



SISTEMA ROBÓTICO AUTÔNOMO PARA MAPEAMENTO DE TAXAS DE
DOSE UTILIZANDO APRENDIZADO ATIVO DE PROCESSOS GAUSSIANOS
E COMPUTAÇÃO EVOLUCIONÁRIA

Paulo Cezar Rocha Silveira

Tese de Doutorado apresentada ao Programa de
Pós-graduação em Engenharia Nuclear, COPPE,
da Universidade Federal do Rio de Janeiro, como
parte dos requisitos necessários à obtenção do
título de Doutor em Engenharia Nuclear.

Orientadores: Roberto Schirru

Cláudio Márcio do Nascimento

Abreu Pereira

Rio de Janeiro
Setembro de 2017

SISTEMA ROBÓTICO AUTÔNOMO PARA MAPEAMENTO DE TAXAS DE
DOSE UTILIZANDO APRENDIZADO ATIVO DE PROCESSOS GAUSSIANOS
E COMPUTAÇÃO EVOLUCIONÁRIA

Paulo Cezar Rocha Silveira

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM
CIÊNCIAS EM ENGENHARIA NUCLEAR.

Examinada por:

Prof. Roberto Schirru, D.Sc.

Prof. Cláudio Márcio do Nascimento Abreu Pereira, D.Sc.

Prof. Eduardo Gomes Dutra do Carmo, D.Sc.

Prof. Ademir Xavier da Silva, D.Sc.

Prof. Celso Marcelo Franklin Lapa, D.Sc.

Prof. César Marques Salgado, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2017

Silveira, Paulo Cezar Rocha

Sistema Robótico Autônomo para Mapeamento de
Taxas de Dose Utilizando Aprendizado Ativo de
Processos Gaussianos e Computação Evolucionária /
Paulo Cezar Rocha Silveira. – Rio de Janeiro:
UFRJ/COPPE, 2017.

XIX, 138 p.: il.; 29,7 cm.

Orientadores: Roberto Schirru

Claudio Márcio do Nascimento Abreu
Pereira

Tese (doutorado) – UFRJ/ COPPE/ Programa de
Engenharia Nuclear, 2017.

Referências Bibliográficas: p.101-108.

1. Robótica. 2. Detecção. 3. Autônomo. 4. Inteligência
Artificial. I. Schirru, Roberto, *et al.*, II. Universidade
Federal do Rio de Janeiro, COPPE, Programa de
Engenharia Nuclear. III. Título.

DEDICATÓRIA

Este trabalho é dedicado a minha esposa
Cristina e aos meus filhos Filipe e Natália.

AGRADECIMENTOS

Ao professor e amigo Roberto Schirru, por sua orientação, ajuda e compreensão, sem as quais a realização desse trabalho não teria sido possível.

Ao professor e amigo Cláudio Márcio, que foi o grande responsável pelo êxito deste trabalho, que apesar dos seus inúmeros afazeres sempre encontra tempo para ajudar e orientar com uma incomparável boa vontade, contribuindo com suas ideias e com sua experiência acadêmica.

Ao aluno Danilo Naiff que contribuiu de forma significativa para a realização das simulações computacionais desta tese.

A minha esposa Cristina pelo apoio, compreensão e paciência por conta dos momentos difíceis ao longo desses cinco anos de trabalho.

Aos funcionários do IEN/CNEN, em especial ao Marcos Santana do laboratório de instrumentação e a Maria Isabel do gerenciamento de rejeitos pelo apoio na utilização de instalações e equipamentos.

Aos funcionários da secretaria do PEN/COPPE/UFRJ, por serem solícitos e atenciosos nos momentos de resolver trâmites administrativos referentes a curso.

Aos funcionários e amigos do Laboratório de Monitoração de Processos (LMP) do Programa de Engenharia Nuclear pelo apoio e a todos que contribuíram de forma direta ou indireta, na realização desse trabalho.

“As pessoas sempre põem a culpa nas circunstâncias por serem quem são. Não acredito em circunstâncias: os indivíduos de sucesso são aqueles que saem e procuram as condições que desejam; e, se não as encontram, criam-nas.”

George Bernard Shaw

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

SISTEMA ROBÓTICO AUTÔNOMO PARA MAPEAMENTO DE TAXAS DE
DOSE UTILIZANDO APRENDIZADO ATIVO DE PROCESSOS GAUSSIANOS
E COMPUTAÇÃO EVOLUCIONÁRIA

Paulo Cezar Rocha Silveira

Setembro/2017

Orientadores: Roberto Schirru

Claudio Márcio do Nascimento Abreu Pereira

Programa: Engenharia Nuclear

Este trabalho desenvolve um sistema robótico autônomo para inspeções e detecções em ambientes nucleares diversos, com capacidade de navegar por obstáculos e coletar informações através de sensoriamento embarcado. Para tal desenvolve um algoritmo computacional capaz de realizar o mapeamento de ambientes radioativos, com capacidade de reconstruir um campo de radiação através da leitura de taxas de dose fornecida via aquisição de dados de um detector de radiação. Para tanto apresenta uma nova aplicação para a metodologia de aprendizagem ativa de processos gaussianos em combinação com a computação evolucionária. Apresenta os resultados alcançados através de simulações computacionais para as diversas variações do algoritmo de mapeamento desenvolvido, bem como a construção do protótipo robótico concebido nos moldes da elaboração deste projeto.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

AUTONOMOUS ROBOTIC SYSTEM FOR MAPPING DOSE RATES USING
ACTIVE LEARNING OF GAUSSIAN PROCESSES AND EVOLUTIONARY
COMPUTATION

Paulo Cezar Rocha Silveira

September/2017

Advisors: Roberto Schirru

Claudio Márcio do Nascimento Abreu Pereira

Department: Nuclear Engineering

This work develops an autonomous robotic system for inspections and detections in diverse nuclear environments, with ability to navigate obstacles and collect information through embedded sensing. To this end, it develops a computational algorithm capable of performing the mapping of radioactive environments, with the ability to reconstruct a radiation field through the reading of dose rates provided via the acquisition of data from a radiation detector. For this, it presents a new application for the active learning methodology of Gaussian processes in combination with evolutionary computation. It presents the results achieved through computational simulations for the various variations of the mapping algorithm developed, as well as the construction of the robotic prototype conceived in the molds of the elaboration of this project.

ÍNDICE

CAPÍTULO 1. INTRODUÇÃO.....	01
1.1. Relevância.....	05
1.2. Originalidade.....	07
1.3. Organização do Trabalho.....	09
CAPÍTULO 2. FUNDAMENTAÇÃO TEÓRICA.....	11
2.1. Complexão Robótica.....	11
2.1.1. Conceitos de Robótica Autônoma.....	12
2.1.1.1 Arquiteturas Deliberativas.....	13
2.1.1.2 Arquiteturas Reativas.....	14
2.1.1.3 Arquiteturas Híbridas.....	15
2.2. Inteligência artificial e seus Aspectos.....	16
2.2.1. Algoritmos Evolucionários.....	18
2.2.1.1 Evolução diferencial.....	19
2.2.2. Sistemas Especialistas.....	21
2.2.3. Processo Gaussiano.....	23
2.2.4. <i>Active Learning</i>	25
CAPÍTULO 3. PROJETO BASE E SUAS ETAPAS.....	27
3.1. Considerações do projeto.....	27

3.2. Visão geral do projeto.....	28
CAPITULO 4. ALGORITMO DE RECONSTRUÇÃO DE SUPERFÍCIE.....	35
4.1. Holística do Problema.....	35
4.2. Abordagem Metodológica.....	36
4.3. Formulação de problema.....	36
4.4. Descritivo Operacional do Algoritmo computacional.....	40
4.4.1 Evolução do Algoritmo Computacional.....	43
4.5. Modelagem do Sistema.....	47
CAPITULO 5. DESEMPENHO DO ALGORITMO DE RECONSTRUÇÃO DE SUPERFÍCIE.....	51
5.1. Critério de penalização e suas variações.....	51
5.2. Performance do ARS 4P SP.....	53
5.3. Performance do ARS 6P SP.....	58
5.4. Performance do ARS 8P SP.....	63
5.5. Performance do ARS 4P Wp80 Lp4.....	67
5.6. Performance do ARS 6P Wp40 Lp8.....	72
5.7. Performance do ARS 8P Wp40 Lp4.....	77
5.8. Resultados comparativos.....	82
CAPITULO 6. PROTOTIPAGEM E RESULTADOS.....	86
6.1. Visão geral do projeto.....	86

6.2. Estrutura mecânica.....	87
6.3. Eletrônica embarcada.....	88
6.4. Programação operacional.....	94
CONCLUSÕES.....	97
REFERÊNCIAS.....	101
ANEXOS.....	109

LISTA DE FIGURAS

Figura 1. Pseudocódigo da evolução diferencial.....	19
Figura 2. Estrutura Básica de um Sistema Especialista.....	22
Figura 3. Liberdade de movimentos.....	28
Figura 4. Diagrama geral do funcionamento.....	29
Figura 5. a) Pontos de medição de tx de dose; b) Levantamento Radiométrico do depósito de rejeitos; c) Planta do depósito de rejeitos radioativos.....	31
Figura 6. Área de cobertura do sensoriamento de proximidade.....	32
Figura 7. Simulação computacional de deslocamento do robô.....	33
Figura 8. Placa de desenvolvimento de testes e simulações.....	34
Figura 9. Função de penalização.....	40
Figura 10. Pseudocódigo ARS.....	40
Figura 11. Fluxograma de sequenciamento de ações.....	42
Figura 12. Distribuição de probabilidade do ponto inicial.....	43
Figura 13. Busca pelo ponto de maior entropia.....	44
Figura 14. Distribuição de probabilidade baseada em dois pontos.....	44
Figura 15. Distribuição de probabilidade baseada em cinco pontos.....	45
Figura 16. Distribuição de probabilidade baseada em dez pontos.....	45
Figura 17. Distribuição de probabilidade baseada em vinte pontos.....	46

Figura 18. Função de otimização penalizada.....	46
Figura 19. Superfície de quatro picos de máximo.....	48
Figura 20. Superfície de seis picos de máximo.....	49
Figura 21. Superfície de oito picos de máximo.....	50
Figura 22. Comparativo do erro estimado de dose entre as rodadas do ARS 4P.....	54
Figura 23. Comparativo de tempos e delineamento do trajeto do robô com o ARS 4P SP.....	54
Figura 24. Resultado da 1ª rodada do ARS 4P SP após 75 avaliações.....	55
Figura 25. Resultado da 2ª rodada do ARS 4P SP após 75 avaliações.....	55
Figura 26. Resultado da 3ª rodada do ARS 4P SP após 75 avaliações.....	55
Figura 27. Evolução do ARS 4P SP.....	57
Figura 28. Comparativo do erro estimado de dose entre as rodadas do ARS 6P SP.....	58
Figura 29. Comparativo de tempos e delineamento do trajeto do robô com o ARS 6P SP.....	59
Figura 30. Resultado da 1ª rodada do ARS 6P SP após 125 avaliações.....	59
Figura 31. Resultado da 2ª rodada do ARS 6P SP após 125 avaliações.....	60
Figura 32. Resultado da 3ª rodada do ARS 6P SP após 125 avaliações.....	60
Figura 33. Evolução do ARS 6P SP.....	62
Figura 34. Comparativo do erro estimado de dose entre as rodadas do ARS 8P SP.....	63

Figura 35. Comparativo de tempos e delineamento do trajeto do robô com o ARS 8P SP.....	64
Figura 36. Resultado da 1ª rodada do ARS 8P após 150 avaliações.....	64
Figura 37. Resultado da 2ª rodada do ARS 8P após 150 avaliações.....	65
Figura 38. Resultado da 3ª rodada do ARS 8P SP após 150 avaliações.....	65
Figura 39. Evolução do ARS 8P SP.....	66
Figura 40. Comparativo do erro estimado de dose entre as rodadas do ARS 4P Wp80 Lp4.....	67
Figura 41. Comparativo de tempos e delineamento do trajeto do robô com o ARS 4P Wp80 Lp4.....	68
Figura 42. Resultado da 1ª rodada do ARS 4P Wp80 Lp4 após 75 avaliações.....	68
Figura 43. Resultado da 2ª rodada do ARS 4P Wp80 Lp4 após 75 avaliações.....	69
Figura 44. Resultado da 3ª rodada do ARS 4P Wp80 Lp4 após 75 avaliações.....	69
Figura 45. Evolução do ARS 4P Wp80 Lp4.....	71
Figura 46. Comparativo do erro estimado de dose entre as rodadas do ARS 6P Wp40 Lp8.....	72
Figura 47. Comparativo de tempos e delineamento do trajeto do robô com o ARS 6P Wp40 Lp8.....	73
Figura 48. Resultado da 1ª rodada do ARS 6P Wp40 Lp8 após 75 avaliações.....	73

Figura 49. Resultado da 2ª rodada do ARS 6P Wp40 Lp8 após 75 avaliações.....	74
Figura 50. Resultado da 3ª rodada do ARS 6P Wp40 Lp8 após 75 avaliações.....	74
Figura 51. Evolução do ARS 6P Wp40 Lp8.....	76
Figura 52. Comparativo do erro estimado de dose entre as rodadas do ARS 8P Wp40 Lp4.....	77
Figura 53. Comparativo de tempos e delineamento do trajeto do robô com o ARS 8P Wp40 Lp4.....	78
Figura 54. Resultado da 1ª rodada do ARS 8P Wp40 Lp4 após 125 avaliações.....	78
Figura 55. Resultado da 2ª rodada do ARS 8P Wp40 Lp4 após 125 avaliações.....	79
Figura 56. Resultado da 3ª rodada do ARS 8P Wp40 Lp4 após 125 avaliações.....	79
Figura 57. Evolução do ARS 8P Wp40 Lp4.....	81
Figura 58. Estrutura do sistema de tracionamento.....	87
Figura 59. Servomotor em visão explodida.....	88
Figura 60. Encoder incremental AB.....	88
Figura 61. Raspberry PI B.....	89
Figura 62. Placa de interface de sensoriamento.....	89
Figura 63. Placa de driver de motores.....	90

Figura 64. Sensor de proximidade.....	91
Figura 65. Sensor de radiação.....	91
Figura 66. Diagrama elétrico do circuito adequador de sinais para o sensor de radiação.....	92
Figura 67. Placa reguladora de tensão.....	92
Figura 68. Pseudocódigo do algoritmo de navegação.....	96

LISTA DE TABELAS

Tabela 1. Variações dos métodos do ARS.....	52
Tabela 2. Média dos resultados do ARS 4P SP após 75 avaliações.....	55
Tabela 3. Média dos resultados do ARS 4P SP após 150 avaliações.....	56
Tabela 4. Média dos resultados do ARS 6P SP após 125 avaliações.....	60
Tabela 5. Média dos resultados do ARS 6P SP após 150 avaliações.....	61
Tabela 6. Média dos resultados do ARS 8P SP após 150 avaliações.....	65
Tabela 7. Média dos resultados do ARS 4P Wp80 Lp4 após 75 avaliações.....	69
Tabela 8. Média dos resultados do ARS 4P Wp80 Lp4 após 150 avaliações.....	70
Tabela 9. Média dos resultados do ARS 6P Wp40 Lp8 após 75 avaliações.....	74
Tabela 10. Média dos resultados do ARS 6P Wp40 Lp8 após 150 avaliações.....	75
Tabela 11. Média dos resultados do ARS 8P Wp40 Lp4 após 125 avaliações.....	79
Tabela 12. Média dos resultados do ARS 8P Wp40 Lp4 após 150 avaliações.....	80
Tabela 13. Comparativo dos métodos experimentados para a topologia de quatro picos.....	82
Tabela 14. Comparativo dos métodos experimentados para a topologia de seis picos	83
Tabela 15. Comparativo dos métodos experimentados para a topologia de oito picos	83
Tabela 16. Apresentação do melhor desempenho para a topologia de quatro picos.....	84

Tabela 17. Apresentação do melhor desempenho para a topologia de seis picos.....	85
Tabela 18. Apresentação do melhor desempenho para a topologia de oito picos.....	85
Tabela 19. Dados técnicos dos sensores.....	93
Tabela 20. Biblioteca de controle do robô.....	94
Tabela 21. Atribuição de parâmetros computacionais.....	95
Tabela 22. Declarações do domínio do problema.....	95

LISTA DE SIGLAS E ABREVIATURAS

A/D	Conversor Analógico/Digital
AL	Active Learning
ANEEL	Agência Nacional de Energia Elétrica
ARM	Advanced RISC Machine
ARS	Algoritmo de Reconstrução de Superfície
ATO	Algoritmo de Transposição de Obstáculo
CI	Circuito Integrado
CNAAA	Central Nuclear Almirante Álvaro Alberto
CNEN	Comissão Nacional de Energia Nuclear
ED	Evolução Diferencial
EPI	Equipamentos de Proteção Individual
GP	Gaussian Process
GPIO	General Purpose Input/Output
IA	Inteligência Artificial
IEN	Instituto de Engenharia Nuclear
IR	Infravermelho
IAEA	Agência de Energia Atômica do Japão
NASA	National Aeronautics and Space Administration

ROV	Remotely Operated Vehicle
TEPCO	Universidade de Tohoku e pela Tóquio Electric Power Company
OCDE	Organização para a Cooperação e Desenvolvimento Econômico
LARO	Laboratório de robótica
LBFGS	Interface para algoritmos de minimização para funções multivariadas
LMP	Laboratório de Monitoração de Processos
NIter	Numero de iterações
SE	Sistemas Especialistas
Modbus	Protocolo de comunicação de dados
POD	Estrutura onde se reúne o sensoriamento embarcado
PPR	Pulsos por Revolução
Python	Linguagem de programação
RCB	Relação Custo Benefício
SoC	System on a Chip
UFRJ	Universidade Federal do Rio de Janeiro

Capítulo 1

INTRODUÇÃO

Dentro de uma visão holística, a construção de um robô dos tempos modernos requer vastos conhecimentos sobre eletrônica, mecânica e computação. Muitos anos de pesquisa e dedicação se fazem necessários para se produzir protótipos viáveis de serem aplicados comercialmente (Silveira, 2012).

O advento da microeletrônica pode ser considerado como grande responsável pelo monumental avanço no desenvolvimento de tecnologias na fabricação de robôs. A miniaturização de componentes eletrônicos proporcionou o desenvolvimento de processadores mais velozes e a criação de uma vasta gama de microcontroladores, que por sua vez, alavancaram o desenvolvimento da informática atual. Juntos contribuíram para a modernização e criação de novas técnicas de aplicação da conhecida mecânica clássica, renovando-a e tornando-a mais adequada aos atuais padrões tecnológicos de exigência.

Apesar dos inúmeros tipos de robôs disponíveis atualmente para diversas aplicações, foi no ramo industrial que se deu início a grande demanda das últimas décadas. A maior parte dos robôs vendidos no mundo é voltada para a indústria, seja ela de transformação, têxtil, cosméticos, alimentícia e etc.

Não há hoje, um único instituto de tecnologia renomado, que não desenvolva ou efetue estudos no desenvolvimento de algum tipo de sistema robótico. Diversas linhas de ação vêm sendo traçadas no intuito de desenvolver aplicações que facilitem a vida do ser humano. Robôs que façam tarefas básicas de limpeza e conservação, robôs que ajudem um cego a atravessar uma rua, robôs que explorem vulcões, geleiras,

profundidades abissais e o espaço sideral. Não há limites para as possibilidades de aplicação que podem ser criadas com esta nova onda de tecnologia. No entanto, muitas destas aplicações ainda continuarão por um bom tempo a povoar apenas a imaginação dos homens. Ainda será necessário trilhar um longo caminho de pesquisa e desenvolvimento de novos materiais e técnicas, que proporcionem resolver problemas robóticos do tipo locomoção, autonomia energética e um grau de inteligência razoável (Silveira, 2012).

Na tentativa de resolver estes tipos de problemas, há tempos cientistas vêm percebendo que a melhor solução para os problemas encontrados está na observação da natureza. Tudo que temos nela já vêm sendo testado e aprovado nos últimos milhões de anos. Então, basta procurar o melhor modelo para cada tipo de aplicação, ou se for o caso, mesclar modelos de acordo com a necessidade. E a partir daí desenvolver os estudos de cinemática e processamento que proporcionem a criação de um modelo mecatrônico da natureza (Werger, 2003).

Ao tomar por base a evolução natural dos seres vivos para aplicações robóticas, percebeu-se que não apenas soluções para padrões de locomoção poderiam ser analisadas, como também, a forma de se raciocinar e de tomar decisão poderiam ser pareadas.

