados.

O segundo capítulo descreve a fundamentação teórica que integra o desenvolvimento estabelecido durante todo o trabalho. O fresamento helicoidal, sua cinemática, vantagens, e aplicações são apresentados. São listadas as respostas de interesse do presente trabalho e os métodos de planejamento e otimização.

O terceiro capítulo aborda o método de pesquisa utilizado, enquanto o quarto apresenta a estruturação do desenvolvimento do trabalho. Desde modo, o quarto capítulo descreve a metodologia utilizada no desenvolvimento do procedimento experimental, considerando os materiais, ferramentas e equipamentos utilizados para a usinagem e medição dos corpos de prova.

O quinto capítulo apresenta os resultados. Inicialmente as características de interesse são discutidas e modeladas de forma individual, com a finalidade de confrontar os resultados obtidos com a literatura. São apresentados e discutidos os gráficos de efeitos principais interações. Posteriormente é realizada a otimização multiobjetivo com a descrição dos resultados observados e análise do espaçamento da fronteira.

Por fim, são apresentadas as conclusões obtidas com o desenvolvimento da pesquisa.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Fresamento helicoidal

No processo de fresamento helicoidal a ferramenta é fixada no eixo árvore e realiza movimento de avanço helicoidal, bem como de rotação em torno de seu próprio eixo. Devido a trajetória helicoidal realizada pela fresa, o processo de fresamento helicoidal viabiliza a obtenção de furos de diâmetros diferentes sem alterar a ferramenta, possibilitando a realização de operações de desbaste e acabamento sem necessidade de troca da fresa ao longo da operação e obtenção de furos acabados em uma única operação (OLVERA *et al.*, 2012).

O processo de fresamento helicoidal proporciona alta qualidade da superfície e furos com maior precisão geométrica com apenas uma execução, além de custos de processamento mais baixos e maior eficiência de processamento comparado à furação convencional (SUN *et al.*, 2020). Como outros benefícios do processo de fresamento helicoidal, podem-se destacar: forças de usinagem mais baixas, boa evacuação de cavacos e pouca rebarba, boas condições de lubri-refrigeração e temperatura mais baixa que a furação convencional (LI, 2014). No geral, podem ocorrer altos esforços de corte e vibrações, durante a furação convencional, o que pode resultar em maior absorção de calor pela aresta de corte e aumento nos mecanismos de desgaste da ferramenta como difusão, abrasão e aderência, além de redução do tempo de vida da broca e comprometimento no acabamento superficial da peça (KORKUT, 2004; POLICENA, 2018). A Figura 1 ilustra as características e apresenta algumas vantagens do processo.

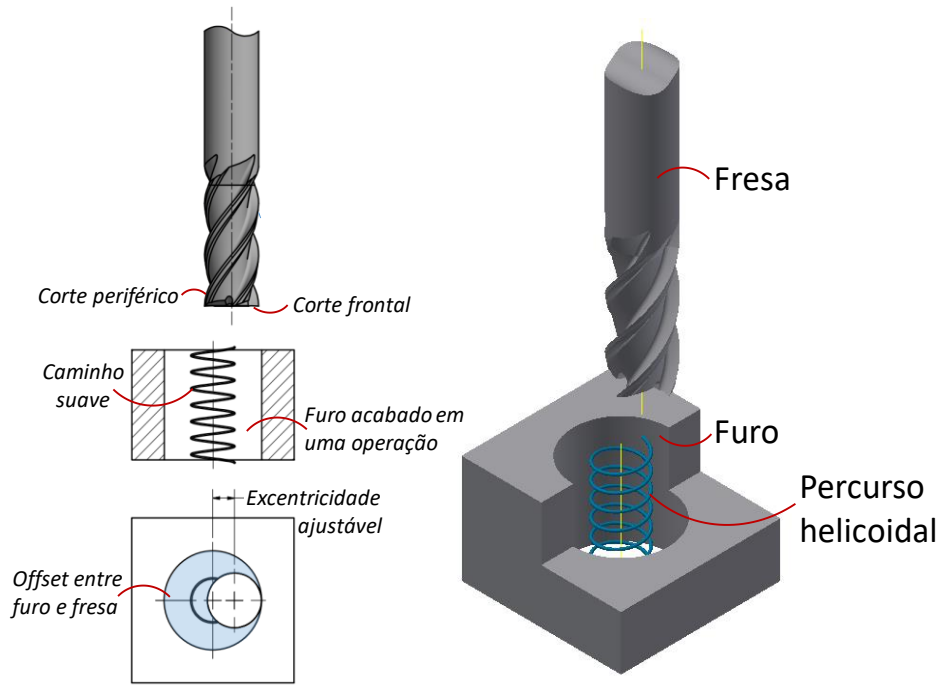


Figura 2.1. Características do processo de fresamento helicoidal (PEREIRA, 2017)

O desempenho de corte é afetado ainda por outros fatores, incluindo parâmetros de corte e geometria da ferramenta (ZHU, 2020). No processo de fresamento helicoidal, menores níveis de força na direção axial também são desenvolvidos devido a trajetória helicoidal e a própria geometria da ferramenta. Existe maior facilidade na evacuação de cavacos e facilidade de lubri-refrigeração por mínima quantidade de fluido (MQL) e refrigeração por ar comprimido são viabilizados na operação de determinados materiais, devido a folga lateral entre a fresa e o furo de usinagem, proporcionada pela diferença entre os diâmetros do furo e da hélice. A trajetória realizada pela fresa possibilita ainda a redução de inventário de ferramentas, proporcionando a realização de furos de diâmetros diferentes com a mesma fresa, melhor precisão geométrica, microgeométrica e dimensional de acordo com os estudos de Denkena *et al.* (2008), Iyer *et al.* (2007) e Brinskmeir *et al.* (2008).

2.1.1 Cinemática do fresamento helicoidal

A cinemática do processo de fresamento helicoidal é realizada através do percurso gerado pela alimentação do eixo z combinado com a interpolação circular dos eixos x e y , simultaneamente, em um centro de usinagem CNC (WANG *et al.*, 2019). A cinemática do

processo viabiliza a obtenção de furos de diferentes diâmetros utilizando apenas uma ferramenta (DENKENA *et al.*, 2011).

De acordo com Denkena (2008) a cinemática do fresamento helicoidal pode ser compreendida através da decomposição da velocidade de avanço v_f na velocidade de avanço axial da hélice v_{fha} em [mm/mim] e na velocidade de avanço tangencial da hélice v_{fht} em [mm/mim] nas direções axial e tangencial, respectivamente, combinando corte periférico e frontal, conforme apresentado na Equação 2.1.

A velocidade de avanço axial da hélice v_{fha} pode ser calculada pelo produto do avanço axial por dente f_{za} , do número de dentes da fresa z e do número de rotações n por minuto do *spindle* em [RPM], representada na Equação 2.2. A velocidade de avanço tangencial da hélice v_{fht} , no centro da fresa, descrita na Equação 2.3 pode ser calculada considerando o diâmetro da hélice D_h em [mm], o diâmetro do furo D_b em [mm] e a velocidade tangencial v_{ft} , na periferia do furo, que por sua vez é expressa em função do avanço tangencial por dente f_{zt} , número de dentes da fresa z e rotação n do *spindle*, compreendida na Equação 2.4, com o avanço tangencial f_{zt} dado em [mm/dente]. Ainda de acordo com Denkena (2008), o diâmetro do furo é determinado pelo diâmetro da ferramenta em combinação com o diâmetro do caminho helicoidal (hélice), o que leva a uma alta flexibilidade no que diz respeito ao diâmetro do furo.

Os principais parâmetros no fresamento helicoidal incluem o avanço axial por dente (f_{za}) em [mm/dente], o avanço tangencial por dente (f_{zt}) em [mm/dente], o diâmetro do furo (D_b) em [mm], o diâmetro da hélice (D_h) em [mm], a velocidade de rotação n e o número de dentes da fresa z (SUN *et al.*, 2020). A cinemática do processo de fresamento helicoidal é ilustrada na Figura 2.2 e as definições do desenvolvimento das equações abaixo estão representadas na Figura 2.3.

$$v_f = \sqrt{v_{fha}^2 + v_{fht}^2} \quad (2.1)$$

$$v_{fha} = f_{za} \cdot z \cdot n \quad (2.2)$$

$$v_{fht} = v_{ft} \frac{D_h}{D_b} \quad (2.3)$$

$$v_{ft} = f_{zt} \cdot z \cdot n \quad (2.4)$$

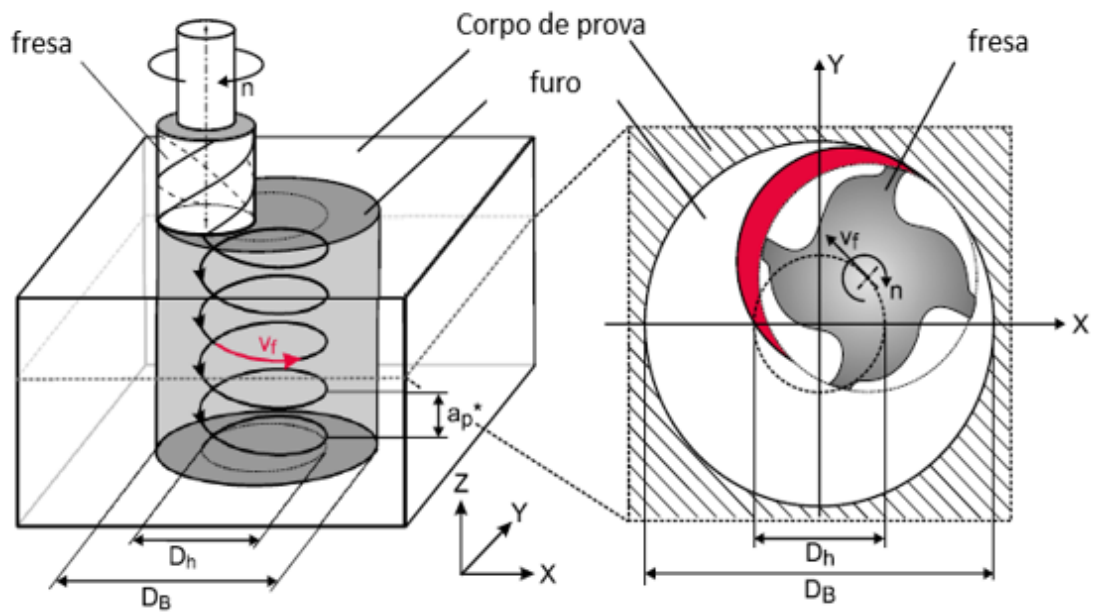


Figura 2.2. Cinemática do fresamento helicoidal (Adaptado de DENKENA *et al.*, 2011)

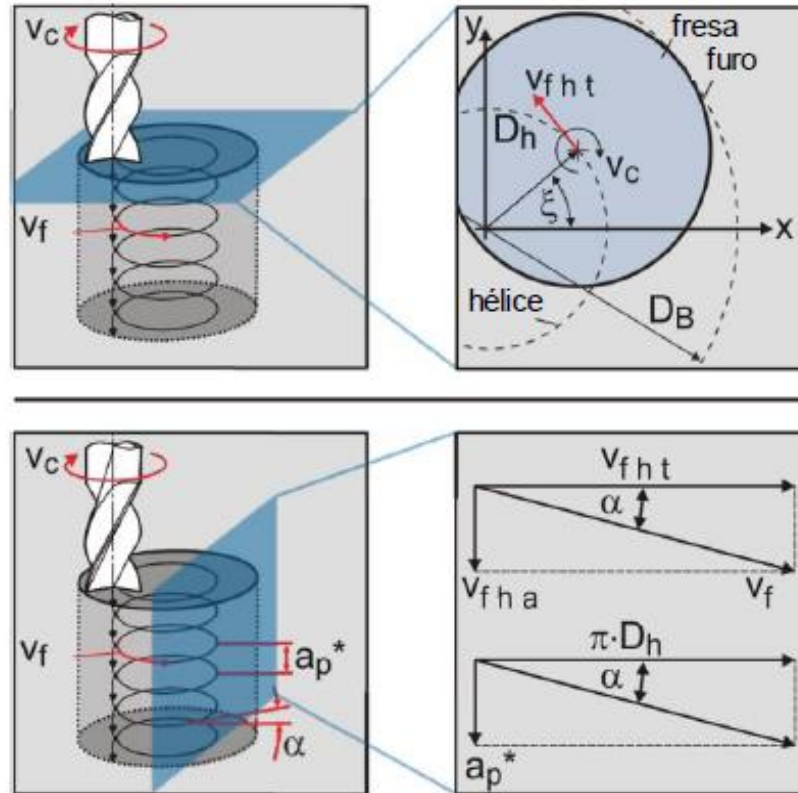


Figura 2.3. Parâmetros relevantes no processo de fresamento helicoidal (Adaptado de DENKENA *et al.*, 2008)

2.1.2 Vantagens do processo de Fresamento Helicoidal

De acordo com Qin *et al.* (2010), como uma das metodologias de usinagem de furos, o fresamento helicoidal possui uma série de vantagens, como suavidade no processo de corte, baixa força de corte e alta precisão com apenas um passe em comparação com a furação convencional. Outra vantagem obtida no processo de fresamento helicoidal são as forças axiais mais baixas quando comparadas a furação convencional, o que permite alcançar tolerâncias mais apertadas (WANG *et al.*, 2012).

O processo de fresamento helicoidal viabiliza ainda a furação do aço em seu estado endurecido. Segundo Iyer *et al.* (2007) trabalhar com o objeto de partida (*stock*) já endurecido resulta em benefícios de redução do tempo de operação, eliminação de posteriores operações de processamento e logística. Proporciona ainda um ganho do produto em termos de qualidade com eliminação dos erros associados que podem ocorrer nos processos de tratamento térmico e de processamento das etapas de operação e altos níveis de processos de *setup*, quando comparado a técnica de desbastar o material em seu estado macio, seguido de tratamento térmico, acabamento por retificação e EDM, entre outros.

O fresamento helicoidal pode ser executado em centros de usinagem CNC (Comando Numérico Computadorizado), aliando as vantagens do fresamento, em detrimento as dificuldades de furação, na realização de obtenção de furos em materiais de baixa usinabilidade. Para viabilizar o uso de novas tecnologias de fresamento, desenvolvedores de *softwares* de Manufatura Auxiliada por Computador (*Computer Aided Manufacturing – CAM*) tem desenvolvido a criação de programação de trajetórias especiais (ALTAN, 2001). O desenvolvimento de máquinas-ferramenta com tecnologia CNC, onde o código pode ser complementado por um ciclo de fresamento helicoidal específico de cada comando numérico ou via CAM, que viabiliza o planejamento, execução e visualização do escopo do projeto da peça bruta e a operação de usinagem até ao acabamento final. Segundo Souza (2004) a tecnologia CAD/CAM deve ser considerada um dos pontos mais importantes do desenvolvimento tecnológico em usinagem com trajetórias complexas e em altas velocidades.

2.1.3 Aplicações

O estudo da investigação do processo de fresamento helicoidal para a furação do aço ferramenta AISI D2 foi realizado por Iyer *et al.* (2007). O processo facilitou a furação deste material em seu estado endurecido com uma ordem de magnitude de aumento na vida útil da ferramenta sob condições de desgaste progressivo da ferramenta. Através da aplicação do método foi identificado que o fresamento helicoidal possui capacidade de usinar furos de qualidade, com um acabamento superficial de tal forma que a necessidade de alargamento é eliminada. A pesquisa identificou o processo como uma tecnologia viável com a vantagem adicional de realizar corte a seco, o que gera benefícios ambientais e de redução de custos significativos.

O mecanismo de formação de cavaco e a usinabilidade dos aços duplex UNS S32205 e super duplex UNS S32750 foram investigados por Nomani *et al.* (2015). Além dos ensaios de vida em furação convencional, os autores conduziram ensaios de torneamento ortogonal utilizando dispositivo de parada rápida para preservar a raiz do cavaco e assim verificar a formação do cavaco, com o intuito de explicar a relação dos resultados aos estudos de furação. Os estudos mostraram que apenas a fase de ferrita aderiu à ferramenta, enquanto a fase de austenita fluiu para fora da zona de formação da aresta postiça de corte. Os dois aços inoxidáveis mostraram mecanismo de formação de cavaco semelhante.

Nomani *et al.* (2013) avaliou a furação convencional dos aços inoxidáveis duplex UNS S32205 e super duplex UNS S32750 utilizando brocas de metal duro com cobertura de TiAlN e +TiN, para obtenção de furos de 12 mm de diâmetro e 30 mm de profundidade. O fim de vida da broca usinando o aço super duplex foi atingido após 26 furos, considerando o critério de fim de vida de 0,15 mm de flanco médio, com mecanismo de desgaste por adesão predominante. Também foram observadas avarias na periferia dos canais helicoidais das brocas. Na avaliação dos resultados, altos níveis de rugosidade foram observados com rugosidade R_a de 2,06 μm para o 25º furo no aço inox super duplex.

Com o intuito de eliminar os efeitos adversos da movimentação excêntrica, força de corte instável ao longo do tempo e outras variáveis durante o processo de usinagem, Fang *et al.* (2015) estudou os efeitos do fresamento helicoidal em ligas de titânio e compósitos de plástico reforçados por fibra de carbono (CFRP), materiais este muito usados na indústria aeronáutica. Os resultados mostraram que o fresamento helicoidal é adequado para usinagem de CFRP e

ligas de Ti com obtenção de níveis de rugosidade abaixo de 3,2 μm e circularidade dos furos usinados abaixo 7 μm . Isto confirma a possibilidade de aplicação do fresamento helicoidal em materiais de baixa usinabilidade.

2.2 Rugosidade

O acabamento superficial, representado geralmente pela rugosidade, consiste em um conjunto de irregularidades com formato padrão cíclico ou uma textura característica de uma superfície (AMORIM, 2002). A superfície gerada após o processo de usinagem usualmente apresenta desvios geométricos no acabamento superficial. Normalmente, a rugosidade é inevitável devido aos aspectos inerentes ao processo de usinagem e influência da geometria da ferramenta durante o processo de remoção de cavaco e se enquadra na categoria de desvios micro geométricos (PETROPOULOS *et al.*, 2010).

O conhecimento das relações dos parâmetros do processo é fundamental para melhora dos ajustes dos modelos. A rugosidade de uma peça usinada compreende um dos fatores determinantes para a vida de um componente em trabalho mecânico (PEREIRA, 2017). A rugosidade final obtida de uma superfície usinada é resultante de vários fatores, como o material, dureza e geometria da peça, material e geometria da ferramenta, e parâmetros de corte (BOEHS, 1999; QIN *et al.*, 2014).

A rugosidade pode ser caracterizada por diversos parâmetros, sendo os mais utilizados a rugosidade média ou desvio médio aritmético (R_a) e a rugosidade total (R_t). De acordo com Agostinho *et al.* (2004) e Machado *et al.* (2009), o parâmetro de rugosidade R_a é o valor da média aritmética dos valores absolutos das ordenadas do perfil efetivo (medido) em relação a linha média em um comprimento de amostragem. Enquanto o valor da rugosidade R_t corresponde à distância vertical entre o pico mais alto e o vale mais profundo no comprimento de avaliação, independente dos valores de rugosidade parcial. Adicionalmente também são relacionadas a rugosidade R_q caracterizada pelo desvio médio quadrático e a rugosidade máxima R_z na avaliação dos parâmetros. No presente trabalho serão apresentados e analisados os valores da rugosidade média R_a .

2.2.1 Forças de corte

A previsão adequada dos componentes de força contribui com as estratégias de corte e escolha dos parâmetros de usinagem (VENTURA e HASSUI, 2013). A previsão das forças de corte é um fator essencial para melhorar o desempenho da usinagem e diminuição dos esforços que podem gerar desvios de forma e baixa qualidade superficial. A previsão confiável das forças de corte é ainda fundamental para uma maior previsão da potência necessária, vibrações, qualidade da superfície da peça, precisão geométrica e estabilidade do processo (WANG *et al.*, 2012).

A previsão das forças de corte através de modelagem e simulação é uma importante área de pesquisa a fim de melhorar o planejamento do processo. Durante a modelagem das forças de corte, uma das características investigadas é que ambas as arestas de corte periféricas e frontais interagem com a peça durante o processo de fresamento helicoidal (LIU, 2012).

O estudo do comportamento e magnitude das forças de corte é importante para o planejamento e otimização do processo de corte na finalidade de reduzir os danos de usinagem (WANG *et al.*, 2011). Geralmente, estudos sobre forças de usinagem no fresamento helicoidal consideram dois sistemas de coordenadas: o sistema de coordenadas da peça, fixo com origem no centro do furo, e o sistema de coordenadas da ferramenta, com deslocamento ao longo do passo helicoidal (DENKENA *et al.*, 2008; HAIYAN e XUDA, 2016; SHEN *et al.*, 2012). No sistema de coordenadas da peça, as forças são representadas nos eixos x, y e z, respectivamente F_x , F_y e F_z e podem ser monitoradas com um dinamômetro estacionário. A força axial é igual à componente F_z , de modo que $F_a = F_z$, no sistema de coordenadas da peça (SHAN *et al.*, 2011, HE *et al.*, 2015). No sistema de coordenadas da ferramenta são abordadas diferentes nomenclaturas. Alguns autores consideram a força axial F_a ao longo do eixo z, a força radial F_r atuando na direção do centro da fresa e a força tangencial F_t atuando na direção oposta da direção de corte, como apresentado na Figura 2.5.

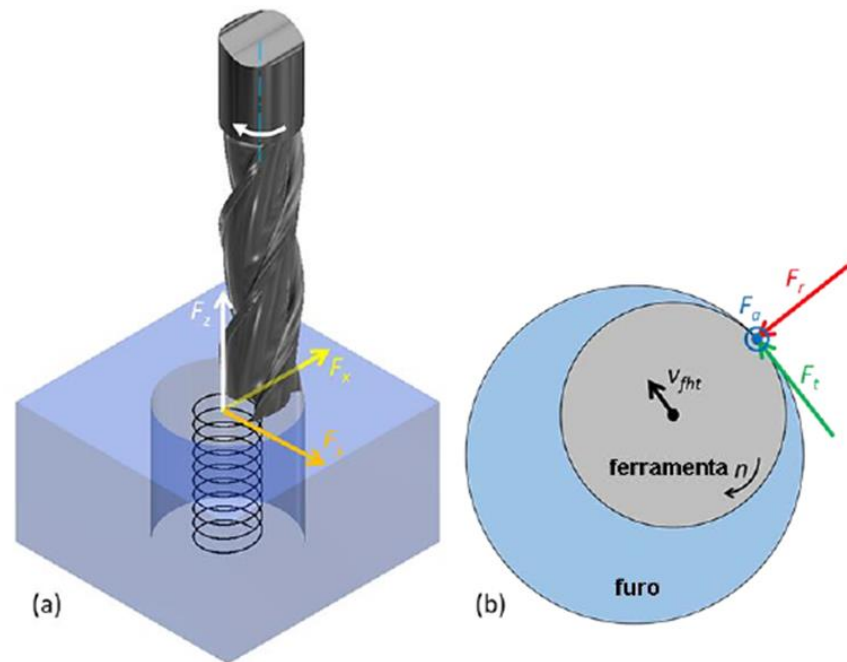


Figura 2.4. Forças de corte no: a) sistema de coordenadas da peça; b) sistema de coordenadas da ferramenta (Adaptado de REY *et al.*, 2016 e PEREIRA *et al.*, 2017)

2.2.2 Circularidade

Os processos de usinagem podem afetar a peça no que corresponde a resistência à fadiga, desgaste e vida do componente em trabalho. Segundo LI *et al.* (2013) produtos que são submetidos a processos de carga cíclica podem apresentar comprometimento na qualidade superficial. Outro aspecto notório que pode ocorrer durante a operação de furação é o erro de circularidade no fresamento helicoidal, uma vez que os desvios de circularidade ou erros de forma, de maneira geral classificados na categoria de desvios macro geométricos, juntamente aos desvios dimensionais e desvios de posição podem dificultar a montagem de componentes (PEREIRA, 2017; PETROPOULOS *et al.*, 2010).

A degradação da qualidade geométrica observada no furo pode ser afetada pela deflexão da ferramenta resultante das forças radiais atuantes no fresamento helicoidal. Para compensar essa deflexão, é possível a utilização de uma fresa de diâmetro maior, porém esse aumento diminui a razão entre corte periférico e axial, diminuindo a eficiência do processo (ALVIM, 2019; IYER *et al.*, 2007). Na medição de circularidade de furos obtidos por fresamento helicoidal também foi observado o erro devido a folga dos eixos x e y do centro de usinagem

CNC em virtude da mudança do sentido de movimento dos eixos durante a interpolação circular (COSTA *et al.*, 2015).

2.2.3 Produtividade

A taxa de remoção de material (*Material Removal Rate* – MRR) é um dos parâmetros para analisar a produtividade na usinagem. A produtividade pode ser definida como a principal forma de avaliação do desempenho de processos produtivos, uma vez que permite a análise da relação entre recursos necessários à produção e o volume produzido (BRAGA, 2000). De maneira geral para o processo de fresamento o valor de MRR pode ser avaliado conforme apresentado na Equação 2.5. A definição da MRR é determinada pelo produto da profundidade de corte (a_p^*) em mm, penetração de trabalho (a_e) em mm e velocidade de avanço mm/min.

$$MRR = a_p^* \cdot a_e \cdot v_f \quad (2.5)$$

2.3 Planejamento de Experimentos

O Planejamento de Experimentos (*Design of Experiments* – DOE) é uma abordagem que atua na investigação da relação entre as variáveis de um processo. Para o entendimento das relações de causa e efeito em um sistema é necessária a investigação das mudanças na saída do processo causadas pela alteração das variáveis de entrada do sistema. Sendo assim, as observações sobre um sistema ou processo podem levar a teorias ou hipóteses sobre o que faz o sistema funcionar, porém o delineamento de experimentos é fundamental para verificar se as hipóteses estão corretas (MONTGOMERY, 2013).

O planejamento de experimentos é ainda uma estratégia utilizada para minimização de custos experimentais, e conseqüente redução na quantidade de experimentos, sendo uma técnica estatística eficaz para modelagem e otimização de experimentos (ALMEIDA *et al.*, 2018). A Metodologia de Superfície de Resposta (*Response Surface Methodology* – RSM) é um dos métodos do DOE mais utilizados para otimização no âmbito da engenharia, atualmente aplicada em diversas áreas onde há a necessidade de modelar e otimizar uma resposta y em função de k

variáveis de controle (PEREIRA, 2017). A metodologia de superfície de resposta foi proposta inicialmente por Box e Wilson (1951) em aplicações em procedimentos químicos. A experimentação pode ser definida com uma ou mais tentativas de avaliação no intuito de verificação de quais variáveis de entrada impactam nas variáveis de saída.

O desenvolvimento de um modelo relacionando a resposta de interesse às variáveis de entrada geralmente abrange dois ou mais fatores a fim de verificar os efeitos em uma variável de resposta. Neste contexto, os planejamentos fatoriais são os mais eficientes já que todas as combinações possíveis dos fatores são investigadas a cada ensaio ou replicação completa (MONTGOMERY, 2013; MYERS *et al.*, 2009).