Diversos tipos de controle vêm sendo desenvolvidos no intuito de proporcionar um grau de inteligência na navegação autônoma de sistemas robóticos. Conceitos como o Deliberativo e o Reativo vêm sendo implementados com êxito em robôs exploradores desenvolvidos pela NASA. O método Deliberativo permite que um veículo robótico se desloque com base em um mapeamento pré-programado, enquanto o método Reativo proporciona que o veículo robótico reaja

com o ambiente de navegação aprendendo e memorizando suas melhores soluções. Estes métodos são possíveis graças ao desenvolvimento de técnicas baseadas no comportamento humano, conhecidas como teoria comportamental (Werger, 2003). A principal característica desta teoria está no emprego de técnicas de inteligência artificial como a lógica *fuzzy* (Ross, 2015) e as redes neurais artificiais (Haykin, 2008). Recursos como estes têm conseguido bons resultados na análise de informações e tomada de decisões em sistemas computacionais robóticos.

Os avanços nas pesquisas de desenvolvimento de técnicas de inteligência artificial vêm proporcionando o desenvolvimento de robôs mais complexos com a capacidade de realizar tarefas com um grau de qualidade aceitável na substituição humana.

A robótica tem sido utilizada com sucesso em ambientes insalubres e inóspitos de exploração, tais como: espacial (Katz, 2003), vulcânica (Muscato, 2003), submarina (Bingham, 2010), cavernas (Morris, 2006). Outro, não menos, importante uso de robôs é substituir humanos em Situações de risco que possam afetar a sua integridade física, tais como o ambiente sujeito a exposição química ou nuclear.

Atualmente existem protótipos robóticos sendo aplicados no campo nuclear, tais como: robôs de inspeção de escalador de parede (Briones, 1994); o *Robug* IIS (Luk, 2005), que é um veículo robótico em pernas para superar obstáculos em terreno mais complexo; *Robot Snake* (Buckingham, 2005) utilizados para reparar tubulações nucleares; os robôs coreanos Kaerot (Kim, 2010) utilizados para inspeção e detecção em instalações nucleares; robôs subaquáticos (Nawaz, 2009) para inspeção e detecção de resíduos nucleares e o veículo robótico denominado

Equipa Nippon (Ohno, 2010) projetado para medir a radiação na Central Nuclear de Fukushima Daiichi, desenvolvido pela Agência de Energia Atômica do Japão (JAEA), Universidade de Tohoku e pela Tóquio Electric Power Company (TEPCO).

Apesar da grande diversidade robótica disponível, poucas são as Usinas Nucleares que utilizam robôs na substituição de tarefas que apresentem grande risco aos seus profissionais, como detecções de vazamentos, inspeções e medições rotineiras de ambientes radioativos e depósitos de rejeitos.

Mesmo durante o funcionamento normal, uma central nuclear exige alguns reparos e manutenções regulares que envolvem riscos de irradiação e contaminação dos trabalhadores. A fim de proteger os trabalhadores na central nuclear, todos os países têm leis e regulamentos que estabelecem os limites de dose por ano. Esta situação exige profissionais especializados e em constante formação, uma vasta gama de equipamentos de proteção individual, vários sistemas e equipamentos envolvendo a segurança coletiva do ambiente de trabalho. O uso da inspeção e detecção de robôs não só diminui os custos (profissionais especializados, equipamentos e sistemas), como também aumenta a proteção individual, evitando a exposição humana à irradiação desnecessária ou prolongada.

Desta forma, este projeto apresenta o desenvolvimento de um veículo robótico autônomo para inspeção, detecção e monitoramento de variáveis como radiação, temperatura, Infravermelho e gases em ambientes nucleares fabris ou de geração de energia e seus depósitos de rejeitos nucleares. Um dos principais objetivos do robô é poder ser utilizado como uma ferramenta de desenvolvimento e pesquisa (*Facility*) para investigações da utilização de robôs em ambientes nucleares.

Para tal, a proposta *facility* apresenta, além dos dispositivos de controle, operação, monitoração e comunicação, a possibilidade de programação por meio de uma linguagem livre e multiplataforma, bem difundida no ambiente acadêmico científico e que favorece o desenvolvimento de algoritmos complexos, outrossim, proporciona um ádito direto a biblioteca (*driver*) de acesso de todo o *hardware* disponível o que torna sua utilização mais facilitada para futuros desenvolvedores.

1.1 Relevância

Atualmente no mundo existem cerca de 440 Centrais nucleares em operação e mais 33 sendo construídas, ao todo são 31 países utilizando de energia nuclear para seu desenvolvimento. Embora tenha começado a ser utilizada com mais ênfase nos anos 70, já se tornou a segunda maior fonte de energia elétrica nos países da OCDE (Organização para a Cooperação e Desenvolvimento Econômico). A geração de eletricidade nuclear se iguala ao total da energia produzida pelas Centrais Hidroelétricas e a gás, são aproximadamente 17% da geração elétrica mundial. (Scientif American, 2012)

Uma considerável parcela dos países desenvolvidos utiliza um percentual elevado de energia elétrica gerada por meio de centrais nucleares. São eles: França 78%, Bélgica 57%, Japão 39%, Coréia do Sul 39%, Alemanha 30%, Suécia 46% e Suíça 40%. Somente nos Estados Unidos existem 104 reatores em funcionamento, gerando 20% de toda energia elétrica consumida no país, são aproximadamente 105.664 MW produzidos. Para fins de comparação o valor total produzido pelo sistema de geração

elétrica brasileiro é de 104.390 MW, segundo ANEEL (Agência Nacional de Energia Elétrica).

Além dos reatores utilizados em centrais de geração elétrica, existem 284 reatores em funcionamento dentro de centros de pesquisa distribuídos por 56 países, sem contar um número estimado de 220 reatores de propulsão em navios e submarinos (Scientif American, 2012).

A matriz energética brasileira possui hoje duas usinas em operação localizadas na CNAAA (Central Nuclear Almirante Álvaro Alberto) em Angra dos Reis, município do Estado do Rio de Janeiro, com potência instalada total de 1.990 MW (ELETRONUCLEAR, 2014). No entanto, em 2008, foi anunciado o Plano Nacional de Energia 2030, que previa a construção de pelo menos quatro novos empreendimentos nucleares com potência de 1.000 MW cada, dois no Nordeste e dois no Sudeste, além da finalização da usina de Angra 3, com capacidade de fornecer 1.405 MW de potência. As usinas seriam construídas duas a duas e cada central teria capacidade de abrigar um total de seis reatores.

Ao analisar o cenário mundial atual, percebe-se a necessidade do desenvolvimento de pesquisas que possibilitem melhorias nas condições de operação e segurança dos diversos tipos de utilização da energia nuclear. É de extrema relevância pesquisas que busquem maximizar a segurança da operação e do trabalhador em plantas nucleares.

Este trabalho visa desenvolver um sistema robótico autônomo capaz de minimizar o emprego de profissionais da área nuclear em tarefas de medições, detecções e levantamentos de taxa de dose em ambientes radioativos, evitando a

exposição humana à irradiação desnecessária ou prolongada, reduzindo os custos com equipamentos de proteção Individual (EPI) e com equipamentos especializados. A utilização de robótica autônoma ainda pode proporcionar um aumento significativo na eficiência, pois o robô não precisa obedecer a um tempo de repouso e nem a um tempo de exposição máxima, podendo fornecer leituras de medições constantemente atualizadas 24hs por dia.

1.2 Originalidade

Atualmente diversas linhas de pesquisa em desenvolvimento de *softwares* para inteligência computacional, esbarram na dificuldade de projetar o *hardware* dedicado a elaboração específica de projeto que envolva robótica aplicada. Quando o desenvolvimento está voltado para área nuclear esta dificuldade aumenta significativamente em virtude das peculiaridades do ambiente.

O presente trabalho foi motivado inicialmente pela necessidade de se criar uma ferramenta (*facility*) para o desenvolvimento de projetos e pesquisas na área de robótica autônoma baseada em inteligência computacional para aplicação em ambientes nucleares.

O foco principal desta tese é o desenvolvimento de um sistema robótico autônomo, que seja capaz de realizar inspeções e detecções em ambientes nucleares diversos, navegando através de obstáculos e coletando informações através de sensoramento, remotamente sem a intervenção humana.

Para tal, foi desenvolvido um algoritmo de mapeamento de radiação em linguagem *Python*, com a finalidade de reconstruir um campo de radiação através da utilização de um método de regressão por processo Gaussiano (do inglês, *Gaussian Process Regression*), doravante designado por *GP*, conjuntamente com um sistema de aprendizado ativo, do inglês, *Active Learning (AL)*, que utiliza otimização auxiliar baseada em Evolução Diferencial (ED).

Esta tese apresenta uma nova forma de construção de algoritmos de busca e navegação para sistemas robóticos móveis, baseadas em uma estrutura de programação ainda não explorada para estes fins.

Tal abordagem é original não apenas pelo fato de ser uma primeira abordagem do emprego de robótica móvel autônoma para reconstrução de perfis de taxas de doses em um ambiente de detecção nuclear, mas também do ponto de vista da adaptação e aplicação dos métodos matemáticos de *GP* em conjunto com técnica de *AL*, embarcado em um sistema robótico para tal finalidade.

Parte dos resultados obtidos nos estudos e investigações feitos nesta tese encontram-se publicados em importantes veículos científicos de circulação internacional (Naiff, Silveira e Pereira, 2017 e Silveira, Naiff, Pereira e Schirru, 2017). Tais resultados corroboram viabilidade e eficiência da utilização da abordagem proposta nesta tese.

Acredita-se que o aprofundamento desta pesquisa, possa permitir uma visão da utilização de sistemas robóticos autônomos em aplicações profissionais atuais e futuras, no intuito de buscar soluções mais eficientes para problemas de detecção, mapeamento

e monitoração em geral, em especial os da área nuclear, que atualmente encontram-se sem solução definitiva.

1.3 Organização do Trabalho

No presente capítulo foi apresentada uma breve descrição do cenário atual quanto à utilização de soluções robóticas para problemas nucleares, buscando-se justificar a necessidade deste projeto, sua relevância e motivação, bem como expor seu objetivo e indicar sua originalidade.

No Capítulo 2 são apresentados os principais conceitos de Robótica Autônoma utilizados nesta tese, bem como, uma explicação geral sobre os conceitos de Inteligência Artificial, Algoritmos Evolucionários, Sistema Especialista, Modelo estatístico *Gaussian process* e o Método *Active Learning*.

No Capítulo 3 são apresentados os métodos e conceitos aplicados durante a abordagem metodológica utilizada na modelagem do sistema.

O Capítulo 4 apresenta a elaboração de um algoritmo computacional, desenvolvido com a finalidade de reconstruir uma superfície de radiação, detalhando as metodologias utilizadas e a modelagem do sistema.

O capítulo 5 aborda as variações do algoritmo de reconstrução de superfície, apresentando os resultados computacionais de desempenho de cada método utilizado e um comparativo do desempenho alcançado.

O Capítulo 6 trata da elaboração do protótipo robótico real, apresentando em detalhes sua construção mecânica, o *hardware*, o *software* e o *firmware* utilizados.

O Capítulo 7 conclui o trabalho e apresenta as oportunidades para futuras melhorias.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é apresentar uma breve descrição dos principais fundamentos teóricos que embasaram o desenvolvimento desta tese. Inicialmente serão introduzidos os principais aspectos do funcionamento de um sistema robótico autônomo, bem como, seus principais tipos e conceitos importantes para o entendimento de estruturas fundamentais da robótica moderna. Em seguida serão apresentados os principais fundamentos da Inteligência Artificial e da Computação Evolucionária, além de uma breve descrição de Sistema Especialista.

2.1 Complexão Robótica

A robótica é uma área da engenharia que engloba diferentes tópicos de conhecimentos da mecânica, da eletrônica e da computação. Na mecânica são levados em consideração os estudos de dimensionamento estrutural, cálculo de engrenagens, resistência de materiais, entre outros. A eletrônica utilizada na robótica reúne conhecimentos da eletrônica analógica, digital, processadores e microcontroladores. Em computação, englobam-se diferentes tópicos que podem variar desde a programação de rotinas de funcionamento até o desenvolvimento de técnicas de inteligência artificial. A escolha adequada do tipo de controle de funcionamento irá determinar a escolha do *software* adequado em relação ao *hardware* desenvolvido para o sistema robótico. Muitas das vezes, por limitação de hardware ou por não haver tecnologia atual suficiente para a implementação de certos modelos de desenvolvimento, alguns robôs são construídos apenas dentro de um ambiente de simulação computacional.

Os conceitos interdisciplinares empregados no desenvolvimento de um robô visam à resolução de problemas como: cinemática, dinâmica, controle, odometria, mapeamento, heurísticas, autonomia e sensoriamento. Estes conceitos são estudados em disciplinas acadêmicas como: Eletrônica avançada, Robótica Industrial, Telerobótica, Sensores e detectores, Interação Homem-máquina, Computação avançada, Sistemas inteligentes, Engenharia de software, Cinemática Robótica, Elementos de máquinas e Resistência de materiais.

2.1.1 Conceitos de Robótica Autônoma

O desenvolvimento de um robô autônomo demanda conhecimentos de diversas áreas da engenharia principalmente da eletrônica, mecânica e computação. No entanto, uma definição compartilhada por (Grassi Jr, 2006) e (Arkin, 1998) afirma que: “uma arquitetura para robôs está mais relacionada a uma arquitetura de *software*, e não tanto à parte de *hardware* do sistema de controle”.

A arquitetura de construção de um software voltado para o controle inteligente de um robô, nada mais é que uma maneira de descrever as etapas que fazem parte da formulação do seu “raciocínio” e como elas interagem entre si.

O tipo de arquitetura a ser implementada, está diretamente ligado a forma de “raciocinar” do robô e deve ser escolhido baseado na aplicação que pretendemos dar ao robô, uma vez que o desencadeamento deste “raciocínio” ocorre mediante um estímulo externo, seja ele uma leitura de sensores ou uma entrada de dados de especificação de objetivo. No universo da robótica autônoma, esta forma de raciocinar encontra-se dividida em três classificações: Deliberativa, Reativa e Híbrida.

2.1.1.1 Arquiteturas Deliberativas

Arquitetura onde o processo de tomada de decisão do ser humano é descrito para um robô, de forma que uma determinada atividade possa ser executada (Grassi Jr, 2006).

O emprego de raciocínio deliberativo exige que o sistema de controle do robô possua o conhecimento completo do mundo, armazenado em um modelo interno do seu sistema, que lhe permita a partir desse conhecimento prévio do ambiente e das informações sensoriais do robô prever o resultado de suas ações e otimizar o seu desempenho relativamente ao seu modelo do mundo.

O raciocínio deliberativo fundamenta-se na confiabilidade, segurança e coerência do conhecimento de sua base de dados, uma vez que requer um forte modelo de hipóteses sobre o mundo (Pieri, 2002). Informações imprecisas, confusas ou não atualizadas, causam sérios erros no resultado do raciocínio, pois o ambiente ao redor do robô pode ser muito dinâmico, onde os objetos podem rapidamente se mover fazendo com que as informações de posição não sejam mais válidas (Arkin, 1998).

As formas mais usuais de um robô representar o modelo do mundo são: o modelo geométrico e modelo simbólico. O modelo geométrico é geralmente utilizado quando se quer que o robô execute atividades de movimentação de forma autônoma, sem colidir com obstáculos representados em regiões previamente definidas, neste modelo o ambiente é representado espacialmente por regiões de espaços livres e regiões com obstáculos (Grassi Jr, 2006). No modelo simbólico não há um mapeamento geométrico do ambiente, a movimentação do robô baseia-se em aplicação de lógica,

geralmente são utilizados conceitos de inteligência artificial para o cálculo de movimentos e trajetórias.

2.1.1.2 Arquiteturas Reativas

A arquitetura reativa é composta por uma série de comportamentos que relacionam condições sensoriais a um conjunto de ações do robô, além de ser dotada de métodos que possibilitam a correta coordenação destes comportamentos (Pieri, 2002). Esta arquitetura evita a utilização de um modelo interno do mundo. Para (Brooks, 1991) “o mundo é a melhor representação dele mesmo”. E por não utilizar um modelo interno do ambiente, este tipo de arquitetura, normalmente interpreta as características do ambiente que podem ser do interesse do robô (Santos, 2009).

O principal objetivo das arquiteturas reativas é possibilitar a implementação de sistemas de controle que respondam rapidamente a uma variedade de ocorrências ou situações no ambiente, fazendo com que robôs operem em ambientes altamente dinâmicos. Esta rapidez se deve a simplicidade no tratamento das informações sensoriais e a forma direta pela qual a percepção, ou estímulo, está associado com uma ação, ou resposta (Grassi Jr, 2006).

A coordenação em uma arquitetura reativa pode ser feita de forma competitiva ou cooperativa. Na coordenação competitiva, dos comportamentos ativos em um dado momento, apenas um deles prevalece, dentro de uma hierarquia ou arbitragem, determinando a ação que o robô deve realizar. Já na coordenação cooperativa, todos os comportamentos ativos contribuem para determinar a ação do robô (Santos, 2009).

2.1.1.3 Arquiteturas Híbridas

Este tipo de arquitetura é a que predomina atualmente, ela utiliza deliberação para planejar as ações do robô a partir de uma representação interna do conhecimento do mundo, de forma que os objetivos do robô possam ser atingidos eficientemente. Quando as ações já foram planejadas, a execução do plano gerado é feita utilizando-se reatividade, que responde em alta velocidade a mudanças dinâmicas no ambiente. Assim, a arquitetura híbrida busca ser apropriada para solução de problemas complexos atingindo objetivos de maneira ótima e eficiente, através do uso da deliberação, e em ambientes dinâmicos que exigem rapidez na resposta através de comportamentos reativos (Santos, 2009).

A arquitetura híbrida é responsável por definir a função da parte deliberativa e reativa dentro do sistema de controle inteligente do robô. Também define como cada uma destas partes está organizada, onde e como é feita a interface de coordenação entre deliberação e reação dentro do sistema (Grassi Jr, 2006).

Em uma outra visão comparativa, a arquitetura híbrida corresponde a uma arquitetura reativa controlada por um plano de execução e de sequenciamento de comportamentos. Através do agrupamento da habilidade de raciocínio que se baseiam em modelos internos do mundo (deliberação), estas arquiteturas permitem a reconfiguração dinâmica de sistemas de controle reativo (Ribeiro, 2001).

2.2 Inteligência artificial e seus Aspectos

A definição de Inteligência Artificial (IA) a primeiro momento pode parecer uma tarefa simples e de analogia direta a escrita do nome. Porém, torna-se difícil citar de uma forma acadêmica, tendo em vista o fato de não haver um senso comum entre seus estudiosos. No entanto, uma definição que parece delimitar bem o conceito de IA é: Capacidade de um dispositivo em simular a aptidão humana de raciocinar, perceber, aprender, tomar decisões e resolver problemas (Silveira, 2012).

A inteligência artificial é um ramo da ciência da computação que se dedica a pesquisar métodos de resolução de problemas, com capacidade similar ou superior ao raciocínio do ser humano (Luger, 2004). Podendo ser definida também como a área da ciência da computação que se dedica ao estudo do comportamento inteligente ou ainda, o estudo de como conceber programas computacionais que consigam realizar tarefas que, atualmente, os humanos fazem melhor (Rich, 1994).

Elaborar uma definição de inteligência que englobe toda a complexidade de raciocínio humano nunca foi uma tarefa fácil, no entanto consegue-se com facilidade identificar a inteligência quando a se vê. Isto se torna ainda mais complexo quando tenta-se traduzir computacionalmente o conceito de inteligência.

O sentido original dado ao termo Inteligência Artificial segundo McCarthy foi: “a capacidade de uma máquina realizar funções que, se realizadas pelo ser humano, seriam consideradas inteligentes”. É um conceito vasto que agrega tantas definições quanto concedemos significados diferentes à palavra Inteligência (Russell, 2003).

De uma forma holística pode-se perceber diversas características humanas agregadas ao conceito de inteligência. Destas as que se apresentam com maior

importância é a capacidade de raciocinar, a capacidade de aprender e a de reconhecer padrões. Para poder transladar estas capacidades para uma modelagem computacional, é preciso definir formas de realização de tarefas análogas a estas capacidades. De uma forma simples pode-se: i) associar o raciocínio à aplicação de regras lógicas sobre um conjunto de dados disponíveis para uma conclusão; ii) associar a aprendizagem à avaliação da aplicação de regras lógicas, guardando o resultado de erros e acertos, de forma não repetir o mesmo erro em uma futura aplicação de regras para uma mesma situação; iii) associar o reconhecimento à identificação de padrões matemáticos.

Diferentemente de sistema computacional clássico, capaz somente de armazenar e manipular dados, um sistema baseado em inteligência artificial ainda aquisita, representa e manipula o conhecimento. A manipulação traduz-se na capacidade de deduzir ou inferir novos conhecimentos ou novas relações a respeito de fatos e conceitos, a partir do conhecimento existente. A representação e a manipulação são recursos utilizados para dar solução a problemas complexos, frequentemente não-quantitativos por natureza. Uma forma muito utilizada em sistemas inteligentes é a de implementação de fatos e regras, que basicamente constituem na obtenção e codificação de regras e fatos que sejam suficientes para um determinado domínio de um problema. Este processo de codificação é chamado de engenharia do conhecimento (Schutzer, 1987).

Diversas soluções computacionais empregadas atualmente em inteligência artificial têm sua origem em observações da natureza. Sistemas baseados em inspiração biológica vêm sendo empregados com sucesso na solução de diversos tipos de problemas. Existem hoje inúmeras técnicas de computação baseadas na evolução das espécies chamadas de evolucionárias e técnicas baseadas em observações do comportamento de espécies chamadas de comportamental.

2.2.1 Algoritmos Evolucionários

Os algoritmos evolucionários englobam um conjunto de métodos computacionais inspirados nos mecanismos evolutivos encontrados na natureza. Esses mecanismos estão diretamente relacionados com a Teoria da Evolução das Espécies (Darwin, 1859), onde afirma que a vida na Terra é o resultado de um processo de seleção feito pelo meio ambiente, em que somente os mais aptos e adaptados possuirão chances de sobreviver e, conseqüentemente, reproduzir-se. Dessa forma, os algoritmos evolucionários possuem auto-organização e comportamento adaptativo, seus elementos chaves são: i) população de indivíduos; ii) noção de aptidão; iii) ciclo de nascimento; iv) morte baseados na aptidão e noção de herança.

Os primeiros passos dados na área da computação evolucionária foram de biólogos e geneticistas. Estes tinham interesse em simular os processos vitais de um ser humano em um computador. Dentre os cientistas destacam-se os nomes de Barricelli, Fraser, Martin e Cockerham. Na década de 60, um grupo de cientistas, em que o nome de Holland se destaca, iniciaram um estudo em que era implementada uma população de n indivíduos onde cada um possuía seu genótipo e estava sujeito a operações de seleção, mutação e recombinação. Tal estudo foi modelado e passou a ser conhecido como Algoritmo Genético (Goldberg, 1989).

2.2.1.1 Evolução diferencial

É um tipo de computação evolutiva e como tal se inspira em princípios da teoria da evolução e seleção natural e utiliza modelos destes processos naturais para a solução de problemas.

A evolução diferencial (ED) é uma meta-heurística de otimização baseada na população (Storn e Price, 1997). O ED é aplicado à minimização de uma função-objetivo $f(x)$, onde x é um vetor variável contínuo. Na Figura 1 é mostrado um pseudocódigo de algoritmo DE simplificado.

Algorithm 2 Differential Evolution

```
1:  $NIter = 0$ ;  
2: Inicialização();  
3: while (NOT critério de parada) do  
4:    $NIter = NIter + 1$   
5:   mutação();  
6:   cruzamento();  
7:   seleção();  
8: end
```

Figura 1. Pseudocódigo da evolução diferencial

Fonte: Autor

Inicialmente, uma população aleatória de candidatos de solução é gerada por função inicializada. Então, o processo de evolução é repetido até que um critério de parada seja satisfeito. A primeira operação dentro do *loop* é a mutação, na qual uma solução de teste é gerada para cada indivíduo i conforme descrito na Equação 1.

$$x'_i = x_{p(1)} + F(x_{p(2)} - x_{p(3)}) \quad (1)$$

onde $p(1)$, $p(2)$ e $p(3)$ são índices aleatórios mutuamente diferentes uns dos outros e diferentes do índice i e F é um fator de escala na faixa $[0, 2]$. A solução correspondente

ao primeiro índice aleatório, $x_{p(1)}$, é conhecida como vetor base. Este vetor é alterado pela adição da diferença ponderada das duas outras soluções com os índices, $p(2)$ e $p(3)$. A operação é repetida enquanto a solução de teste x'_i estiver fora do domínio.

Após a mutação, a população passa pelo cruzamento, em que o componente j da prole y_i é encontrado de seus pais x_i e x'_i de acordo com a regra:

$$y_i^j = \begin{cases} x'^j_i, & \text{if } R^j \leq CR \text{ or } j = I_i, \\ x^j_i, & \text{otherwise} \end{cases} \quad (2)$$

Onde I_i é um número inteiro aleatório no intervalo $[1, n]$, R^j é aleatório em $[0, 1]$ e a taxa de cruzamento CR , também em $[0, 1]$, controla a fração dos valores dos parâmetros que são copiados do teste solução x'_i . Note que a alternativa $j = I_i$ assegura que pelo menos um componente receberá um valor mutado.

Finalmente, ocorre o processo de seleção, que define a população da próxima geração como segue:

$$x_i^{NIter+1} = \begin{cases} y_i^{NIter}, & \text{if } f(y_i^{NIter}) \leq f(x_i^{NIter}), \\ x_i^{NIter}, & \text{otherwise} \end{cases} \quad (3)$$

A solução de teste só irá substituir sua contrapartida na população atual se for igual ou melhor que a última.

Como o critério de término utilizado foi o número de gerações ($NIter = 1000$).

2.2.2 Sistemas Especialistas

Os sistemas especialistas foram desenvolvidos nos anos 60 como uma técnica de inteligência artificial para a solução de problemas complexos em um campo específico, como o diagnóstico médico de doenças. Os sistemas especialistas têm aumentado grandemente em popularidade desde sua introdução comercial no início dos anos 80. Atualmente, sistemas especialistas são utilizados em negócios, ciência, engenharia, processos de fabricação e muitos outros campos onde existe um domínio do problema bem definido. Muitas abordagens híbridas existem combinando sistemas especialistas com outras técnicas, como, por exemplo, algoritmos genéticos e redes neurais artificiais (Salmon, 2013).

Os sistemas especialistas fazem uso intensivo de conhecimento especializado para resolver problemas no nível de um especialista humano. Um especialista é uma pessoa que tem perícia em certa área. O especialista tem conhecimento ou habilidades especiais que não são conhecidas ou não estão disponíveis para a maioria das pessoas e pode resolver problemas de um campo específico que a maioria das pessoas não consegue resolver ou pode resolvê-los de forma mais eficiente.

Um dos fundamentos da teoria que envolve o desenvolvimento dos sistemas especialistas é a separação entre a base de conhecimento e o programa que executa as inferências sobre esta base. Na figura 2 é ilustrada a estrutura simplificada de um sistema especialista baseado em conhecimento, retratando a inter-relação entre os componentes do sistema.

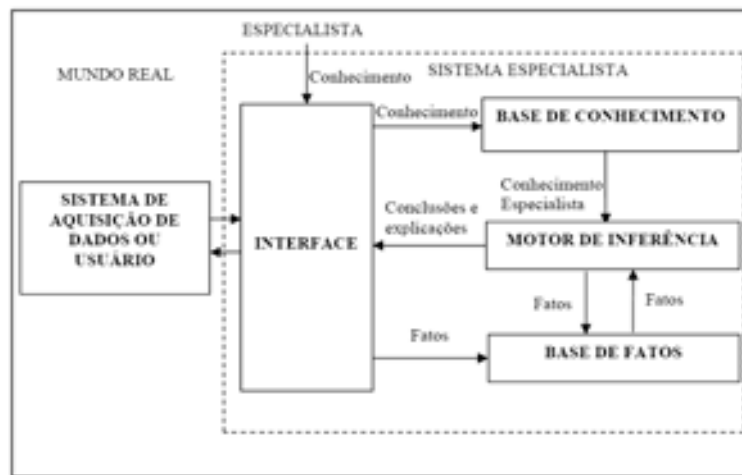


Figura 2. Estrutura Básica de um Sistema Especialista

Fonte: SALMON, 2013

O usuário ou um sistema de aquisição de dados supre fatos ou outra informação para o sistema especialista através da interface. O sistema especialista responde ao usuário com conhecimento especialista na forma de conclusões, diagnósticos, explicações ou perguntas.

A base de conhecimento pode ser considerada como o principal componente do sistema especialista. Ela armazena o conhecimento teórico e prático do domínio do problema particular. Em um sistema especialista baseado em regras este conhecimento é comumente representado na forma de regras do tipo *Se...então...* A base de conhecimento pode armazenar tanto conhecimento genérico como informações de casos específicos do domínio-problema (Salmon, 2013).

O motor de inferência aplica o conhecimento para a solução dos problemas reais. Ele é o componente que efetua a busca da solução do problema, “raciocinando” sobre as verdades apresentadas na base de fatos, tomando como referência o conhecimento armazenado na base de conhecimento. O motor de inferência pode ser entendido como um intérprete deste conhecimento e as inferências efetuadas alteram a base de fatos.

2.2.3 Processo Gaussiano

Seja $g: X \rightarrow \mathbb{R}^+$ uma função de dose de radiação definida em um domínio de dimensão d . No caso do problema de mapeamento de radiação considerado, $d = 2$. O objetivo do problema é reconstruir a função g com a maior fidelidade possível, dado um conjunto de medidas $\mathcal{D}_n = \{(\mathbf{x}_i, z_i)\}_{i=1:n}$ desta função.

Um processo Gaussiano (GP) é um método Bayesiano de regressão que consiste em uma extensão da distribuição Gaussiana multivariada para um espaço de funções de X para $\mathbb{R}$, com a propriedade de que qualquer subcoleção finita de pontos desta distribuição seguirá uma distribuição Gaussiana multivariada (Rasmussen, 2006; MacKay, 1998). Assim, f é uma variável aleatória sobre um espaço de funções de X para $\mathbb{R}$ e sendo $m(\mathbf{x}) := \mathbb{E}[f(\mathbf{x})]$ a função média, e $k(\mathbf{x}, \mathbf{x}') := \text{cov}(f(\mathbf{x}), f(\mathbf{x}'))$ a função de covariância, o processo Gaussiano é completamente definido por estas funções, e tem-se que $f \sim \mathcal{GP}(m, k)$. Normalmente se considera $m = 0$. Para poder usar o processo Gaussiano para regressão da função de taxa de dose g , este é aplicado sobre o logaritmo de g , logo sendo $f(\mathbf{x}) = \log(g(\mathbf{x}))$, assim sendo g também uma variável aleatória no processo Gaussiano.

Com f sendo distribuída de acordo com um processo Gaussiano com $m = 0$, e sendo $\mathcal{C}_n = \{(\mathbf{x}_i, \log(z)_i)\}_{i=1:n} = \{(\mathbf{x}_i, y_i)\}_{i=1:n}$, e sendo $y_{n+1} = \log(z_{n+1})$, com $z_{n+1} = g(\mathbf{x}_{n+1})$ sendo o valor da função no ponto $\mathbf{x}_{n+1}$, y_{n+1} é uma variável aleatória que segue uma distribuição normal $p(y_{n+1}|\mathcal{C}_n) = \mathcal{N}(\mu_n(\mathbf{x}_{n+1}), \sigma_n^2(\mathbf{x}_{n+1}))$ e z_{n+1} é uma variável aleatória que segue uma distribuição log-normal $p(z_{n+1}|\mathcal{D}_n) = \text{Lognormal}(\mu_n(\mathbf{x}_{n+1}), \sigma_n^2(\mathbf{x}_{n+1}))$ de acordo com as equações 4 e 5.

$$\mu_n(\mathbf{x}_{n+1}) = \mathbf{k}(\mathbf{x}_{n+1})^T \mathbf{K}^{-1} \mathbf{y} \quad (4)$$

$$\sigma_n^2(\mathbf{x}_{n+1}) = k(\mathbf{x}_{n+1}, \mathbf{x}_{n+1}) + \mathbf{k}(\mathbf{x}_{n+1})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}_{n+1}) \quad (5)$$

Onde $\mathbf{k}(\mathbf{x}_{n+1}) := (k(\mathbf{x}_1, \mathbf{x}_{n+1}), \dots, k(\mathbf{x}_n, \mathbf{x}_{n+1}))^T$ é o vetor de covariância entre $\mathbf{x}_{n+1}$ e $\mathbf{x}_{1:n}$, $\mathbf{y} := (y_1, \dots, y_n)^T$ é o vetor de valores observados, e $\mathbf{K} = (K_{i,j})_{1:n,1:n}$, $K_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$ é a matriz de covariância.

A função de covariância k determina a estrutura das funções no espaço de funções no processo Gaussiano, logo sendo o fator principal deste. Estas funções também são chamadas de *kernels*, e entre eles, são comumente usados os *kernels* da classe Matérn, parametrizados por um parâmetro $\nu > 0$. Nesse trabalho foi usado este *kernel* de forma isotrópica, com $\nu = 3/2$, sendo este dado por $k(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp\left(-\frac{\sqrt{3}}{l} |\mathbf{x} - \mathbf{x}'|\right) \left(1 + \frac{\sqrt{3}}{l} |\mathbf{x} - \mathbf{x}'|\right)$, onde θ_0^2 é o parâmetro de amplitude e l o parâmetro de distância. Sendo estes hiperparâmetros coletivamente Θ , estes podem ser otimizados através da maximização da *log-likelihood* apresentada na equação 6 (Snelson, 2003).

$$\log p(\mathbf{z}|\mathbf{x}_{1:n}, \Theta) = -\frac{1}{2} \log(\det \mathbf{K}) - \frac{1}{2} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} - \frac{n}{2} \log(2\pi) - \sum_{i=1}^n \log z_i \quad (6)$$

A inferência do valor de $g(\mathbf{x}_{n+1})$ é feita a partir de uma medida pontual da distribuição $p(z_{n+1}|\mathcal{D}_n)$. Neste trabalho foi usada a mediana desta distribuição, dada por $\exp(\mu_{n+1})$.

2.2.4 Active Learning

Com o uso do processo Gaussiano, pode se determinar sequencialmente os pontos onde é feita a medição da função g , com o objetivo de estimá-la com fidelidade com o menor número de medições possíveis. Este processo de determinação automática de pontos de medição entra no contexto de aprendizagem ativa, em que o agente escolhe seus próprios dados de aprendizagem (Settles, 2010; Kapoor, 2007).

De forma mais específica, dado conjunto de dados já medidos $\mathcal{D}_n$, se escolhe o próximo ponto de medição $\mathbf{x}_{n+1}$ de forma que $\mathcal{D}_{n+1}$ possa em média descrever g da melhor forma possível. Isso pode ser feito a partir do critério de entropia da informação, em que é escolhido o ponto de máxima entropia condicional apresentado na equação 7 (Krause, 2007).

$$h(z_{n+1}|\mathcal{D}_n) = \int p(z_{n+1}|\mathcal{D}_n) \log(p(z_{n+1}|\mathcal{D}_n)) dz_{n+1} \quad (7)$$

Como $p(z_{n+1}|\mathcal{D}_n)$ é uma distribuição log-normal, tem-se a equação 8.

$$h(z_{n+1}|\mathcal{D}_n) = \mu_n(\mathbf{x}_{n+1}) + \frac{1}{2} \log(2\pi e \sigma_n(\mathbf{x}_{n+1})) \quad (8)$$

No contexto de robótica, é desejável que, na medição do campo de radiação, o robô não se desloque de forma que este frequentemente percorra grandes distâncias, visto que o tempo que o robô se encontrará no campo será dominado por esse deslocamento. Visto isso, é acoplada a este critério de informação um outro critério de proximidade do próximo ponto de medida ao ponto medido imediatamente onde se encontra o robô. Assim, é usada uma penalização de distância entre os pontos medidos $P(\mathbf{x}_{n+1}, \mathbf{x}_n) = P(|\mathbf{x}_{n+1} - \mathbf{x}_n|)$. Esta por sua vez é parametrizada por um limite inferior de penalização w_P , $0 \leq w_P \leq 1$, e um parâmetro de distância característica de

penalização l_p . A função Matérn com $\nu = \frac{3}{2}$ é uma função radial que pode ser usada para esse fim. Usando esta, temos que a penalização de distância se torna a equação 9.

$$\begin{aligned}
 P_n(\mathbf{x}_{n+1}) &= P(|\mathbf{x}_{n+1} - \mathbf{x}_n|) \\
 &= (1 - w_p) \exp\left(-\frac{\sqrt{3}}{l_p} |\mathbf{x} - \mathbf{x}'|\right) \left(1 + \frac{\sqrt{3}}{l_p} |\mathbf{x} - \mathbf{x}'|\right) + w_p
 \end{aligned} \tag{9}$$

Assim, a função a ser otimizada para cada medição $n + 1$ é dada por $\alpha_n(\mathbf{x}_{n+1}) = P_n(\mathbf{x}_{n+1})h(z_{n+1}|\mathcal{D}_n)$, sendo o ponto escolhido de medição $\mathbf{x}_{n+1}$ dado a partir da otimização desta, que é feita por um método externo de otimização.

Capítulo 3

PROJETO BASE E SUAS ETAPAS

Este capítulo aborda a dinâmica das etapas de desenvolvimento do projeto, apresentando de forma geral conceitos relacionados ao planejamento, desenvolvimento, teste e validação, desde o ambiente de simulação computacional até sua validação prática em um ambiente real.

3.1 Considerações do projeto

Os fatores decisivos para efetivação deste projeto envolvem prever o emprego de robótica autônoma na área nuclear, tanto para a indústria de geração de energia, como também em fábricas de materiais radioativos, em aplicações de mapeamento dosimétrico, detecções e inspeção diversas, bem como dispor para pesquisadores e desenvolvedores da área computacional uma ferramenta de *hardware* (*Facility*) que dinamize a implementação de testes práticos em ambientes reais e corroborem suas simulações computacionais.

No intuito de demonstrar a viabilidade da elaboração de aplicações de controle utilizando este projeto como uma *facility* para desenvolvedores, foi criado um algoritmo computacional para mapeamento de radiação com a capacidade de reconstruir um campo de radiação através da leitura de taxa de dose de radiação fornecida via aquisição de dados de um sensor de radiação disposto na estrutura do robô.

3.2 Visão geral do projeto

A busca de opções estruturais e de configurações robóticas que possibilitasse a elaboração de um robô com as características necessárias para realização das aplicações práticas mencionadas, orientou para a escolha de um sistema robótico com condições de se locomover tanto em ambientes internos como externos, com capacidade de manobrabilidade para se desvencilhar de obstáculos em espaços reduzidos, com possibilidade de adquirir dados de sensoriamento embarcado em tempo real, com capacidade de processamento suficiente para execução de suas funções e com boa robustez física.

Optou-se, então pelo desenvolvimento de uma plataforma sobre esteiras que possibilitasse movimentos de pivoteamento e de rápidas mudanças de direção. Na figura 3 são apresentados esses movimentos.

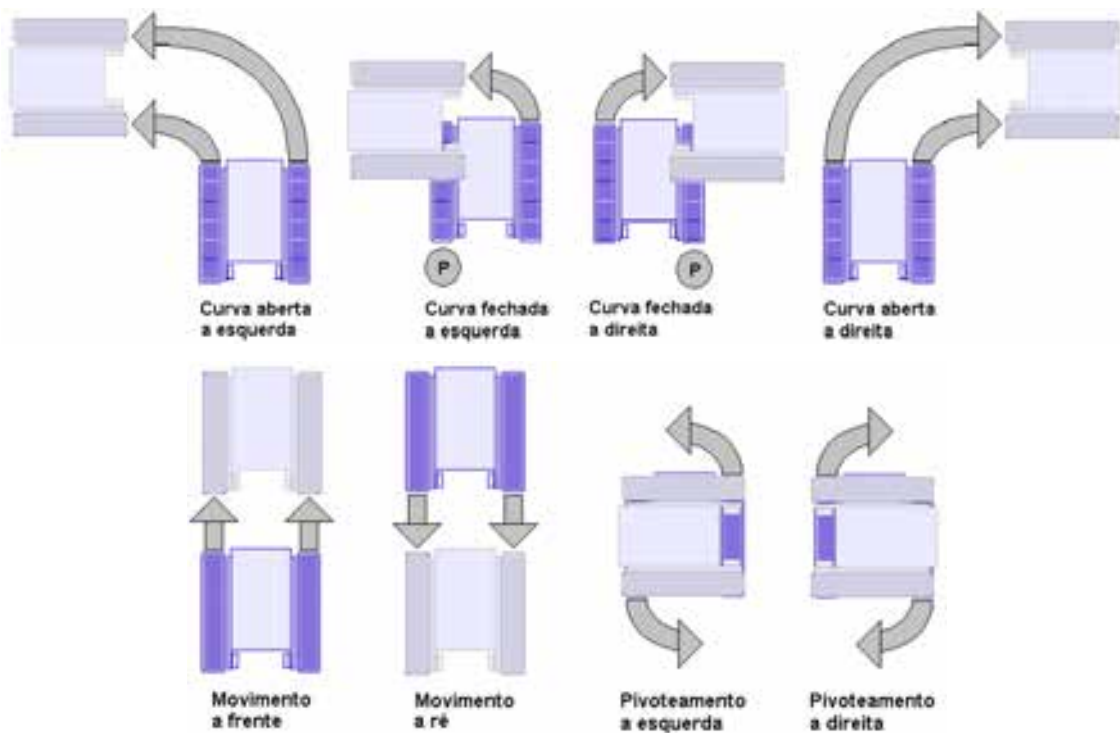


Figura 3. Liberdade de movimentos

Fonte: Autor

Sensores embarcados proporcionam condições de leitura de variáveis externas como: proximidade de objetos e/ou obstáculos, leituras de infravermelho (IR), luminosidade, temperatura, distância, radiação e gases (CO₂ e H). Além de sensores internos que atuam no funcionamento direto das funções vitais de controle do sistema robótico. O diagrama em blocos apresentado na figura 4 mostra uma visão geral do funcionamento interno do robô, representando o caminho transcorrido pela informação de cada etapa de funcionamento.