2.3.1 Metodologia de Superfície de Resposta e Planejamento Composto Central

A metodologia de superfície de resposta (*Response Surface Methodology* – RSM) é um conjunto de técnicas estatísticas e matemáticas úteis para análise, modelagem e otimização nas aplicações em que a resposta de interesse é influenciada por diversas variáveis (MONTGOMERY, 2005). A RSM sugere a construção de um parâmetro para a resposta esperada, com o intuito de encontrar uma aproximação adequada para a relação verdadeira entre a resposta de interesse e as variáveis do processo (CASTILHO, 2007).

A RSM pode ser aplicada quando as variáveis de interesse são quantitativas e influenciam potencialmente as medidas de desempenho ou características de qualidade do produto ou processo (MYERS *et al.*, 2016). Considerando as variáveis de entrada como fatores mensuráveis chamados também de variáveis independentes ou variáveis de processo, a partir de N combinações dos níveis das variáveis de processo, realização de experimentos e medição da resposta de interesse, pode-se obter um modelo de resposta para y em função de x_1, \dots, x_k , representado pela Equação 2.6.

$$y = f(\mathbf{x}) + \varepsilon = f(x_1, x_2, \dots, x_k) + \varepsilon \quad (2.6)$$

Como observado na Equação 2.6, em casos onde a aproximação é adequada e a resposta mantém uma relação linear com as variáveis do processo, a função e o modelo apresentados

são lineares. No entanto, nos casos que apresentam relações não lineares, a função deverá ser um polinomial de grau maior. A partir de realizações experimentais das combinações dos níveis das variáveis de processo e medição da resposta de interesse pode-se obter um modelo de segunda ordem para a resposta em função das variáveis ou fatores de controle. Deste modo, para esses casos que requerem um modelo de segunda ou maior ordem, a função pode ser aproximada por um modelo de segunda ordem conforme a Equação 2.7.

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i<j} \beta_{ij} x_i x_j + \varepsilon \quad (2.7)$$

O planejamento composto central (*Central Composite Design – CCD*) é o arranjo experimental mais utilizado para coleta de dados para a modelagem das funções de superfície de resposta. O planejamento composto central tem o objetivo de estimar modelos de segunda ordem, com boa capacidade de previsão e permite a experimentação sequencial, além de garantir economia na experimentação (BOX, 1951). No CCD, os pontos fatoriais ou fracionados são utilizados para estimar os efeitos lineares e interações de segunda ordem. Os pontos centrais são usados para estimar o erro experimental de maneira independente, e, assim como os pontos axiais, contribuem na estimativa dos termos quadráticos. No presente trabalho, o planejamento experimental foi desenvolvido por um projeto fatorial a meia fração com $n_f = 2^{(k+r-1)} = 2^{(3+3)-1} = 32$ pontos fatoriais, $n_a = 2k = 2 \times 3 = 6$ pontos axiais, $n_c = 9$ pontos centrais, resultando um total de 47 experimentos, realizados de forma aleatória. A Figura 2.4 apresenta dois Planejamentos Composto Central com diferentes números de fatores.

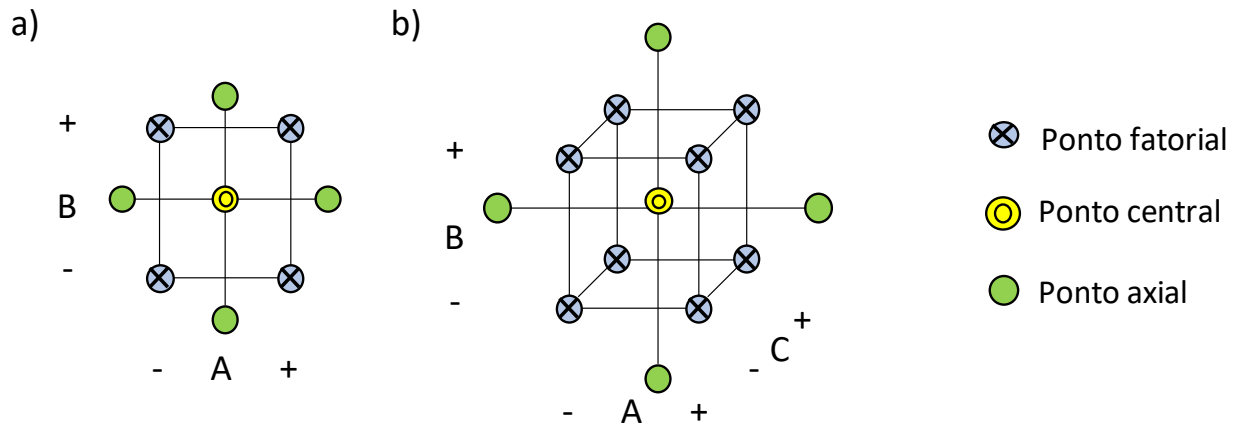


Figura 2.5. Planejamento Composto Central (a) 2 Fatores e (b) 3 Fatores

2.3.2 Projeto de Parâmetro Robusto e Erro Quadrático Médio

O projeto de parâmetro robusto (*Robust Parameter Design – RPD*) é um método que viabiliza a análise e melhoria de processos através da identificação dos fatores controláveis e fatores incontrolláveis (MONTGOMERY, 2005), e a execução do planejamento experimental considerando as variáveis de ruído como variáveis controláveis e efeitos fixos no planejamento experimental. A partir do RPD é possível obter níveis das variáveis de processo que minimizam a influência das variáveis de ruído. O RPD visa determinar os níveis apropriados dos parâmetros que proporcionam menor sensibilidade às fontes de variações (TAGUCHI *et al.*, 2000).

A metodologia de superfície de resposta é ainda uma alternativa eficiente para utilização de projeto de parâmetro robusto para definição de modelagem de média e variância pontualmente, conforme sugerido por Myers *et al.* (1992). De acordo com Taguchi (1986) uma possibilidade de otimização robusta seria resumir os dados de um experimento de matriz cruzada com a média de cada observação na matriz interna através de todas as execuções na matriz externa, sendo combinado na relação sinal-ruído. Outra possibilidade consiste na execução de um planejamento constituído de variáveis de processo e ruído, viabilizando economia experimental e permitindo o estudo das interações entre variáveis de processo e ruído. O primeiro passo para analisar o RPD considerando uma matriz combinada é alcançar um modelo de superfície de resposta em função das variáveis de processo e ruído de acordo com a

Equação 2.8, onde β_0 é uma constante ou intercepto, $\mathbf{x} = [x_1, x_2, \dots, x_k]^T$ é um vetor de variáveis de decisão $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_k]^T$ é um vetor de coeficientes lineares relacionado as variáveis de processo, \mathbf{B} é uma matriz de termos quadráticos de interação relaciona as variáveis de processo, $\mathbf{z} = [z_1, z_2, \dots, z_r]^T$ é um vetor de variáveis de ruído, $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_r]^T$ é um vetor de coeficientes lineares relacionado as variáveis de ruído e $\boldsymbol{\Delta}$ é uma matriz de termos de interação entre variáveis de processo e ruído.

$$\hat{y}(\mathbf{x}, \mathbf{z}) = \beta_0 + \mathbf{x}^T \boldsymbol{\beta} + \mathbf{x}^T \mathbf{B} \mathbf{x} + \mathbf{z}^T \boldsymbol{\gamma} + \mathbf{x}^T \boldsymbol{\Delta} \mathbf{z} \quad (2.8)$$

A matriz \mathbf{B} e a matriz $\boldsymbol{\Delta}$ podem ser escritas conforme segue:

$$\mathbf{B} = \begin{bmatrix} \beta_{11} & \beta_{12}/2 & \dots & \beta_{1k}/2 \\ \beta_{21}/2 & \beta_{22} & \dots & \beta_{2k}/2 \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{k1}/2 & \beta_{k2}/2 & \dots & \beta_{kk} \end{bmatrix}, \boldsymbol{\Delta} = \begin{bmatrix} \delta_{11} & \delta_{12} & \dots & \delta_{1r} \\ \delta_{21} & \delta_{22} & \dots & \delta_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{k1} & \delta_{k2} & \dots & \delta_{kr} \end{bmatrix}$$

A equação de média pode ser obtida como apresentado na Equação 2.9, considerando as variáveis de ruído com média nula, enquanto o modelo de variância é apresentado na Equação 2.10, sendo obtido por propagação de erro em relação às variáveis de ruído. Os modelos de média e variância apresentados envolvem somente as variáveis do processo, portanto o modelo de equação de média apresentado representa o valor esperado para a resposta, enquanto o modelo de variância explica a variabilidade da resposta em relação ao ruído propagada para as variáveis de processo.

$$E_{\mathbf{z}, \varepsilon}[\hat{y}(\mathbf{x}, \mathbf{z})] = \beta_0 + \mathbf{x}^T \boldsymbol{\beta} + \mathbf{x}^T \mathbf{B} \mathbf{x} \quad (2.9)$$

$$Var_{\mathbf{z}, \varepsilon}[\hat{y}(\mathbf{x}, \mathbf{z})] = (\boldsymbol{\gamma} + \boldsymbol{\Delta}^T \mathbf{x})^T \mathbf{V} (\boldsymbol{\gamma} + \boldsymbol{\Delta}^T \mathbf{x}) + \sigma_\varepsilon^2 \quad (2.10)$$

Neste cenário, superfícies de resposta para média e variância podem ser estimadas utilizando um arranjo cruzado Welch *et al.* 1990 e Vining e Myers (1990), ou um arranjo combinado (MYERS *et al.*, 1992). Para essa execução, os planejamentos em RPD que combinam em um único arranjo variáveis de processo e variáveis de ruído, propostos por Welch

et al. (1990), permitem a redução do número de ensaios quando comparados aos arranjos cruzados para estimar o efeito das interações de primeira e segunda ordem relevantes ao RPD. Nos arranjos combinados, há a vantagem de possibilidade do estudo das interações entre as variáveis de processo e ruído importante na propagação do erro e no estudo da robustez do processo (SHOEMAKER *et al.*, 1991).

O Erro Quadrático Médio (*Mean Square Error* – MSE) é aplicado na estatística para medir a esperança dos desvios quadráticos de um estimador, sendo, portanto, uma medida de aproximação ou ajuste de um valor estimado (MOOD, 1974). A minimização do erro quadrático médio, aliada ao projeto de parâmetro robusto, tem a finalidade de aproximar do alvo a modelagem da média de uma resposta de interesse e minimizar a variância desta mesma resposta (BOX E JONES, 1992). De acordo com Vining e Myers (1990) a aplicação é mais recomendada com uma abordagem de matriz combinada, uma vez que no contexto de otimização robusta resultante por arranjo combinado, há uma otimização dual, pois busca-se otimizar média e variância simultaneamente. A otimização do erro quadrático médio é usada com RPD para permitir a otimização deste viés e variância. O erro quadrático pode ser definido com $\hat{y}(\mathbf{x})$ como um estimador do parâmetro y no ponto $\mathbf{x} = [x_1, \dots, x_k]$ com função densidade de probabilidade $f(\mathbf{x})$, conforme Equação 2.11.

$$MSE = E\{[\hat{y}(\mathbf{x}) - y]^2\} = \int (\hat{y}(\mathbf{x}) - y)^2 f(\mathbf{x}) d\mathbf{x} \quad (2.11)$$

Neste sentido, o presente trabalho busca a possibilidade de estudar experimentalmente o comprimento em balanço da ferramenta, a altura medida na superfície do furo e a vazão do fluido de corte como variáveis de ruído no fresamento helicoidal do aço inoxidável super duplex UNS S32760 de forma a minimizar o efeito da variabilidade destas variáveis de ruído nas respostas avaliadas no processo.

2.4 Otimização Multiobjetivo

A otimização multiobjetivo tem por finalidade a otimização de múltiplas funções conflitantes de forma simultânea possibilitando a obtenção de um conjunto de soluções ditas Pareto ótimas. Os métodos clássicos de otimização multiobjetivo por escalarização sugerem a conversão do problema de otimização em um problema de otimização de objetivo único, buscando uma solução Pareto ótima por vez (DEB *et al.*, 2002). Nestes algoritmos cada solução na fronteira de Pareto é obtida a partir da predefinição de preferências (pesos) para as funções objetivo em otimização. Ademais, em tais métodos as soluções não são previamente comparadas, visto que são obtidas de forma independente, sendo necessária uma filtragem adicional para evitar a presença de soluções não dominadas. Os métodos bioinspirados e evolutivos para otimização multiobjetivo são meta-heurísticas baseadas em mecanismos biológicos e evolutivos, respectivamente, que buscam evoluir a fronteira de Pareto em uma única corrida. Tais métodos consideram a comparação das soluções da fronteira em evolução de forma a eliminar soluções não dominadas. Ademais, os mecanismos de evolução previstos nestes métodos garantem uma aproximação melhor da fronteira de Pareto verdadeira, desconhecida nos problemas práticos de otimização multiobjetivo.

2.4.1 Otimização de enxame de partículas multiobjetivo

O método otimização de enxame de partículas multiobjetivo (*Multi-objective Particle Swarm Optimization* – MOPSO) é um método bio-inspirado no movimento praticado pelos pássaros. A abordagem pode ser compreendida como um algoritmo comportamental distribuído que institui uma pesquisa multidimensional. Na simulação, o algoritmo considera que cada movimento de um enxame de partículas é determinado pelo comportamento individual, considerando sua melhor posição armazenada e informações sociais considerando a melhor posição de todas as partículas. Os ajustes são realizados para armazenar as soluções não dominadas encontradas em cada interação (COELLO E LECHUGA, 2002; RAQUEL E NAVAL, 2005).

O algoritmo proposto otimização de enxame de partículas multiobjetivo com distância de aglomeração (*Multi-objective Particle Swarm Optimization with Crowding Distance* –

MOPSO-CD) incorpora o mecanismo de computação de distância de aglomeração no algoritmo de enxame de partículas, de forma específica na melhor seleção global e no método de exclusão de um arquivo externo de soluções não dominadas. O mecanismo de distância de aglomeração em conjunto com o operador de mutação mantém a diversidade de soluções não dominadas no arquivo externo. Inicialmente, a distância de aglomeração é calculada classificando o conjunto de soluções em valores crescentes da função objetivo. O valor da distância de aglomeração de uma solução fornece uma estimativa da densidade de soluções ao redor dessa solução. A Figura 2.5 mostra o cálculo da distância de aglomeração do ponto i como sendo uma estimativa do tamanho do maior cuboide que envolve i sem incluir nenhum outro ponto (RAQUEL E NAVAL, 2005). O Quadro 2.1 apresenta um algoritmo simplificado do método MOPSO-CD.

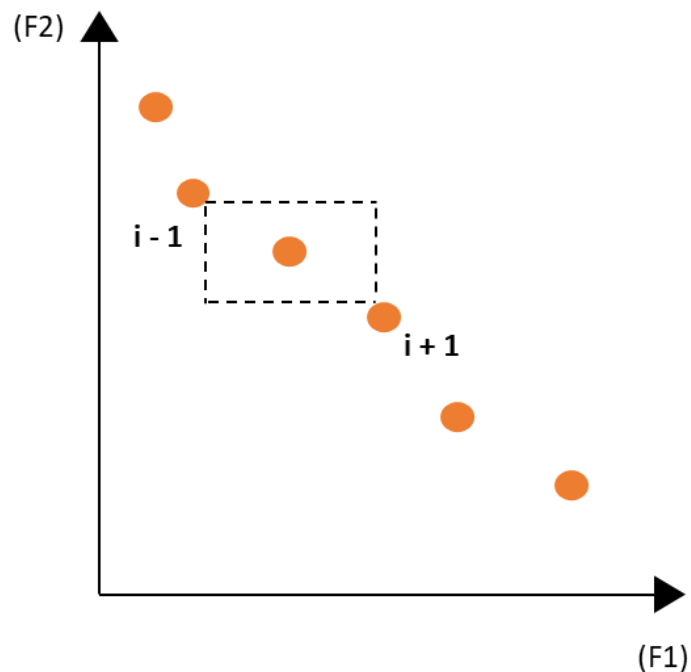


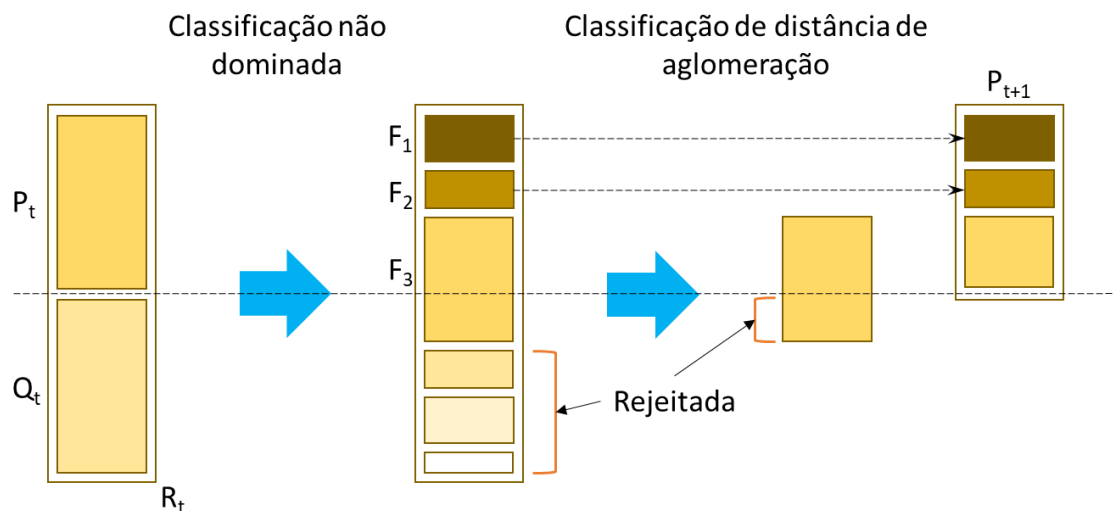
Figura 2.6. Cálculo da distância de aglomeração (Adaptado de DEB *et al.*, 2002)

1. Para $i = 1$ até M (M é o tamanho da população)
 - a. Inicialize $P[i]$ aleatoriamente (P é a população de partículas)
 - b. Inicialize $V[i] = 0$ (V é a velocidade de cada partícula)
 - c. Avalie $P[i]$
 - d. Inicialize o melhor resultado individual de cada partícula, $PBEST[i] = P[i]$
 - e. $GBEST$ = melhor partícula encontrada em $P[i]$
2. Fim Para
3. Inicialize o contador de iterações, $t = 0$
4. Guarde os vetores não dominados encontrados em P em A (A é um arquivo externo para guardar as soluções não dominadas)
5. Repita
 - a. Calcule as distâncias de aglomeração para as soluções não-dominadas do arquivo A
 - b. Ordene as soluções não-dominadas em A em valores decrescentes de distância de aglomeração
 - c. Para $i = 1$ até M
 - i. Selecione aleatoriamente o melhor guia global para $P[i]$ de uma porção top especificada (por exemplo 10% melhores) do arquivo A e guarde sua posição para $GBEST$
 - ii. Compute a nova velocidade, $V[i] = W \times V[i] + R1 \times (PBEST[i] - P[i]) + R2 \times (A[GBEST] - P[i])$
(W é o peso de inércia igual a 0,4; $R1$ e $R2$ são números aleatórios na distribuição uniforme entre 0 e 1)
($PBEST[i]$ é a melhor posição que a partícula alcançou e $A[GBEST]$ é o melhor guia global para cada solução não dominada)
 - iii. Calcule a nova posição de $P[i]$: $P[i] = P[i] + V[i]$
 - iv. Se $P[i]$ vai além das fronteiras, então ele é modificado, tendo a variável de decisão sendo igualada ao seu valor limite (inferior ou superior) com sua velocidade multiplicada por -1 de forma que este caminhe na direção contrária
 - v. Se ($t < tmax \times Pmut$), então faça uma mutação em $P[i]$
($tmax$ é o número máximo de iterações)
($Pmut$ é a probabilidade de mutação)
 - vi. Avalie $P[i]$
 - d. Fim Para
 - e. Insira todas soluções não dominadas de P em A caso não sejam dominadas pelas já armazenadas. Elimine as soluções anteriores dominadas pelas novas do arquivo A . Em caso de o arquivo estar cheio, as soluções que ficam são determinadas conforme segue:
 - i. Calcule a distância de aglomeração para as soluções em avaliação
 - ii. Ordene as soluções em valores decrescentes de distância de aglomeração
 - iii. Selecione aleatoriamente a partícula de uma especificada parte inferior (por exemplo, 10% mais baixo) o qual compreende as partículas mais aglomeradas das em avaliação e as substitua com a nova solução
 - f. Atualize o melhor resultado individual de cada partícula em P . Se o $PBEST$ corrente dominar a posição em memória, as posições das partículas são atualizadas, $PBEST[i] = P[i]$
 - g. $t = t + 1$
6. Repita até $tmax$

Quadro 2.1. Código do algoritmo MOPSO (Adaptado de LI, 2015)

2.4.2 Algoritmo genético de ordenação não-dominado

O método NSGA-II (*Non-dominated Sorting Genetic Algorithm*) trabalha o deslocamento na direção da região ótima de Pareto para encontrar várias soluções ótimas de Pareto em uma única corrida. No método, uma população inicial P_0 de tamanho N é gerada aleatoriamente, seguida de uma população descendente inicial Q_0 também de tamanho N criada através de operações genéticas. As soluções obtidas das populações são ordenadas considerando a não dominância e as primeiras frentes são selecionadas para a próxima população evoluída. Posteriormente, a distância de aglomeração seleciona as últimas soluções das populações posicionadas nas regiões menos aglomeradas. A nova população $P_{(t+1)}$ é formada a partir das soluções melhores classificadas e bem distribuídas na fronteira. O processo é repetido até um critério de parada e pode ser efetivo em problemas simulados ou reais (DEB *et al.*, 2002; TSOU, 2015).

Figura 2.7. Procedimento NSGA-II (Adaptado de DEB *et al.*, 2002)

Para análise de comparação entre os métodos de otimização podem ser considerados o número de soluções geradas, o número de soluções Pareto ótimas e o hiper volume. O hiper

volume é considerado o espaço n-dimensional dominado pelos pontos (soluções) em uma fronteira. Uma fronteira com um hiper volume maior normalmente apresenta um conjunto de opções melhores para o usuário do que uma fronteira com um hiper volume menor (HAO *et al.*, 2012).

3 MÉTODO DE PESQUISA

A pesquisa consistiu em realizar a otimização do fresamento helicoidal do aço inoxidável super duplex UNS S32760. De acordo com Gil (2008) os estudos científicos são classificados quanto à natureza, ao objetivo à abordagem e ao método. O presente trabalho foi classificado como sendo de natureza aplicada com objetivo explicativo. A abordagem da pesquisa é quantitativa com aplicação de método de pesquisa experimental, utilizando experimentos projetados de forma a produzir os dados necessários. Para atingir os objetivos determinados neste trabalho, é projetada a utilização de técnicas de estatística, planejamento de experimentos e métodos de otimização.

4 PROCEDIMENTO EXPERIMENTAL

O presente trabalho propôs a otimização do fresamento helicoidal do aço inoxidável super duplex UNS S32760. Os procedimentos experimentais de fresamento, as medições da rugosidade dos corpos de prova e as medições da força de usinagem foram realizados no Laboratório de Manufatura e Automação (LMAUT) do Instituto de Engenharia de Produção e Gestão da Universidade Federal de Itajubá (UNIFEI). O processo de medição da circularidade foi realizado no Laboratório de Metrologia do Departamento de Engenharia Mecânica e Produção (DEMEP) da Universidade Federal de São João del Rei (UFSJ).

4.1 Equipamentos e materiais

Os experimentos planejados foram realizados no Laboratório de Manufatura e Automação (LMAUT) da Universidade Federal de Itajubá (UNIFEI) em um centro de usinagem

vertical *ROMI*[®] *D 600*, com potência máxima no eixo-árvore de 15 kW e rotação máxima de 10.000 RPM. O Centro de Usinagem possui curso longitudinal da mesa (eixo X) de 600 mm, latitudinal de 530 mm (eixo Y) e 580 mm no curso do cabeçote (eixo Z), além de um cabeçote vertical com cone do eixo-árvore ISO 40, avanço rápido de 30 m/min nos três eixos de programação (X, Y e Z) e é equipado com comando numérico *GE*[®] *FANUC 0i-MD*. O porta-ferramentas utilizado foi um mandril hidráulico *CoroChuck 930*, cone BT-40 normas JIS B 6339/DIN ISO 7388-2 código 930-B40-S-12-085 da *Sandvik Coromant*[®]. Para medição das forças axiais de avanço foi utilizado um dinamômetro estacionário com três componentes modelo 9257BA, com amplificador de carga de sinais 5233A e software *Dynoware* ambos da *Kistler*[®]. A Figura 4.1 expõe o *setup* experimental.



Figura 4.1. *Setup* experimental

As medições de força foram realizadas a uma frequência de 500 Hz. A região de interesse relativa à posição do furo para cada experimento foi selecionada posteriormente, uma vez que a altura a ser medida na superfície do furo foi considerada como fator de ruído. Em virtude da grandeza da obtenção de pontos em cada frequência, foram consideradas pequenas amostras, de tamanhos idênticos em cada posição.

A ferramenta de corte utilizada para a execução dos experimentos foi uma fresa inteiriça da linha *Coromill Plura* código ISO/ANSI R215.H4-10050DAC0P 1620 da *Sandvik Coromant*[®] com 10 mm de diâmetro, $z = 4$ dentes, restrição de $\alpha_p(\text{máx}) = 0,7$ mm e ângulo de saída axial de 6° , ângulo de saída radial de -18° , e ângulo de hélice de 50° . A fresa é de metal

duro, classe ISO M, com cobertura de TiAlN depositada por PVD. O sentido de corte foi concordante. A fresa não apresenta quebra-cavacos, sendo mostrada na Figura 4.2.



Figura 4.2. Fresa *Coromill Plura* código ISO/ANSI R215.H4-10050DAC0P 1620 da *Sandvik®*

O aço inoxidável super duplex UNS S32760 utilizado nos ensaios apresenta valor PRE entre 41,47 a 42,67 e foi fornecido pela empresa Villares Metals de acordo com o certificado de qualidade CQ 1526190. A composição química do aço inoxidável super duplex liga UNS S32760 é mostrada na Tabela 4.1.

Tabela 4.1. Composição química (% peso) do aço inoxidável super duplex UNS S32760

Aço	Quant.	C	Si	Mn	Cr	Ni	Mo	W	Cu	N	PRE
UNS S32760	Min.	0,00	0,00	0,00	24,00	6,00	3,0	0,50	0,50	0,20	≥ 40
	Máx.	0,03	1,00	1,00	26,00	8,00	4,0	1,00	1,00	0,30	
Amostra CQ - 1526190	-	0,019	0,41	0,51	25,36	6,90	3,66	0,73	0,57	0,252	41,47- 42,67

Os corpos de prova do aço inoxidável super duplex UNS S32760 foram utilizados nos ensaios com dureza de 26 HRC (258 HBW). As dimensões dos corpos de prova são 24 mm de diâmetro e 15 mm de altura. O diâmetro dos furos usinados foi de 18 mm. A Figura 4.3

apresenta o corpo de prova montado no dispositivo de fixação, um corpo de prova antes da usinagem, e um corpo de prova depois de furado, respectivamente.