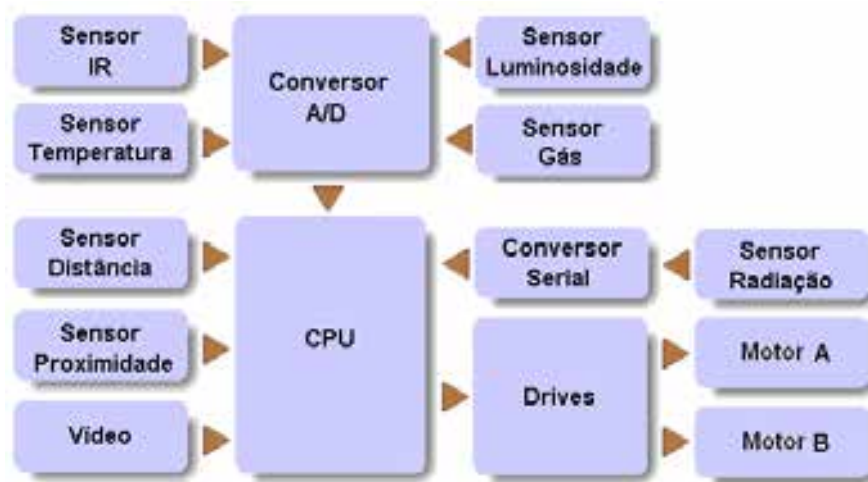


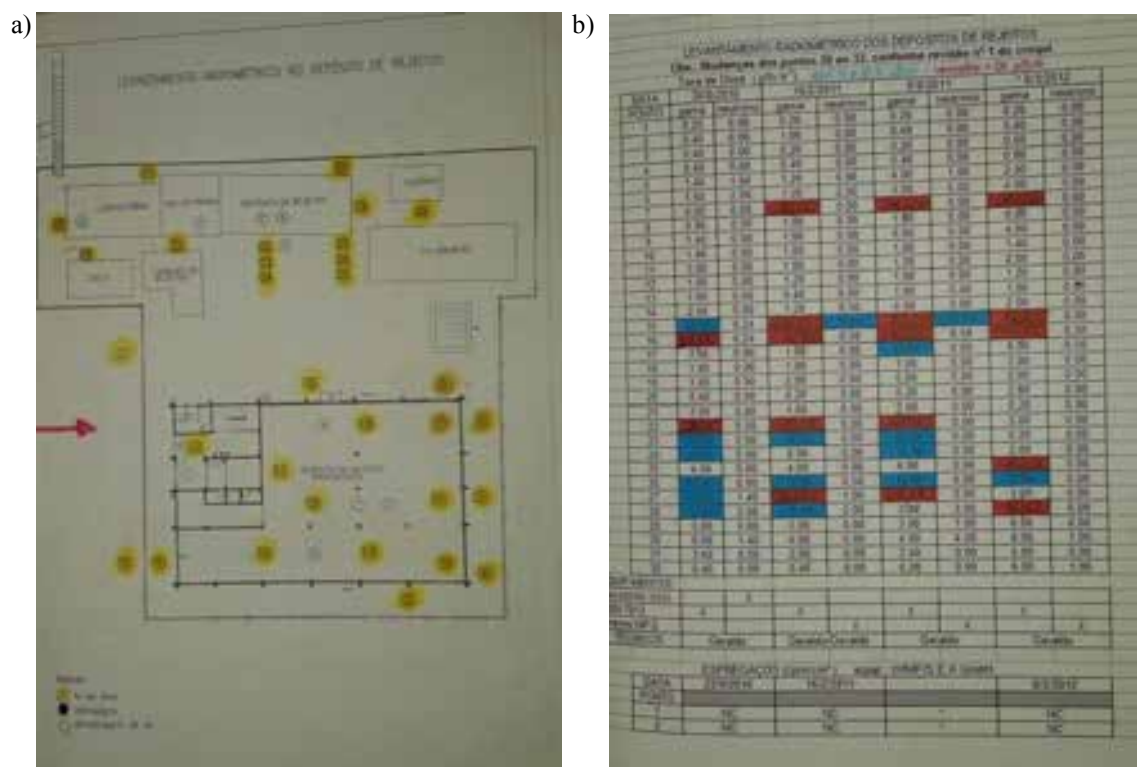
Figura 4. Diagrama geral do funcionamento

Fonte: Autor

A aplicação computacional deste trabalho, denominada de Algoritmo de reconstrução de superfície (ARS), foi desenvolvida em linguagem *Python* (Langtangen, 2009) e está focada na reconstrução de um campo de radiação através da utilização de um sistema inteligente de aprendizagem *Active Learning* (Settles, 2010; Kapoor, 2007), que através do uso de um *Gaussian Process* (Rasmussen, 2006; MacKay, 1998), que é um método Bayesiano de regressão que consiste em uma extensão da distribuição Gaussiana multivariada, consegue reconstruir computacionalmente um ambiente de incidência radiológica através de um método de regressão, esta aplicação também faz uso de outro processo como otimizador auxiliar, conhecido como *Differential Evolution* (DE) (Storn e Price, 1997), que é um tipo de computação evolutiva que se inspira em

princípios da teoria da evolução e seleção natural e utiliza modelos destes processos naturais para a solução de problemas.

No intuito de validar o algoritmo (ARS) desenvolvido, foi necessário criar um ambiente de simulação computacional que retratasse fidedignamente as condições reais de um campo de radiação. Com o apoio do Instituto de Engenharia Nuclear (IEN), órgão subordinado a Comissão Nacional de Energia Nuclear (CNEN), pode-se obter acesso ao levantamento radiométrico do depósito de rejeito radioativo de sua unidade (Figura 5), o que propiciou através da aquisição de dados de levantamento dosimétrico, gerar uma equação matemática (capítulo 4) que reproduzisse tal ambiente.



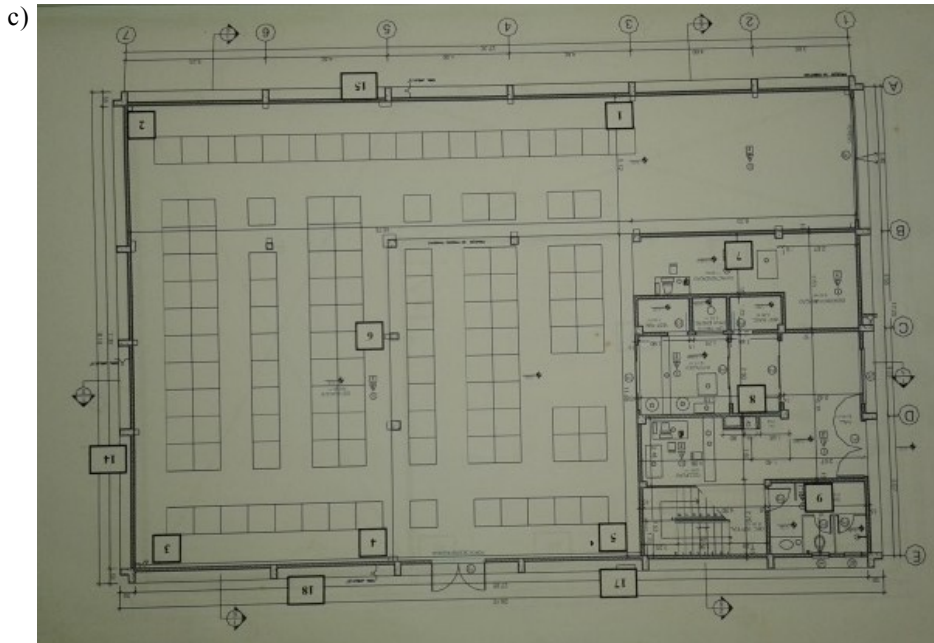


Figura 5. a) Pontos de medição de tx de dose; b) Levantamento Radiométrico do depósito de rejeitos; c) Planta do depósito de rejeitos radioativos

Fonte: Instituto de Engenharia Nuclear

Uma simulação computacional é o processo de compor matematicamente um modelo abstrato com base em uma situação real com intuito de investigar o impacto de modificações de um modelo e o efeito de introduzir diferentes perturbações no ambiente investigado. Neste caso específico a utilização de simulação computacional não só permite prever o que acontecerá a uma situação real de aplicação sob algumas hipóteses simplificadoras, como também mitiga a necessidade de intervenções humanas na realização de testes e levantamentos necessários para montagem de uma aplicação prática, uma vez que o exemplo de aplicação se trata de depósito de rejeito radioativo.

Para tal, criou-se uma equação matemática de taxa de dose onde foram estimadas algumas condições de distribuição de taxa de dose através da elaboração de três funções que apresentam exemplos de superfícies com topologia de diferentes níveis de dificuldade para análise do ARS. Essas funções foram utilizadas como base para execução de diversas simulações computacionais visando investigar a viabilidade do

algoritmo desenvolvido e poder realizar ajustes necessários para seu pleno funcionamento.

No entanto, a implementação de um protótipo em ambiente real requer que o robô possua recursos que possibilitem a transposição de obstáculos, durante sua navegação exploratória. A ultrapassagem desses obstáculos de forma circulatoria, não pode influenciar de forma prejudicial na trajetória estipulada previamente pelo ARS. Para isso o robô conta com três sensores de proximidade instalados à frente de sua estrutura, um na posição central, uma na lateral direita e outro na lateral esquerda, de modo realizar uma varredura de 90° de cobertura durante seu deslocamento, conforme ilustrado na figura 6.

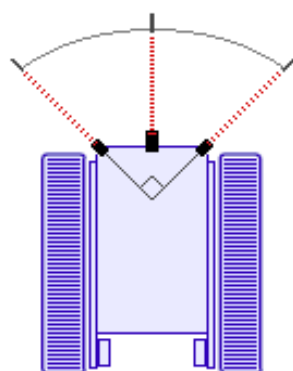


Figura 6. Área de cobertura do sensoriamento de proximidade

Fonte: Autor

Para gerenciar o deslocamento de forma eficiente, foi desenvolvido um algoritmo de navegação, denominado como Algoritmo de Transposição de Obstáculo (ATO), que tem seu funcionamento firmado na premissa dos sistemas especialistas, onde o processamento de soluções viáveis para o problema da transposição de obstáculos detectáveis é realizado através da aplicação de um motor de inferência sobre uma base de conhecimento. O funcionamento deste algoritmo será apresentado de forma detalhada no capítulo 6.

Para investigar o comportamento do ARS em um ambiente que requeresse a utilização de recursos para transposição de obstáculos, gerou-se uma simulação computacional de forma a integrar a utilização do ATO. A figura 7 apresenta o deslocamento do robô através de obstáculos gerados computacionalmente, onde o robô encontra-se representado em escala pelo retângulo de cor rosa, os obstáculos a serem transplantados encontram-se representados pelos polígonos irregulares de cor azul, a detecção dos sensores de proximidade representados pela cor vermelha e o alcance dos sensores pelo arco amarelo.

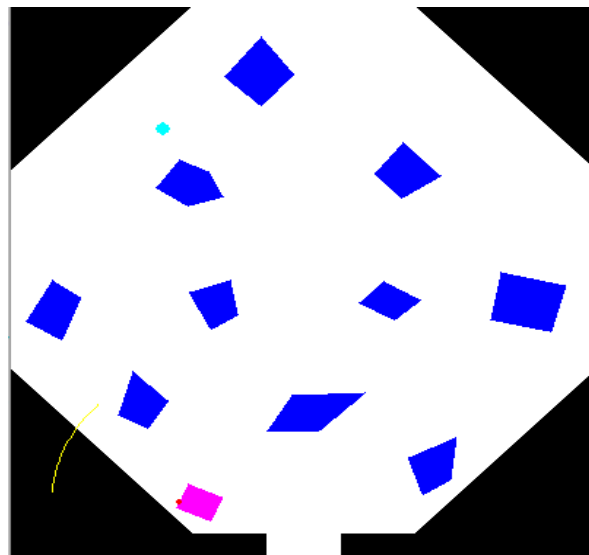


Figura 7. Simulação computacional de deslocamento do robô
Fonte: Autor

De posse dos resultados obtidos através da totalidade de simulações computacionais, pode-se levantar os valores ideais de diversas variáveis de interesse que integram o algoritmo de reconstrução de superfície desenvolvido, assim como, determinar de forma mais apropriada alguns parâmetros do modelo computacional, de modo a adaptar sua implementação em um protótipo prático real.

Com a finalidade de criar um ambiente em bancada que reproduzisse todo o hardware necessário para a integração do algoritmo proposto e com um processamento similar ao utilizado pelo protótipo robótico elaborado neste projeto, foi desenvolvida

pelo LARO (Laboratório de robótica) do LMP (Laboratório de Monitoração de Processos) da UFRJ (Universidade Federal do Rio de Janeiro) uma placa de testes capaz de reproduzir o funcionamento do sistema robótico real e colher respostas de sensoriamento, com o intuito de analisar a eficiência, diagnosticar falhas e plotar possíveis pontos de melhoria no hardware, no software e nos firmwares de controle. A figura 8 apresenta uma imagem desta placa desenvolvida.



Figura 8. Placa de desenvolvimento de testes e simulações
Fonte: Autor

A efetivação de ensaios em bancada, focados na aquisição de dados de sensoriamento externo, transmissão e recepção de sinais de controle e na velocidade de acionamento de hardware, propiciou rumar para a etapa de prototipagem.

Capítulo 4

ALGORITMO DE RECONSTRUÇÃO DE SUPERFÍCIE

Este capítulo apresenta a elaboração do algoritmo computacional, desenvolvido com a finalidade de reconstruir uma superfície de radiação, bem como detalha as metodologias utilizadas, o descritivo operacional e a modelagem do sistema.

4.1 Holística do Problema

A navegação autônoma de um sistema robótico é um problema complexo que demanda a análise de diversas variáveis de forma integrada, suscetível a diversas incertezas e de difícil solução genérica. Para o caso específico apresentado neste trabalho, foi utilizada uma combinação de conceitos de arquiteturas de controle inteligente de forma a adequar os recursos tradicionais às necessidades específicas desta aplicação.

O problema geral apresentado nesta aplicação é o da detecção otimizada de fontes radiológicas em ambientes não conhecidos, realizada através de um robô confeccionado em plataforma multiterreno. Para tal, o sistema robótico conta com um *POD* de sensores embarcados, responsáveis pela leitura de intensidade de taxa de dose.

O algoritmo computacional desenvolvido para o robô deve ser capaz de realizar um levantamento radiométrico de um ambiente em um menor espaço de tempo e de distância percorrida possíveis, reconstruindo a superfície de radiação do local através da leitura de pontos escolhidos computacionalmente.

4.2 Abordagem Metodológica

A metodologia utilizada para elaboração deste algoritmo faz uso de um processo gaussiano utilizado num contexto de aprendizagem ativa (*Active learning*), que é definida pelo contraste com o modelo passivo de aprendizagem supervisionada convencional, onde todos os dados para aprendizagem são obtidos sem referência ao algoritmo de aprendizagem, enquanto na aprendizagem ativa o algoritmo escolhe interativamente os pontos de identificação para avaliá-los (Dasgupta, 2009). A expectativa da aprendizagem ativa é que a interação possa reduzir substancialmente o número de pontos e validações necessárias, tornando mais simples a solução de problemas através do aprendizado da máquina. Com o uso de uma regressão de Processo Gaussiano (*GP - Gaussian Processe*), pode se determinar sequencialmente os pontos onde é feita a medição da função, a partir da definição de uma função de aquisição, a qual define o quão desejável é a medição de um ponto dada a sua distribuição de probabilidade. O GP é um método Bayesiano de regressão que consiste em uma extensão da distribuição Gaussiana multivariada para um espaço de funções sobre X . Formalmente, o processo Gaussiano é uma coleção de variáveis aleatórias com suporte em $\mathbb{R}$ (cada uma associada a um ponto de X), onde qualquer subcoleção finita desta função seguirá uma distribuição Gaussiana multivariada (Rasmussen, 2006).

4.3 Formulação de problema

Seja $X \subset \mathbb{R}^2$ o mapa que contém uma distribuição de taxa de dose, e $g: X \rightarrow \mathbb{R}^+$ a função da distribuição de taxa de dose desconhecida. O problema é projetar sequencialmente um conjunto de n pontos de medição de g , resultando nos dados de

$\mathcal{D}_n = \{(\mathbf{x}_i, g(\mathbf{x}_i))\}_{i=1:n}$, e através destas medições poder estimar uma Função g^* que se assemelhe o mais próximo possível de g .

O problema de estimativa é tratado com um Processo Gaussiano (GP) sobre o logaritmo de g . O GP é uma generalização da distribuição gaussiana multivariada onde consideramos a função desconhecida $f = \log g$, função aleatória a partir do espaço de funções diferenciáveis de X para $\mathbb{R}$, como por exemplo, para cada subconjunto finito de pontos $\{\mathbf{x}_i\}_{i=1:m} \subset X$, $\{f(\mathbf{x}_i)\}_{i=1:m}$ que segue uma distribuição gaussiana multivariada (Rasmussen, 2006; MacKay, 1998). O GP é completamente especificado pela sua função média $m(\mathbf{x}) := \mathbb{E}[f(\mathbf{x})]$, geralmente assumida como zero, e pela função de covariância $k(\mathbf{x}, \mathbf{x}') := \text{cov}(f(\mathbf{x}), f(\mathbf{x}'))$ que define a estrutura de GP. Neste trabalho, a função de covariância utilizada é a função de covariância isotrópica Matérn com parâmetro $\nu = 3/2$, como apresentada na equação 9:

$$k(\mathbf{x}, \mathbf{x}') = C_{3/2}(\mathbf{x}, \mathbf{x}') = \theta_0^2 \exp\left(-\frac{1}{l}\sqrt{3}r\right)\left(1 + \frac{1}{l}\sqrt{3}r\right) \quad (9)$$

Onde $r = \|\mathbf{x} - \mathbf{x}'\|$

Os parâmetros θ_0^2 e l são chamados, respectivamente, de parâmetro de amplitude e de parâmetro de forma.

Com f sendo distribuído de acordo com um Processo Gaussiano com a média $m(\mathbf{x}) = 0$ e $\mathcal{C}_n = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1:n} = \{(\mathbf{x}_i, \log(g(\mathbf{x}_i)))\}_{i=1:n}$ sendo o logaritmo dos dados adquiridos anteriormente, temos que $f(\mathbf{x})$ é uma variável aleatória normalmente distribuída, como $f(\mathbf{x}) \sim p(f(\mathbf{x})|\mathcal{C}_n) = \mathcal{N}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x}))$, onde:

$$\mu_n(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{y} \quad (10)$$

$$\sigma_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) + \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}) \quad (11)$$

Aqui $\mathbf{k}(\mathbf{x}) := (k(\mathbf{x}_1, \mathbf{x}), \dots, k(\mathbf{x}_n, \mathbf{x}))^T$ é o vetor de covariância entre $\mathbf{x}$ e $\{\mathbf{x}_i\}_{i=1:n}$, $\mathbf{y} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^T$ é o vetor de valores observados e $\mathbf{K} = (k(\mathbf{x}_i, \mathbf{x}_j))_{1:n, 1:n} \in \mathbb{R}^{n \times n}$ é a matriz de covariância entre $\{\mathbf{x}_i\}_{i=1:n}$. Uma vez que o GP é aplicado ao logaritmo de g , obtém-se que: dado $\mathcal{D}_n$, $g(\mathbf{x})$ é uma variável aleatória log-normalmente distribuída como $g(\mathbf{x}) \sim \text{Lognormal}(\mu_n(\mathbf{x}), \sigma_n^2(\mathbf{x}))$, onde $\mu_n(\mathbf{x})$ e $\sigma_n^2(\mathbf{x})$ são como definidos acima.

Os parâmetros θ_0^2 e l são otimizados ao se maximizar o *log-likelihood* $\mathcal{L}$ do $\mathcal{D}_n$ (Snelson, 2003):

$$\mathcal{L} = \log p(\mathbf{z}|\mathbf{x}_{1:n}, \Theta) = -\frac{1}{2} \log(\det \mathbf{K}) - \frac{1}{2} \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} - \frac{n}{2} \log(2\pi) - \sum_{i=1}^n \log z_i \quad (12)$$

Finalmente, a estimativa $g^*(\mathbf{x})$ de $g(\mathbf{x})$ pode ser feita por uma medida pontual de $g(\mathbf{x})$. Neste trabalho, optou-se pela utilização da mediana, minimizando desta forma o erro absoluto esperado, de modo que se infere $g^*(\mathbf{x}) = e^{\mu_n(\mathbf{x})}$.

Técnicas de aprendizagem ativa podem ser usadas juntamente com o GP para determinar sequencialmente o conjunto de pontos a serem medidos, para uma melhor estimativa de g . A determinação automática de pontos de medição para fins de estimativa também é conhecida como aprendizagem ativa (Settles, 2010; Kapoor, 2007). Essa determinação sequencial pode ser feita de maneira gulosa, de modo que, com dados $\mathcal{D}_n$ previamente adquiridos, pretende-se escolher o próximo ponto de medição $\mathbf{x}_{n+1}$ para que o dado esperado $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}_{n+1}, g(\mathbf{x}_{n+1}))\}$ possa

melhorar o conhecimento de g , tanto quanto possível. O critério usado neste trabalho é o de entropia de informação (Krause, 2007; Krause, 2008), em que $\mathbf{x}_{n+1}$ é escolhido para que $g(\mathbf{x}_{n+1})$ tenha uma entropia diferencial máxima. Uma vez que $g(\mathbf{x}_{n+1})$ é log-normalmente distribuído, maximiza-se a equação 13.

$$h(g(\mathbf{x}_{n+1})) = \mu_n(\mathbf{x}_{n+1}) + \frac{1}{2} \log(2\pi e \sigma_n^2(\mathbf{x}_{n+1})) \quad (13)$$

No contexto da robótica, também é desejável controlar a distância do caminho percorrido pelo robô, favorecendo a seleção de pontos mais próximos da localização atual do robô. Isto é feito adicionando uma penalização $P(\mathbf{x}_{n+1}, \mathbf{x}_n) = P(||\mathbf{x}_{n+1} - \mathbf{x}_n||)$ ao critério de seleção de ponto acima, de modo que se escolheu $\mathbf{x}_{n+1}$ que maximiza a função de aquisição $\alpha_n(\mathbf{x}_{n+1}) = P(||\mathbf{x}_{n+1} - \mathbf{x}_n||) h(g(\mathbf{x}_{n+1}))$. Neste trabalho, a penalização é baseada no mesmo kernel Matérn usado no GP.

$$P(||\mathbf{x}_{n+1} - \mathbf{x}_n||) = P(r_p) = (1 - w_p) \exp\left(-\frac{1}{l_p} \sqrt{3} r_p\right) \left(1 + \frac{1}{l_p} \sqrt{3} r_p\right) + w_p \quad (14)$$

Onde w_p é uma constante entre 0 e 1 determinando a penalização máxima, e l_p é a escala de comprimento da função de penalização, conforme pode-se verificar na figura 9.

A função de aquisição α_n deve ser otimizada por um algoritmo externo, para determinar o próximo ponto de medição. Para este trabalho foi escolhido um algoritmo de otimização de evolução diferencial.

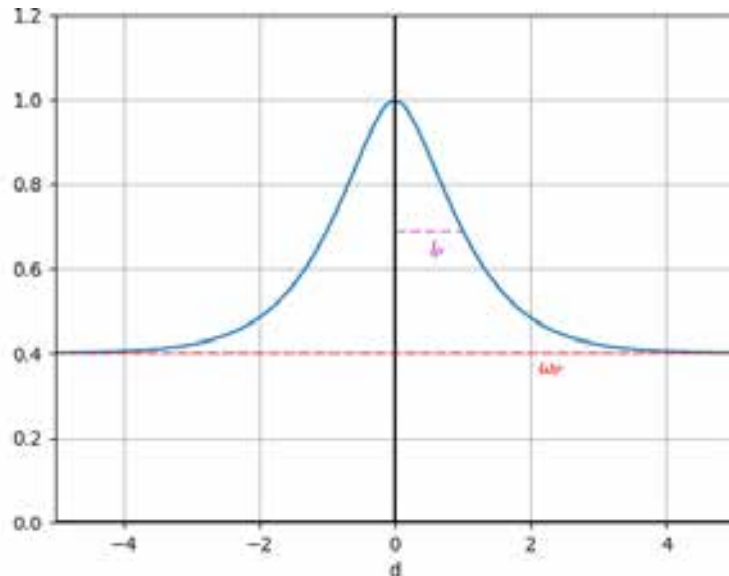


Figura 9. Função de penalização
Fonte: Autor

4.4 Descritivo Operacional do Algoritmo computacional

Para a problematização abordada neste trabalho foi desenvolvido um algoritmo computacional o qual se encontra representado através do pseudocódigo ilustrado na figura 10.

Pseudocódigo	Algoritmo de Reconstrução de Superfície
1:	$GP \leftarrow$ Inicializa o processo gaussiano
2:	$Wp \leftarrow$ Inicializa o limite inferior de penalização
3:	$Lp \leftarrow$ Inicializa a distância de penalização
4:	$n \leftarrow$ Número de avaliações
5:	$x \leftarrow$ Inicializa posição do robô
6:	$y \leftarrow$ MENSURA_RADIAÇÃO ()
7:	$GP. ADICIONA_DADOS (\{x, y\})$
8:	$counter \leftarrow 1$
9:	Enquanto não finalizado faça
10:	$\alpha_n^P \leftarrow$ OBTENHA_FUNCAO_AQUISICAO (GP, x)
11:	$x \leftarrow$ OTIMIZAR (α_n^P)
12:	MOVER (x)
13:	$y \leftarrow$ MENSURA_RADIAÇÃO ()
14:	$GP. ADICIONA_DADOS (\{x, y\})$
15:	$counter \leftarrow counter + 1$
16:	Se $i \bmod 10 = 0$ então $GP. ATUALIZA_PARAMETROS ()$
17:	Se valor n atingido então $GP. SALVA ()$
18:	Fim

Figura 10. Pseudocódigo ARS
Fonte: Autor

Detalhamento das etapas do pseudocódigo:

- Carrega as variáveis de inicialização: Wp, Lp, n e x
- Carrega as dimensões do ambiente e segue para a posição inicial x_0 .
- Carrega o *Gaussian Process*¹ com hiperparâmetros iniciais do kernel $\theta_0^2 = 1$ e $l = 1 m$.
- Mensura a taxa de dose de radiação y_0 na posição atual e adiciona uma tupla (x_0, y_0) ao GP.
- Inicializa o contador de dados $n = 1$.
- Enquanto a medição de campo não finalizar:
 - Utiliza o Otimizador¹ Diferencial de Entropia para encontrar o ponto de entropia diferencial máxima, penalizado pela distância, com a dimensão da sala como limites. Deixar este ponto ser x_n .
 - Altera para x_n e mande a taxa de dose y_n . Adicionar uma tupla (x_n, y_n) ao GP. Torna $n = n + 1$.
 - Se $n \equiv 0 \pmod{10}$, atualiza os hiperparâmetros do *kernel* GP ao otimizar a estimativa de verossimilhança, usando o otimizador LBFGS³.
- Estima a taxa da dose do ambiente usando a mediana da distribuição lognormal estimada pelo GP.

¹ Objeto da classe *Gaussian Process (GP)* desenvolvido em linguagem python.

² Utiliza o *DE (Differential Evolution)* do *Scipy* (Coleção de softwares de código aberto para computação científica em Python) como otimizador auxiliar.

³ Utiliza o otimizador *LBFGS* (Interface para algoritmos de minimização para funções multivariadas) do *Scipy*.

O fluxograma de sequenciamento de ações apresentado na figura 11 ilustra de forma gráfica a evolução do algoritmo computacional detalhado neste pseudocódigo.

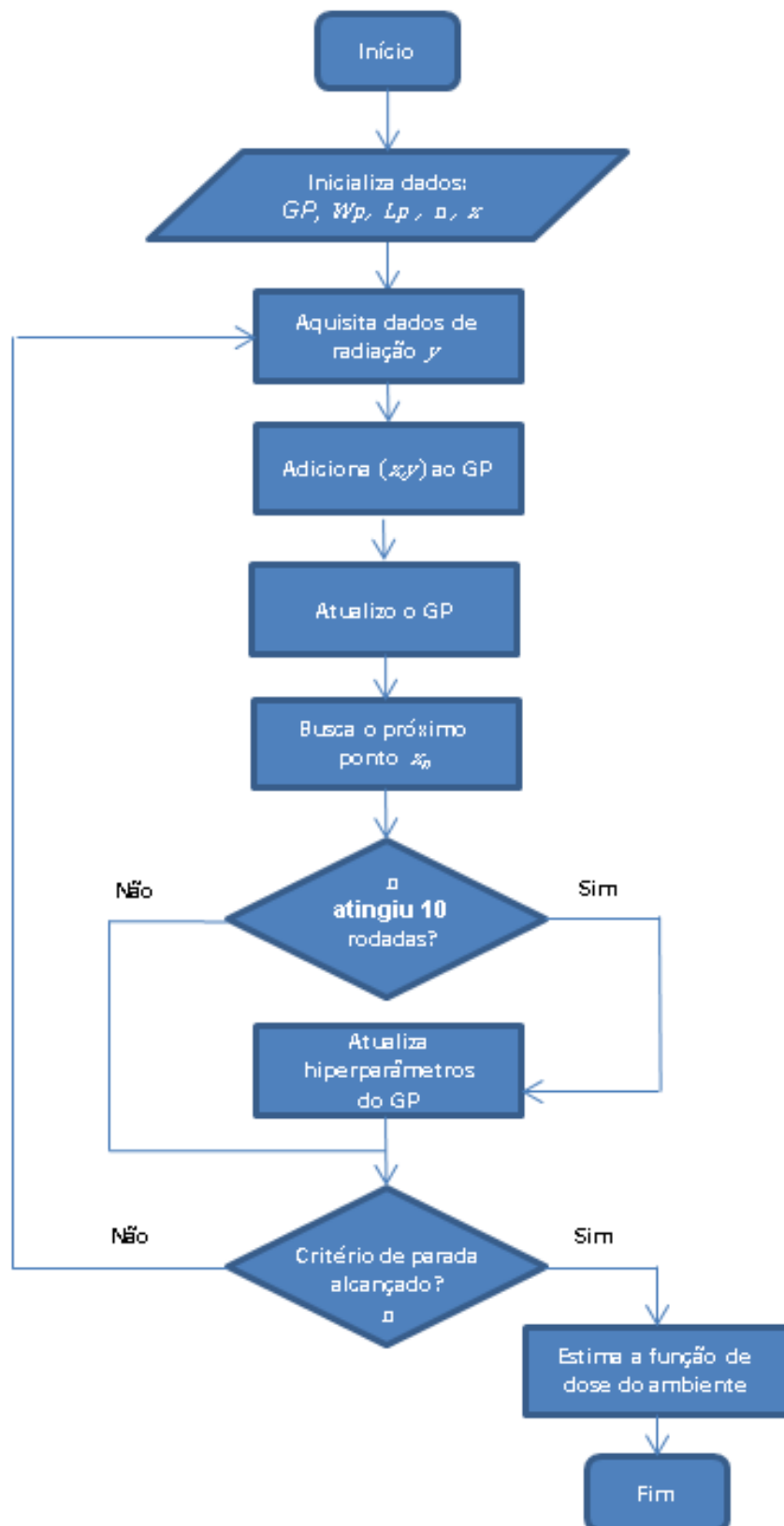


Figura 11. Fluxograma de sequenciamento de ações
Fonte: Autor

4.4.1 Evolução do Algoritmo computacional

A evolução computacional do algoritmo é baseada na busca de caminhos de maior entropia da informação, selecionados através da otimização de uma distribuição de probabilidade, gerada a partir da medição de um ponto.

Com base na medição do primeiro ponto, o GP gera uma distribuição de probabilidade. Conforme ilustração apresentada na figura 12.

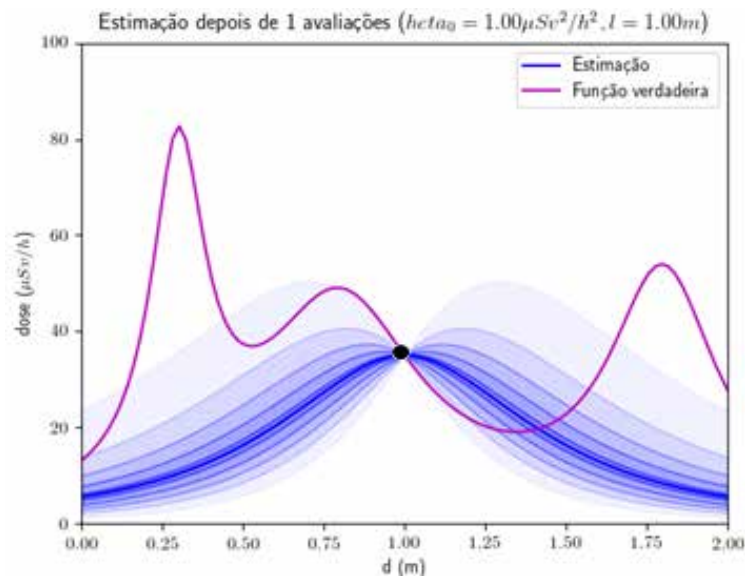


Figura 12. Distribuição de probabilidade do ponto inicial

Fonte: Autor

A determinação do próximo ponto é realizada com base na distribuição de probabilidade do ponto inicial através da utilização de um algoritmo de evolução diferencial na busca pelo ponto de maior entropia (maior desconhecimento da informação), essa busca visa regiões de maior variância e média mais alta. A função de otimização deste algoritmo encontra-se apresentada na equação 13 e a figura 13 apresenta a ilustração gráfica da busca pelo próximo ponto.

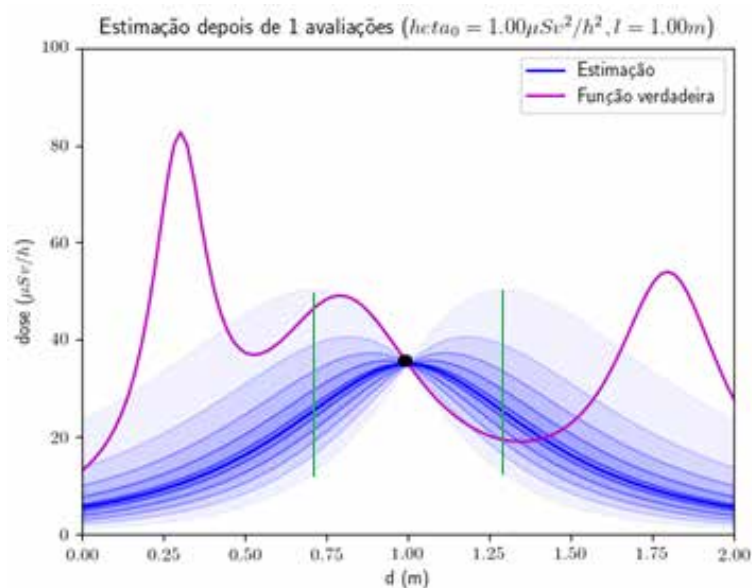


Figura 13. Busca pelo ponto de maior entropia

Fonte: Autor

Após a determinação do segundo ponto realizada através da otimização produzida pelo algoritmo de evolução diferencial. Será então realizada a medição da radiação deste novo ponto e na sequência gerada uma nova distribuição de probabilidade, conforme apresenta graficamente na figura 14.

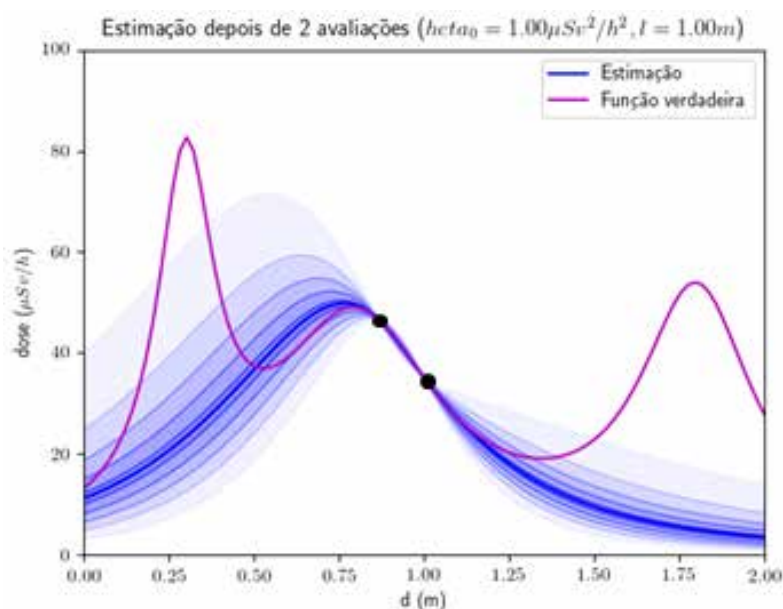


Figura 14. Distribuição de probabilidade baseada em dois pontos

Fonte: Autor

Sequencialmente o algoritmo continua sua evolução através da seleção e medição de novos pontos, conforme apresentado graficamente nas figuras 15, 16 e 17.