Figura 4.3. (a) Dispositivo de fixação montado com o corpo de prova; (b) corpo de prova e (c) corpo de prova usinado do aço inoxidável super duplex UNS S32760

A medição da rugosidade R_a dos furos usinados foi realizada com a utilização de um rugosímetro portátil da *Mitutoyo*[®] modelo SJ-210, com deslocamento transversal (Z) de 350 μm e longitudinal (X) de 16 mm, velocidade de medição de até 0,75 mm/s, velocidade de retorno de 1 mm/s. O medidor é auxiliado por um computador e pelo software *USB communication tool* também da *Mitutoyo*[®]. Foi considerado um *cut-off* de 0,25 mm para as medições de rugosidade. A Figura 4.4 ilustra o *setup* experimental da medição de rugosidade.

Para a realização das medições foram considerados 2 operadores, cada um realizando três medições, realizadas em três posições radiais equidistantes de 120° no furo; e mais três posições no sentido da altura do furo, medida de 4 mm cada uma e com espaçamento entre as posições de 0,5 mm.



Figura 4.4. *Setup* de medição da rugosidade

A análise da circularidade foi realizada com a utilização de um medidor de forma *Talyround 131* da *Taylor Hobson®* com apalpador de rubi, alta gama de 2 mm, resolução normal de 30 nm e alta resolução de 6 nm. O medidor de forma também é auxiliado por um computador e pelo *software ultra* da *Taylor Hobson®*. Na Figura 4.5 é mostrado o circularímetro.



Figura 4.5. Circularímetro *Taylor Hobson®*

4.2 Planejamento experimental e métodos de otimização

Os experimentos foram conduzidos com base em um planejamento composto central (CCD) com a finalidade de modelagem e estimativa dos efeitos, combinado com as variáveis de controle e de ruído nas respostas consideradas de forma a viabilizar a obtenção do modelo de segunda ordem. O planejamento experimental é composto por um projeto fatorial meia fração com $n_f = 2^{(k+r-1)} = 2^{(3+3)-1} = 32$ pontos fatoriais, $n_c = 9$ pontos centrais e $n_a = 2k = 2 \times 3 = 6$ pontos axiais, totalizando 47 experimentos.

O planejamento composto central permite garantir economia na experimentação e busca de região de curvatura e máximo crescimento da função estimada (BOX, 1951). No CCD, os pontos fatoriais ou fracionados são usados para estimar os efeitos lineares e interações de segunda ordem. Os pontos centrais são usados para estimar o erro experimental de maneira independente, e, junto com os pontos axiais, permitem a estimativa dos termos quadráticos.

Os fatores de controle avaliados foram a velocidade de corte (v_c) variando entre 35 e 65 m/min, o avanço axial por dente (f_{za}) com variação entre 0,10 e 0,20 $\mu\text{m}/\text{dente}$ e avanço tangencial por dente (f_{zt}) com variação entre 0,10 e 0,20 mm/dente. Os fatores de ruído considerados foram o comprimento em balanço (l_{to}) em mm, a altura medida do furo (l_b) em

mm e a vazão do fluido de corte (Q) em l/min. A Tabela 4.2 apresenta as variáveis de controle e ruído adotadas no fresamento helicoidal do aço inoxidável super duplex UNS S32760 e os níveis fatoriais, centrais e axiais aos quais foram testados.

Tabela 4.2. Variáveis de controle e ruído adotados e seus níveis

Variáveis de controle	Axial	Fatorial	Níveis			Unidade
			Central	Fatorial	Axial	
f_{za}	0,03	0,10	0,15	0,20	0,27	$\mu\text{m/dente}$
f_{zt}	0,03	0,10	0,15	0,20	0,27	mm/dente
v_c	14,32	35	50	65	85,68	m/min
Variáveis de ruído						
l_o	-	27	29	31	-	mm
l_b	-	início	meio	fim	-	-
Q	-	5	12,5	20	-	l/min

As análises estatísticas foram realizadas na linguagem de programação *R* (*R CORE TEAM*, 2020) através dos pacotes: *rsm* (Lenth, 2009) para obtenção dos modelos de superfície de resposta; *olsrr* (HEBBALI, 2020) para teste de homoscedasticidade; *ggplot2* (WICKHAM, 2016) e *ggpubr* (KASSAMBARA, 2020) para os gráficos de efeitos principais e de comparação entre as fronteiras de Pareto; *mopsocd* (NAVAL, 2013) e *mco* (MERSMANN, 2014) para otimização multi-objetivo.

5. RESULTADOS E DISCUSSÕES

5.1 Resultados experimentais

Os resultados experimentais do fresamento helicoidal do aço inoxidável super duplex UNS S32760 obtidos a partir do planeamento composto central CCD combinando variáveis de processo e ruído são apresentados na Tabela 5.1. São apresentados os resultados das medições da rugosidade média (R_a), força axial de avanço (F_a) e os desvios de circularidade (R_{ont}), além da resposta determinística MRR.

Tabela 5.1. Planeamento experimental com respostas e resultados

Ordem Padrão	f_{za}	f_{zt}	v_c	l_{to}	l_b	Q	R_a	F_a	R_{ont}	MRR
	[$\mu\text{m}/\text{dente}$]	[mm/dente]	[m/min]	[mm]	[-]	[l/min]	[μm]	[kN]	[μm]	[mm^3/min]
1	0,1	0,1	35	27	-1	-1	0,16	0,29	19,11	113,40
2	0,2	0,1	35	27	-1	1	0,23	0,58	13,08	226,80
3	0,1	0,2	35	27	-1	1	0,24	0,56	19,14	113,40
4	0,2	0,2	35	27	-1	-1	0,22	0,57	12,91	226,80
5	0,1	0,1	65	27	-1	1	0,19	0,09	22,12	210,60
6	0,2	0,1	65	27	-1	-1	0,19	0,30	32,48	421,20
7	0,1	0,2	65	27	-1	-1	0,20	0,46	19,42	113,40
8	0,2	0,2	65	27	-1	1	0,25	0,72	25,49	226,80
9	0,1	0,1	35	31	-1	1	0,20	0,17	16,44	113,40
10	0,2	0,1	35	31	-1	-1	0,15	0,24	12,80	226,80
11	0,1	0,2	35	31	-1	-1	0,24	0,25	20,14	210,60
12	0,2	0,2	35	31	-1	1	0,28	0,71	22,70	421,20
13	0,1	0,1	65	31	-1	-1	0,19	0,36	17,77	210,60
14	0,2	0,1	65	31	-1	1	0,20	0,70	14,67	421,20
15	0,1	0,2	65	31	-1	1	0,24	0,25	18,61	113,40
16	0,2	0,2	65	31	-1	-1	0,24	0,57	24,95	226,80
17	0,1	0,1	35	27	1	1	0,14	0,54	15,67	210,60
18	0,2	0,1	35	27	1	-1	0,15	0,31	13,08	421,20
19	0,1	0,2	35	27	1	-1	0,21	0,26	22,33	210,60
20	0,2	0,2	35	27	1	1	0,22	0,79	20,01	421,20
21	0,1	0,1	65	27	1	-1	0,19	0,67	15,46	113,40
22	0,2	0,1	65	27	1	1	0,17	0,49	16,97	226,80
23	0,1	0,2	65	27	1	1	0,16	0,64	23,23	113,40

24	0,2	0,2	65	27	1	-1	0,21	0,52	23,99	226,80
25	0,1	0,1	35	31	1	-1	0,18	0,59	13,15	210,60
26	0,2	0,1	35	31	1	1	0,18	0,85	18,41	421,20
27	0,1	0,2	35	31	1	1	0,24	0,75	13,97	210,60
28	0,2	0,2	35	31	1	-1	0,26	0,55	21,79	421,20
29	0,1	0,1	65	31	1	1	0,19	0,47	19,85	113,40
30	0,2	0,1	65	31	1	-1	0,17	0,60	16,08	226,80
31	0,1	0,2	65	31	1	-1	0,24	0,65	19,16	113,40
32	0,2	0,2	65	31	1	1	0,19	0,62	19,14	226,80
33	0,03	0,15	50	29	0	0	0,15	0,41	15,17	210,60
34	0,27	0,15	50	29	0	0	0,17	0,66	18,31	421,20
35	0,15	0,03	50	29	0	0	0,10	0,11	11,72	210,60
36	0,15	0,27	50	29	0	0	0,20	0,65	20,71	421,20
37	0,15	0,15	14,32	29	0	0	0,26	0,86	10,79	50,34
38	0,15	0,15	85,68	29	0	0	0,21	0,60	18,41	435,65
39	0,15	0,15	50	29	0	0	0,16	0,62	15,87	243,01
40	0,15	0,15	50	29	0	0	0,13	0,07	19,38	243,00
41	0,15	0,15	50	29	0	0	0,14	0,49	18,35	69,61
42	0,15	0,15	50	29	0	0	0,15	0,64	17,81	416,38
43	0,15	0,15	50	29	0	0	0,17	0,79	13,58	243,00
44	0,15	0,15	50	29	0	0	0,17	0,66	14,54	243,00
45	0,15	0,15	50	29	0	0	0,15	0,31	14,46	243,00
46	0,15	0,15	50	29	0	0	0,16	0,73	23,77	243,00
47	0,15	0,15	50	29	0	0	0,14	0,52	14,47	243,00

5.2 Rugosidade média R_a

A rugosidade média R_a foi medida na direção axial da superfície dos furos. Os valores obtidos variaram entre 0,0982 e 0,2838 μm , que são considerados rugosidades relativamente baixas para uma operação de furação em aço inoxidável. As forças axiais de avanço obtidas foram relativamente baixas em função dos baixos avanços por dente na direção axial.

A modelagem dos dados de rugosidade R_a foi realizada através do método dos mínimos quadrados ponderados. A Tabela 5.2 apresenta o modelo de regressão codificado e o teste **t** para os coeficientes. Os efeitos significativos foram destacados na última coluna de acordo com o nível de significância. Considerando $\alpha = 0,05$, os efeitos significativos foram de f_{zt} , l_{to} , l_b , v_c^2 , $f_{zt} \times v_c$ e $v_c \times Q$.

Tabela 5.2. Teste t para os coeficientes do modelo de rugosidade

	Estimativa	Erro padrão	Valor t	Pr(> t)	
(Intercept)	0,168871	0,002924	57,756	< 2e-16	***
f_{za}	0,002374	0,002042	1,163	0,255965	
f_{zt}	0,020807	0,001897	10,966	4,83e-11	***
v_c	-0,003603	0,001883	-1,913	0,067263	.
l_{to}	0,008379	0,002009	4,171	0,000319	***
l_b	-0,009706	0,003430	-2,830	0,009047	**
Q	0,004964	0,003467	1,432	0,164598	
$I(f_{za}^2)$	0,005448	0,003953	1,378	0,180340	
$I(f_{zt}^2)$	0,002736	0,003950	0,693	0,494882	
$I(v_c^2)$	0,017270	0,003407	5,069	3,12e-05	***
$f_{za}: f_{zt}$	0,003694	0,003548	1,041	0,307838	
$f_{za}: v_c$	-0,002577	0,003350	-0,769	0,448968	
$f_{za}: l_{to}$	-0,004110	0,003442	-1,194	0,243710	
$f_{za}: l_b$	-0,003913	0,003383	-1,157	0,258376	
$f_{za}: Q$	0,004829	0,003327	1,452	0,159046	
$f_{zt}: v_c$	-0,007824	0,003357	-2,331	0,028127	*
$f_{zt}: l_{to}$	0,006584	0,003358	1,961	0,061096	.
$f_{zt}: l_b$	-0,001768	0,003464	-0,510	0,614205	
$f_{zt}: Q$	-0,004320	0,003406	-1,268	0,216366	
$v_c: l_{to}$	-0,002457	0,003642	-0,675	0,506102	
$v_c: l_b$	-0,001107	0,003303	-0,335	0,740257	
$v_c: Q$	-0,007241	0,003357	-2,157	0,040808	*
R^2	0,9739				
R^2_{aj}	0,9521				

Significância: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

O modelo codificado de regressão que descreve R_a considerando os níveis das variáveis testados é dado na Equação 5.1. Pelo coeficiente de determinação ajustado R^2_{aj} , o modelo para rugosidade explica 95,21% da variabilidade dos dados.

$$\begin{aligned}
 R_a = & 0,1688 + 0,024 f_{za} + 0,021 f_{zt} - 0,0036 v_c + 0,0084 l_{to} - 0,0097 l_b + 0,0049 Q + 0,0054 f_{za}^2 \\
 & + 0,0027 f_{zt}^2 + 0,0049 v_c^2 + 0,0037 f_{za} f_{zt} - 0,0026 f_{za} v_c - 0,0041 f_{za} l_{to} - 0,0039 f_{za} l_b + 0,0048 \\
 & f_{za} Q - 0,0078 f_{zt} v_c + 0,0066 f_{zt} l_{to} - 0,0018 f_{zt} l_b
 \end{aligned} \quad (5.1)$$

Os gráficos dos efeitos principais são mostrados na Figura 5.1. Considerando os efeitos significativos, em relação às variáveis de processo pode-se observar um efeito linear positivo de f_{zt} com aumento do avanço provocando aumento nos níveis da rugosidade média R_a , e um efeito quadrático de v_c com concavidade para cima, sendo observado uma redução nos níveis de R_a nos níveis intermediários de velocidade. Já em relação às variáveis de ruído significativas, observa-se um efeito linear positivo de l_{to} , já que o aumento do comprimento em balanço pode causar um aumento da deflexão da ferramenta e a obtenção de uma superfície mais rugosa. O efeito linear negativo de l_b confronta a literatura, uma vez que o alisamento da superfície pelas arestas de corte tangenciais proporciona um número maior de voltas no início do furo (LI e LIU, 2013). No caso de tais variáveis é importante entender como estas interagem com as variáveis de processo.

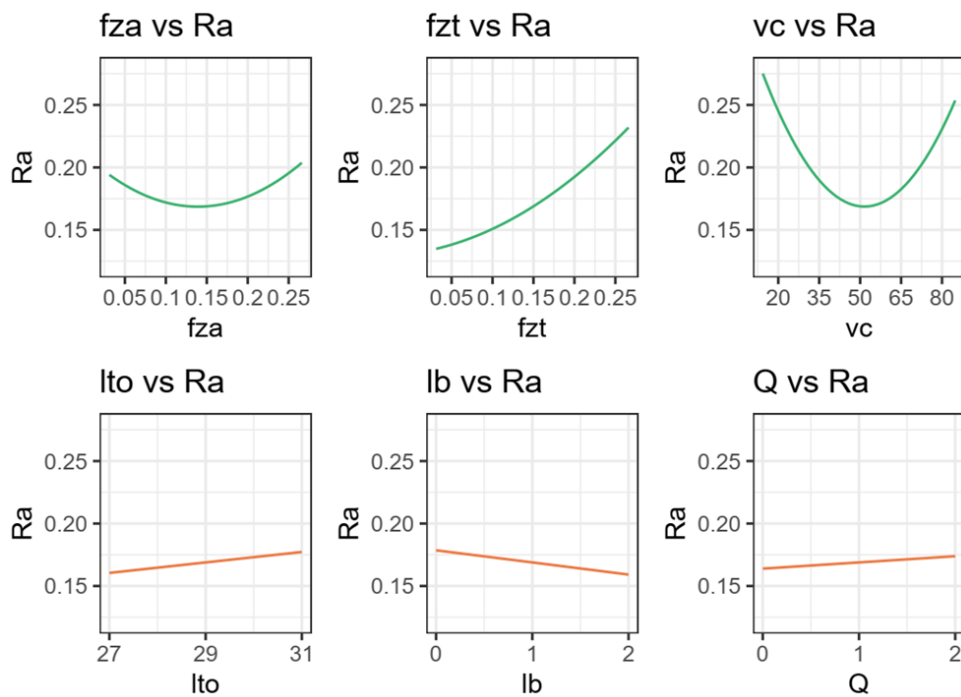


Figura 5.1. Gráficos de efeitos principais das respostas da rugosidade R_a

A Figura 5.2 apresenta os gráficos de interações das variáveis de controle e ruído sobre a rugosidade média R_a . Considerando a interação de processo significativa $f_{zt} \times v_c$, pode-se observar que o efeito linear de f_{zt} é maior em velocidades de corte mais baixas. Tomando a interação processo \times ruído significativa $v_c \times Q$, pode-se observar que um nível de velocidade próximo de 63 m/min viabiliza a minimização do efeito da vazão na rugosidade, ou seja o processo fica menos dependente do uso de fluido de corte e portanto, mais sustentável. A variável f_{za} apresentou leve curvatura e robustez com a vazão no nível 0,1 $\mu\text{m}/\text{dente}$, enquanto

f_{zt} apresentou uma inclinação maior e robustez próximo a 0,23 mm/dente. A interação $f_{zt} \times l_{to}$ apresentou nível de significância igual a 0,06, o que pode ser considerado significativo, ou seja para f_{zt} de 0,1 mm/dente, para os parâmetros de entrada este processo pode ser considerado robusto e independente do balanço da fresa. Graficamente, há outras interações processo×ruído que se destacam, porém não foram estatisticamente significativas. Entretanto, uma vez que o modelo de resposta contém tais coeficientes, tais interações também ditarão o comportamento do modelo de variância e os melhores cenários para minimizar o erro propagado serão obtidos por otimização. É importante destacar que em alguns casos é difícil obter robustez, dado ausência de interação, por exemplo $f_{zt} \times l_b$. Pode-se observar em níveis mais altos de f_{za} proporcionam robustez em relação à variação do comprimento em balanço na resposta R_a , já para níveis mais baixos de f_{za} proporcionam robustez em relação à altura de medição da rugosidade do furo (l_b).

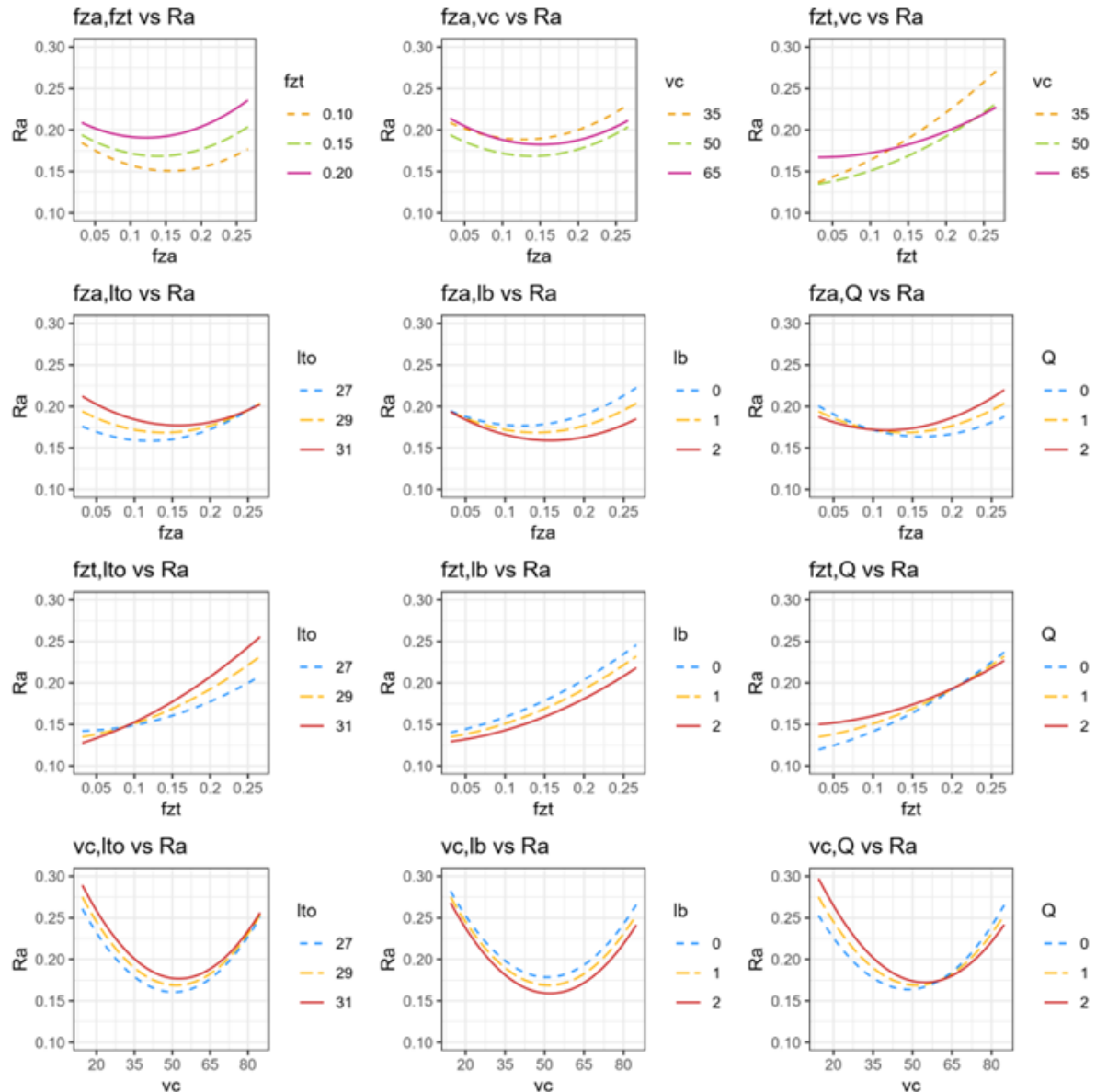


Figura 5.2. Gráficos de interações entre as variáveis de controle e ruído

A partir do modelo de resposta de R_a foram derivados os modelos de média e variância, conforme Equações 2.07 e 2.08. Neste sentido, são apresentados a seguir os vetores e matrizes dos coeficientes que compõem tais equações. Para um melhor entendimento e interpretação dos modelos são apresentados gráficos de contorno para as funções de média e desvio-padrão da rugosidade R_a . A Figura 5.3 apresenta estes gráficos considerando os efeitos significativos das variáveis de processo, percebe-se na combinação com f_{zt} e v_c um efeito linear positivo do avanço, com efeito quadrático da velocidade de corte, acarretando em níveis mínimos de avanço tangencial e intermediário de velocidade de corte para minimizar a rugosidade. Entretanto, no RPD deseja-se não apenas otimizar a média, mas também tornar o processo robusto a partir da

minimização da variância, considerando a propagação de erro a partir dos coeficientes das variáveis de ruído. Por exemplo, na combinação entre f_{za} e f_{zt} no desvio-padrão de R_a percebe-se níveis mais baixos destes parâmetros garantem uma menor variância relativa às variáveis de ruído na rugosidade média. Entretanto, tais gráficos de contorno são cortes, uma vez que se tem três variáveis independentes e uma resposta, sendo a discussão destes limitada.

$$\beta_{Ra} = [0.0024 \quad 0.0208 \quad -0.0036]^T$$

$$\mathbf{B}_{Ra} = \begin{bmatrix} 0.0054 & 0.0018 & -0.0013 \\ 0.0018 & 0.0027 & -0.0039 \\ -0.0013 & -0.0039 & 0.017 \end{bmatrix}$$

$$\gamma_{Ra} = [0.0084 \quad -0.0097 \quad 0.0050]^T$$

$$\Delta_{Ra} = \begin{bmatrix} -0.0041 & -0.0039 & 0.0048 \\ 0.0066 & -0.0018 & -0.0043 \\ -0.0024 & -0.0011 & -0.0072 \end{bmatrix}$$

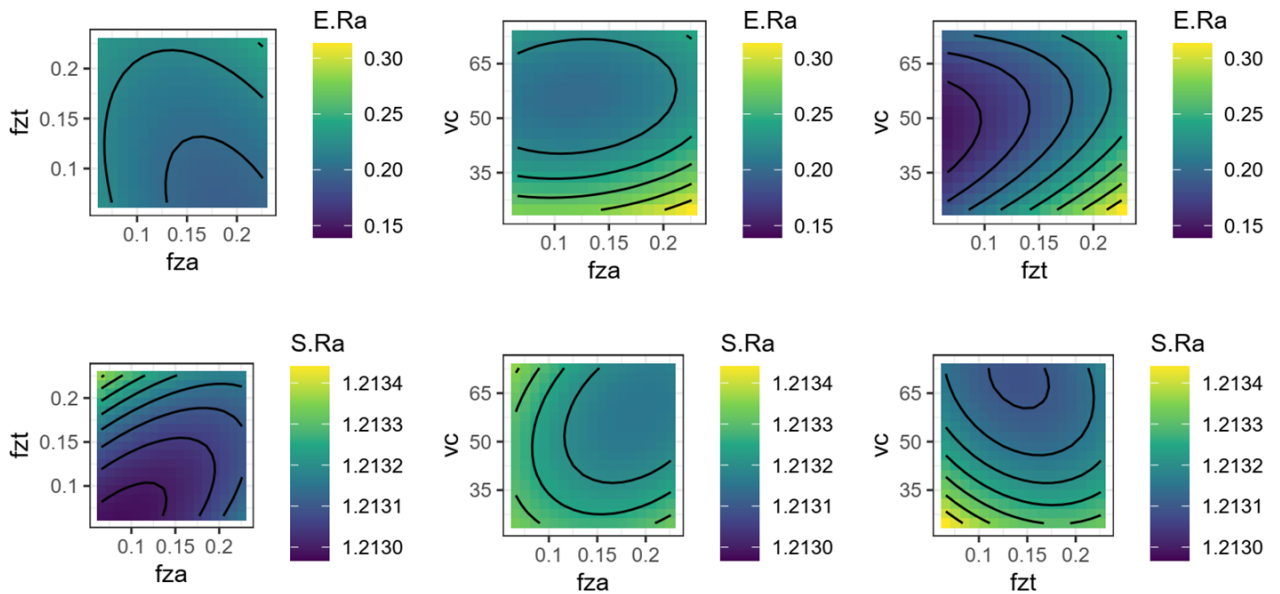


Figura 5.3. Gráficos de Contorno para média (E) e Desvio-padrão (S) da rugosidade R_a

5.3 Força axial de avanço

As forças de corte podem ser medidas na direção de alimentação x, direção normal y e direção axial z. Quando as forças de corte são medidas durante o corte de uma peça fixada no dinamômetro, os componentes medidos são paralelos aos três eixos da máquina ferramenta

(WANG *et al.*, 2012). Os valores da força axial F_a variaram entre 73,7 a 86,4 N, conforme apresentado na Tabela 5.1.

A modelagem dos dados da força axial F_a foi realizada através do método dos mínimos quadrados ponderados. A Tabela 5.3 apresenta o modelo de regressão codificado e o teste **t** para os coeficientes. Os efeitos significativos foram destacados na última coluna de acordo com o nível de significância.