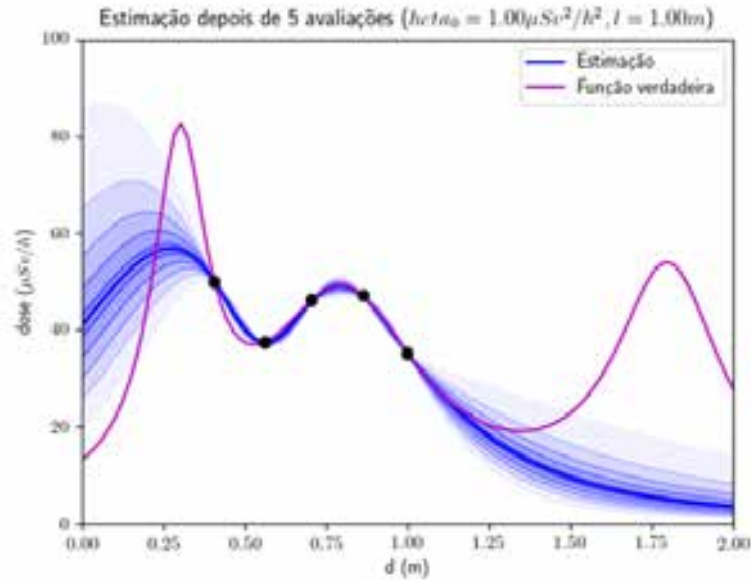


Figura 15. Distribuição de probabilidade baseada em cinco pontos
Fonte: Autor

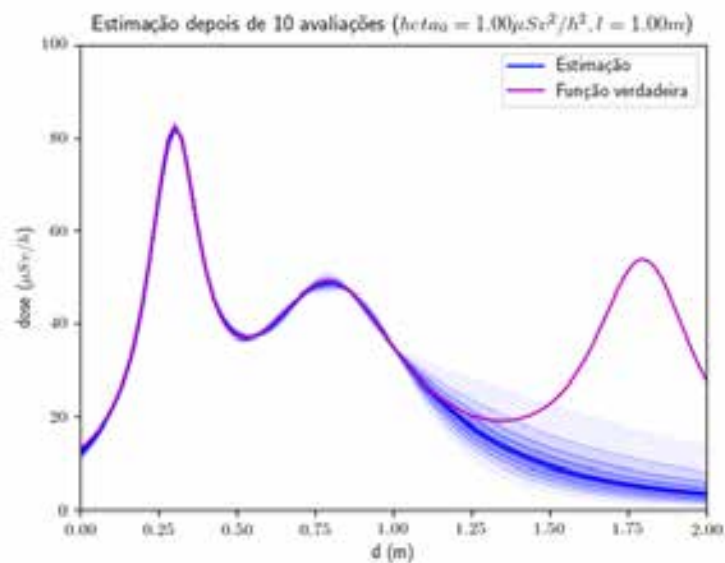


Figura 16. Distribuição de probabilidade baseada em dez pontos
Fonte: Autor

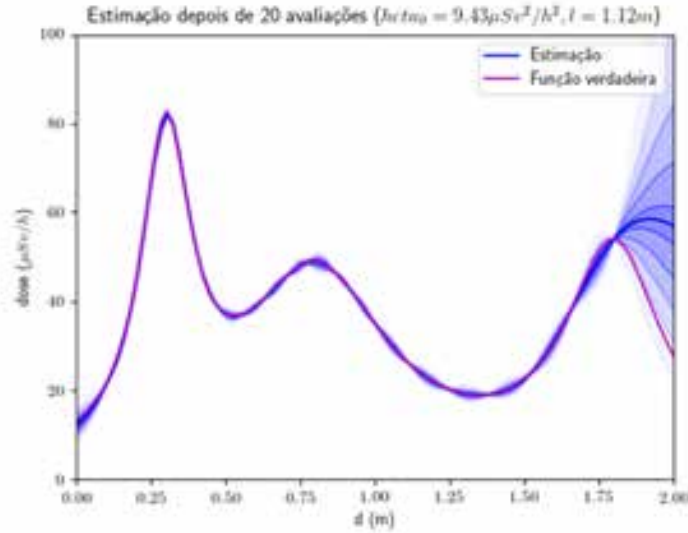


Figura 17. Distribuição de probabilidade baseada em vinte pontos
Fonte: Autor

No contexto da robótica, também é desejável controlar a distância do caminho percorrido pelo robô, favorecendo a seleção de pontos mais próximos da localização atual do robô. Isto é feito adicionando uma penalização (equação 14) ao critério de seleção de ponto (equação 13), o que resulta em uma função de otimização penalizada apresentada na equação 15 e ilustrada graficamente na Figura 18.

$$P(||x_{n+1} - x_n||) * h(g(x_{n+1})) \quad (15)$$

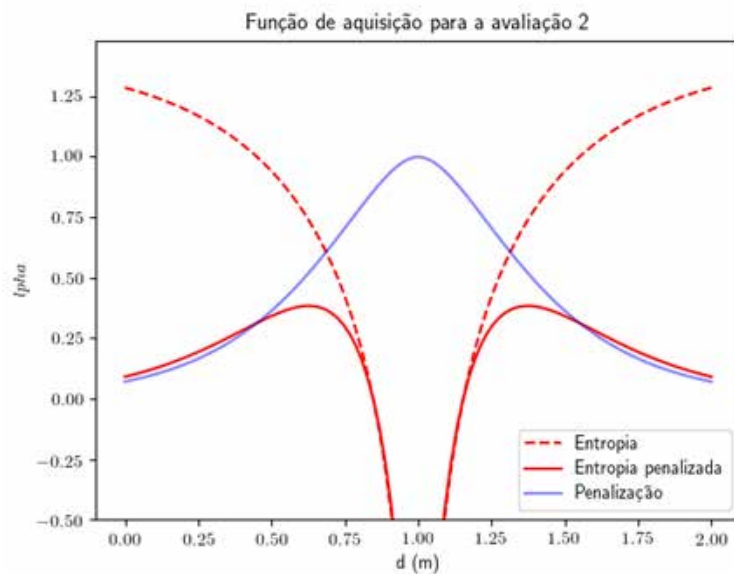


Figura 18. Função de otimização penalizada
Fonte: Autor

4.5 Modelagem do Sistema

Uma modelagem de sistemas é o processo de desenvolvimento de modelos abstratos de um sistema, de modo que cada modelo apresente uma visão ou perspectiva diferente do sistema. É uma perspectiva comportamental, onde se modela dinamicamente o comportamento do sistema e sua resposta a eventos. Geralmente utilizados durante a engenharia de requisitos para esclarecer o que o sistema existente faz, bem como subsidiar o levantamento das suas potencialidades e fragilidades (Sommerville, 2011).

De maneira a modelar uma ferramenta de simulação computacional para descrever as condições reais de um ambiente de distribuição de taxa de dose, estimou-se com base nos valores radiométricos de referência a seguinte função de taxa de dose g :

$$(x, y; x_0, y_0, k, h) = \frac{K}{(h^2 + (x - x_0)^2 + (y - y_0)^2)} \quad (15)$$

Onde:

$$\left\{ \begin{array}{l} k \text{ em } \mu Sv * m^2 \\ h \text{ em metros} \\ x_0 \text{ em metros} \\ y_0 \text{ em metros} \end{array} \right.$$

Para a observação e a investigação do modelo desenvolvido, foram estipulados três níveis graduais de dificuldade de distribuição de taxa de dose, representadas através das superfícies topológicas de quatro, seis e oito picos de máximo.

Figura 19 ilustra a função de quatro picos apresentada na equação 16:

$$f(x, y) = g_1(x, y) + g_2(x, y) + g_3(x, y) + g_4(x, y) \quad (16)$$

Onde:

$$\left\{ \begin{array}{l} g_1 = g(x, y; 4.0, 4.0, 96.0, 1.0) \\ g_2 = g(x, y; 14.0, 14.0, 112.0, 1.2) \\ g_3 = g(x, y; 15.2, 5.6, 96.0, 1.0) \\ g_4 = g(x, y; 4.8, 16.6, 80.0, 0.8) \end{array} \right.$$

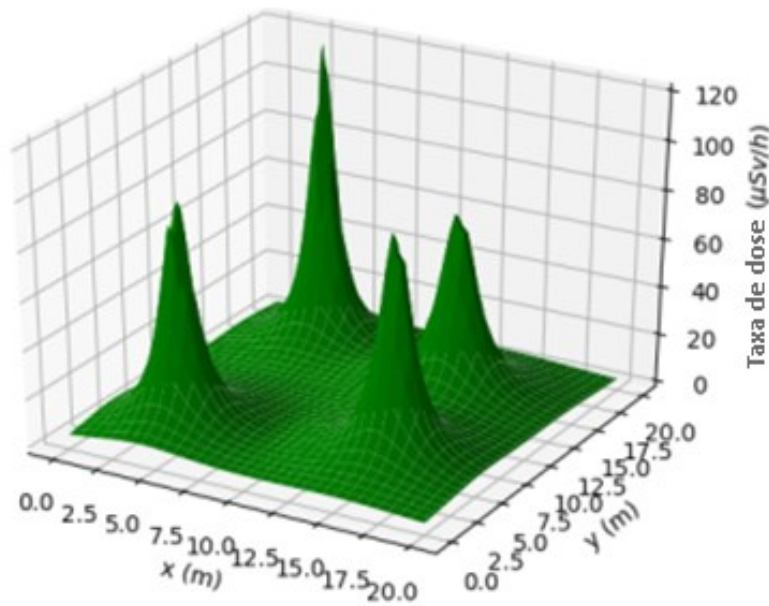


Figura 19. Superfície de quatro picos de máximo

Fonte: Autor

Figura 20 ilustra a função de seis picos apresentada na equação 17:

$$f(x, y) = g_1(x, y) + g_2(x, y) + g_3(x, y) + g_4(x, y) + g_5(x, y) + g_6(x, y) \quad (17)$$

Onde:

$$\left\{ \begin{array}{l} g1 = g(x, y; 4.0, 4.0, 160.0, 1.6) \\ g2 = g(x, y; 10.0, 16.0, 320.0, 2.8) \\ g3 = g(x, y; 11.2, 8.0, 80.0, 1.0) \\ g4 = g(x, y; 14.0, 2.0, 1280.0, 6.0) \\ g5 = g(x, y; 3.0, 16.6, 67.2, 0.8) \\ g6 = g(x, y; 16.8, 12.8, 640.0, 4.0) \end{array} \right.$$

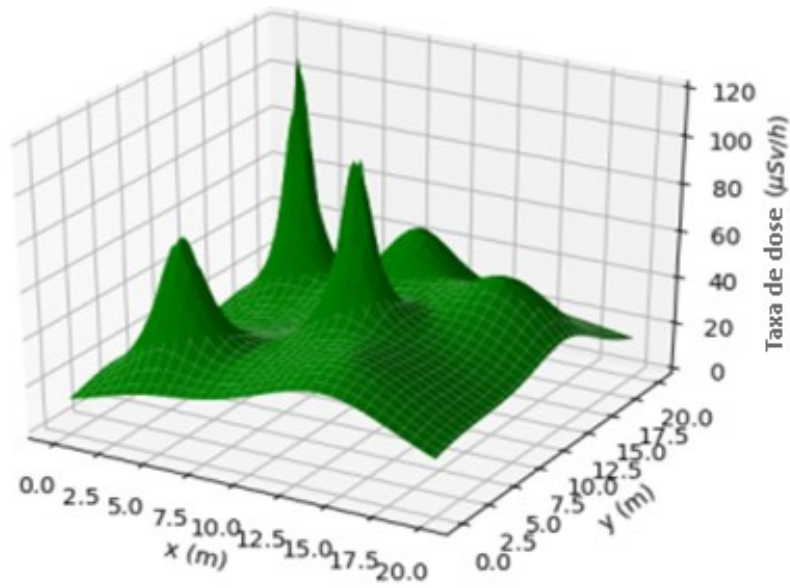


Figura 20. Superfície de seis picos de máximo

Fonte: Autor

Figura 21 ilustra a função de oito picos apresentada na equação 18:

$$f(x, y) = g_1(x, y) + g_2(x, y) + g_3(x, y) + g_4(x, y) + g_5(x, y) + g_6(x, y) + g_7(x, y) + g_8(x, y) \quad (18)$$

Onde:

$$\left\{ \begin{array}{l} g1 = g(x, y; 4.0, 4.0, 160.0, 1.6) \\ g2 = g(x, y; 10.0, 16.0, 320.0, 2.8) \\ g3 = g(x, y; 11.2, 8.0, 80.0, 1.0) \\ g4 = g(x, y; 14.0, 2.0, 1280.0, 1.0) \\ g5 = g(x, y; 3.0, 16.6, 67.2, 0.8) \\ g6 = g(x, y; 16.8, 5.6, 640.0, 4.0) \\ g7 = g(x, y; 15.2, 5.6, 96.0, 1.0) \\ g8 = g(x, y; 14.0, 14.0, 80.0, 1.2) \end{array} \right.$$

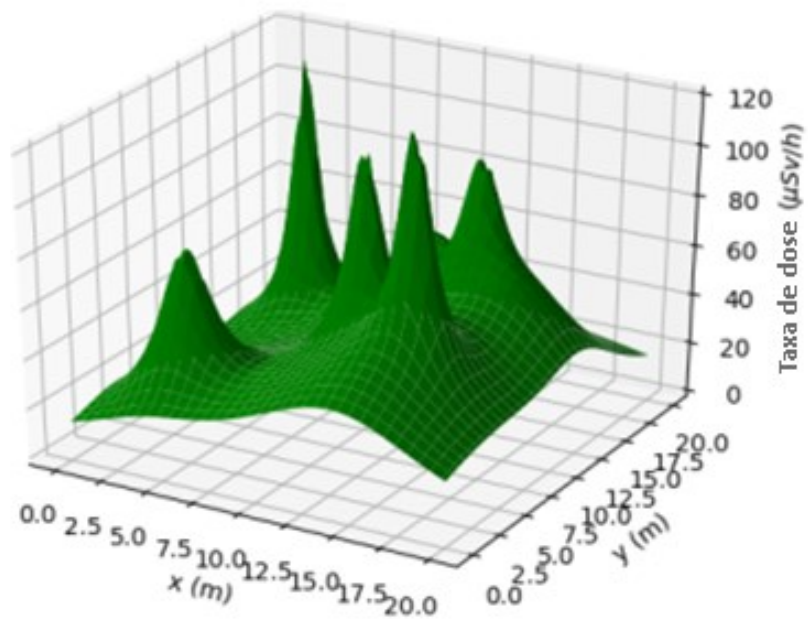


Figura 21. Superfície de oito picos de máximo

Fonte: Autor

Capítulo 5

DESEMPENHO DO ALGORITMO DE RECONSTRUÇÃO DE SUPERFÍCIE

Este capítulo aborda as variações do algoritmo de reconstrução de superfície, apresentando de forma comparativa os resultados obtidos através das simulações computacionais.

5.1 Critério de penalização e suas variações

Com o objetivo de verificar a robustez do método, foram executadas três rodadas de simulações computacionais para cada um das quinze variações do algoritmo, alternando os cenários hipotéticos das superfícies de 4, 6 e 8 picos, o limite inferior de penalização w_p nos valores de 1, 0,8 e 0,4 e o parâmetro de distância de penalização l_p nos valores 4 m e 8 m. Estas variações receberam as seguintes denominações conforme ilustra a Tabela 1.

Tabela 1. Variações dos métodos do ARS

Fonte: Autor

Sigla	Descrição
ARS 4P SP	ARS para topologia de 4 picos de máximo sem penalização
ARS 4P Wp40 Lp4	ARS para topologia de 4 picos de máximo, com limite inferior de penalização $w_p = 0,4$ e parâmetro de distância de penalização $l_p = 4m$
ARS 4P Wp40 Lp8	ARS para topologia de 4 picos de máximo, com limite inferior de penalização $w_p = 0,4$ e parâmetro de distância de penalização $l_p = 8m$
ARS 4P Wp80 Lp4	ARS para topologia de 4 picos de máximo, com limite inferior de penalização $w_p = 0,8$ e parâmetro de distância de penalização $l_p = 4m$
ARS 4P Wp80 Lp8	ARS para topologia de 4 picos de máximo, com limite inferior de penalização $w_p = 0,8$ e parâmetro de distância de penalização $l_p = 8m$
ARS 6P SP	ARS, topologia de 6 picos de máximo, sem penalização
ARS 6P Wp40 Lp4	ARS para topologia de 6 picos de máximo, com limite inferior de penalização $w_p = 0,4$ e parâmetro de distância de penalização $l_p = 4m$
ARS 6P Wp40 Lp8	ARS para topologia de 6 picos de máximo, com limite inferior de penalização $w_p = 0,4$ e parâmetro de distância de penalização $l_p = 8m$
ARS 6P Wp80 Lp4	ARS para topologia de 6 picos de máximo, com limite inferior de penalização $w_p = 0,8$ e parâmetro de distância de penalização $l_p = 4m$
ARS 6P Wp80 Lp8	ARS para topologia de 6 picos de máximo, com limite inferior de penalização $w_p = 0,8$ e parâmetro de distância de penalização $l_p = 8m$
ARS 8P SP	ARS, topologia de 8 picos de máximo, sem penalização
ARS 8P Wp40 Lp4	ARS para topologia de 8 picos de máximo, com limite inferior de penalização $w_p = 0,4$ e parâmetro de distância de penalização $l_p = 4m$
ARS 8P Wp40 Lp8	ARS para topologia de 8 picos de máximo, com limite inferior de penalização $w_p = 0,4$ e parâmetro de distância de penalização $l_p = 8m$
ARS 8P Wp80 Lp4	ARS para topologia de 8 picos de máximo, com limite inferior de penalização $w_p = 0,8$ e parâmetro de distância de penalização $l_p = 4m$
ARS 8P Wp80 Lp8	ARS para topologia de 8 picos de máximo, com limite inferior de penalização $w_p = 0,8$ e parâmetro de distância de penalização $l_p = 8m$

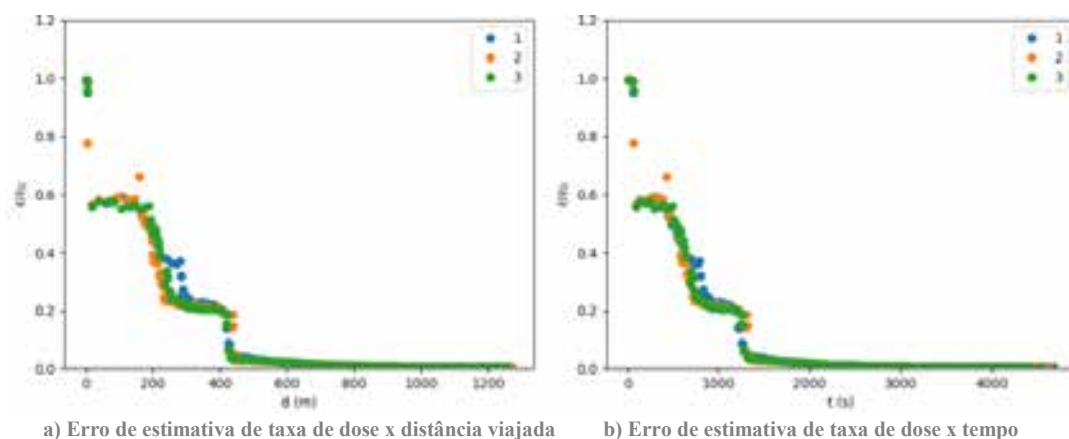
Para cada uma das três superfícies de radiação modeladas (topologia de 4, 6 e 8 picos), foram realizadas um total de 150 avaliações (pontos de medição) para cada uma das 3 rodadas do ARS efetuadas, sendo as avaliações: 5, 10, 25, 50, 75, 100, 125 e 150 tomadas como referência para análise da evolução do algoritmo.

As versões ARS 4P SP, ARS 6P SP e ARS 8P SP possuem os valores de $w_p = 1$, o que caracteriza a não utilização do critério de penalização. Cabe ressaltar que a utilização do critério de penalização tem como objetivo minimizar a distância percorrida pelo robô no decorrer do levantamento radiométrico do ambiente, favorecendo a seleção de pontos mais próximos da localização atual do robô e resultando em um menor tempo de reconstrução da superfície.

Nas próximas seções serão apresentadas de forma detalhada o desempenho do algoritmo sem a utilização do critério de penalização (ARS 4P SP, ARS 6P SP e ARS 8P SP), bem como as melhores performances para cada cenário hipotético, utilizando o critério de penalização que foram o ARS 4P Wp80 Lp4, ARS 6P Wp80 Lp8 e ARS 8P Wp40 Lp4. Ao final deste capítulo será apresentado um mapa comparativo de todas as versões de algoritmos testadas computacionalmente. As evoluções de todas as variações do ARS testadas podem ser observadas de maneira mais detalha no anexo desta tese.

5.2 Performance do ARS 4P SP

Com base nos dados obtidos após a realização de 3 rodadas de simulações, foram gerados gráficos visando comparar o progresso do algoritmo, bem como analisar sua repetitividade e reprodutibilidade de resultados. Para tal foram elaborados comparativos do erro de estimativa de taxa de dose por distância viajada pelo robô, do erro de estimativa de taxa de dose pelo tempo gasto durante o deslocamento e pelo erro de estimativa de taxa de dose por cada avaliação realizada, conforme apresentado na figura 22.



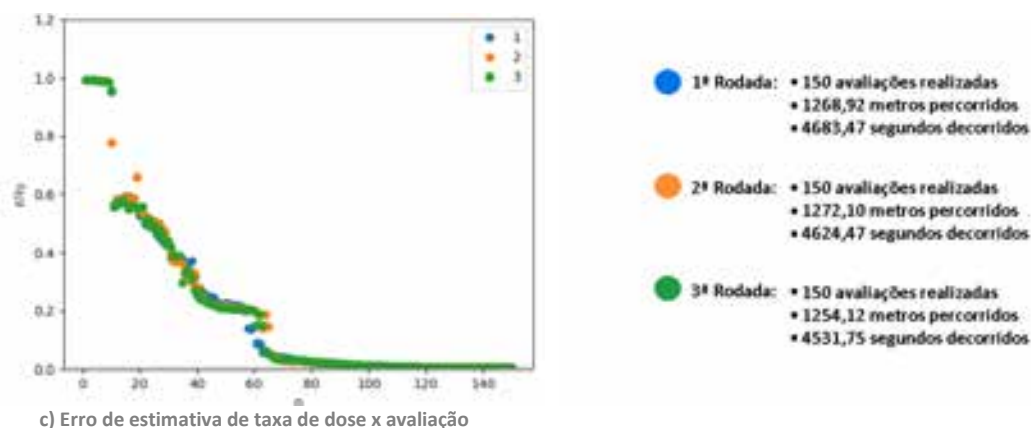


Figura 22. Comparativo do erro estimado de taxa de dose entre as rodadas do ARS 4P SP
Fonte: Autor

A figura 23 apresenta de forma gráfica a reprodução do tempo decorrido através das 150 avaliações realizadas, exibindo as curvas correspondentes ao Tempo total da simulação, o Tempo de viagem do robô, Tempo de cálculo de rota e o Tempo de medida, bem como o delineamento do trajeto realizado pelo robô no decorrer da navegação.