Tabela 5.3. Teste t para os coeficientes do modelo de força

	Estimativa	Erro Padrão	Valor t	Pr(> t)	
(Intercept)	0,516042	0,002729	189,090	< 2e-16	***
f_{za}	0,061409	0,007569	8,114	1,81e-08	***
f_{zt}	0,071194	0,009265	7,684	4,86e-08	***
v_c	-0,018402	0,008821	-2,086	0,04732	*
l_{to}	0,014779	0,011519	1,283	0,21128	
l_b	0,082799	0,009742	8,499	7,66e-09	***
Q	0,053235	0,011065	4,811	6,08e-05	***
$I(f_{za}^2)$	-0,000575	0,003497	-0,164	0,87073	
$I(f_{zt}^2)$	-0,026353	0,010291	-2,561	0,01687	*
$I(v_c^2)$	0,0364653	0,010315	3,535	0,00162	**
$f_{za} \cdot f_{zt}$	0,009043	0,009387	0,963	0,34460	
$f_{za} \cdot v_c$	-0,003323	0,011584	-0,287	0,77660	
$f_{za} \cdot l_{to}$	0,030117	0,010245	2,939	0,00698	**
$f_{za} \cdot l_b$	-0,069617	0,009850	-7,068	2,08e-07	***
$f_{za} \cdot Q$	0,0734969	0,009576	7,675	4,96e-08	***
$f_{zt} \cdot v_c$	-0,012152	0,011631	-1,045	0,30610	
$f_{zt} \cdot l_{to}$	-0,036886	0,010785	-3,420	0,00216	**
$f_{zt} \cdot l_b$	-0,023506	0,011626	-2,022	0,05403	
$f_{zt} \cdot Q$	0,025897	0,010658	2,430	0,02263	*
$v_c \cdot l_{to}$	0,005702	0,012910	0,442	0,66254	
$v_c \cdot l_b$	0,003206	0,011505	0,279	0,78278	
$v_c \cdot Q$	-0,061853	0,012841	-4,817	5,99e-05	***
R^2	0,9782				
R^2_{aj}	0,9598				
Significância: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1					

O modelo codificado de regressão que descreve F_a considerando os níveis das variáveis testados é dado na Equação 5.2. Pelo coeficiente de determinação ajustado R^2_{aj} , o modelo para força axial explica 95,98% da variabilidade dos dados.

$$F_a = 83,9397 - 1,916 f_{za} - 3,48 f_{zt} - 7,298 v_c - 0,555 l_{to} + 0,9360 l_b + 0,1711 Q + 0,009978 f_{za}^2 + 0,6598 f_{zt}^2 + 5,8465 v_c^2 + 2,4408 f_{za} f_{zt} + 2,2569 f_{za} v_c + 0,9068 f_{za} l_{to} + 0,6892 f_{za} l_b + 0,1463 f_{za} Q + 1,7019 f_{zt} v_c + 1,6175 f_{zt} l_{to} + 1,1094 f_{zt} l_b + 0,6741 f_{zt} Q + 1,6053 v_c l_{to} + 1,4596 f_{zt} l_b + 0,2501 v_c Q \quad (5.2)$$

Os gráficos dos efeitos principais são mostrados na Figura 5.4. Considerando os efeitos significativos, em relação às variáveis de processo pode-se observar um efeito linear positivo de f_{za} em F_a devido ao fato do aumento do avanço axial proporcionar um aumento na profundidade de corte axial, medida na mesma direção da força. Já a variável f_{zt} apresentou efeito linear positivo até aproximadamente 0,2 mm/dente, proporcionando aumento nos níveis de força, e um efeito quadrático de v_c podendo ser observado um aumento dos níveis de força nos níveis extremos de velocidade. Em relação as variáveis de ruído, observa-se um efeito linear positivo de l_b e um efeito linear positivo de Q , que pode ser explicado por um aumento da área de contato fresa/peça no final do furo no caso de l_b , e um aumento da força axial em função de maior resfriamento da peça com maior vazão de fluido no caso de Q .

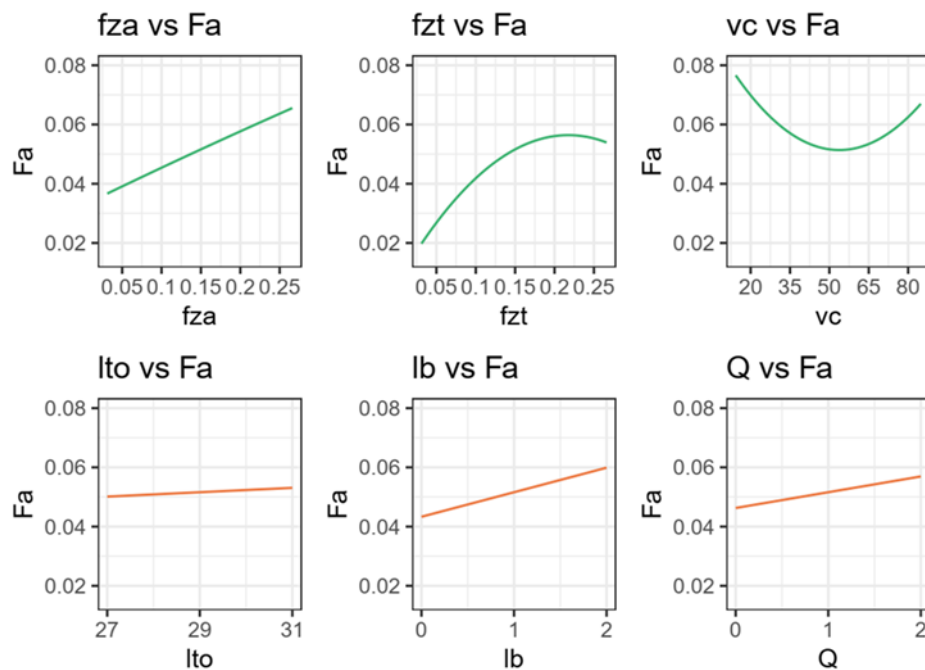


Figura 5.4. Gráfico de efeitos principais das respostas da força axial F_a

A Figura 5.5 apresenta os gráficos de interações para F_a . Considerando as interações de processo não significativas $f_{za} \times f_{zt}$ e $f_{zt} \times v_c$, na interação $f_{za} \times f_{zt}$ observa-se um efeito positivo mais alto de f_{za} quando f_{zt} está no nível mais alto, enquanto na interação $f_{zt} \times v_c$ observa-se um efeito mais alto de f_{zt} com velocidade de corte mais baixa. Em relação as interações processo×ruído significativas $f_{za} \times l_{to}$, $f_{za} \times l_b$, $f_{za} \times Q$, na interação $f_{za} \times l_{to}$ o nível de f_{za} que garante a robustez de F_a em relação a l_{to} é aproximadamente 0,12 $\mu\text{m}/\text{dente}$, já na interação $f_{za} \times l_b$ o nível de f_{za} que torna F_a robusto em relação ao ruído é aproximadamente 0,21 $\mu\text{m}/\text{dente}$, e na interação $f_{za} \times Q$ a robustez ocorre próximo a f_{za} no ponto 0,12 $\mu\text{m}/\text{dente}$. As interações $f_{zt} \times l_{to}$, $f_{zt} \times Q$ também foram estatisticamente significativas. Na interação $f_{zt} \times l_{to}$ o nível de f_{zt} que tornou F_a robusta em relação à l_{to} ocorreu em aproximadamente em 0,17 mm/dente e na interação $f_{zt} \times Q$ próximo à 0,06 mm/dente. Na interação $v_c \times Q$ pode-se observar que a redução dos níveis de força nos níveis de velocidade até 63 m/min, e neste ponto tem-se a minimização do efeito da vazão na força axial.

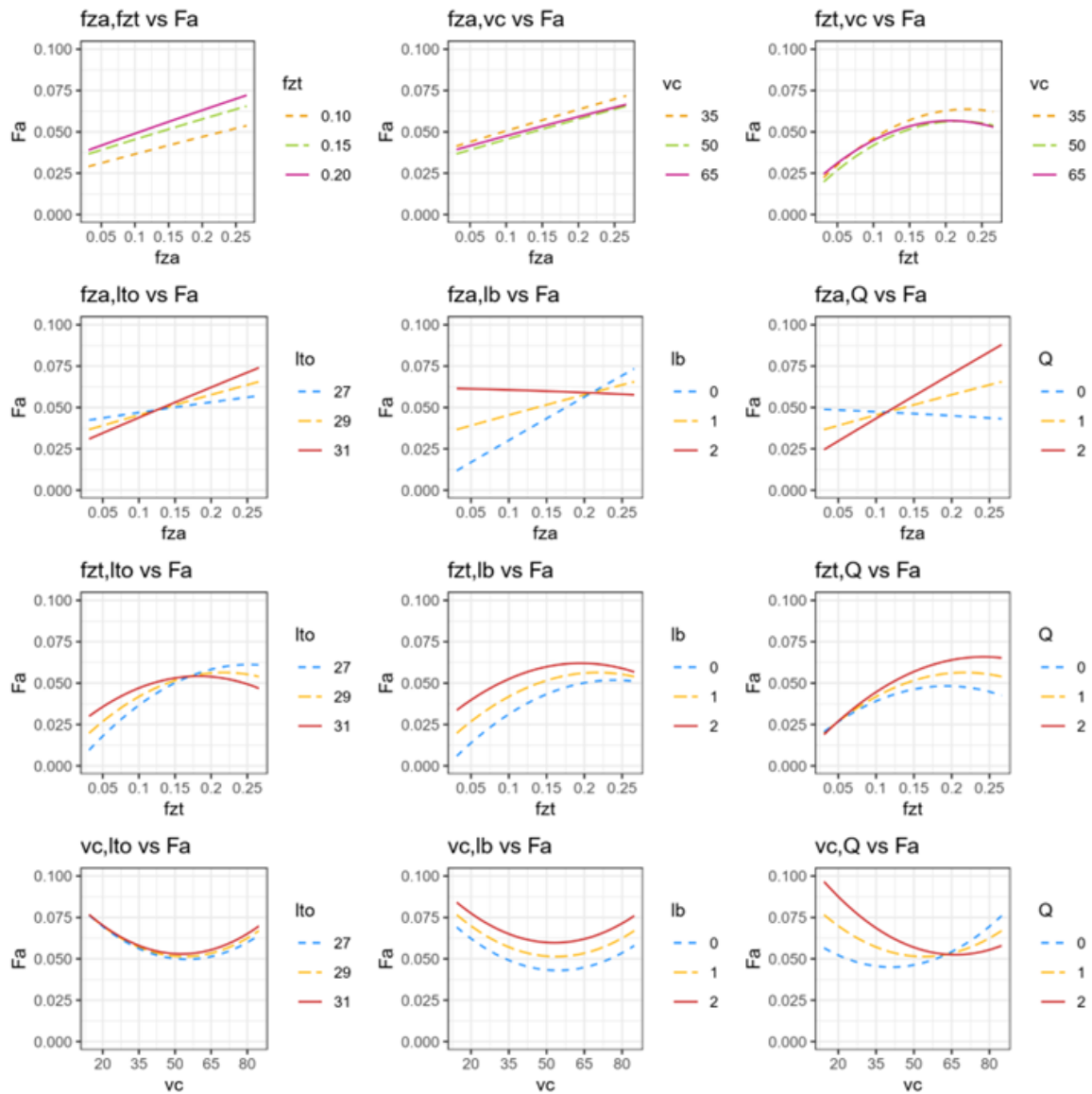


Figura 5.5. Gráficos de interações entre as variáveis de controle e ruído

Através do modelo de resposta de F_a foram derivados os modelos de média e variância. A Figura 5.6 apresenta os gráficos de média e desvio-padrão para a força axial F_a . Na avaliação dos efeitos significativos, percebe-se que nos níveis mais baixos de avanço $f_{za} \times f_{zt}$ tem-se menores valores da força axial F_a . Já em níveis intermediários de velocidade tem-se menor desvio-padrão de F_a , ou seja, uma menor variância relativa às variáveis de ruído.

$$\beta_{F_a} = [0.0061 \quad 0.0071 \quad -0.0018]^T$$

$$\mathbf{B}_{F_a} = \begin{bmatrix} -5.7490 & 0.0005 & -0.0002 \\ 4.5215 & -0.0026 & -0.0006 \\ -1.6613 & -0.0006 & 0.0036 \end{bmatrix}$$

$$\gamma_{F_a} = [0.0015 \quad 0.0083 \quad 0.0053]^T$$

$$\Delta_{Fa} = \begin{bmatrix} -0.0041 & -0.0039 & 0.0048 \\ 0.0066 & -0.0018 & -0.0043 \\ -0.0025 & -0.0011 & -0.0072 \end{bmatrix}$$

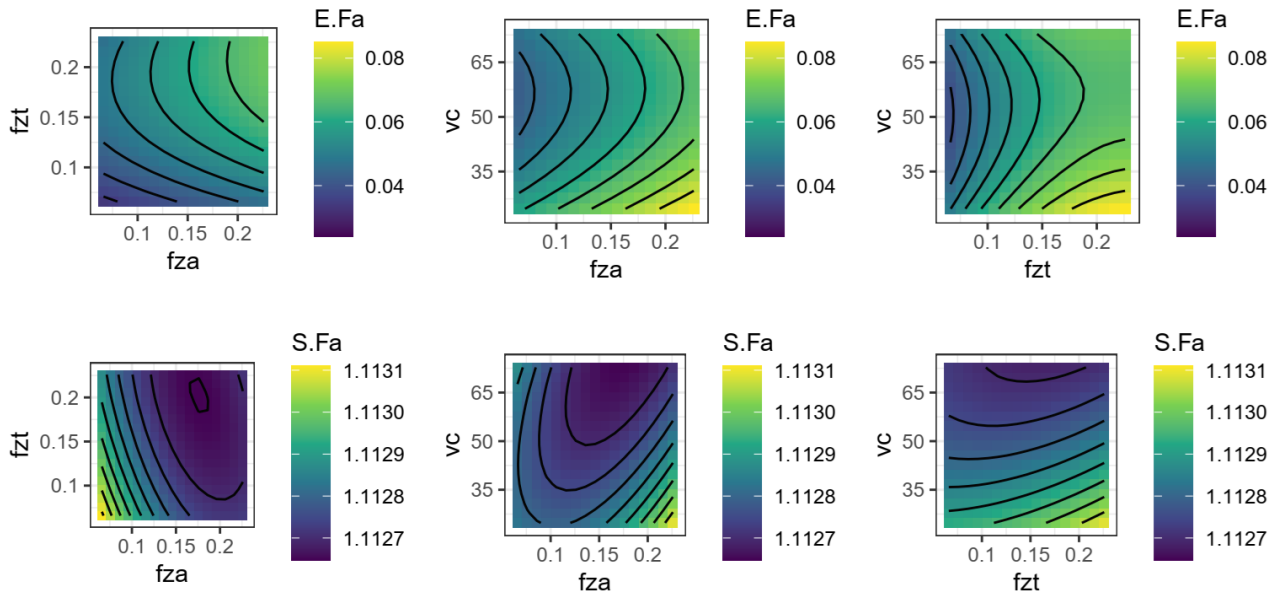


Figura 5.6. Gráficos de Contorno para média (E) e Desvio-padrão (S) de F_a

5.4 Desvios de Circularidade

A circularidade total Ron_t foi considerada neste trabalho como avaliação do erro de forma dos furos do aço inoxidável super duplex. Conforme apresentado na Tabela 5.1 os valores da resposta Ron_t variaram de 10,79 a 32,48 [μm] dentro do planejamento experimental. A Tabela 5.4 apresenta a análise dos resultados e o teste t de Ron_t .

A modelagem dos dados de circularidade Ron_t foi realizada através do método dos mínimos quadrados ponderados. A Tabela 5.4 apresenta o modelo de regressão codificado e o teste t para os coeficientes. Os efeitos significativos foram destacados na última coluna de acordo com o nível de significância.

Tabela 5.4. Teste t para os coeficientes de Ron_t

	Estimativa	Erro Padrão	Valor t	Pr(> t)	
(Intercept)	18,30853	0,15619	117,220	< 2e-16	***
f_{za}	0,42727	0,31199	1,369	0,1830	

f_{zt}	1,85600	0,20883	8,887	3,28e-09	***
v_c	1,46811	0,22526	6,517	7,94e-07	***
l_{to}	-0,68720	0,25734	-2,670	0,0131	*
l_b	-0,73109	0,35390	-2,066	0,0494	*
Q	-0,14191	0,32072	-0,442	0,6619	
$I(f_{za}^2)$	0,04609	0,26236	0,176	0,8620	
$I(f_{zt}^2)$	-0,10323	0,24445	-0,422	0,6764	
$I(v_c^2)$	-0,34892	0,26591	-1,312	0,2014	
$f_{za}: f_{zt}$	0,62340	0,52764	1,181	0,2485	
$f_{za}: v_c$	0,82093	0,59678	1,376	0,1811	
$f_{za}: l_{to}$	0,17849	0,36236	0,493	0,6266	
$f_{za}: l_b$	0,23133	0,39996	0,578	0,5682	
$f_{za}: Q$	-0,61010	0,36157	-1,687	0,1040	.
$f_{zt}: v_c$	-0,55598	0,38478	-1,445	0,1609	
$f_{zt}: l_{to}$	0,48294	0,58038	0,832	0,4132	
$f_{zt}: l_b$	0,61955	0,55479	1,117	0,2747	
$f_{zt}: Q$	0,08530	0,57024	0,150	0,8823	
$v_c: l_{to}$	-0,99886	0,51306	-1,947	0,0629	.
$v_c: l_b$	-0,92134	0,54137	-1,702	0,1012	.
$v_c: Q$	-0,55861	0,51260	-1,090	0,2862	
R^2	0,9997				
R^2_{aj}	0,9994				
Significância: 0 '****' 0.001 '***' 0.01 '**' 0.05 '.' 0.1 ' ' 1					

O modelo codificado de regressão que descreve Ron_t considerando os níveis das variáveis testados é dado na Equação 5.3, com coeficiente de determinação ajustado R^2_{aj} para o modelo de circularidade explicando 99,94% da variabilidade dos dados.

$$\begin{aligned}
 Ron_t = & 18,31 + 0,4273 f_{za} + 1,8560 f_{zt} - 1,4681 v_c - 0,6872 l_{to} - 0,7311 l_b - 1,1419 Q - 0,0461 \\
 & f_{za}^2 - 0,1032 f_{zt}^2 - 0,3489 v_c^2 + 0,6234 f_{za} f_{zt} + 0,8209 f_{za} v_c + 0,1785 f_{za} l_{to} + 0,2313 f_{za} l_b - \\
 & 0,6101 f_{za} Q - 0,5560 f_{zt} v_c + 0,4829 f_{zt} l_{to} + 0,6195 f_{zt} l_b + 0,0853 f_{zt} Q - 0,9989 v_c l_{to} - 0,9213 \\
 & f_{zt} l_b - 0,5586 v_c Q
 \end{aligned} \tag{5.3}$$

Os gráficos dos efeitos principais de Ron_t são mostrados na Figura 5.7. Observou-se que a faixa de variação do avanço por dente na direção axial não foi significativa para a circularidade. Considerando os efeitos significativos, em relação às variáveis de processo pode-se observar um efeito linear positivo de f_{zt} e aumento nas respostas de circularidade com o

aumento de v_c . Já em relação às variáveis de ruído significativas, observa-se um efeito linear negativo em l_{to} e l_b . Uma possível explicação para o efeito de l_b é que na medida da saída do furo tem-se um maior comprimento de contato da fresa com o furo e assim, menor valor do desvio de circularidade. Apesar de não ter sido observado significância nas interações processo×ruído, os termos são considerados no modelo de resposta de forma a possibilitar a propagação do erro e o estudo da robustez do processo de forma a minimizar a variância relativa das variáveis de ruído consideradas, mesmo que essa seja mínima.

Os gráficos de interações de Ron_t são apresentados na Figura 5.8. Onde observa-se pontos de robustez para circularidade nas interações das variáveis de controle e ruído $f_{za} \times Q$, $v_c \times l_{to}$ e $v_c \times l_b$. O ponto de f_{za} igual a $0,14 \mu\text{m/dente}$ é robusto, pois negligencia o efeito da vazão de fluido na circularidade do furo. Já para a velocidade de corte, a robustez do processo acontece em baixos valores baixos desta variável próximos de 37 m/min , onde o comprimento de balanço da fresa e a altura de medição no furo não influenciam a circularidade. No entanto, para avanços tangenciais acima de $0,22 \text{ mm/dente}$, estes ruídos também não interferem na circularidade. E por fim, em velocidade de corte de 50 m/min há robustez da circularidade em função da variação da vazão de fluido.

Neste trabalho também são apresentados os modelos de média e variância de Ron_t . Para um melhor entendimento e interpretação dos modelos são apresentados gráficos de contorno para as funções de média e desvio-padrão de Ron_t , conforme a Figura 5.9. Onde observa-se menores valores de circularidade em baixas velocidades de corte e avanço tangencial por dente da fresa. Por outro lado, observa-se uma correlação positiva de f_{za} e f_{zt} com o desvio-padrão de Ron_t , ou seja, há uma variância relativa maior das variáveis de ruído na circularidade. Em relação a v_c , os níveis mais baixos deste parâmetro garantem uma menor variância relativa das variáveis de ruído na circularidade.

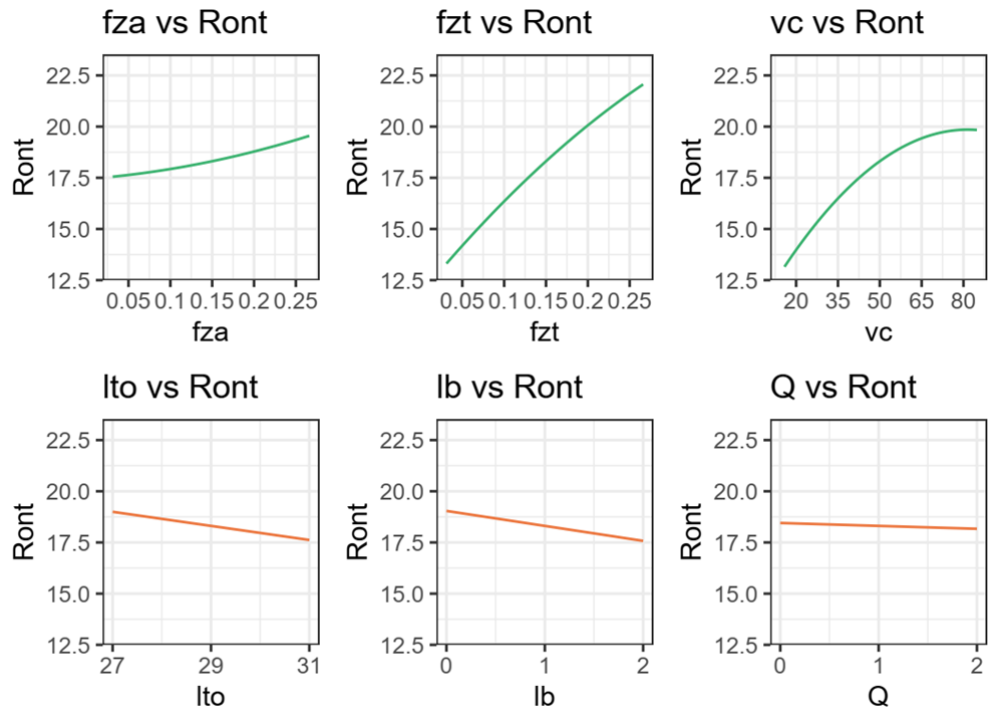


Figura 5.7. Gráfico de efeitos principais das respostas da circularidade total $Ront_t$

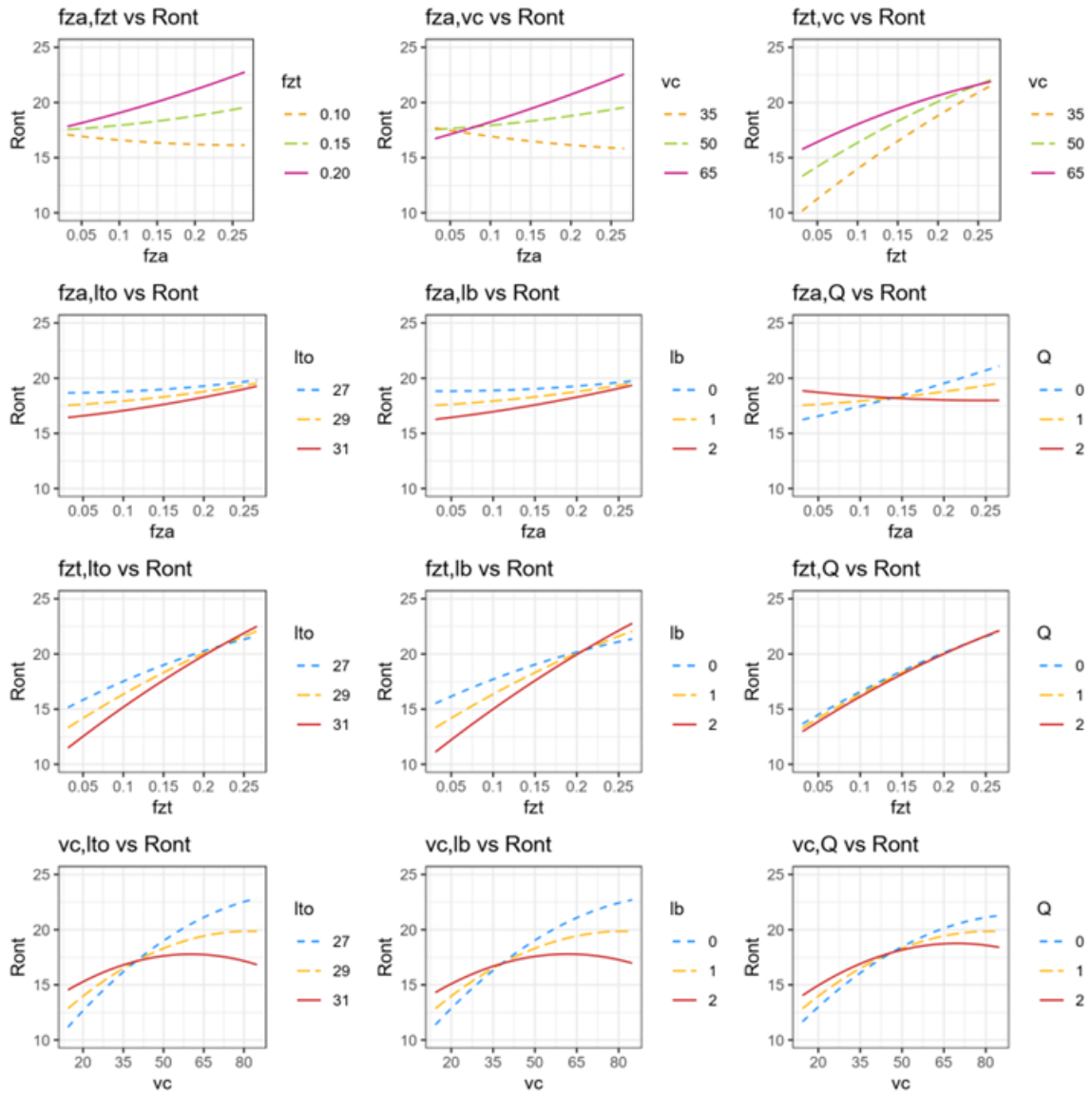


Figura 5.8. Gráficos de interações entre as variáveis de controle e ruído

$$\boldsymbol{\beta}_{Ront} = [0.4273 \quad 1.8560 \quad 1.4681]^T$$

$$\mathbf{B}_{Ront} = \begin{bmatrix} 0.0461 & 0.3117 & 0.4105 \\ 0.3117 & -0.1032 & -0.2780 \\ 0.4105 & -0.2780 & -0.3489 \end{bmatrix}$$

$$\boldsymbol{\gamma}_{Ront} = [-0.6872 \quad -0.7311 \quad -0.1419]^T$$

$$\boldsymbol{\Delta}_{Ront} = \begin{bmatrix} 0.1785 & 0.2313 & -0.6101 \\ 0.4829 & 0.6195 & 0.0853 \\ -0.9989 & -0.9213 & -0.5586 \end{bmatrix}$$

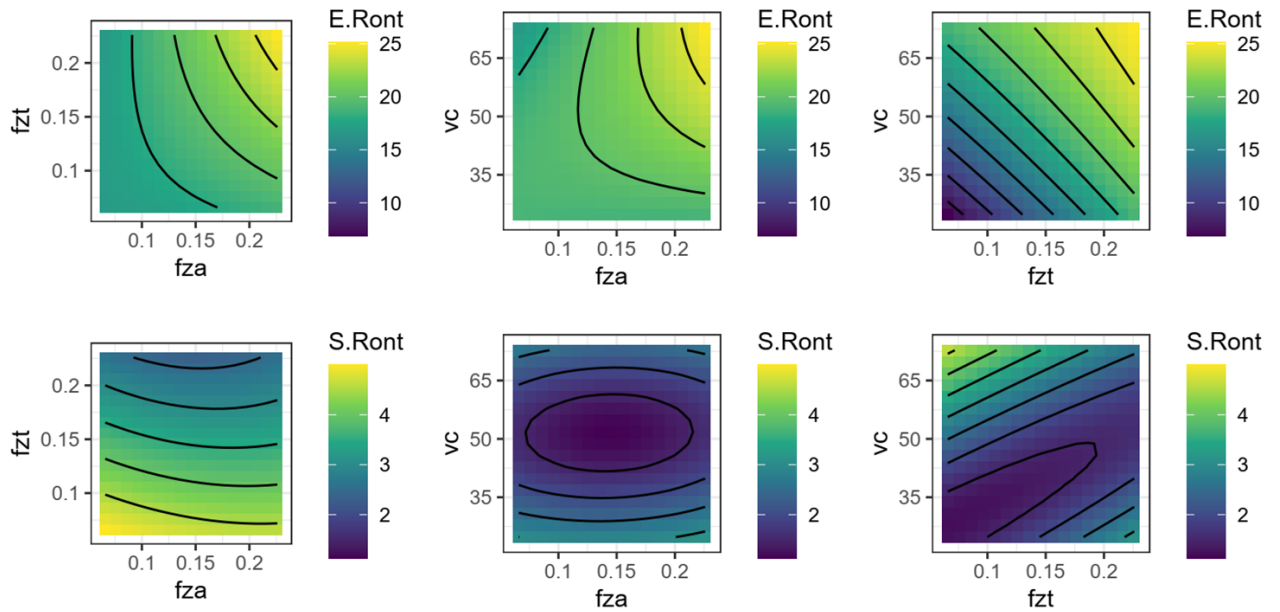


Figura 5.9. Gráficos de Contorno para média (E) e Desvio-padrão (S) de Ron_t

5.5 Otimização evolutiva robusta multiobjetivo através do MOPSO-CD e NSGA-II

A otimização evolutiva robusta multiobjetivo foi realizada neste trabalho através dos métodos MOPSO-CD e NSGA-II. A análise multiobjetivo considerou a qualidade probabilística robusta medida através do erro quadrático médio das respostas de R_a , F_a e Ron_t e uma função de probabilidade determinística que considera a taxa de remoção de material (MRR). Os métodos forneceram um número satisfatório de soluções próximo das 200 seleções desejadas. A otimização multiobjetivo é empregada para entregar soluções ótimas de Pareto que gerenciem o *trade-off* entre as quatro funções objetivo.

A otimização da resposta determinística MRR foi realizada com a finalidade de avaliar e otimizar a produtividade do processo. A otimização robusta multiobjetivo para discutir o *trade-off* entre os resultados de R_a , F_a e Ron_t e MRR foi realizada através dos métodos NSGA-II e MOPSO-CD.

A Figura 5.10 mostra a fronteira de Pareto para o problema de otimização multiobjetivo das respostas obtida pelo método NSGA-II. Nesta fronteira de Pareto é explorado o *trade-off* do erro quadrático médio de rugosidade (R_a), força axial de avanço (F_a) desvios de circularidade (Ron_t) e produtividade (MRR), considerando viés e variância. Como pode ser observado, a abordagem de escalarização apresenta uma região de solução linear limitada. Para melhor discussão dos resultados, os níveis dos parâmetros são apresentados em escala codificada e

original e são apresentadas as soluções ótimas de Pareto robustas, plotadas em diferentes cores na escala. Como pode ser verificado na avaliação dos pesos e os níveis dos parâmetros, através da minimização dos modelos de MSE, é possível alcançar um bom equilíbrio entre a aproximação média do alvo e a minimização da variância, garantindo a minimização do viés e a robustez do processo. Desta forma, os resultados possibilitam a decisão de uma solução adequada para situações específicas de planejamento.

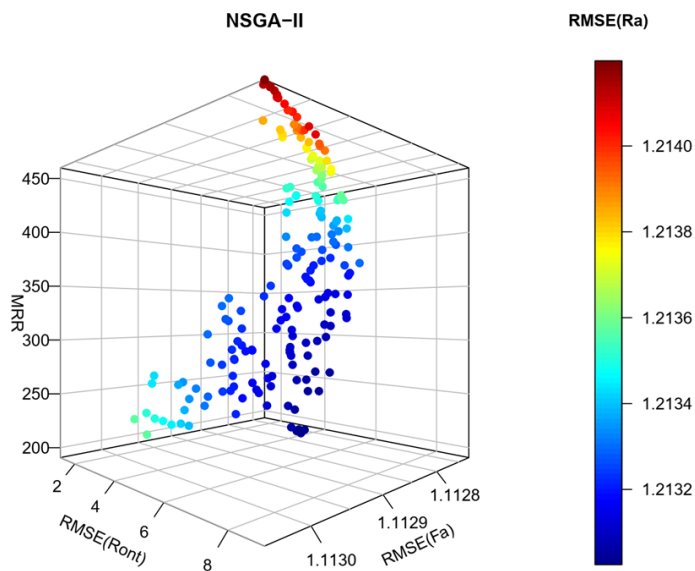


Figura 5.10. Fronteira de Pareto para a otimização de quatro objetivos

A Figura 5.11 mostra a fronteira de Pareto para o problema de otimização obtido pelo método MOPSO-CD. Pode ser verificado a semelhança representada referente ao NSGA-II, com uma região de solução linear limitada e pontos ótimos mais espaçados, apresentando resultados competitivos. Geralmente o MOPSO-CD supera outros métodos comumente utilizados considerando o número de soluções não dominadas. O método MOPSO-CD também foi desenvolvido como estratégia para a resolução do problema multiobjetivo para aumentar a diversidade populacional e para fins de comparação. Na análise da fronteira de Pareto pode-se observar que o MOPSO-CD explora melhor o espaço multiobjetivo. Apesar das fronteiras

apresentarem aspecto semelhante, uma exploração mais extensa da fronteira de Pareto foi obtida através do método MOPSO-CD.

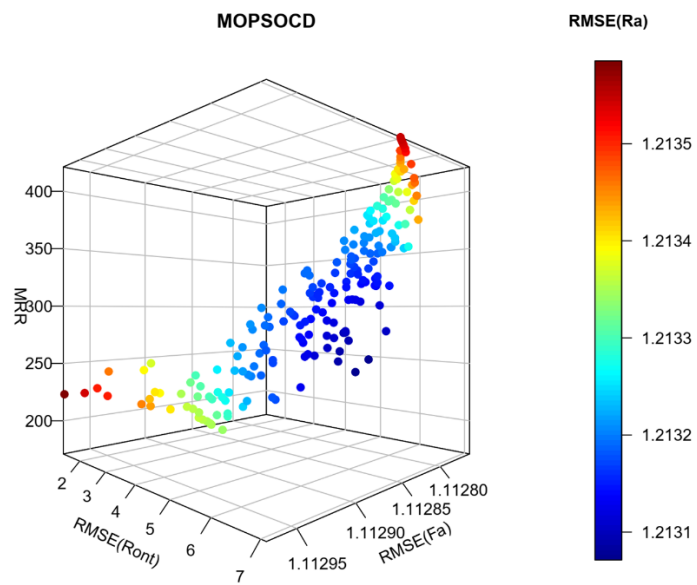


Figura 5.11. Fronteira de Pareto para a otimização de quatro objetivos

6. CONCLUSÕES

A presente pesquisa apresentou os modelos de rugosidade, força de avanço e circularidade considerando a variação do avanço tangencial, avanço axial, velocidade de corte, comprimento em balanço da fresa, altura de medição do furo e vazão do fluido de corte no processo de fresamento helicoidal do aço inoxidável super duplex UNS S32760. A combinação de fatores de processo e ruído do projeto permitiram a modelagem e otimização robusta dos resultados apresentados.

Os valores de rugosidade média R_a medidos variaram entre 0,0982 e 0,2838 μm , valores estes considerados bastante satisfatórios para acabamento de furos neste material de baixa usinabilidade. Através da redução do modelo por mínimos quadrados ponderados foi obtido um coeficiente de determinação ajustado R^2_{aj} de 95,21%, com boa capacidade de ajuste e previsão do modelo. Os valores da força axial F_a variaram entre 73,7 a 86,4 N, com coeficiente de determinação ajustado R^2_{aj} de 95,98%. Valores relativamente baixos comparados às elevadas forças de avanço do processo convencional de furação de aços inoxidáveis super duplex. Em relação aos desvios de circularidade dos furos, os valores de Ron_t apresentaram variação de 10,79 a 32,48 μm e o modelo de previsão ajuste R^2_{aj} de 99,94%, sendo obtidos modelos com ótima capacidade de previsão para avaliação dos desvios de circularidade.

O projeto de parâmetro robusto do processo de fresamento helicoidal do aço inoxidável super duplex UNS S32760 possibilitou o entendimento e avaliação da interação entre as variáveis de processo. Considerando os efeitos significativos, em relação às variáveis de processo pode-se observar um efeito linear positivo de f_{zt} na resposta R_a . Em relação à resposta de força axial F_a , os níveis de avanço foram estatisticamente significativos na análise dos fatores que influenciam esta resposta. A faixa de variação do avanço por dente na direção axial não foi significativa para a análise de Ron_t . Foram obtidas equações dos modelos robustos considerando o viés e variância da função nas variáveis do processo, em que se observou vários pontos de robustez onde as variáveis de ruído, comprimento de balanço da fresa, altura de medição do furo e vazão de fluido não influenciaram as respostas rugosidade R_a , força de avanço e o desvio de circularidade.

A otimização robusta multiobjetivo para discutir o *trade-off* entre os resultados de R_a , F_a e Ron_t e MRR foi realizada e discutida através dos métodos NSGA-II e MOPSO-CD. As soluções ótimas de Pareto permitiram a avaliação dos níveis dos fatores ótimos de processo e

entendimento do *trade-off* entre as respostas em termos de variação dos níveis dos fatores de processo. Apesar dos gráficos apresentarem aspecto semelhante, na avaliação da disposição das soluções na fronteira, observa-se que o MOPSO-CD apresenta um alcance melhor na região de solução em relação ao NSGA-II, com uma exploração mais extensa da fronteira de Pareto sendo obtida através do método MOPSO-CD.

6.1 Sugestões para trabalhos futuros

Como sugestão para o desenvolvimento de trabalhos futuros poderiam ser verificadas as possibilidades relacionadas ao processo de fresamento helicoidal e aos métodos utilizados. Algumas sugestões são apresentadas a seguir:

- Estudar o processo de fresamento helicoidal do aço inoxidável super duplex UNS S32760 a altas velocidades de corte;
- Abordar o fresamento helicoidal em outros aços inoxidáveis;
- Empregar métodos de otimização com definição ótima para os pesos atribuídos a cada resposta de interesse;
- Utilizar outros métodos de otimização de modo que seja realizada a comparação entre eles.

REFERÊNCIAS

- AGOSTINHO, O. L., RODRIGUES. A. C. S., LIRANI, J. **Tolerâncias, Ajustes, Desvios e Análise de Dimensões**. Editora Edgard Blucher. S Paulo, 2004.
- ALTAN, T.; LILLY, B.; YEN, Y. C. **Manufacturing of dies and molds**. CIRP Annals-Manufacturing Technology, v. 50, n. 2, p. 404-422, 2001.
- AMORIM, H. J. **Estudo da relação entre velocidade de corte, desgaste de ferramenta, rugosidade e forças de usinagem no torneamento com ferramenta de metal duro**. Dissertação de Mestrado. Universidade Federal do Rio Grande do Sul. Porto Alegre, p. 114. 2002.
- ALMEIDA, D O. **Investigação de desvios geométricos no alargamento de ferro fundido com ferramentas revestidas**. 2008. 100p. Dissertação de Mestrado – Universidade Federal de Uberlândia, Uberlândia.
- ALMEIDA, F. A.; GOMES, G. F; PAULA, V. R.; **A weighted mean square error approach to the robust optimization of the surface roughness in na AISI 12L14 free-machining steel-turning process**. Journal of Mechanical Engineering, v. 64, p. 147-156, 2018.
- ALVIM, A. C. **Otimização robusta do fresamento helicoidal em altas velocidades do aço ABNT H13 endurecido**. 2019. 131p. Dissertação de Mestrado – Universidade Federal de Itajubá, Itajubá.
- BOEHS, L. **Análise de superfície usinada**. Apostila do curso de Pós-graduação em Engenharia Mecânica. Universidade Federal de Santa Catarina. Florianópolis. 1999.
- BOX, G. E. P.; DRAPER, N. R. **Response surfaces, mixtures, and ridge analyses**. New Jersey: John Wiley & Sons, 2007.
- BOX, G. E. P.; WILSON, K. B. **On the Experimental Attainment of Optimum Conditions**. Journal of the Royal Statistical Society. Series B (Methodological), v. 13, n. 1, p. 1-45, 1951.
- BOX, G. E. P.; JONES, S. **Designing products that are robust to the environment**. Total Quality Management, v. 3, n. 3, p. 265-282, 1992.

- BRINKSMEIER, E.; FANGMANN, S.; MEYER, I. **Orbital drilling kinematics**. Production engineering, v. 2, n. 3, p. 277-283, 2008.
- CARBÓ, H. M. **Aços inoxidáveis: aplicações e especificações**. São Paulo: Arcelor Mittal Inox Brasil, 2008. 29 p.
- COELLO, C.; LECHUGA, M. S.; **MOPSO: A proposal for multiple objective particle swarm optimization**, IEEE, v. 2, p. 1051-1056, 2002.
- COSTA, D. D. Da; MARQUES, A.; AMORIM, F. L. **Hole quality and cutting time evaluation in the interpolated helical milling**. International Journal of Manufacturing Research, v. 10, n. 4, p. 313-327, 2015.
- DAMIN, K. V. S.; GENEROSO, D. J.; NETO, F. C. **Aços inoxidáveis duplex: uma breve visão**. Revista Liberato, v. 18, n. 29, p. 01-132, 2017.
- DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. **A fast and Elitist Multiobjective Genetic Algorithm: NSGA-II**, IEEE Transactions on Evolutionary Computation, v. 6, n. 2, 2002.
- DEB, K.; SINDHYA, K.; HAKANEN, J. **Multi-Objective Optimization**. In: SENGUPTA, R. N., GUPTA, A. DUTTA, J. (Org.) Decision Sciences: Theory and Practice. CRC PRESS, 2016.
- DENKENA, B.; BOEHNKE, D.; DEGE, J. H. **Helical milling of CFRP–titanium layer compounds**. CIRP Journal of Manufacturing Science and Technology, v. 1, n. 2, p. 64–69, jan. 2008.
- DENKENA, B.; NESPOR, D.; REHE, M., DEGE, J. H. **Process force prediction in orbital drilling of TiAl6V4**. In: 9th International Conference on Advanced Manufacturing Systems and Technology. 2011. p. 16-17.
- EGUTI, C. C. A.; TRABASSO, L. G. **Design of a robotic orbital driller for assembling aircraft structures**. Mechatronics, v. 24, n. 5, p. 533-545, 2014.
- FANG, Q.; PAN, Z.; FEI, S.; XIE, X.; KE, Y. **A Novel Helical Milling End-Effector and Its Application**. IEEE/ASME Transactions on Mechatronics, v. 20, n. 6, p. 3112-3122, 2015a.

- GAMARRA, J. R.; DINIZ, A. E. **Taper turning of super duplex stainless steel: tool life, tool wear and workpiece surface roughness**. Journal of the Brazilian Society of Mechanical Sciences and Engineering, v. 40:39, p. 1-13, 2018. <https://doi.org/10.1007/s40430-018-0991-1>.
- GIL, A. C. **Métodos e Técnicas de Pesquisa Social**. 6ª Edição, São Paulo: Atlas, 2008. 220 p.
- HAIYAN, W.; XUDA, Q. **A mechanistic model for cutting force in helical milling of carbon fiber-reinforced polymers**. The International Journal of Advanced Manufacturing Technology, v. 82, n. 9-12, p. 1485-1494, 2016.
- HE, G.; LI, H.; JIANG, Y.; QIN, X.; ZHANG, X.; GUAN, Y. **Helical milling of CFRP/Ti-6Al-4V stacks with varying machining parameters**. Transactions of Tianjin University, v. 21, n. 1, p. 56-63, 2015.
- HEBBALI, A., 2020. **“olsrr: Tools for Building OLS Regression Models”**. *R package version 0.5.3*. Jun. 2020 <<https://CRAN.R-project.org/package=olsrr>>.
- IYER, R.; KOSHY, P.; NG, E. **Helical milling: an enabling technology for hard machining precision holes in AISI D2 tool steel**. International Journal of Machine Tools and Manufacture, v. 47, n. 2, p. 205-210, 2007.
- JI, C., LI, Y.; QIN, X.; ZHAO, Q.; SUN, D.; JIN, Y. **3D FEM simulation of helical milling hole process for titanium alloy Ti-6Al-4V**. The International Journal of Advanced Manufacturing Technology, v. 81, n. 9-12, p. 1733-1742, 2015.
- KASSAMBARA, A., 2020. **“ggpubr: 'ggplot2' Based Publication Ready Plots”**. *R package version 0.4.0*. Jun. 2020 <<https://CRAN.R-project.org/package=ggpubr>>.
- KOYEE, R. D., HEISEL, U., EISSELER, R., & SCHMAUDER, S. (2014). **Modeling and optimization of turning duplex stainless steels**. Journal of Manufacturing Processes, 16(4), 451-467.
- KORKUT, I.; KASAP, M.; CIFTICI, I.; SEKER, U. **Determination of optimum cutting parameters during machining of AISI 304 austenitic stainless steel**. *Materials & Design*, v.25, p.303-305, 2004.
- KROLCZYK, G.; LEGUTKO, S.; STOIC, A. **Influence of cutting parameters and conditions onto surface hardness of duplex stainless steel after turning process**, 2013.

- KURT, M; BAGCI, E. **Evaluation of drilled hole quality in Al 2024 alloy**. International Journal of Advanced Manufacturing Technology, v. 37, p. 1051–1060, 2008.
- LENTH, R.V., 2009. “**Response-Surface Methods in R, Using rsm**”. *Journal of Statistical Software*, 32(7), 1-17. Jun. 2020 <<http://www.jstatsoft.org/v32/i07/>>.
- LI, Z.; LIU, Q.; PENG, C.; SUN, X. **Cutting force modeling and simulation for hole-making process by helical milling**. In: Fourth International Seminar on Modern Cutting and Measuring Engineering. International Society for Optics and Photonics, 2010. p. 799703-799703-6.
- LI, Z.; LIU, Q. **Surface topography and roughness in hole-making by helical milling**. The International Journal of Advanced Manufacturing Technology, v. 66, n. 9-12, p. 1415-1425, 2013.
- LI, H. et al. **Tool wear and hole quality investigation in dry helical milling of Ti-6Al-4V alloy**. International Journal of Advanced Manufacturing Technology, v. 71, n. 5–8, p. 1511–1523, 2014.
- LIU, C.; WANG, G.; DARGUSCH, M. S. **Modelling, simulation and experimental investigation of cutting forces during helical milling operations**. The International Journal of Advanced Manufacturing Technology, v. 63, n. 9, p. 839-850, 2012.
- MACHADO, A. R., COELHO, R.T., ABRÃO, A.M.; DA SILVA, M.B. **Teoria da Usinagem dos Materiais**. Ed. Edgard Blücher, São Paulo, p. 384, 2009.
- MARTINHO, R. P. et al. *Comparative study of PVD and CVD cutting tools performance in milling of duplex stainless steel*. *International Journal of Advanced Manufacturing Technology*, 2019.
- MERSMANN, O., 2014. “**mco: Multiple Criteria Optimization Algorithms and Related Functions**”. *R package version 1.0-15.1*. Jun. 2020 <<https://CRAN.R-project.org/package=mco>>.
- MERSMANN, O., 2020. “**emoa: Evolutionary Multiobjective Optimization Algorithms**”. *R package version 0.5-0.1*. Jun. 2020 <<https://CRAN.R-project.org/package=emoa>>.
- MYERS, R. H.; KHURI, A. I.; VINING, G. **Response surface alternatives to the Taguchi robust parameter design approach**. The American Statistician, v. 46, n. 2, p. 131-139, 1992.

- MYERS, R. H.; MONTGOMERY, D. C.; CHRISTINE, M.; ANDERSON-COOK, C. M. **Response Surface Methodology: Process and Product Optimization Using Designed Experiments**. John Wiley & Sons, 2009.
- MYERS, R. H.; MONTGOMERY, D. C.; ANDERSON-COOK, C. M. **Response surface methodology: process and product optimization using designed experiments**. John Wiley & Sons, 2016.
- MONTGOMERY, D. C. **Design and Analysis of Experiments**. New Jersey: John Wiley & Sons, 2005, 643p.
- MONTGOMERY, D. C. **Design and Analysis of Experiments**. 8th ed. United States of America: John Wiley & Sons, 2013.
- NAVAL, P., 2013. “**Multi-objective Particle Swarm Optimization with Crowding Distance**”. *R package version 0.5.1*. Jun. 2020 <<https://CRAN.R-project.org/package=mopsocd>>.
- NILSSON, J. O.; **Super duplex stainless steel “overview”**. *Materials Science and Technology*, v.8, 1992.
- NOMANI, J. et al. **Chip formation mechanism and machinability of wrought duplex stainless steel alloys**. *International Journal of Advanced Manufacturing Technology*, v. 80, n. 5–8, p. 1127–1135, 2015.
- NOMANI, J., PRAMANIK, A., HILDITCH, T. E LITTLEFAIR, G., 2017. “**Stagnation zone during the turning of Duplex SAF 2205 stainless steels alloy**”. *Materials and Manufacturing Processes*, Vol. 32, p. 1486-1489.
- OLIVEIRA Jr., C. A. **Torneamento do aço inoxidável super duplex UNS S32750 e influência na resistência à corrosão**. Dissertação (Mestrado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, p. 18, 2013.
- OLVERA, D.; DE LACALLE, L. N. L.; URBIKAIN, G.; LAMIKIZ, A.; RODAL, P., & ZAMAKONA, I. **Hole making using ball helical milling on titanium alloys**. *Machining Science and Technology*, v. 16, n. 2, p. 173-188, 2012.
- PEREIRA, R. B. D.; BRANDÃO, L. C.; PAIVA, A. P.; FERREIRA, J. R.; DAVIM, J. P. **A review of helical milling process**. *International Journal of Machine Tools and Manufacture*, v. 120, 2017.

- PEREIRA, R. B. D. **Otimização robusta multi-objetivo do processo de fresamento helicoidal do aço AISI H13 endurecido pelo método da restrição normal**. 2017. 321p. Tese de Doutorado – Universidade Federal de Itajubá, Itajubá.
- PETROPOULOS, G. P.; PANDAZARAS, C. N.; DAVIM, J. P. **Surface Texture Characterization and Evaluation Related to Machining**. Em: Davim J. (ed.), *Surface Integrity in Machining*. Springer, London, 2010.
- POLICENA, M. R., DEVITTE, C., FRONZA, G., GARCIA, R. F. E SOUZA, A. J., 2018. **“Surface Roughness Analysis in Finishing End-Milling of Duplex Stainless Steel UNS S32205”**. *International Journal of Advanced Manufacturing Technology*, Vol. 98, p. 1617-1625.
- QIN XD, HUA S, JI XL, LIU WC CHEN SM (2010) **Research on the surface roughness model for helical milling of die-steel based on response surface methodology**, *Key Eng Mater* 431-432:346-350.
- QIN, X.; ZHANG, X.; LI, H.; RONG, B.; WANG, D.; ZHANG, H.; ZUO, G. **Comparative analyses on tool wear in helical milling of Ti-6Al-4V using diamond-coated tool and TiAlN-coated tool**. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, v. 8, n. 1, p. 1-14, 2014.
- R CORE TEAM, 2020. **“R: A language and environment for statistical computing”**. *R Foundation for Statistical Computing*, Vienna, Austria. Jun. 2020 <<https://www.R-project.org/>>.
- RAQUEL, C. E NAVAL, P., 2005. **“An effective use of crowding distance in multiobjective particle swarm optimization”**. *GECCO 2005 - Genet Evol Comput Conf*, 257–264. Jul. 2020 <<https://doi.org/10.1145/1068009.1068047>>.
- REY, P. A.; LEDREF, J.; SENATORE, J.; LANDON, Y. **Modelling of cutting forces in orbital drilling of titanium alloy Ti-6Al-4V**. *International Journal of Machine Tools and Manufacture*, v. 106, p. 75-88, 2016.
- SMIDERLE, J. et al. **Premature failure of superduplex stainless steel pipe by pitting in sea water environment**. *Engineering Failure Analysis*, v. 46, p. 134–139, 2014. Disponível em: <<http://dx.doi.org/10.1016/j.engfailanal.2014.08.001>>.
- SHAN, Y.; HE, N.; LI, L.; ZHAO, W.; Qin, X. **Orbital milling hole of aerospace Al-alloy with big pitch**. *Transactions of Tianjin University*, v. 17, n. 5, p. 329-335, 2011.