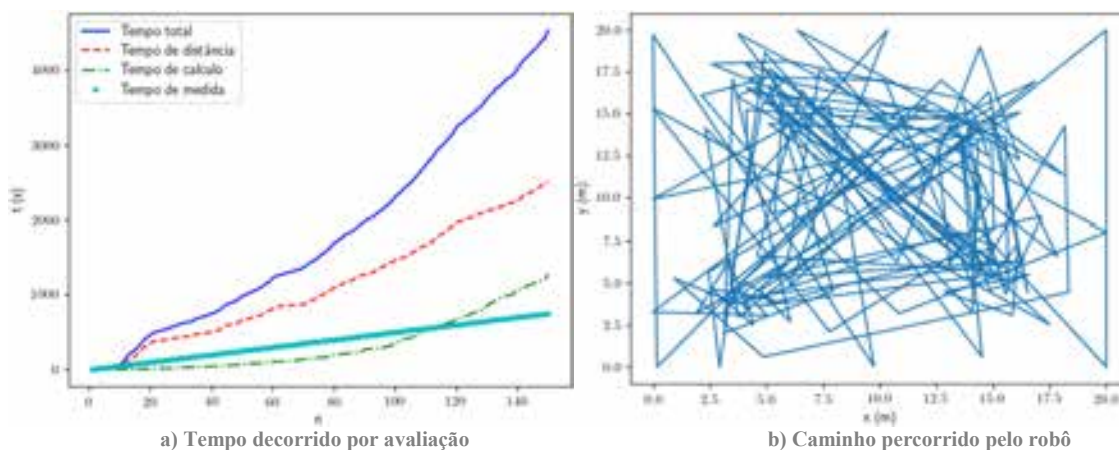


Figura 23. Comparativo de tempos e delineamento do trajeto do robô com o ARS 4P SP
Fonte: Autor

A análise da evolução do ARS 4P SP, demonstra que após a septuagésima quinta avaliação de cada rodada o algoritmo consegue reconstruir o campo de radiação real com um erro estimado de taxa de dose consideravelmente baixo, conforme a evolução demonstrada nas figuras 24, 25 e 26.

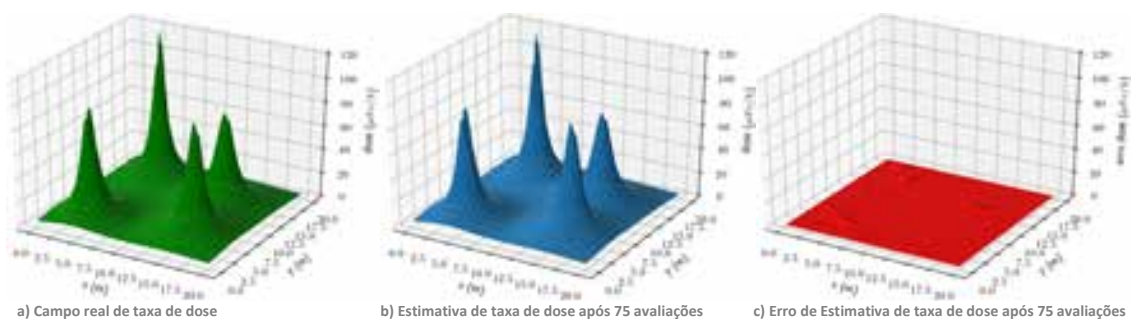


Figura 24. Resultado da 1ª rodada do ARS 4P SP após 75 avaliações

Fonte: Autor

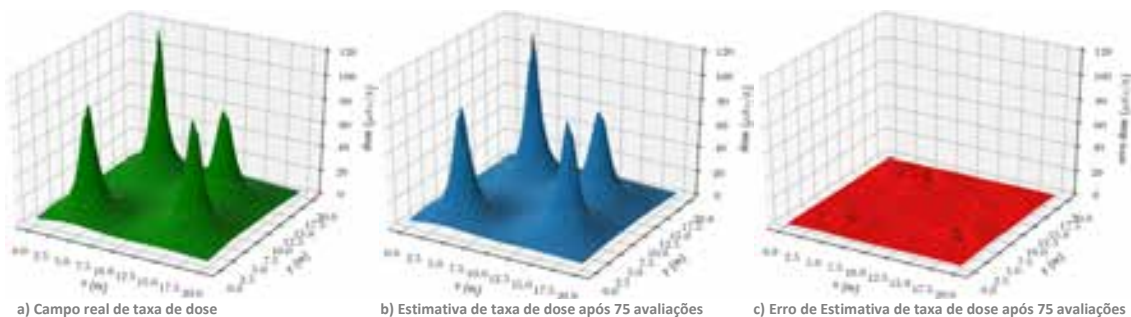


Figura 25. Resultado da 2ª rodada do ARS 4P SP após 75 avaliações

Fonte: Autor

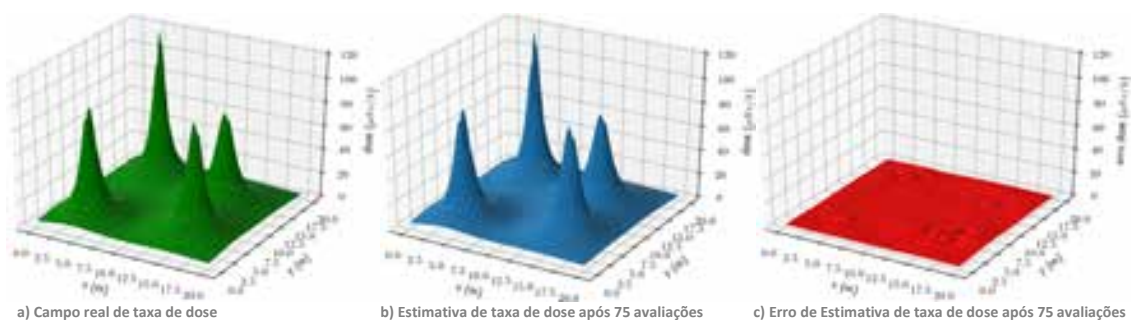


Figura 26. Resultado da 3ª rodada do ARS 4P SP após 75 avaliações

Fonte: Autor

Os resultados das performances da 1ª, 2ª e 3ª rodadas do ARS 4P SP alcançaram um erro médio de 2,91 conforme os valores apresentados na Tabela 2:

Tabela 2. Média dos resultados do ARS 4P SP após 75 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	75	2,91	1667,4s	500,6m
2ª	75	2,91	1670,2s	501,7m
3ª	75	2,90	1672,7s	503,1m
Média	75	2,91	1670,1s	501,7m

A evolução do algoritmo após a decorrência das 150 avaliações, não apresenta uma melhora expressiva dos resultados, pois apesar do erro de estimativa de taxa de dose reduzir em 92,1% em comparação a avaliação 75, o tempo decorrido aumenta em 176% e há um acréscimo de 152% na distância percorrida, conforme apresentado na Tabela 3.

Tabela 3. Média dos resultados do ARS 4P SP após 150 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	150	0,23	4683,4s	1268,9m
2ª	150	0,23	4624,4s	1272,1m
3ª	150	0,22	4531,7s	1254,1m
Média	150	0,23	4613,1s	1265,0m

A Figura 27 mostra as capturas de tela do número intermediário de pontos visitados (5, 10, 25, 50, 75, 100, 125 e 150) e a melhora seqüencial obtida com o decorrer da evolução do algoritmo.

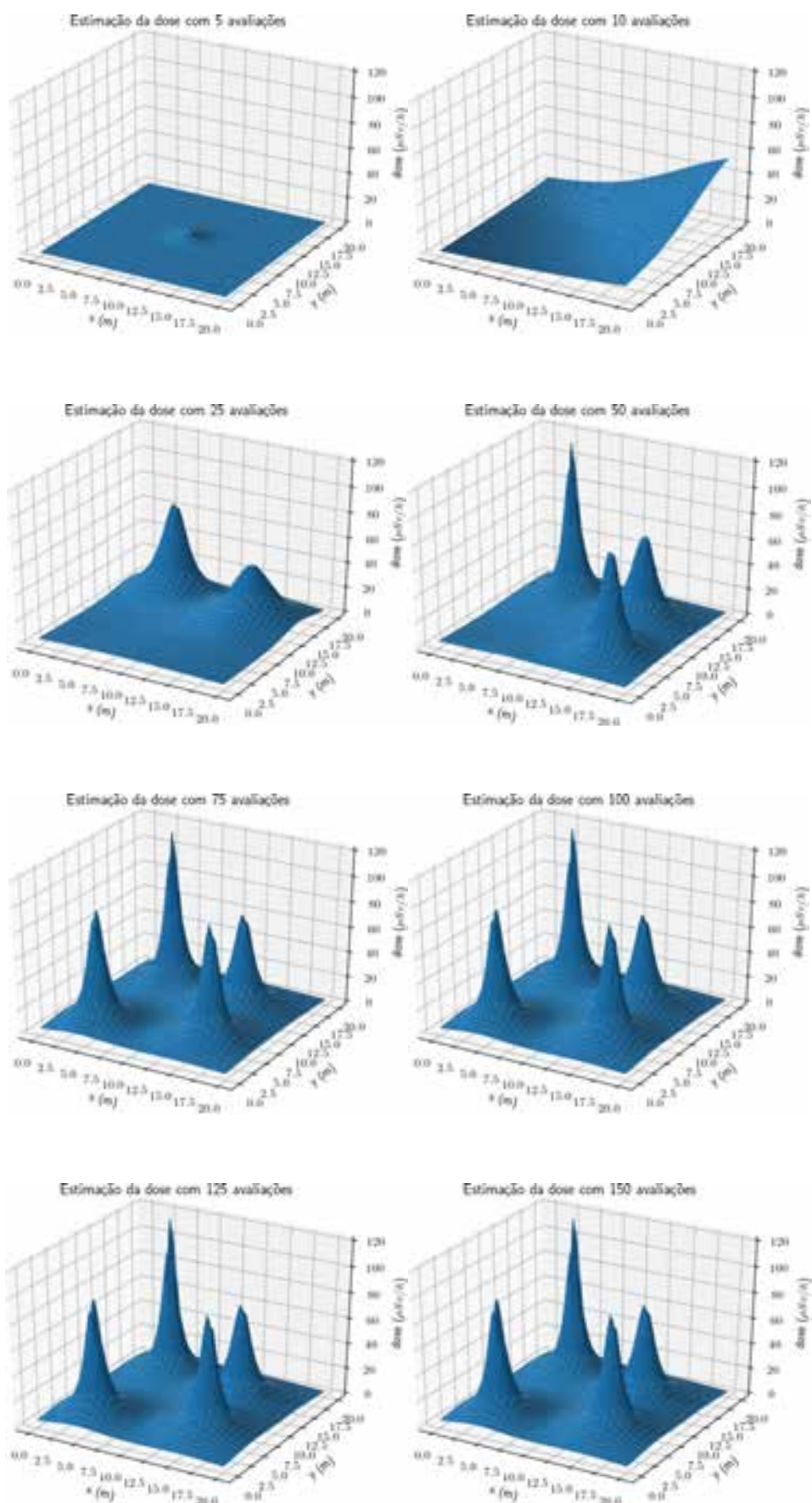


Figura 27. Evolução do ARS 4P SP
Fonte: Autor

O evolução do desempenho de todas as 150 avaliações de cada uma das 3 rodadas de simulação computacional do ARS 4P SP, encontra-se melhor detalhada no Anexo desta tese.

5.3 Performance do ARS 6P SP

Os gráficos apresentados na figura 28 trazem de forma comparativa a relação do erro de estimativa de taxa de dose por distância viajada pelo robô, do erro de estimativa de taxa de dose pelo tempo gasto durante o deslocamento e do erro de estimativa de taxa de dose por cada avaliação realizada.

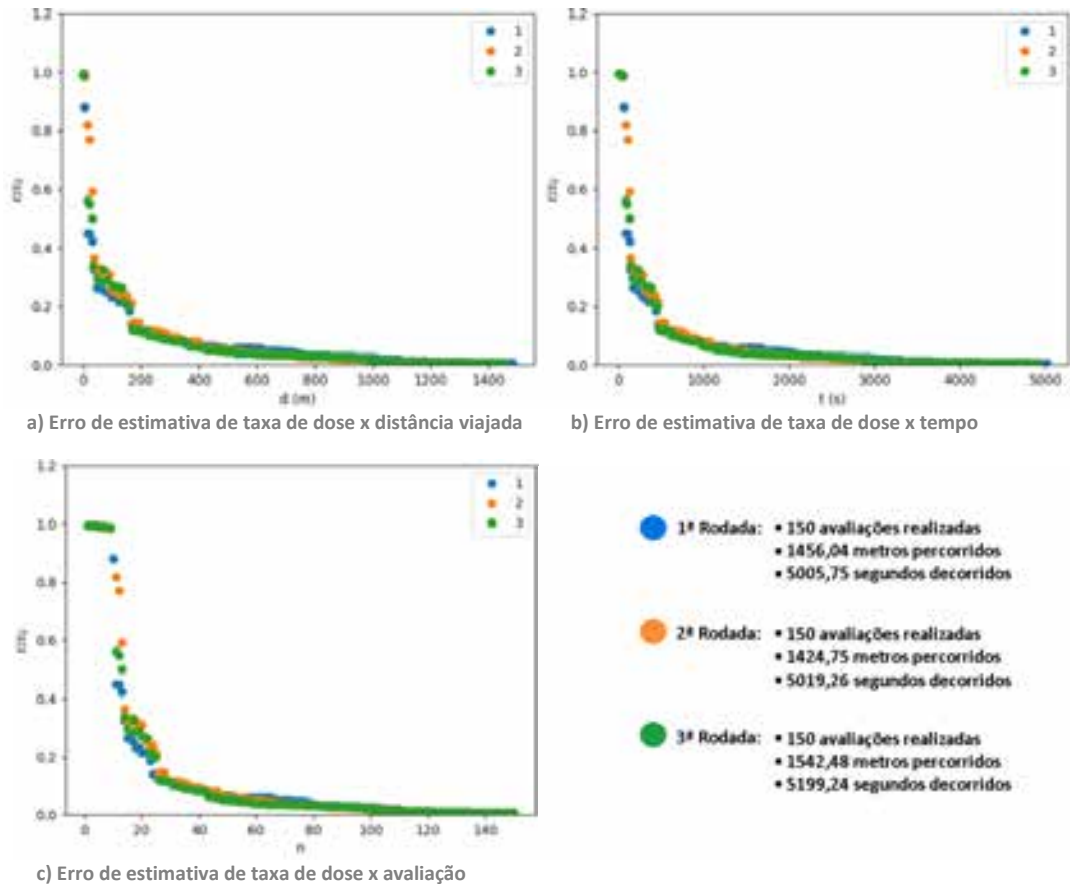


Figura 28. Comparativo do erro estimado de taxa de dose entre as rodadas do ARS 6P SP

Fonte: Autor

A visualização gráfica da evolução dos tempos pode ser observada na figura 29, a qual exibe o excursionamento das curvas referentes ao Tempo total da simulação, o Tempo de viagem do robô, Tempo de cálculo de rota e o Tempo de medida e apresenta traçado do caminho percorrido pelo robô durante toda a execução do algoritmo.

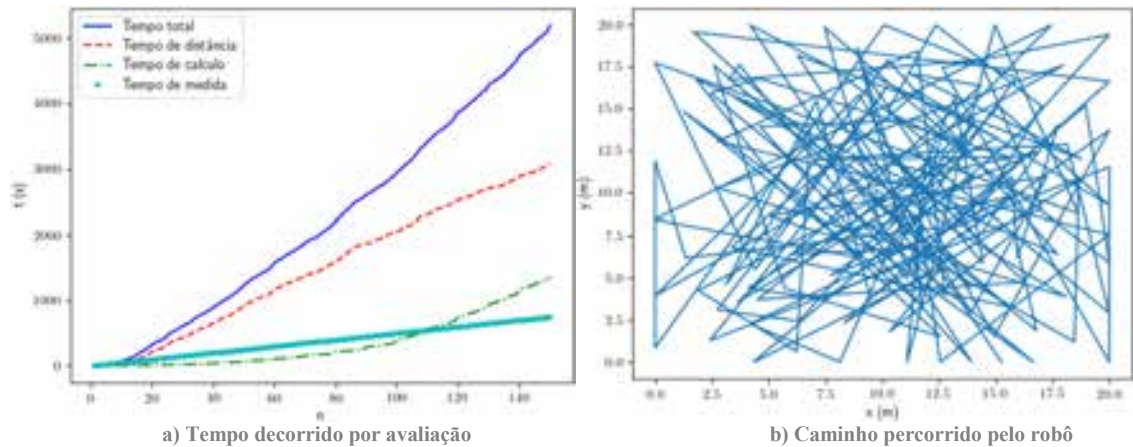


Figura 29. Comparativo de tempos e delineamento do trajeto do robô com o ARS 6P SP
Fonte: Autor

A evolução do ARS 6P SP no decorrer das três rodadas de simulações computacionais, apresenta uma boa eficiência a partir da centésima vigésima quinta avaliação, onde alcança um erro estimado de taxa de dose significativamente reduzido durante a reconstrução do campo de radiação real. As figuras 30, 31 e 32 apresentam graficamente este resultado.

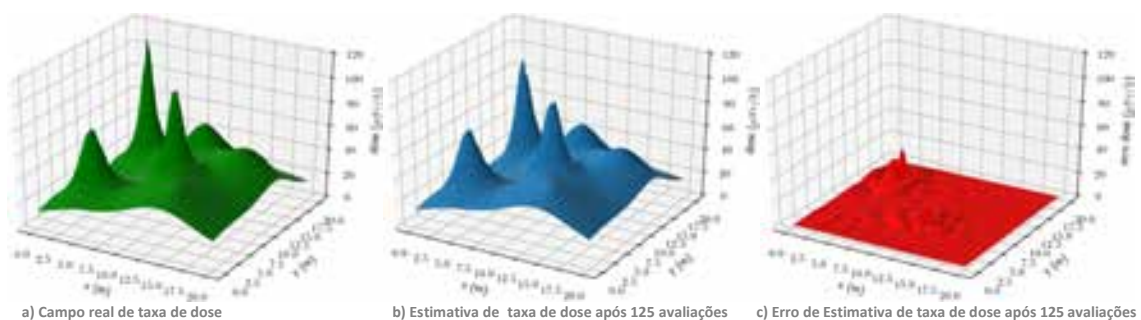


Figura 30. Resultado da 1ª rodada do ARS 6P SP após 125 avaliações
Fonte: Autor

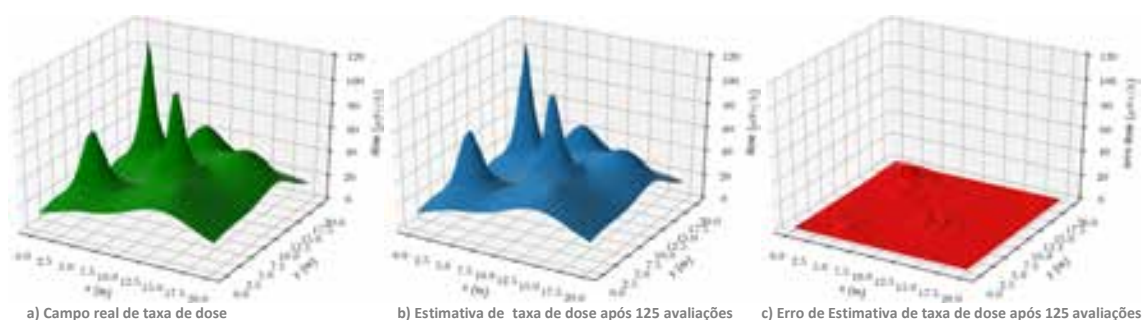


Figura 31. Resultado da 2ª rodada do ARS 6P SP após 125 avaliações

Fonte: Autor

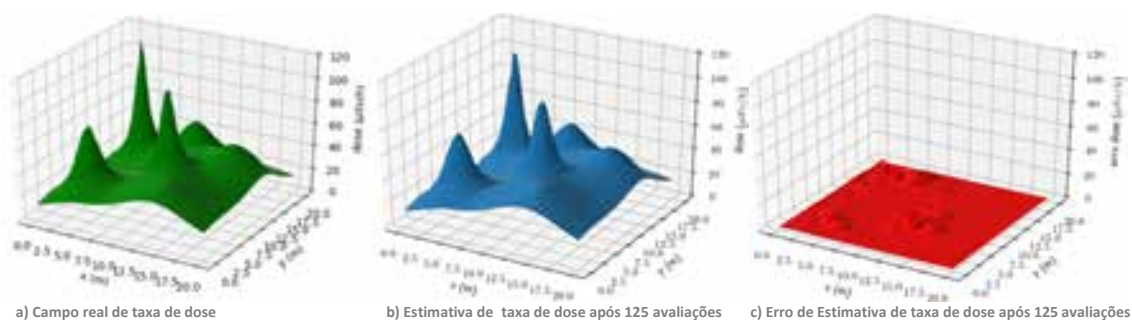


Figura 32. Resultado da 3ª rodada do ARS 6P SP após 125 avaliações

Fonte: Autor

Os resultados dos desempenhos das 1ª, 2ª e 3ª rodadas do ARS 6P SP alcançaram um erro médio de 1,35 conforme os valores apresentados na tabela 4:

Tabela 4. Média dos resultados do ARS 6P SP após 125 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	125	1,36	3931,5s	1355,1m
2ª	125	1,35	3815,6s	1201,3m
3ª	125	1,35	3774,5s	1132,5m
Média	125	1,35	3840,0s	1229,6m

A evolução do algoritmo após a decorrência das 150 avaliações, apresenta pouca melhora nos resultados, pois apesar do erro de estimativa de taxa de dose reduzir em 54% em comparação a avaliação 125, o tempo decorrido aumenta em 32% e há um acréscimo de 20% na distância percorrida, conforme apresentado na tabela 5.