- SHEN, J. L.; LIU, Q.; YUAN, S. M.; YANG, Y. Q. **Research on a Generalized Simulation Algorithm for Cutting Forces Prediction of Helical Milling and its Implementation.** *Advanced Materials Research*. v. 562-564, p. 793-798, 2012.
- SHOEMAKER, A. C.; TSUI, K.-L.; WU, C. F. J. **Economical experimentation methods for robust design.** *Technometrics*, v. 33, n. 4, p. 415-427, 1991.
- SOUZA, A. F. **Contribuições ao fresamento de geometrias complexas aplicando a tecnologia de usinagem com altas velocidades.** Tese de Doutorado. Escola de Engenharia de São Carlos, Universidade de São Paulo. São Carlos, p. 186. 2004.
- SUN, L.; GAO, H.; WANG, B.; BAO, Y.; WANG, M.; MA, S. **Mechanism of reduction of damage during helical milling of titanium/CFRP/aluminium stacks.** *The International Journal of Advanced Manufacturing Technology*, v.107, p.4741-4753, 2020.
- TAGUCHI, G. **Introduction to Quality Engineering: Designing Quality into Products and Process.** Tokyo, Japan: Asian Productivity Organization, 1986.
- TAGUCHI, G., CHOWDHURY, S., and TAGUCHI, S., 2000. **Robust engineering – learn how to boost quality while reducing costs and time to market.** New York: McGraw-Hill.
- TAVARES, S. S. M. et al. **Failure of superduplex stainless steel flange due to inadequate microstructure and fabrication process.** *Engineering Failure Analysis*, v. 84, n. October 2017, p. 1-10, 2018
- TONSHOFF, H. K.; SPINTIG, W.; KÖNIG, W.; NEISES, A. **Machining of holes developments in drilling technology.** *CIRP Annals-Manufacturing Technology*, v. 43, n. 2, p. 551-561, 1994.
- TSOU, C., 2015 **“Elitist Non-dominated Sorting Genetic Algorithm” based on R. R package version 1.0.**
- VENTURA, C. E. H.; HASSUI, A. **Modeling of cutting forces in helical milling by analysis of tool contact angle and respective depths of cut.** *International Journal Of Advanced Manufacturing Technology*, v. 68, 2013.
- VINING, G.; MYERS, R. **Combining Taguchi and response surface philosophies- A dual response approach.** *Journal of quality technology*, v. 22, p. 38-45, 1990.

- WANG, H.; QIN, X.; REN, C.; WANG, Q. **Prediction of cutting forces in helical milling process.** *International Journal of Advanced Manufacturing Technology*, v. 58, p. 849-859, 2012.
- WANG, H.; QIN, X.; LI, H. **Machinability analysis on helical milling of carbon fiber reinforced polymer.** *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, v. 9, n. 5, p. 1-11, 2015.
- WANG, C.; ZHAO, J.; ZHOU, Y. **Mechanics and dynamics study of helical milling process for nickel-based superalloy.** *The International Journal of Advanced Manufacturing Technology*, v.106, p.2305-2316, 2019.
- WELCH, W.; YU, T.; KANG, S. M.; SACKS, J. **Computer experiments for quality control by parameter design.** *Journal of Quality Technology*, v. 22, n. 1, p. 15-22, 1990
- WICKHAM, H., 2016. **“ggplot2: Elegant Graphics for Data Analysis”.** *Springer-Verlag New York.*

APÊNDICE A. Código para modelagem robusta da rugosidade R_a

```

library(rsm)

library(olsrr)

library(ggplot2)

library(ggpubr)

library(NlcOptim)

plan.ccd <- ccd(~x1+x2+x3+z1+z2,

               generators = z3~x1*x2*x3*z1*z2,

               n0 = c(0,9),

               alpha = "rotatable",

               randomize = F,

               oneblock = T,

               coding = list(x1~(fza-0.15)/0.05,

                             x2~(fzt-0.15)/0.05,

                             x3~(vc-50)/15,

                             z1~(lto-29)/2,

                             z2~(lb-1)/1,

                             z3~(Q-1)/1))

plan.ccd <- rbind(plan.ccd[1:38,],plan.ccd[45:53,])

plan.ccd

setwd("C:\\\\...")

respostas <- read.csv("resultados.csv", header = T)

Ra <- respostas$Ra

plan.ccd$Ra <- Ra

plan.ccd$Ra <- Ra

plan.ccd

```

```

res.Ra <- lm(formula = Ra ~ x1+x2+x3+z1+z2+z3 +
             x1*x2+x1*x3+x1*z1+x1*z2+x1*z3+x2*x3+x2*z1+x2*z2+x2*z3+x3*z1+x3*z2+x3*z3 +
             + I(x1^2) + I(x2^2) + I(x3^2),
             data = plan.ccd)

summary(res.Ra)

rsm.Ra <- rsm(formula = Ra ~ FO(x1,x2,x3,z1,z2,z3) +
              TWI(x1,x2) + TWI(x1,x3) + TWI(x1,z1) + TWI(x1,z2) + TWI(x1,z3) + TWI(x2,x3) +
              TWI(x2,z1) + TWI(x2,z2) + TWI(x2,z3) + TWI(x3,z1) + TWI(x3,z2) + TWI(x3,z3) +
              + PQ(x1,x2,x3),
              data = plan.ccd)

summary(rsm.Ra)

shapiro.test(res.Ra$residuals)

ols_test_breusch_pagan(res.Ra, rhs = T, multiple = T)

par(mfrow = c(2, 2))

plot(res.Ra)

par(mfrow = c(1, 1))

w.Ra <- 1/res.Ra$residuals^2

res.Ra.w <- lm(formula = Ra ~ x1+x2+x3+z1+z2+z3 +
              x1*x2+x1*x3+x1*z1+x1*z2+x1*z3+x2*x3+x2*z1+x2*z2+x2*z3+x3*z1+x3*z2+x3*z3 +
              + I(x1^2) + I(x2^2) + I(x3^2),
              data = plan.ccd,
              weights = w.Ra)

summary(res.Ra.w)

rsm.Ra.w <- rsm(formula = Ra ~ FO(x1,x2,x3,z1,z2,z3) +
                TWI(x1,x2) + TWI(x1,x3) + TWI(x1,z1) + TWI(x1,z2) + TWI(x1,z3) + TWI(x2,x3) +
                TWI(x2,z1) + TWI(x2,z2) + TWI(x2,z3) + TWI(x3,z1) + TWI(x3,z2) + TWI(x3,z3) +

```

```

+ PQ(x1,x2,x3),

data = plan.ccd,

weights = w.Ra)

summary(rsm.Ra.w)

shapiro.test(res.Ra.w$residuals)

ols_test_breusch_pagan(res.Ra.w, rhs = T, multiple = T)

par(mfrow = c(2, 2))

plot(res.Ra.w)

par(mfrow = c(1, 1))

x1_grid <- seq(min(plan.ccd$x1), max(plan.ccd$x1), 0.1)

pp1 <- ggplot() +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,

                                                                    x2 = 0,

                                                                    x3 = 0,

                                                                    z1 = 0,

                                                                    z2 = 0,

                                                                    z3 = 0))))),

  colour = 'mediumseagreen') +

  ggtitle('fza vs Ra') +

  xlab('fza') +

  ylab('Ra') +

  ylim(.12,.28) +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  theme_bw()

pp2 <- ggplot() +

```

```

geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))))),
          colour = 'mediumseagreen') +
ggtitle('fzt vs Ra') +
xlab('fzt') +
ylab('Ra') +
ylim(.12,.28) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp3 <- ggplot() +
geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = x1_grid,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))))),
          colour = 'mediumseagreen') +
ggtitle('vc vs Ra') +
xlab('vc') +
ylab('Ra') +
ylim(.12,.28) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

```

```

theme_bw()

z1_grid <- seq(min(plan.ccd$z1), max(plan.ccd$z1), 0.1)

pp4 <- ggplot() +

  geom_line(aes(x = z1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,

                                                                    x2 = 0,

                                                                    x3 = 0,

                                                                    z1 = z1_grid,

                                                                    z2 = 0,

                                                                    z3 = 0))))),

            colour = 'sienna2') +

  ggtitle('lto vs Ra') +

  xlab('lto') +

  ylab('Ra') +

  ylim(.12,.28) +

  scale_x_continuous(breaks = c(-1, 0, 1), label = c(27, 29, 31)) +

  theme_bw()

pp5 <- ggplot() +

  geom_line(aes(x = z1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,

                                                                    x2 = 0,

                                                                    x3 = 0,

                                                                    z1 = 0,

                                                                    z2 = z1_grid,

                                                                    z3 = 0))))),

            colour = 'sienna2') +

  ggtitle('lb vs Ra') +

  xlab('lb') +

```



```

colour = '0.15', linetype = '0.15')) +

  ggtitle('fza,fzt vs Ra') +

  xlab('fza') +

  ylab('Ra') +

  ylim(.10,30) +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_12 <- pp_12a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,

                                                                    x2 = -1,

                                                                    x3 = 0,

                                                                    z1 = 0,

                                                                    z2 = 0,

                                                                    z3 = 0))),

              colour = '0.10', linetype = '0.10')) +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,

                                                                    x2 = 1,

                                                                    x3 = 0,

                                                                    z1 = 0,

                                                                    z2 = 0,

                                                                    z3 = 0))),

              colour = '0.20', linetype = '0.20')) +

  scale_color_manual(name = "fzt",

                    values = c("0.10" = "orange2",

```

```

    "0.15" = "olivedrab3",
    "0.20" = "mediumvioletred")) +
scale_linetype_manual(name = "fzt",
    values = c("0.10" = "dashed",
    "0.15" = "longdash",
    "0.20" = "solid"))
pp_13a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
    colour = '50', linetype = '50')) +
  ggtitle('fza,vc vs Ra') +
  xlab('fza') +
  ylab('Ra') +
  ylim(.10,30) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()
pp_13 <- pp_13a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = -1,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
    colour = '50', linetype = '50')) +
  ggtitle('fza,vc vs Ra') +
  xlab('fza') +
  ylab('Ra') +
  ylim(.10,30) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()

```

```

                                z3 = 0))),
    colour = '35', linetype = '35')) +
geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 1,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '65', linetype = '65')) +
scale_color_manual(name = "vc",
    values = c("35" = "orange2",
               "50" = "olivedrab3",
               "65" = "mediumvioletred")) +
scale_linetype_manual(name = "vc",
    values = c("35" = "dashed",
               "50" = "longdash",
               "65" = "solid"))
pp_23a <- ggplot() +
    geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '50', linetype = '50')) +

```

```

ggtitle('fzt,vc vs Ra') +

xlab('fzt') +

ylab('Ra') +

ylim(.10,.30) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_23 <- pp_23a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = -1,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
                colour = '35', linetype = '35')) +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 1,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
                colour = '65', linetype = '65')) +

  scale_color_manual(name = "vc",
                     values = c("35" = "orange2",
                                "50" = "olivedrab3",
                                "65" = "mediumvioletred")) +

  scale_linetype_manual(name = "vc",

```

```

        values = c("35" = "dashed",
                  "50" = "longdash",
                  "65" = "solid"))

pp_14a <- ggplot() +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '29', linetype = '29')) +

  ggtitle('fza,lto vs Ra') +

  xlab('fza') +

  ylab('Ra') +

  ylim(.10,30) +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  theme_bw()

pp_14 <- pp_14a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = -1,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '27', linetype = '27')) +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,

```

```

x2 = 0,
x3 = 0,
z1 = 1,
z2 = 0,
z3 = 0))),
  colour = '31', linetype = '31')) +
  scale_color_manual(name = "lto",
    values = c("27" = "dodgerblue",
               "29" = "darkgoldenrod1",
               "31" = "firebrick3")) +
  scale_linetype_manual(name = "lto",
    values = c("27" = "dashed",
               "29" = "longdash",
               "31" = "solid"))
pp_15a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
x2 = 0,
x3 = 0,
z1 = 0,
z2 = 0,
z3 = 0))),
  colour = '1', linetype = '1')) +
  ggtitle('fza,lb vs Ra') +
  xlab('fza') +
  ylab('Ra') +
  ylim(.10,.30) +

```

```

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp_15 <- pp_15a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = -1,
                                                                    z3 = 0))),
              colour = '0', linetype = '0')) +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 1,
                                                                    z3 = 0))),
              colour = '2', linetype = '2')) +
  scale_color_manual(name = "lb",
                    values = c("0" = "dodgerblue",
                              "1" = "darkgoldenrod1",
                              "2" = "firebrick3")) +
  scale_linetype_manual(name = "lb",
                       values = c("0" = "dashed",
                                   "1" = "longdash",
                                   "2" = "solid"))
pp_16a <- ggplot() +

```

```

geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '1', linetype = '1')) +
ggtitle('fza,Q vs Ra') +
xlab('fza') +
ylab('Ra') +
ylim(.10,.30) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp_16 <- pp_16a +
geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = -1))),
              colour = '0', linetype = '0')) +
geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = -1))),
              colour = '0', linetype = '0')) +

```



```

                                z3 = 1))),
                                colour = '2', linetype = '2')) +
scale_color_manual(name = "Q",
                   values = c("0" = "dodgerblue",
                                "1" = "darkgoldenrod1",
                                "2" = "firebrick3")) +
scale_linetype_manual(name = "Q",
                      values = c("0" = "dashed",
                                   "1" = "longdash",
                                   "2" = "solid"))
pp_24a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
                                colour = '29', linetype = '29')) +
  ggtitle('fzt,lto vs Ra') +
  xlab('fzt') +
  ylab('Ra') +
  ylim(.10,.30) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()
pp_24 <- pp_24a +

```



```

                z2 = 0,
                z3 = 0))),
    colour = '1', linetype = '1')) +
  ggtitle('fzt,lb vs Ra') +
  xlab('fzt') +
  ylab('Ra') +
  ylim(.10,.30) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()
pp_25 <- pp_25a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                x2 = x1_grid,
                x3 = 0,
                z1 = 0,
                z2 = -1,
                z3 = 0))),
    colour = '0', linetype = '0')) +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                x2 = x1_grid,
                x3 = 0,
                z1 = 0,
                z2 = 1,
                z3 = 0))),
    colour = '2', linetype = '2')) +
  scale_color_manual(name = "lb",
    values = c("0" = "dodgerblue",

```

```

      "1" = "darkgoldenrod1",
      "2" = "firebrick3")) +
scale_linetype_manual(name = "lb",
      values = c("0" = "dashed",
      "1" = "longdash",
      "2" = "solid"))
pp_26a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
      x2 = x1_grid,
      x3 = 0,
      z1 = 0,
      z2 = 0,
      z3 = 0))),
      colour = '1', linetype = '1')) +
  ggtitle('fzt,Q vs Ra') +
  xlab('fzt') +
  ylab('Ra') +
  ylim(.10,.30) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()
pp_26 <- pp_26a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
      x2 = x1_grid,
      x3 = 0,
      z1 = 0,
      z2 = 0,

```

```

                                z3 = -1))),
    colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 1))),
    colour = '2', linetype = '2')) +
scale_color_manual(name = "Q",
    values = c("0" = "dodgerblue",
              "1" = "darkgoldenrod1",
              "2" = "firebrick3")) +
scale_linetype_manual(name = "Q",
    values = c("0" = "dashed",
              "1" = "longdash",
              "2" = "solid"))
pp_34a <- ggplot() +
    geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '29', linetype = '29')) +

```

```

ggtitle('vc,lto vs Ra') +

xlab('vc') +

ylab('Ra') +

ylim(.10,.30) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

theme_bw()

pp_34 <- pp_34a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,

                                                                    x2 = 0,

                                                                    x3 = x1_grid,

                                                                    z1 = -1,

                                                                    z2 = 0,

                                                                    z3 = 0))),

              colour = '27', linetype = '27')) +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,

                                                                    x2 = 0,

                                                                    x3 = x1_grid,

                                                                    z1 = 1,

                                                                    z2 = 0,

                                                                    z3 = 0))),

              colour = '31', linetype = '31')) +

  scale_color_manual(name = "lto",

                    values = c("27" = "dodgerblue",

                               "29" = "darkgoldenrod1",

                               "31" = "firebrick3")) +

  scale_linetype_manual(name = "lto",

```

```

        values = c("27" = "dashed",
                  "29" = "longdash",
                  "31" = "solid"))

pp_35a <- ggplot() +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = x1_grid,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '1', linetype = '1')) +

  ggtitle('vc,lb vs Ra') +

  xlab('vc') +

  ylab('Ra') +

  ylim(.10,.30) +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

  theme_bw()

pp_35 <- pp_35a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = x1_grid,
                                                                    z1 = 0,
                                                                    z2 = -1,
                                                                    z3 = 0))),
              colour = '0', linetype = '0')) +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,

```

```

x2 = 0,
x3 = x1_grid,
z1 = 0,

z2 = 1,
z3 = 0))),

  colour = '2', linetype = '2')) +
scale_color_manual(name = "lb",
  values = c("0" = "dodgerblue",
            "1" = "darkgoldenrod1",
            "2" = "firebrick3")) +
scale_linetype_manual(name = "lb",
  values = c("0" = "dashed",
            "1" = "longdash",
            "2" = "solid"))
pp_36a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
x2 = 0,
x3 = x1_grid,
z1 = 0,
z2 = 0,
z3 = 0))),

  colour = '1', linetype = '1')) +
  ggtitle('vc,Q vs Ra') +
  xlab('vc') +
  ylab('Ra') +

```



```

ylim(.10,.30) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

theme_bw()

pp_36 <- pp_36a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = x1_grid,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = -1))),
              colour = '0', linetype = '0')) +

  geom_line(aes(x = x1_grid, y = (predict(res.Ra.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = x1_grid,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 1))),
              colour = '2', linetype = '2')) +

  scale_color_manual(name = "Q",
                    values = c("0" = "dodgerblue",
                              "1" = "darkgoldenrod1",
                              "2" = "firebrick3")) +

  scale_linetype_manual(name = "Q",
                      values = c("0" = "dashed",
                                "1" = "longdash",

```

```

"2" = "solid"))

ggarrange(pp_12,pp_13,pp_23,
          pp_14,pp_15,pp_16,
          pp_24,pp_25,pp_26,
          pp_34,pp_35,pp_36, ncol = 3, nrow = 4)

mse_Ra <- (deviance(res.Ra.w)/df.residual(res.Ra.w))

coef_Ra <- res.Ra.w$coefficients

coef_Ra

b0_Ra <- coef_Ra[1]

b_Ra <- c(coef_Ra[2:4])

B_Ra <- matrix(c(coef_Ra[8], coef_Ra[11]/2, coef_Ra[12]/2,
                coef_Ra[11]/2, coef_Ra[9], coef_Ra[16]/2,
                coef_Ra[12]/2, coef_Ra[16]/2, coef_Ra[10]),
              nrow = 3,
              ncol = 3)

g_Ra <- coef_Ra[5:7]

coefs_xz_Ra <- c(coef_Ra[13:15], coef_Ra[17:22])

D_Ra <- matrix(coefs_xz_Ra,
              nrow = 3,
              ncol = 3,
              byrow = T)

x1 <- seq(-(2^3)^0.25, (2^3)^0.25, .2)

x2 <- x1

x3 <- x1

xx <- expand.grid(x1,x2,x3)

```

```

Ra.mean = function(x) {b0_Ra + t(x)%*%b_Ra + t(x)%*%B_Ra%*%x}

x.test <- c(-.8,-.2,0)

Ra.mean(x.test)

z1 <- apply(xx, 1, Ra.mean)

dataRaplo11 <- cbind(xx,z1)

colnames(dataRaplo11) <- c("fza", "fzt", "vc", "E.Ra")

p1 <- ggplot(data = dataRaplo11,
             mapping = aes(x = fza, y = fzt, z = E.Ra, fill = E.Ra)) +
  geom_tile() +
  scale_fill_viridis_c(option = "viridis") +
  geom_contour(color = "grey2") +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  coord_equal() + theme_bw()

p2 <- ggplot(data = dataRaplo11,
             mapping = aes(x = fza, y = vc, z = E.Ra, fill = E.Ra)) +
  geom_tile() +
  scale_fill_viridis_c(option = "viridis") +
  geom_contour(color = "grey2") +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
  coord_equal() + theme_bw()

p3 <- ggplot(data = dataRaplo11,
             mapping = aes(x = fzt, y = vc, z = E.Ra, fill = E.Ra)) +
  geom_tile() +

```

```

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

coord_equal() + theme_bw()

Ra.var = function(x) {mse_Ra + t(g_Ra + t(D_Ra)%*%x)%*%(g_Ra + t(D_Ra)%*%x)}

Ra.var(x.test)

ww1 <- apply(xx, 1, Ra.var)

dataRaplo11$S.Ra <- sqrt(ww1)

p4 <- ggplot(data = dataRaplo11,

             mapping = aes(x = fza, y = fzt, z = S.Ra, fill = S.Ra)) +

             geom_tile() +

             scale_fill_viridis_c(option = "viridis") +

             geom_contour(color = "grey2") +

             scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

             scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

             coord_equal() + theme_bw()

p5 <- ggplot(data = dataRaplo11,

             mapping = aes(x = fza, y = vc, z = S.Ra, fill = S.Ra)) +

             geom_tile() +

             scale_fill_viridis_c(option = "viridis") +

             geom_contour(color = "grey2") +

             scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

             scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

             coord_equal() + theme_bw()

p6 <- ggplot(data = dataRaplo11,

```

```

mapping = aes(x = fzt, y = vc, z = S.Ra, fill = S.Ra) +

geom_tile() +

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

coord_equal() + theme_bw()

ggarrange(p1,p2,p3,p4,p5,p6)

ro <- (2^3)^0.25      # raiz quarta de 2^k, k = numero de fatores de processo

g <- function(x){

  g1 <- x[1]^2 + x[2]^2 + x[3]^2 - ro^2

  return(list(ceq=NULL,c=g1))

}

x0 <- rep(0, 3)

target_Ra <- solnl(X = x0, objfun = Ra.mean, confun = g)

T_Ra <- target_Ra$fn

Ra.rmse = function(x, T_Ra) {

  E <- (b0_Ra + t(x)%*%b_Ra + t(x)%*%B_Ra)%*%x)

  V <- (mse_Ra + t(g_Ra + t(D_Ra)%*%x)%*%(g_Ra + t(D_Ra)%*%x))

  rmse <- sqrt((E - T_Ra)^2 + V)

  return(rmse)

}

Ra.rmse(x.test, T_Ra)

kk1 <- rep(0, nrow(xx))

for(i in 1: nrow(xx)){

```

```

x <- c(xx[i,1],xx[i,2],xx[i,3])

kk1[i] <- Ra.rmse(x,T_Ra)

}

dataRaplo11$RMSE.Ra <- kk1

p7 <- ggplot(data = dataRaplo11,

             mapping = aes(x = fza, y = fzt, z = RMSE.Ra, fill = RMSE.Ra)) +

             geom_tile() +

             scale_fill_distiller(palette = "RdYlGn",

                                   direction = -1) +

             geom_contour(color = "grey2") +

             scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

             scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

             coord_equal() + theme_bw()

p8 <- ggplot(data = dataRaplo11,

             mapping = aes(x = fza, y = vc, z = RMSE.Ra, fill = RMSE.Ra)) +

             geom_tile() +

             scale_fill_distiller(palette = "RdYlGn",

                                   direction = -1) +

             geom_contour(color = "grey2") +

             scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

             scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

             coord_equal() + theme_bw()

p9 <- ggplot(data = dataRaplo11,

             mapping = aes(x = fzt, y = vc, z = RMSE.Ra, fill = RMSE.Ra)) +

```

```
geom_tile() +  
  
scale_fill_distiller(palette = "RdYlGn",  
                     direction = -1) +  
  
geom_contour(color = "grey2") +  
  
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +  
  
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +  
  
coord_equal() + theme_bw()  
ggarrange(p7,p8,p9)
```

Código para modelagem robusta da força F_a

```

library(rsm)

library(olsrr)

library(ggplot2)

library(ggpubr)

library(NlcOptim)

library(lattice)

plan.ccd <- ccd(~x1+x2+x3+z1+z2,

               generators = z3~x1*x2*x3*z1*z2,

               n0 = c(0,9),

               alpha = "rotatable",

               randomize = F,

               oneblock = T,

               coding = list(x1~(fza-0.15)/0.05,

                             x2~(fzt-0.15)/0.05,

                             x3~(vc-50)/15,

                             z1~(lto-29)/2,

                             z2~(lb-1)/1,

                             z3~(Q-1)/1))

plan.ccd <- rbind(plan.ccd[1:38,],plan.ccd[45:53,])

plan.ccd

setwd("C:\\...")

respostas <- read.csv("resultados.csv", header = T)

Fa <- respostas$Fz

plan.ccd$Fa <- Fa

res.Fa <- lm(formula = Fa ~ x1+x2+x3+z1+z2+z3 +

```



```

x1*x2+x1*x3+x1*z1+x1*z2+x1*z3+x2*x3+x2*z1+x2*z2+x2*z3+x3*z1+x3*z2+x3*z3 +
+ I(x1^2) + I(x2^2) + I(x3^2),

data = plan.ccd)

summary(res.Fa)

shapiro.test(res.Fa$residuals)

ols_test_breusch_pagan(res.Fa, rhs = T, multiple = T)

par(mfrow = c(2, 2))

plot(res.Fa)

par(mfrow = c(1, 1))

w.Fa <- 1/res.Fa$residuals^2

res.Fa.w <- lm(formula = Fa ~ x1+x2+x3+z1+z2+z3 +

x1*x2+x1*x3+x1*z1+x1*z2+x1*z3+x2*x3+x2*z1+x2*z2+x2*z3+x3*z1+x3*z2+x3*z3 +

+ I(x1^2) + I(x2^2) + I(x3^2),

data = plan.ccd,

weights = w.Fa)

summary(res.Fa.w)

shapiro.test(res.Fa.w$residuals)

ols_test_breusch_pagan(res.Fa.w, rhs = T, multiple = T)

par(mfrow = c(2, 2))

plot(res.Fa.w)

par(mfrow = c(1, 1))

x1_grid <- seq(min(plan.ccd$x1), max(plan.ccd$x1), 0.1)

pp1 <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

x2 = 0,

x3 = 0,

z1 = 0,

```

```

                                z2 = 0,
                                z3 = 0))))),
    colour = 'mediumseagreen') +
ggtitle('fza vs Fa') +
xlab('fza') +
ylab('Fa') +
ylim(0.015,0.08) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp2 <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))))),
    colour = 'mediumseagreen') +
ggtitle('fzt vs Fa') +
xlab('fzt') +
ylab('Fa') +
ylim(0.015,0.08) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp3 <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,

```

```

x3 = x1_grid,

z1 = 0,

z2 = 0,

z3 = 0))))),

  colour = 'mediumseagreen') +

ggtitle('vc vs Fa') +

xlab('vc') +

ylab('Fa') +

ylim(0.015,0.08) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

theme_bw()

z1_grid <- seq(min(plan.ccd$z1), max(plan.ccd$z1), 0.1)

pp4 <- ggplot() +

  geom_line(aes(x = z1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

x2 = 0,

x3 = 0,

z1 = z1_grid,

z2 = 0,

z3 = 0))))),

  colour = 'sienna2') +

ggtitle('lto vs Fa') +

xlab('lto') +

ylab('Fa') +

ylim(0.015,0.08) +

scale_x_continuous(breaks = c(-1, 0, 1), label = c(27, 29, 31)) +

theme_bw()

```

```

pp5 <- ggplot() +
  geom_line(aes(x = z1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = z1_grid,
                                                                    z3 = 0))))),
            colour = 'sienna2') +
  ggtitle('lb vs Fa') +
  xlab('lb') +
  ylab('Fa') +
  ylim(0.015,0.08) +
  scale_x_continuous(breaks = c(-1, 0, 1), label = c(0, 1, 2)) +
  theme_bw()
pp6 <- ggplot() +
  geom_line(aes(x = z1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = z1_grid))))),
            colour = 'sienna2') +
  ggtitle('Q vs Fa') +
  xlab('Q') +
  ylab('Fa') +
  ylim(0.015,0.08) +

```

```

scale_x_continuous(breaks = c(-1, 0, 1), label = c(0, 1, 2)) +

theme_bw()

ggarrange(pp1, pp2, pp3, pp4, pp5, pp6)

pp_12a <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

                                x2 = 0,

                                x3 = 0,

                                z1 = 0,

                                z2 = 0,

                                z3 = 0))),

          colour = '0.15', linetype = '0.15')) +

ggtitle('fza,fzt vs Fa') +

xlab('fza') +

ylab('Fa') +

ylim(0, 0.1) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_12 <- pp_12a +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

                                x2 = -1,

                                x3 = 0,

                                z1 = 0,

                                z2 = 0,

                                z3 = 0))),

          colour = '0.10', linetype = '0.10')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

```

```

                                x2 = 1,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '0.20', linetype = '0.20')) +
scale_color_manual(name = "fzt",
                    values = c("0.10" = "orange2",
                                "0.15" = "olivedrab3",
                                "0.20" = "mediumvioletred")) +
scale_linetype_manual(name = "fzt",
                      values = c("0.10" = "dashed",
                                  "0.15" = "longdash",
                                  "0.20" = "solid"))
pp_13a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
              colour = '50', linetype = '50')) +
  ggtitle('fza,vc vs Fa') +
  xlab('fza') +
  ylab('Fa') +
  ylim(0, 0.1) +

```

```

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_13 <- pp_13a +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

                                x2 = 0,

                                x3 = -1,

                                z1 = 0,

                                z2 = 0,

                                z3 = 0))),

          colour = '35', linetype = '35')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

                                x2 = 0,

                                x3 = 1,

                                z1 = 0,

                                z2 = 0,

                                z3 = 0))),

          colour = '65', linetype = '65')) +

scale_color_manual(name = "vc",

                   values = c("35" = "orange2",

                               "50" = "olivedrab3",

                               "65" = "mediumvioletred")) +

scale_linetype_manual(name = "vc",

                      values = c("35" = "dashed",

                                  "50" = "longdash",

                                  "65" = "solid"))

pp_23a <- ggplot() +

```

```

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
            colour = '50', linetype = '50')) +
ggtitle('fzt,vc vs Fa') +
xlab('fzt') +
ylab('Fa') +
ylim(0, 0.1) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp_23 <- pp_23a +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = -1,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
            colour = '35', linetype = '35')) +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 1,
                                z1 = 0,
                                z2 = 0,

```



```

                                z3 = 0))),

    colour = '65', linetype = '65')) +
scale_color_manual(name = "vc",
                    values = c("35" = "orange2",
                                "50" = "olivedrab3",
                                "65" = "mediumvioletred")) +
scale_linetype_manual(name = "vc",
                       values = c("35" = "dashed",
                                   "50" = "longdash",
                                   "65" = "solid"))

pp_14a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
               colour = '29', linetype = '29')) +
  ggtitle('fza,lto vs Fa') +
  xlab('fza') +
  ylab('Fa') +
  ylim(0, 0.1) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()

pp_14 <- pp_14a +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

```

```

x2 = 0,

x3 = 0,

z1 = -1,

z2 = 0,

z3 = 0))),

colour = '27', linetype = '27')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

x2 = 0,

x3 = 0,

z1 = 1,

z2 = 0,

z3 = 0))),

colour = '31', linetype = '31')) +

scale_color_manual(name = "Ito",

values = c("27" = "dodgerblue",

"29" = "darkgoldenrod1",

"31" = "firebrick3")) +

scale_linetype_manual(name = "Ito",

values = c("27" = "dashed",

"29" = "longdash",

"31" = "solid"))

pp_15a <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,

x2 = 0,

x3 = 0,

z1 = 0,

```

```

                                z2 = 0,
                                z3 = 0))),
    colour = '1', linetype = '1')) +
ggtitle('fza,lb vs Fa') +
xlab('fza') +
ylab('Fa') +
ylim(0, 0.1) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp_15 <- pp_15a +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = -1,
                                z3 = 0))),
    colour = '0', linetype = '0')) +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = 1,
                                z3 = 0))),
    colour = '2', linetype = '2')) +
scale_color_manual(name = "lb",
    values = c("0" = "dodgerblue",

```

```

    "1" = "darkgoldenrod1",
    "2" = "firebrick3")) +
scale_linetype_manual(name = "lb",
    values = c("0" = "dashed",
    "1" = "longdash",
    "2" = "solid"))
pp_16a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
    x2 = 0,
    x3 = 0,
    z1 = 0,
    z2 = 0,
    z3 = 0))),
    colour = '1', linetype = '1')) +
  ggtitle('fza,Q vs Fa') +
  xlab('fza') +
  ylab('Fa') +
  ylim(0, 0.1) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()
pp_16 <- pp_16a +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
    x2 = 0,
    x3 = 0,
    z1 = 0,
    z2 = 0,

```

```

                                z3 = -1))),
    colour = '0', linetype = '0')) +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 1))),
    colour = '2', linetype = '2')) +
scale_color_manual(name = "Q",
    values = c("0" = "dodgerblue",
               "1" = "darkgoldenrod1",
               "2" = "firebrick3")) +
scale_linetype_manual(name = "Q",
    values = c("0" = "dashed",
               "1" = "longdash",
               "2" = "solid"))
pp_24a <- ggplot() +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '29', linetype = '29')) +
ggtitle('fzt,lto vs Fa') +

```

```

xlab('fzt') +

ylab('Fa') +

ylim(0, 0.1) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_24 <- pp_24a +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

                                x2 = x1_grid,

                                x3 = 0,

                                z1 = -1,

                                z2 = 0,

                                z3 = 0))),

           colour = '27', linetype = '27')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

                                x2 = x1_grid,

                                x3 = 0,

                                z1 = 1,

                                z2 = 0,

                                z3 = 0))),

           colour = '31', linetype = '31')) +

scale_color_manual(name = "lto",

                   values = c("27" = "dodgerblue",

                              "29" = "darkgoldenrod1",

                              "31" = "firebrick3")) +

scale_linetype_manual(name = "lto",

```

```

      values = c("27" = "dashed",
                "29" = "longdash",
                "31" = "solid"))

pp_25a <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '1', linetype = '1')) +

ggtitle('fzt,lb vs Fa') +

xlab('fzt') +

ylab('Fa') +

ylim(0, 0.1) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_25 <- pp_25a +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = -1,
                                                                    z3 = 0))),
              colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

```

```

                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 1,
                                z3 = 0))),
    colour = '2', linetype = '2')) +
scale_color_manual(name = "lb",
    values = c("0" = "dodgerblue",
              "1" = "darkgoldenrod1",
              "2" = "firebrick3")) +
scale_linetype_manual(name = "lb",
    values = c("0" = "dashed",
              "1" = "longdash",
              "2" = "solid"))
pp_26a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '1', linetype = '1')) +
  ggtitle('fzt,Q vs Fa') +
  xlab('fzt') +
  ylab('Fa') +
  ylim(0, 0.1) +

```



```

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_26 <- pp_26a +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

                                x2 = x1_grid,

                                x3 = 0,

                                z1 = 0,

                                z2 = 0,

                                z3 = -1))),

          colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

                                x2 = x1_grid,

                                x3 = 0,

                                z1 = 0,

                                z2 = 0,

                                z3 = 1))),

          colour = '2', linetype = '2')) +

scale_color_manual(name = "Q",

                   values = c("0" = "dodgerblue",

                               "1" = "darkgoldenrod1",

                               "2" = "firebrick3")) +

scale_linetype_manual(name = "Q",

                      values = c("0" = "dashed",

                                  "1" = "longdash",

                                  "2" = "solid"))

pp_34a <- ggplot() +

```

```

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
            colour = '29', linetype = '29')) +
ggtitle('vc,lto vs Fa') +
xlab('vc') +
ylab('Fa') +
ylim(0, 0.1) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
theme_bw()
pp_34 <- pp_34a +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = -1,
                                z2 = 0,
                                z3 = 0))),
            colour = '27', linetype = '27')) +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 1,
                                z2 = 0,

```

```

                                z3 = 0))),

    colour = '31', linetype = '31')) +
scale_color_manual(name = "lto",
    values = c("27" = "dodgerblue",
               "29" = "darkgoldenrod1",
               "31" = "firebrick3")) +
scale_linetype_manual(name = "lto",
    values = c("27" = "dashed",
               "29" = "longdash",
               "31" = "solid"))

pp_35a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '1', linetype = '1')) +
  ggtitle('vc,lb vs Fa') +
  xlab('vc') +
  ylab('Fa') +
  ylim(0, 0.1) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
  theme_bw()

pp_35 <- pp_35a +
  geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

```

```

x2 = 0,

x3 = x1_grid,

z1 = 0,

z2 = -1,

z3 = 0))),

colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

x2 = 0,

x3 = x1_grid,

z1 = 0,

z2 = 1,

z3 = 0))),

colour = '2', linetype = '2')) +

scale_color_manual(name = "lb",

values = c("0" = "dodgerblue",

"1" = "darkgoldenrod1",

"2" = "firebrick3")) +

scale_linetype_manual(name = "lb",

values = c("0" = "dashed",

"1" = "longdash",

"2" = "solid"))

pp_36a <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,

x2 = 0,

x3 = x1_grid,

z1 = 0,

```

```

                                z2 = 0,
                                z3 = 0))),
    colour = '1', linetype = '1')) +
ggtitle('vc,Q vs Fa') +
xlab('vc') +
ylab('Fa') +
ylim(0, 0.1) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
theme_bw()
pp_36 <- pp_36a +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 0,
                                z3 = -1))),
    colour = '0', linetype = '0')) +
geom_line(aes(x = x1_grid, y = (predict(res.Fa.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 0,
                                z3 = 1))),
    colour = '2', linetype = '2')) +
scale_color_manual(name = "Q",
    values = c("0" = "dodgerblue",

```

```

        "1" = "darkgoldenrod1",
        "2" = "firebrick3")) +
scale_linetype_manual(name = "Q",
        values = c("0" = "dashed",
        "1" = "longdash",
        "2" = "solid"))
ggarrange(pp_12,pp_13,pp_23,
        pp_14,pp_15,pp_16,
        pp_24,pp_25,pp_26,
        pp_34,pp_35,pp_36, ncol = 3, nrow = 4)
mse_Fa <- (deviance(res.Fa.w)/df.residual(res.Fa.w))
coef_Fa <- res.Fa.w$coefficients
coef_Fa
b0_Fa <- coef_Fa[1]
b_Fa <- c(coef_Fa[2:4])
B_Fa <- matrix(c(coef_Fa[8],  coef_Fa[11]/2, coef_Fa[12]/2,
        coef_Fa[11]/2, coef_Fa[9],  coef_Fa[16]/2,
        coef_Fa[12]/2, coef_Fa[16]/2, coef_Fa[10]),
        nrow = 3,
        ncol = 3)
g_Fa <- coef_Fa[5:7]
coefs_xz1_Fa <- c(coef_Fa[13:15], coef_Fa[17:22])
D_Fa <- matrix(coefs_xz1_Fa,
        nrow = 3,
        ncol = 3,
        byrow = T)

```

```

x1 <- seq(-(2^3)^0.25, (2^3)^0.25, .2)

x2 <- x1

x3 <- x1

xx <- expand.grid(x1,x2,x3)

Fa.mean = function(x) {b0_Fa + t(x)%*%b_Fa + t(x)%*%B_Fa%*%x}

x.test <- c(-.8,-.2,0)

Fa.mean(x.test)

z1 <- apply(xx, 1, Fa.mean)

dataFaplo11 <- cbind(xx,z1)

colnames(dataFaplo11) <- c("fza", "fzt", "vc", "E.Fa")

p1 <- ggplot(data = dataFaplo11,
             mapping = aes(x = fza, y = fzt, z = E.Fa, fill = E.Fa)) +
  geom_tile() +
  scale_fill_viridis_c(option = "viridis") +
  geom_contour(color = "grey2") +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  coord_equal() + theme_bw()

p2 <- ggplot(data = dataFaplo11,
             mapping = aes(x = fza, y = vc, z = E.Fa, fill = E.Fa)) +
  geom_tile() +
  scale_fill_viridis_c(option = "viridis") +
  geom_contour(color = "grey2") +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
  coord_equal() + theme_bw()

```

```

p3 <- ggplot(data = dataFaplo11,

             mapping = aes(x = fzt, y = vc, z = E.Fa, fill = E.Fa)) +

geom_tile() +

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

coord_equal() + theme_bw()

Fa.var = function(x) {mse_Fa + t(g_Fa + t(D_Fa)%*%x)%*%(g_Fa + t(D_Fa)%*%x)}

Fa.var(x.test)

ww1 <- apply(xx, 1, Fa.var)

dataFaplo11$S.Fa <- sqrt(ww1)

p4 <- ggplot(data = dataFaplo11,

             mapping = aes(x = fza, y = fzt, z = S.Fa, fill = S.Fa)) +

geom_tile() +

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

coord_equal() + theme_bw()

p5 <- ggplot(data = dataFaplo11,

             mapping = aes(x = fza, y = vc, z = S.Fa, fill = S.Fa)) +

geom_tile() +

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

```



```

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
coord_equal() + theme_bw()

p6 <- ggplot(data = dataFaplo11,
             mapping = aes(x = fzt, y = vc, z = S.Fa, fill = S.Fa)) +
geom_tile() +
scale_fill_viridis_c(option = "viridis") +
geom_contour(color = "grey2") +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
coord_equal() + theme_bw()

ggarrange(p1,p2,p3,p4,p5,p6)

ro <- (2^3)^0.25 # raiz quarta de 2^k, k = numero de fatores de processo

g_x <- function(x){
  g1 <- x[1]^2 + x[2]^2 + x[3]^2 - ro^2
  return(list(ceq=NULL,c=g1))
}

x0 <- rep(0, 3)

target_Fa <- solnl(X = x0, objfun = Fa.mean, confun = g_x)

T_Fa <- target_Fa$fn

Fa.rmse = function(x, T_Fa) {
  E <- (b0_Fa + t(x)%*%b_Fa + t(x)%*%B_Fa)%*%x)
  V <- (mse_Fa + t(g_Fa + t(D_Fa)%*%x)%*%(g_Fa + t(D_Fa)%*%x))

  rmse <- sqrt((E - T_Fa)^2 + V)

  return(rmse)
}

Fa.rmse(x.test, T_Fa)

```

```

kk1 <- rep(0, nrow(xx))

for(i in 1: nrow(xx)){

  x <- c(xx[i,1],xx[i,2],xx[i,3])

  kk1[i] <- Fa.rmse(x,T_Fa)

}

dataFaplo11$RMSE.Fa <- kk1

p7 <- ggplot(data = dataFaplo11,

             mapping = aes(x = fza, y = fzt, z = RMSE.Fa, fill = RMSE.Fa)) +

  geom_tile() +

  scale_fill_distiller(palette = "RdYlGn",

                      direction = -1) +

  geom_contour(color = "grey2") +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  coord_equal() + theme_bw()

p8 <- ggplot(data = dataFaplo11,

             mapping = aes(x = fza, y = vc, z = RMSE.Fa, fill = RMSE.Fa)) +

  geom_tile() +

  scale_fill_distiller(palette = "RdYlGn",

                      direction = -1) +

  geom_contour(color = "grey2") +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

  coord_equal() + theme_bw()

p9 <- ggplot(data = dataFaplo11,

```

```
mapping = aes(x = fzt, y = vc, z = RMSE.Fa, fill = RMSE.Fa)) +  
geom_tile() +  
scale_fill_distiller(palette = "RdYlGn",  
direction = -1) +  
geom_contour(color = "grey2") +  
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +  
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +  
coord_equal() + theme_bw()  
ggarrange(p7,p8,p9)
```

Código para modelagem robusta da circularidade Ront

```

library(rsm)

library(olsrr)

library(ggplot2)

library(ggpubr)

library(NlcOptim)

library(lattice)

plan.ccd <- ccd(~x1+x2+x3+z1+z2,

               generators = z3~x1*x2*x3*z1*z2,

               n0 = c(0,9),

               alpha = "rotatable",

               randomize = F,

               oneblock = T,

               coding = list(x1~(fza-0.15)/0.05,

                             x2~(fzt-0.15)/0.05,

                             x3~(vc-50)/15,

                             z1~(lto-29)/2,

                             z2~(lb-1)/1,

                             z3~(Q-1)/1))

plan.ccd <- rbind(plan.ccd[1:38,],plan.ccd[45:53,])

plan.ccd

setwd("C:\\...")

respostas <- read.csv("resultados.csv", header = T)

Ront <- respostas$Ront

plan.ccd$Ront <- Ront

res.Ront <- lm(formula = Ront ~ x1+x2+x3+z1+z2+z3 +

```

```

x1*x2+x1*x3+x1*z1+x1*z2+x1*z3+x2*x3+x2*z1+x2*z2+x2*z3+x3*z1+x3*z2+x3*z3 +
  + I(x1^2) + I(x2^2) + I(x3^2),
data = plan.ccd)
summary(res.Ront)
shapiro.test(res.Ront$residuals)
ols_test_breusch_pagan(res.Ront, rhs = T, multiple = T)
par(mfrow = c(2, 2))
plot(res.Ront)
par(mfrow = c(1, 1))
w.Ront <- 1/res.Ront$residuals^2
res.Ront.w <- lm(formula = Ront ~ x1+x2+x3+z1+z2+z3 +
  x1*x2+x1*x3+x1*z1+x1*z2+x1*z3+x2*x3+x2*z1+x2*z2+x2*z3+x3*z1+x3*z2+x3*z3 +
  + I(x1^2) + I(x2^2) + I(x3^2),
data = plan.ccd,
weights = w.Ront)
summary(res.Ront.w)
shapiro.test(res.Ront.w$residuals)
ols_test_breusch_pagan(res.Ront.w, rhs = T, multiple = T)
par(mfrow = c(2, 2))
plot(res.Ront.w)
par(mfrow = c(1, 1))
x1_grid <- seq(min(plan.ccd$x1), max(plan.ccd$x1), 0.1)
pp1 <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
x2 = 0,

```

```

x3 = 0,
z1 = 0,
z2 = 0,
z3 = 0))))),

  colour = 'mediumseagreen') +

ggtitle('fza vs Ront') +

xlab('fza') +

ylab('Ront') +

ylim(13,23) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp2 <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

x2 = x1_grid,

x3 = 0,

z1 = 0,

z2 = 0,

z3 = 0))))),

  colour = 'mediumseagreen') +

ggtitle('fzt vs Ront') +

xlab('fzt') +

ylab('Ront') +

ylim(13,23) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

```

```

pp3 <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = x1_grid,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))))),
            colour = 'mediumseagreen') +
  ggtitle('vc vs Ront') +
  xlab('vc') +
  ylab('Ront') +
  ylim(13,23) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
  theme_bw()

z1_grid <- seq(min(plan.ccd$z1), max(plan.ccd$z1), 0.1)

pp4 <- ggplot() +
  geom_line(aes(x = z1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = z1_grid,
                                                                    z2 = 0,
                                                                    z3 = 0))))),
            colour = 'sienna2') +
  ggtitle('lto vs Ront') +
  xlab('lto') +
  ylab('Ront') +

```

```

ylim(13,23) +

scale_x_continuous(breaks = c(-1, 0, 1), label = c(27, 29, 31)) +

theme_bw()

pp5 <- ggplot() +

geom_line(aes(x = z1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

                                x2 = 0,

                                x3 = 0,

                                z1 = 0,

                                z2 = z1_grid,

                                z3 = 0))))),

          colour = 'sienna2') +

ggtitle('lb vs Ront') +

xlab('lb') +

ylab('Ront') +

ylim(13,23) +

scale_x_continuous(breaks = c(-1, 0, 1), label = c(0, 1, 2)) +

theme_bw()

pp6 <- ggplot() +

geom_line(aes(x = z1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

                                x2 = 0,

                                x3 = 0,

                                z1 = 0,

                                z2 = 0,

                                z3 = z1_grid))))),

          colour = 'sienna2') +

ggtitle('Q vs Ront') +

```



```

xlab('Q') +

ylab('Ront') +

ylim(13,23) +

scale_x_continuous(breaks = c(-1, 0, 1), label = c(0, 1, 2)) +

theme_bw()

ggarrange(pp1, pp2, pp3, pp4, pp5, pp6)

pp_12a <- ggplot() +

  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '0.15', linetype = '0.15')) +

  ggtitle('fza,fzt vs Ront') +

  xlab('fza') +

  ylab('Ront') +

  ylim(10,25) +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  theme_bw()

pp_12 <- pp_12a +

  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = -1,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '0.15', linetype = '0.15')) +

```

```

                                z3 = 0))),
    colour = '0.10', linetype = '0.10')) +
geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 1,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '0.20', linetype = '0.20')) +
scale_color_manual(name = "fzt",
    values = c("0.10" = "orange2",
              "0.15" = "olivedrab3",
              "0.20" = "mediumvioletred")) +
scale_linetype_manual(name = "fzt",
    values = c("0.10" = "dashed",
              "0.15" = "longdash",
              "0.20" = "solid"))
pp_13a <- ggplot() +
geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '50', linetype = '50')) +
ggtitle('fza,vc vs Ront') +

```

```

xlab('fza') +

ylab('Ront') +

ylim(10,25) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_13 <- pp_13a +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = -1,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '35', linetype = '35')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                                                    x2 = 0,
                                                                    x3 = 1,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 0))),
              colour = '65', linetype = '65')) +

scale_color_manual(name = "vc",
                   values = c("35" = "orange2",
                              "50" = "olivedrab3",
                              "65" = "mediumvioletred")) +

scale_linetype_manual(name = "vc",

```

```

values = c("35" = "dashed",
           "50" = "longdash",
           "65" = "solid"))

pp_23a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                                                           x2 = x1_grid,
                                                                           x3 = 0,
                                                                           z1 = 0,
                                                                           z2 = 0,
                                                                           z3 = 0))),
              colour = '50', linetype = '50')) +
  ggtitle('fzt,vc vs Ront') +
  xlab('fzt') +
  ylab('Ront') +
  ylim(10,25) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()

pp_23 <- pp_23a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                                                           x2 = x1_grid,
                                                                           x3 = -1,
                                                                           z1 = 0,
                                                                           z2 = 0,
                                                                           z3 = 0))),
              colour = '35', linetype = '35')) +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

```

```

                                x2 = x1_grid,
                                x3 = 1,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '65', linetype = '65')) +
scale_color_manual(name = "vc",
                    values = c("35" = "orange2",
                                "50" = "olivedrab3",
                                "65" = "mediumvioletred")) +
scale_linetype_manual(name = "vc",
                      values = c("35" = "dashed",
                                  "50" = "longdash",
                                  "65" = "solid"))
pp_14a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
              colour = '29', linetype = '29')) +
  ggtitle('fza,lto vs Ront') +
  xlab('fza') +
  ylab('Ront') +
  ylim(10,25) +

```

```

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_14 <- pp_14a +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,

                                x2 = 0,

                                x3 = 0,

                                z1 = -1,

                                z2 = 0,

                                z3 = 0))),

          colour = '27', linetype = '27')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,

                                x2 = 0,

                                x3 = 0,

                                z1 = 1,

                                z2 = 0,

                                z3 = 0))),

          colour = '31', linetype = '31')) +

scale_color_manual(name = "lto",

                  values = c("27" = "dodgerblue",

                             "29" = "darkgoldenrod1",

                             "31" = "firebrick3")) +

scale_linetype_manual(name = "lto",

                    values = c("27" = "dashed",

                               "29" = "longdash",

                               "31" = "solid"))

pp_15a <- ggplot() +

```



```

                                z3 = 0))),

    colour = '2', linetype = '2')) +
scale_color_manual(name = "lb",
    values = c("0" = "dodgerblue",
               "1" = "darkgoldenrod1",
               "2" = "firebrick3")) +
scale_linetype_manual(name = "lb",
    values = c("0" = "dashed",
               "1" = "longdash",
               "2" = "solid"))

pp_16a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
                                x2 = 0,
                                x3 = 0,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
    colour = '1', linetype = '1')) +
  ggtitle('fza,Q vs Ront') +
  xlab('fza') +
  ylab('Ront') +
  ylim(10,25) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()

pp_16 <- pp_16a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,

```



```

x2 = 0,
x3 = 0,
z1 = 0,
z2 = 0,
z3 = -1))),
  colour = '0', linetype = '0')) +
geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = x1_grid,
x2 = 0,
x3 = 0,
z1 = 0,
z2 = 0,
z3 = 1))),
  colour = '2', linetype = '2')) +
scale_color_manual(name = "Q",
  values = c("0" = "dodgerblue",
            "1" = "darkgoldenrod1",
            "2" = "firebrick3")) +
scale_linetype_manual(name = "Q",
  values = c("0" = "dashed",
            "1" = "longdash",
            "2" = "solid"))
pp_24a <- ggplot() +
geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
x2 = x1_grid,
x3 = 0,
z1 = 0,

```

```

                                z2 = 0,
                                z3 = 0))),
    colour = '29', linetype = '29')) +
ggtitle('fzt,lto vs Ront') +
xlab('fzt') +
ylab('Ront') +
ylim(10,25) +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
theme_bw()
pp_24 <- pp_24a +
geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = -1,
                                z2 = 0,
                                z3 = 0))),
    colour = '27', linetype = '27')) +
geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                x2 = x1_grid,
                                x3 = 0,
                                z1 = 1,
                                z2 = 0,
                                z3 = 0))),
    colour = '31', linetype = '31')) +
scale_color_manual(name = "lto",
                    values = c("27" = "dodgerblue",

```

```

      "29" = "darkgoldenrod1",
      "31" = "firebrick3")) +
scale_linetype_manual(name = "lto",
      values = c("27" = "dashed",
      "29" = "longdash",
      "31" = "solid"))
pp_25a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
      x2 = x1_grid,
      x3 = 0,
      z1 = 0,
      z2 = 0,
      z3 = 0))),
      colour = '1', linetype = '1')) +
  ggtitle('fzt,lb vs Ront') +
  xlab('fzt') +
  ylab('Ront') +
  ylim(10,25) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
  theme_bw()
pp_25 <- pp_25a +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
      x2 = x1_grid,
      x3 = 0,
      z1 = 0,
      z2 = -1,

```

```

                                z3 = 0))),

  colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

                                x2 = x1_grid,

                                x3 = 0,

                                z1 = 0,

                                z2 = 1,

                                z3 = 0))),

  colour = '2', linetype = '2')) +

scale_color_manual(name = "lb",

  values = c("0" = "dodgerblue",

            "1" = "darkgoldenrod1",

            "2" = "firebrick3")) +

scale_linetype_manual(name = "lb",

  values = c("0" = "dashed",

            "1" = "longdash",

            "2" = "solid"))

pp_26a <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

                                x2 = x1_grid,

                                x3 = 0,

                                z1 = 0,

                                z2 = 0,

                                z3 = 0))),

  colour = '1', linetype = '1')) +

ggtitle('fzt,Q vs Ront') +

```

```

xlab('fzt') +

ylab('Ront') +

ylim(10,25) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

theme_bw()

pp_26 <- pp_26a +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = -1))),
              colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                                                    x2 = x1_grid,
                                                                    x3 = 0,
                                                                    z1 = 0,
                                                                    z2 = 0,
                                                                    z3 = 1))),
              colour = '2', linetype = '2')) +

scale_color_manual(name = "Q",
                  values = c("0" = "dodgerblue",
                             "1" = "darkgoldenrod1",
                             "2" = "firebrick3")) +

scale_linetype_manual(name = "Q",
                     values = c("0" = "dashed",

```



```

                                x3 = x1_grid,

                                z1 = 1,

                                z2 = 0,

                                z3 = 0))),

    colour = '31', linetype = '31')) +
scale_color_manual(name = "lto",
                    values = c("27" = "dodgerblue",
                                "29" = "darkgoldenrod1",
                                "31" = "firebrick3")) +
scale_linetype_manual(name = "lto",
                      values = c("27" = "dashed",
                                  "29" = "longdash",
                                  "31" = "solid"))
pp_35a <- ggplot() +
  geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 0,
                                z3 = 0))),
              colour = '1', linetype = '1')) +
  ggtitle('vc,lb vs Ront') +
  xlab('vc') +
  ylab('Ront') +
  ylim(10,25) +
  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

```

```

theme_bw()

pp_35 <- pp_35a +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = -1,
                                z3 = 0))),
            colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,
                                x2 = 0,
                                x3 = x1_grid,
                                z1 = 0,
                                z2 = 1,
                                z3 = 0))),
            colour = '2', linetype = '2')) +

scale_color_manual(name = "lb",
                   values = c("0" = "dodgerblue",
                               "1" = "darkgoldenrod1",
                               "2" = "firebrick3")) +

scale_linetype_manual(name = "lb",
                      values = c("0" = "dashed",
                                  "1" = "longdash",
                                  "2" = "solid"))

pp_36a <- ggplot() +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

```



```

x2 = 0,

x3 = x1_grid,

z1 = 0,

z2 = 0,

z3 = 0))),

  colour = '1', linetype = '1')) +

ggtitle('vc,Q vs Ront') +

xlab('vc') +

ylab('Ront') +

ylim(10,25) +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

theme_bw()

pp_36 <- pp_36a +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

x2 = 0,

x3 = x1_grid,

z1 = 0,

z2 = 0,

z3 = -1))),

  colour = '0', linetype = '0')) +

geom_line(aes(x = x1_grid, y = (predict(res.Ront.w, newdata = data.frame(x1 = 0,

x2 = 0,

x3 = x1_grid,

z1 = 0,

z2 = 0,

z3 = 1))),

```

```

    colour = '2', linetype = '2')) +
scale_color_manual(name = "Q",
    values = c("0" = "dodgerblue",
              "1" = "darkgoldenrod1",
              "2" = "firebrick3")) +
scale_linetype_manual(name = "Q",
    values = c("0" = "dashed",
              "1" = "longdash",
              "2" = "solid"))

ggarrange(pp_12,pp_13,pp_23,
    pp_14,pp_15,pp_16,
    pp_24,pp_25,pp_26,
    pp_34,pp_35,pp_36, ncol = 3, nrow = 4)

mse_Ront <- (deviance(res.Ront.w)/df.residual(res.Ront.w))

coef_Ront <- res.Ront.w$coefficients

coef_Ront

b0_Ront <- coef_Ront[1]

b_Ront <- c(coef_Ront[2:4])

B_Ront <- matrix(c(coef_Ront[8],  coef_Ront[11]/2, coef_Ront[12]/2,
    coef_Ront[11]/2, coef_Ront[9],  coef_Ront[16]/2,
    coef_Ront[12]/2, coef_Ront[16]/2, coef_Ront[10]),
    nrow = 3,
    ncol = 3)

g_Ront <- coef_Ront[5:7]

coefs_xz_Ront <- c(coef_Ront[13:15], coef_Ront[17:22])

```

```

D_Ront <- matrix(coefs_xz_Ront,

                nrow = 3,

                ncol = 3,

                byrow = T)

x1 <- seq(-(2^3)^0.25, (2^3)^0.25, .2)

x2 <- x1

x3 <- x1

xx <- expand.grid(x1,x2,x3)

Ront.mean = function(x) {b0_Ront + t(x)%*%b_Ront + t(x)%*%B_Ront%*%x}

x.test <- c(-.8,-.2,0)

Ront.mean(x.test)

z1 <- apply(xx, 1, Ront.mean)

dataRontplo11 <- cbind(xx,z1)

colnames(dataRontplo11) <- c("fza", "fzt", "vc", "E.Ront")

p1 <- ggplot(data = dataRontplo11,

            mapping = aes(x = fza, y = fzt, z = E.Ront, fill = E.Ront)) +

  geom_tile() +

  scale_fill_viridis_c(option = "viridis") +

  geom_contour(color = "grey2") +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  coord_equal() + theme_bw()

p2 <- ggplot(data = dataRontplo11,

            mapping = aes(x = fza, y = vc, z = E.Ront, fill = E.Ront)) +

  geom_tile() +

  scale_fill_viridis_c(option = "viridis") +

```

```

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

coord_equal() + theme_bw()

p3 <- ggplot(data = dataRontplo11,

             mapping = aes(x = fzt, y = vc, z = E.Ront, fill = E.Ront)) +

geom_tile() +

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +

coord_equal() + theme_bw()

Ront.var = function(x) {mse_Ront + t(g_Ront + t(D_Ront)**x)**(g_Ront + t(D_Ront)**x)}

Ront.var(x.test)

ww1 <- apply(xx, 1, Ront.var)

dataRontplo11$S.Ront <- sqrt(ww1)

p4 <- ggplot(data = dataRontplo11,

             mapping = aes(x = fza, y = fzt, z = S.Ront, fill = S.Ront)) +

geom_tile() +

scale_fill_viridis_c(option = "viridis") +

geom_contour(color = "grey2") +

scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

coord_equal() + theme_bw()

p5 <- ggplot(data = dataRontplo11,

             mapping = aes(x = fza, y = vc, z = S.Ront, fill = S.Ront)) +

```

```

geom_tile() +
scale_fill_viridis_c(option = "viridis") +
geom_contour(color = "grey2") +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
coord_equal() + theme_bw()

p6 <- ggplot(data = dataRontplo11,
             mapping = aes(x = fzt, y = vc, z = S.Ront, fill = S.Ront)) +
geom_tile() +
scale_fill_viridis_c(option = "viridis") +
geom_contour(color = "grey2") +
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +
coord_equal() + theme_bw()

ggarrange(p1,p2,p3,p4,p5,p6)

ro <<- (2^3)^0.25      # raiz quarta de 2^k, k = numero de Fatores de processo

g_x <- function(x){
  g1 <- x[1]^2 + x[2]^2 + x[3]^2 - ro^2
  return(list(ceq=NULL,c=g1))
}

x0 <- rep(-0.5, 3)

target_Ront <- solnl(X = x.test, objfun = Ront.mean, confun = g_x)

T_Ront <<- target_Ront$fn

Ront.rmse = function(x, T_Ront) {
  E <- (b0_Ront + t(x)%*%b_Ront + t(x)%*%B_Ront)%*%x)
  V <- (mse_Ront + t(g_Ront + t(D_Ront)%*%x)%*%(g_Ront + t(D_Ront)%*%x))

```

```

rmse <- sqrt((E - T_Ront)^2 + V)

return(rmse)

}

Ront.rmse(x.test, T_Ront)

kk1 <- rep(0, nrow(xx))

for(i in 1: nrow(xx)){

  x <- c(xx[i,1],xx[i,2],xx[i,3])

  kk1[i] <- Ront.rmse(x,T_Ront)

}

dataRontplo11$RMSE.Ront <- kk1

p7 <- ggplot(data = dataRontplo11,

             mapping = aes(x = fza, y = fzt, z = RMSE.Ront, fill = RMSE.Ront)) +

  geom_tile() +

  scale_fill_distiller(palette = "RdYIGn",

                      direction = -1) +

  geom_contour(color = "grey2") +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

  coord_equal() + theme_bw()

p8 <- ggplot(data = dataRontplo11,

             mapping = aes(x = fza, y = vc, z = RMSE.Ront, fill = RMSE.Ront)) +

  geom_tile() +

  scale_fill_distiller(palette = "RdYIGn",

                      direction = -1) +

  geom_contour(color = "grey2") +

  scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +

```

```
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +  
coord_equal() + theme_bw()  
p9 <- ggplot(data = dataRontplo11,  
            mapping = aes(x = fzt, y = vc, z = RMSE.Ront, fill = RMSE.Ront)) +  
geom_tile() +  
scale_fill_distiller(palette = "RdYlGn",  
                    direction = -1) +  
geom_contour(color = "grey2") +  
scale_x_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(0.05, 0.1, 0.15, 0.2, 0.25)) +  
scale_y_continuous(breaks = c(-2, -1, 0, 1, 2), label = c(20, 35, 50, 65, 80)) +  
coord_equal() + theme_bw()  
ggarrange(p7,p8,p9)
```

Código do algoritmo MOPSO

```

mopsocd <-
function(fn,gn,varcnt,fnctnt,lowerbound,upperbound,opt,popsize,maxgen,archivesize,verbosity,pMut,w,c1,c2)
UseMethod("mopsocd")

mopsocd <-
function(fn,gn=NULL,varcnt,fnctnt,lowerbound,upperbound,opt,popsize=100,maxgen=100,archivesize=250,verb
osity=0,pMut=0.5,w=0.4,c1=1.0,c2=1.0)
{
  varcnt <- as.integer(varcnt)
  fnctnt <- as.integer(fnctnt)
  lowerbound <- as.integer(lowerbound)
  upperbound <- as.integer(upperbound)
  opt <- as.integer(opt)
  popsize <- as.integer(popsize)
  maxgen <- as.integer(maxgen)
  archivesize <- as.integer(archivesize)
  verbosity <- as.integer(verbosity)
  pMut <- as.numeric(pMut)
  w <- as.numeric(w)
  c1 <- as.numeric(c1)
  c2 <- as.numeric(c2)

  minvalue <- vector(length=varcnt)
  maxvalue <- vector(length=varcnt)
  archiveVar <- matrix(,nrow=archivesize,ncol=varcnt)
  archiveFit <- matrix(,nrow=archivesize,ncol=fnctnt)
  popVar <- matrix(,nrow=popsize,ncol=varcnt)
  popFit <- matrix(,nrow=popsize,ncol=fnctnt)
  pbestsVar <- matrix(,nrow=popsize,ncol=varcnt)
  pbestsFit <- matrix(,nrow=popsize,ncol=fnctnt)
  velocity <- matrix(,nrow=popsize,ncol=varcnt)
  crowdDist <- vector(length=archivesize)
  minvaluetemp <- matrix(,nrow=popsize,ncol=varcnt)
  maxvaluetemp <- matrix(,nrow=popsize,ncol=varcnt)
  ndomCtr <- 0
  maxvalue <- upperbound

```



```

minvalue <- lowerbound
initialize_pop <- function() {

  for (i in 1:varcnt) {
    popVar[1:popsi, i] <<- runif(popsi, min=minvalue[i], max=maxvalue[i])
  }
}

initialize_vel <- function() {

  velocity[1:popsi, 1:varcnt] <<- 0.0
}

evaluate <- function() {

  popFit[1:popsi, ] <<- t(apply(popVar, 1, fn))
}

store_pbests <- function() {
  pbestsVar[1:popsi, ] <<- popVar[1:popsi, ]
  pbestsFit[1:popsi, ] <<- popFit[1:popsi, ]
}

insert_nondom <- function() {

  if(!is.null(gn)) {
    z <- rowSums(t(matrix(!apply(popVar, 1, gn), ncol=popsi)*1))
    minz <- min(z)
    ix <- which(z == minz)
    lenix <- length(ix)

    ixfpop <- NULL
    for (i in ix) {
      if (opt == 1) {
        x <- (t(matrix(rep(popFit[i, ], lenix), nrow=fncnt))-popFit[ix, ] < 0.0)*1
      } else {
        x <- (t(matrix(rep(popFit[i, ], lenix), nrow=fncnt))-popFit[ix, ] > 0.0)*1
      }
    }
    if((max(rowSums(x)) == fncnt) == FALSE){

```

```

ixfpop <- c(ixfpop,i)
}
}
} else {
ixfpop =seq(popsize)
}
lenfpop <- length(ixfpop)

if(lenfpop != 0){
if (ndomCtr + lenfpop <= archivesize) {
archiveVar[(ndomCtr+1):(ndomCtr+lenfpop),] <<- popVar[ixfpop,]
archiveFit[(ndomCtr+1):(ndomCtr+lenfpop),] <<- popFit[ixfpop,]
ndomCtr <<- ndomCtr + lenfpop
} else {
cat("FATAL ERROR:\nArchive Size Value ",archivesize," too small\n")
cat("Recommended Archive Size: 2.5*population size\n")
cat("Press Enter to exit\n")
scan(quiet=TRUE)
cat("Goodbye !\n")
stop()
}
} else {
cat("No solution (Unreasonable constraints)\n")
cat("Press Enter to exit\n")
scan(quiet=TRUE)
cat("Goodbye !\n")
stop()
}
}
}

crowding <- function() {

crowdDist[1:ndomCtr] <<- 0.0

for (i in seq(fncnt)) {

```

```

sorted_archive <- sort(archiveFit[1:ndomCtr,i],index.return=TRUE)
valsa <- sorted_archive$x
ixsa <- sorted_archive$ix
crowdDist[ixsa[2:(ndomCtr-1)]] <<- crowdDist[ixsa[2:(ndomCtr-1)]] + valsa[3:ndomCtr] - valsa[1:(ndomCtr-2)]
maxcd <- which.max(crowdDist[1:ndomCtr])
crowdDist[ixsa[1]] <<- crowdDist[ixsa[1]] + crowdDist[maxcd]
crowdDist[ixsa[ndomCtr]] <<- crowdDist[ixsa[ndomCtr]] + crowdDist[maxcd]
mincd <- which.min(crowdDist[1:ndomCtr])
crowdDist[1:ndomCtr] <<- crowdDist[1:ndomCtr]/(crowdDist[maxcd]-crowdDist[mincd])
}

sorted_cd <- sort(crowdDist[1:ndomCtr],decreasing=TRUE,index.return=TRUE)
archiveVar[1:length(sorted_cd$ix),] <<- archiveVar[sorted_cd$ix,]
archiveFit[1:length(sorted_cd$ix),] <<- archiveFit[sorted_cd$ix,]

}

update_pbests <- function() {

if(opt == 1){
  x <- rowSums((popFit-pbestsFit > 0.0)*1)
} else {
  x <- rowSums((popFit-pbestsFit < 0.0)*1)
}

ix1 <- which(x != 0 & x != fncnt)
ix2 <- sample(ix1,floor(length(ix1)/2),replace=FALSE)
ix <- union(which(x == fncnt),ix2)

pbestsVar[ix,] <<- popVar[ix,]
pbestsFit[ix,] <<- popFit[ix,]

}

update_archive <- function() {

```

```

if(!is.null(gn)) {
  z <- rowSums(t(matrix(!apply(popVar,1,gn),ncol=popsize)*1))
  minz <- min(z)
  ix <- which(z == minz)
  lenix <- length(ix)

  ixfpop <- NULL
  for (i in ix) {
    if (opt == 1) {
      x <- (t(matrix(rep(popFit[i,],lenix),nrow=fncnt))-popFit[ix,]) < 0.0)*1
    } else {
      x <- (t(matrix(rep(popFit[i,],lenix),nrow=fncnt))-popFit[ix,]) > 0.0)*1
    }
    if((max(rowSums(x)) == fncnt) == FALSE){
      ixfpop <- c(ixfpop,i)
    }
  }
} else {
  ixfpop =seq(popsize)
}
lenfpop <- length(ixfpop)

ixinspop <- NULL
for (i in ixfpop) {
  if (opt == 1) {
    x <- (t(matrix(rep(popFit[i,],ndomCtr),nrow=fncnt))-archiveFit[1:ndomCtr,]) < 0.0)*1
  } else {
    x <- (t(matrix(rep(popFit[i,],ndomCtr),nrow=fncnt))-archiveFit[1:ndomCtr,]) > 0.0)*1
  }
  if((max(rowSums(x)) == fncnt) == FALSE) {
    ixinspop <- c(ixinspop,i)
  }
}
leninspop <- length(ixinspop)
if(leninspop != 0){

```

```

if (ndomCtr + leninspop <= archivesize) {
  archiveVar[(ndomCtr+1):(ndomCtr+leninspop),] <<- popVar[ixinspop,]
  archiveFit[(ndomCtr+1):(ndomCtr+leninspop),] <<- popFit[ixinspop,]
  ndomCtr <<- ndomCtr + leninspop
} else {
  crowding()
  bottom <- floor(ndomCtr*0.9)
  cnt <- min((archivesize-bottom),leninspop)
  ixinssel <- sample(ixinspop,cnt,replace=F)
  ixreplace <- sample((archivesize-cnt):archivesize,cnt,replace=F)
  archiveVar[(archivesize-cnt+1):archivesize,] <<- popVar[ixinssel,]
  archiveFit[(archivesize-cnt+1):archivesize,] <<- popFit[ixinssel,]
  ndomCtr <<- archivesize
}
} else {
  return(0)
}

ixarc <- seq(ndomCtr)
ixndomarc <- NULL
rmove <- NULL
dup <- NULL
for (i in ixarc) {
  if (i %in% dup) {
    if(verbosity >= 3) cat("Duplicate Particle Found ",i,"\n")
    next
  }
  if (opt == 1) {
    x <- (t(matrix(rep(archiveFit[i,],ndomCtr),nrow=fncnt))-archiveFit[ixarc,] < 0.0)*1
  } else {
    x <- (t(matrix(rep(archiveFit[i,],ndomCtr),nrow=fncnt))-archiveFit[ixarc,] > 0.0)*1
  }
  rmove <- c(setdiff(which(rowSums(x) == 0),i),rmove)
  wich1 <- setdiff(which(rowSums(x)==0),i)
  if(length(wich1) != 0) {

```

```

if(!(i %in% wich1)){
  for (j in wich1) {
    if(all(archiveFit[j,] == archiveFit[i,])){
      dup <- c(j,dup)
    }
  }
}

}

remove <- union(remove,dup)
lenrem <- length(remove)
if(lenrem != 0){
  cor = setdiff(seq(ndomCtr),remove)
  ndomCtr <<- length(cor)
  archiveVar[1:ndomCtr,] <<- archiveVar[cor,]
  archiveFit[1:ndomCtr,] <<- archiveFit[cor,]
}

}

compute_velocity <- function(){

  top <- max(floor((ndomCtr*0.1)),1)
  gBest <- sample(1:top,1,replace=TRUE)
  gBestmat <- t(matrix(rep(archiveVar[gBest,],popsize),nrow=varcnt))
  randum1 <- matrix(runif(popsize*varcnt),popsize,varcnt)
  randum2 <- matrix(runif(popsize*varcnt),popsize,varcnt)
  velocity[1:popsize,] <<- w*velocity[1:popsize,] + c1*randum1*(pbestsVar-popVar) + c2*randum2*(gBestmat-
popVar)
  popVar[1:popsize,] <<- popVar[1:popsize,] + velocity[1:popsize,]

}

maintain_particles <- function(){

  for (j in seq(varcnt)) {

```

```

wich1 <- which(popVar[,j] < minvalue[j])
wich2 <- which(popVar[,j] > maxvalue[j])
popVar[wich1,j] <<- minvalue[j]
popVar[wich2,j] <<- maxvalue[j]
wich12 <- union(wich1,wich2)
velocity[wich12,j] <<- -1.0*velocity[wich12,j]
}

}

mutate <- function() {

if(t >= maxgen*pMut)
  return(0)

rangev <- vector(length=varcnt)
flip <- ifelse(runif(1) < (1-(t/(maxgen*pMut)))^1.5,1,0)
if(flip) {
  dim <- sample(1:varcnt,1,replace=TRUE)
  rangev[1:varcnt] <- (maxvalue[1:varcnt] - minvalue[1:varcnt])*0.5*(1-(t/(maxgen*pMut)))^1.5
  wich1 <- which((popVar[,dim] - rangev[dim]) < minvalue[dim])
  wich2 <- which((popVar[,dim] + rangev[dim]) > maxvalue[dim])
  if(length(wich1) != 0) {
    minvaluetemp[wich1,dim] <<- minvalue[dim]
    minvaluetemp[-wich1,dim] <<- popVar[-wich1,dim] - rangev[dim]
  } else {
    minvaluetemp[,dim] <<- popVar[,dim] - rangev[dim]
  }
  if(length(wich2) != 0) {
    maxvaluetemp[wich2,dim] <<- maxvalue[dim]
    maxvaluetemp[-wich2,dim] <<- popVar[-wich2,dim] + rangev[dim]
  } else {
    maxvaluetemp[,dim] <<- popVar[,dim] + rangev[dim]
  }
  popVar[,dim] <<- runif(popsizem, min=minvaluetemp[,dim], max=maxvaluetemp[,dim])
}

```

```
}  
  
}  
if(verbosity >= 1) cat("Welcome to MOPSO-CD\n")  
  
t1 <- Sys.time()  
t <- 0  
  
initialize_pop()  
initialize_vel()  
evaluate()  
store_pbests()  
insert_nondom()  
  
while (t < maxgen) {  
  t <- t + 1  
  
  if(verbosity >= 1)  
    cat("Generation ",t,"\n")  
  if (ndomCtr>2*fnct)  
    crowding()  
  
  compute_velocity()  
  maintain_particles()  
  mutate()  
  evaluate()  
  update_archive()  
  update_pbests()  
  
  if(verbosity >= 3){  
    cat("\nVariables\n")  
    print(archiveVar[1:ndomCtr,])  
    cat("Objective Function Values:\n")  
    print(archiveFit[1:ndomCtr,])  
    cat("\n")  
  }  
}
```



```

}
}

if(verbosity >= 2){
  cat("#####\n")
  cat("    FINAL VALUES\n")
  cat("#####\n")
  cat("Variables:\n")
  print(archiveVar[1:ndomCtr,])
  cat("\nObjective Function Values:\n")
  print(archiveFit[1:ndomCtr,])
}

if(verbosity >= 1){
  cat("\nNondominated Solutions: ",ndomCtr,"\n")
  cat("Number of Generations: ",t,"\n")
  cat("Computation Time: ",Sys.time() - t1,"\n")
}

if(!is.null(gn)) {
  pareto <-
list("paramvalues"=archiveVar[1:ndomCtr,],"objfnvalues"=archiveFit[1:ndomCtr,],"numsols"=ndomCtr,

"fn"=fn,"gn"=gn,"varcnt"=varcnt,"fncnt"=fncnt,"lowerbound"=lowerbound,"upperbound"=upperbound,"opt"=opt,
"popsize"=popsize,"maxgen"=maxgen,"archivesize"=archivesize,"pMut"=pMut,"w"=w,"c1"=c1,"c2"=c2)
} else {
  pareto <-
list("paramvalues"=archiveVar[1:ndomCtr,],"objfnvalues"=archiveFit[1:ndomCtr,],"numsols"=ndomCtr,

"fn"=fn,"gn"=NULL,"varcnt"=varcnt,"fncnt"=fncnt,"lowerbound"=lowerbound,"upperbound"=upperbound,"opt"=opt,
"popsize"=popsize,"maxgen"=maxgen,"archivesize"=archivesize,"pMut"=pMut,"w"=w,"c1"=c1,"c2"=c2)
}
warnings()
pareto$call <- match.call()
class(pareto) <- "pareto"
pareto
}

```

Código da otimização via MOPSO-CD

```

MOO_FH_SD2 <- function(x){

  E_Fa <- (b0_Fa + t(x)**b_Fa + t(x)**B_Fa**x)

  V_Fa <- (mse_Fa + t(g_Fa + t(D_Fa)**x)**(g_Fa + t(D_Fa)**x))

  Fa.rmse <- sqrt((E_Fa - T_Fa)^2 + V_Fa)

  E_Ra <- (b0_Ra + t(x)**b_Ra + t(x)**B_Ra**x)

  V_Ra <- (mse_Ra + t(g_Ra + t(D_Ra)**x)**(g_Ra + t(D_Ra)**x))

  Ra.rmse <- sqrt((E_Ra - T_Ra)^2 + V_Ra)

  E_Ront <- (b0_Ront + t(x)**b_Ront + t(x)**B_Ront**x)

  V_Ront <- (mse_Ront + t(g_Ront + t(D_Ront)**x)**(g_Ront + t(D_Ront)**x))

  Ront.rmse <- sqrt((E_Ront - T_Ront)^2 + V_Ront)

  z <- 4

  Db <- 18

  Dt <- 10

  Dh <- Db-Dt

  fza <- (x[1]*(0.2-0.1)/2 + 0.15)*10^-3

  fzt <- x[2]*(0.2-0.1)/2 + 0.15

  vc <- x[3]*(65-35)/2 + 50

  mrr <- -250*z*(Db^3/(Dh*Dt))*vc*(fza/fzt)*sqrt(fza^2 + (fzt*Dh/Db)^2)

  return(c(Fa.rmse,Ra.rmse,Ront.rmse, mrr))

}

source("mopsocd.R")

opt1 <- mopsocd(fn = MOO_FH_SD2,

  gn = g,

  varcnt = 3,

  fncnt = 4,

```

```
lowerbound = rep(-ro,3),  
upperbound = rep(ro,3),  
opt = 0,  
archivesize = 200)  
n1 <- opt1$numvals  
sol_opt1 <- opt1$objfnvalues  
colnames(sol_opt1) <- c("Fa.rmse", "Ra.rmse", "Ront.rmse", "mrr")  
niveis_opt1 <- opt1$paramvalues # niveis otimos de x  
colnames(niveis_opt1) <- c("fza", "fzt", "vc") # nomes das colunas  
library(plot3D)  
par(mfrow = c(1,1))  
f1.04.3d <- scatter3D(x = sol_opt1[,1],  
y = sol_opt1[,3],  
z = -sol_opt1[,4],  
colvar = sol_opt1[,2],  
main = "MOPSOCD",  
ticktype = "detailed",  
phi = 0, theta = 135,  
bty = "b2", pch = 20, cex = 1.5,  
xlab = "RMSE(Fa)", ylab = "RMSE(Ront)",  
clab = "RMSE(Ra)", zlab = "RMRR")
```

Código da otimização via NSGA-II

```
source("FH_SD_Fa.R")
source("FH_SD_Ra.R")
source("FH_SD_Ront.R")
opt2 <- nsga2(fn = MOO_FH_SD2,
             idim = 3,
             odim = 4,
             constraints = gg,
             cdim = 1,
             lower.bounds = rep(-ro,4),
             upper.bounds = rep(ro,4),
             popsize = 200, generations = 100)

sol_opt2 <- opt2$value

colnames(sol_opt2) <- c("Fa.rmse", "Ra.rmse", "Ront.rmse", "mrr")

niveis_opt2 <- opt2$par

colnames(niveis_opt2) <- c("fza", "fzt", "vc")

par(mfrow = c(1,1))
f1.04.3d <- scatter3D(x = sol_opt2[,1],
                    y = sol_opt2[,3],
                    z = -sol_opt2[,4],
                    colvar = sol_opt2[,2],
                    main = "NSGA-II",
                    ticktype = "detailed",
                    phi = 0, theta = 135,
                    bty = "b2", pch = 20, cex = 1.5,
                    xlab = "RMSE(Fa)", ylab = "RMSE(Ront)",
                    clab = "RMSE(Ra)", zlab = "RMRR")
```