Tabela 5. Média dos resultados do ARS 6P SP após 150 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	150	0,62	5005,7s	1456,0m
2ª	150	0,61	5019,2s	1424,7m
3ª	150	0,62	5199,2s	1542,4m
Média	150	0,62	5074,7 s	1474,4m

A Figura 33 mostra as capturas de tela do número intermediário de pontos visitados (5, 10, 25, 50, 75, 100, 125 e 150) e a melhoria seqüencial obtida com o decorrer da evolução do algoritmo.

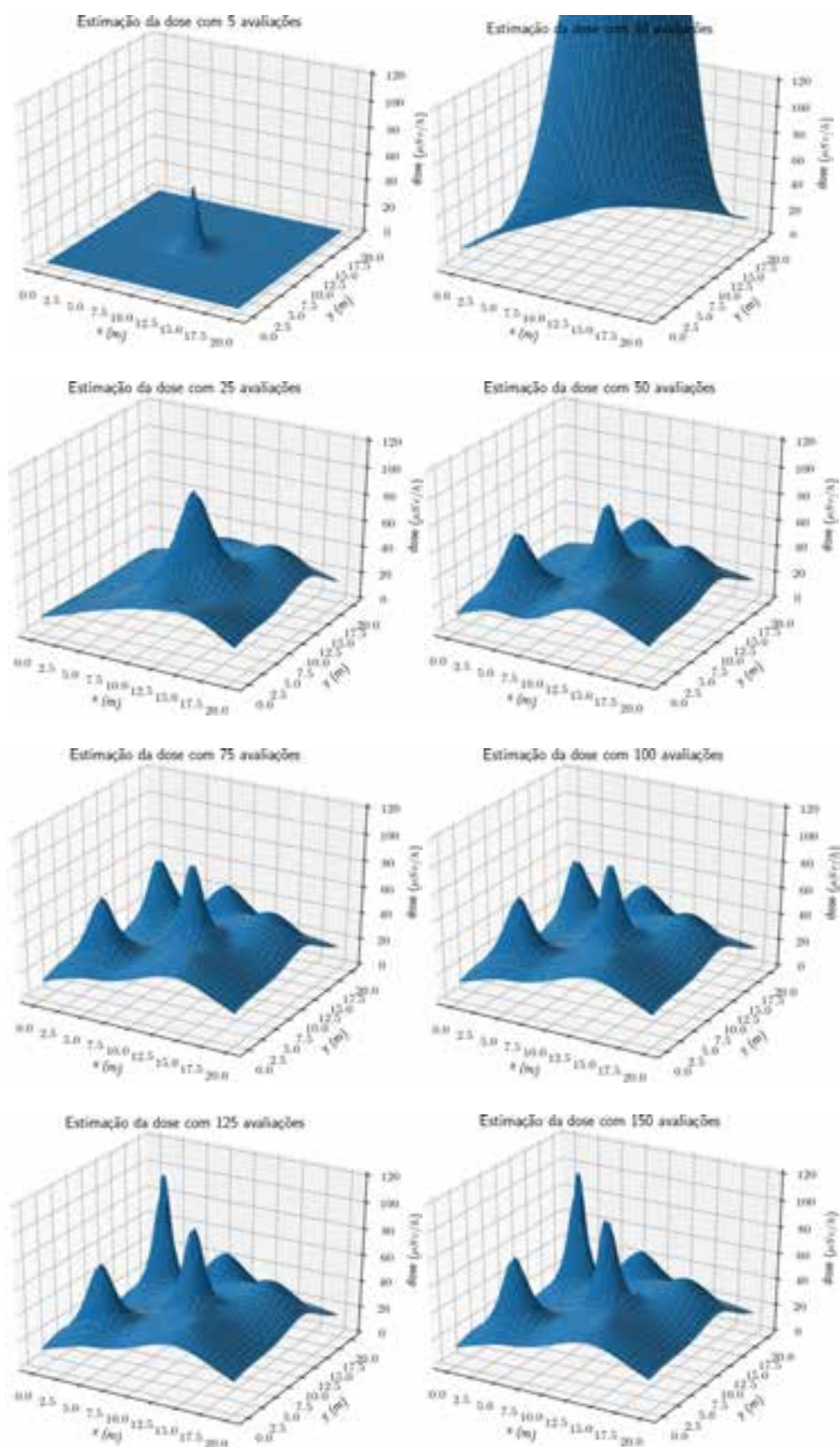


Figura 33. Evolução do ARS 6P SP

Fonte: Autor

A evolução sequencial obtida no decorrer das simulações realizadas com ARS 6P SP, juntamente com suas tabelas de performance, encontra-se melhor detalhada no Anexo desta tese.

5.4 Performance do ARS 8P SP

A figura 34 apresenta de forma gráfica, uma comparativa da relação do erro de estimativa de taxa de dose por distância viajada pelo robô, do erro de estimativa de taxa de dose pelo tempo gasto durante o deslocamento e do erro de estimativa de taxa de dose verificado através das três rodadas de simulações computacionais do ARS 8P SP.

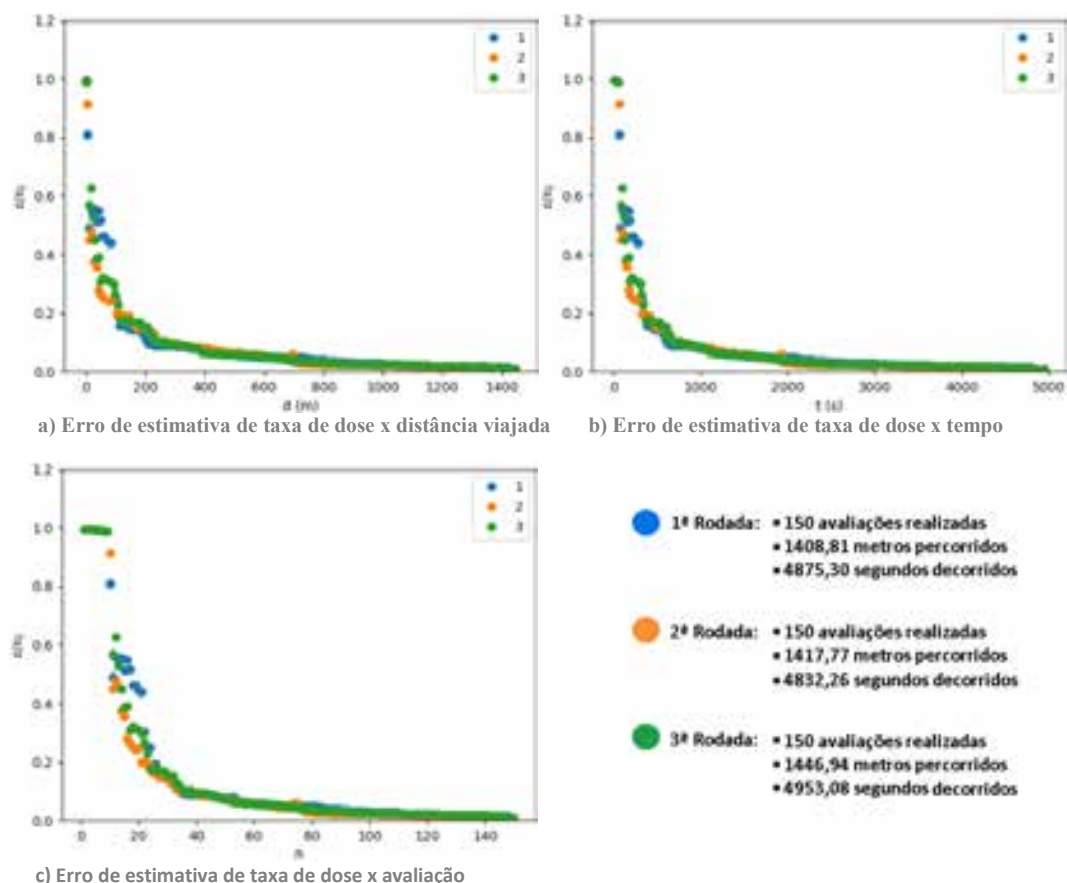


Figura 34. Comparativo do erro estimado de taxa de dose entre as rodadas do ARS 8P SP

Fonte: Autor

Na figura 35 são apresentadas as curvas pertencentes ao Tempo total da simulação, o Tempo de viagem do robô, Tempo de cálculo de rota, o Tempo de medida e também o traçado do caminho percorrido pelo robô durante toda a execução do algoritmo.

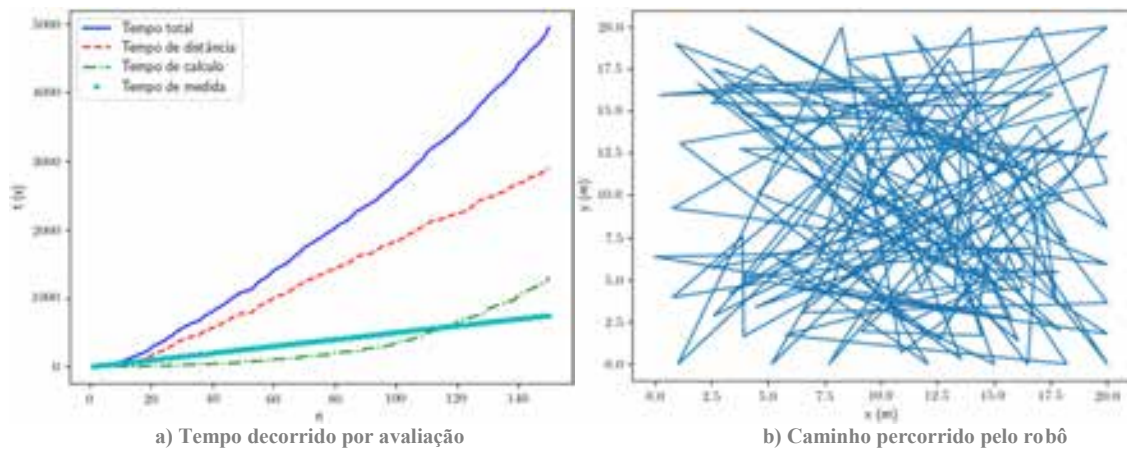


Figura 35. Comparativo de tempos e delineamento do trajeto do robô com o ARS 8P SP
Fonte: Autor

Durante a análise da evolução do ARS 8P SP pode-se perceber que o algoritmo alcançou uma boa eficiência após a última rodada de avaliação, onde atingiu um erro estimado de taxa de dose próximo aos resultados alcançados anteriormente pelo ARS 4P SP e ARS 6P SP. As figuras 36, 37 e 38 retratam graficamente este resultado.

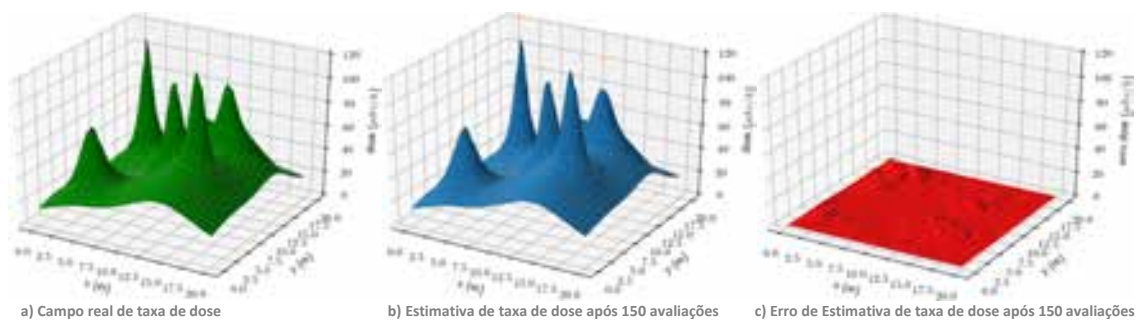


Figura 36. Resultado da 1ª rodada do ARS 8P após 150 avaliações
Fonte: Autor

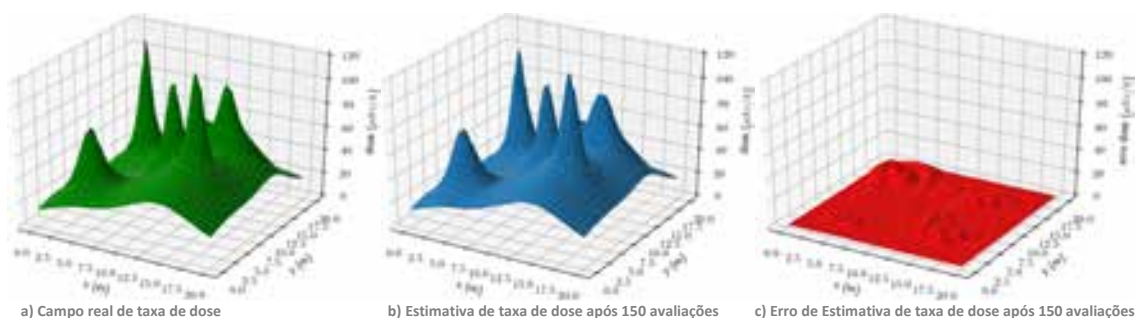


Figura 37. Resultado da 2ª rodada do ARS 8P após 150 avaliações

Fonte: Autor

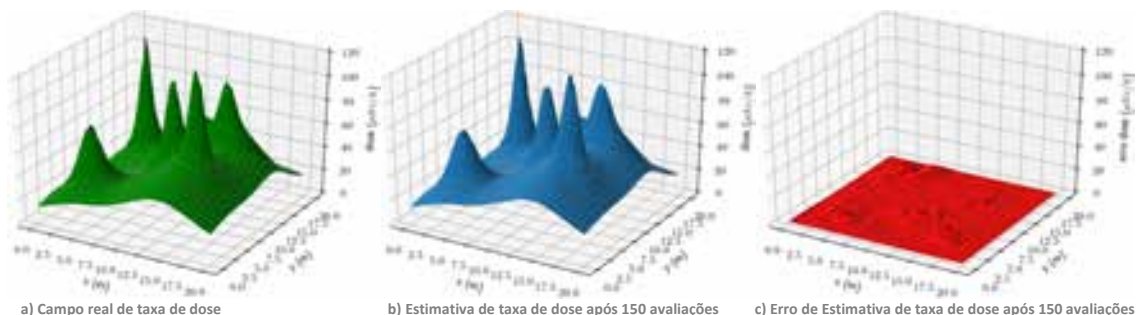


Figura 38. Resultado da 3ª rodada do ARS 8P SP após 150 avaliações

Fonte: Autor

Os resultados das performances da 1ª, 2ª e 3ª rodadas do ARS 8P SP, após a execução das 150 avaliações, alcançaram um erro médio de 0,89 conforme os valores apresentados na tabela 6:

Tabela 6. Média dos resultados do ARS 8P SP após 150 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	150	0,89	4875,3s	1408,8m
2ª	150	0,89	4832,2s	1417,7m
3ª	150	0,88	4953,0s	1446,9m
Média	150	0,89	4886,8s	1424,4m

A Figura 39 mostra as capturas de tela do número intermediário de pontos visitados (5, 10, 25, 50, 75, 100, 125 e 150) e a melhoria seqüencial obtida com o decorrer da evolução do algoritmo.

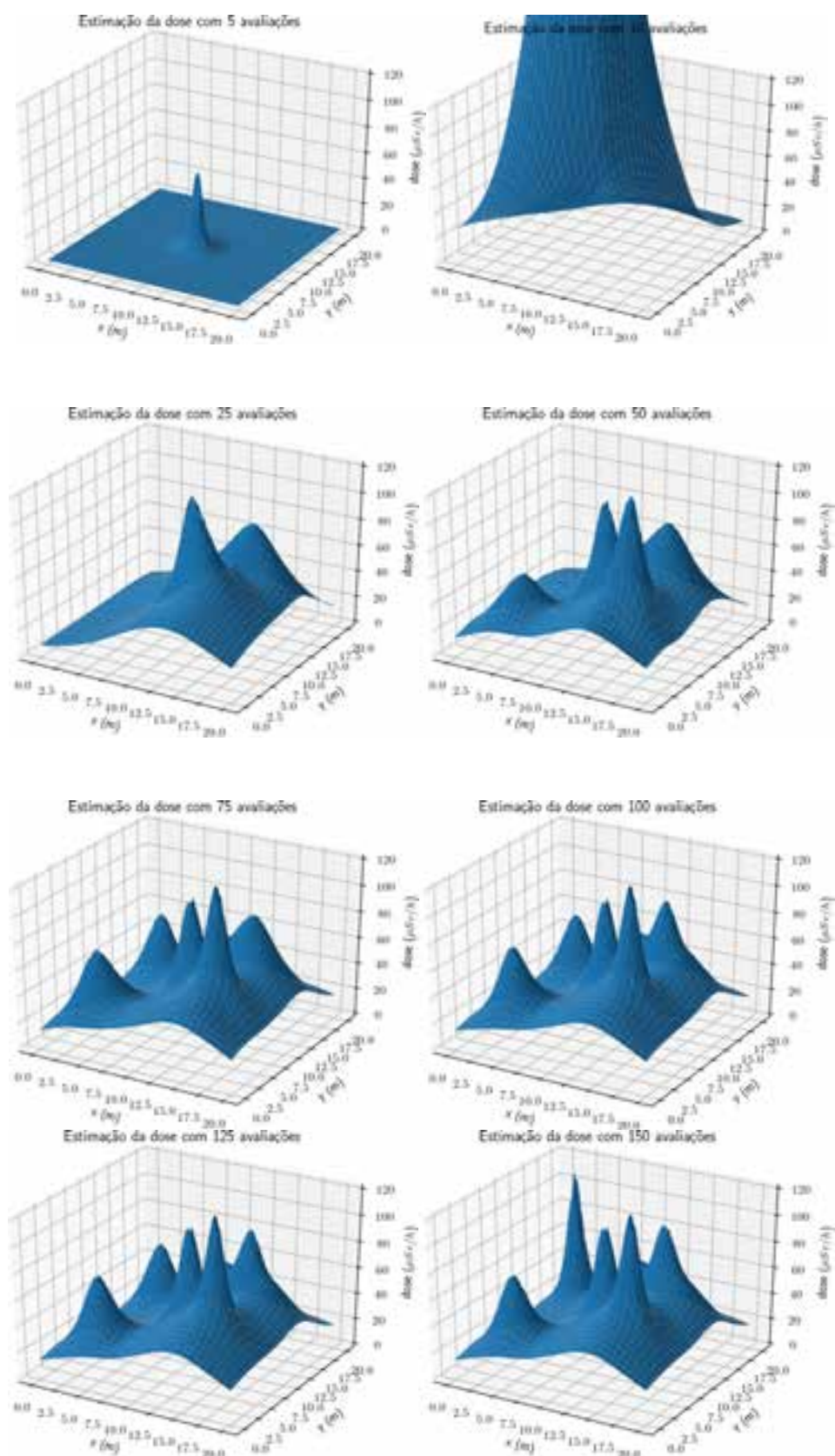


Figura 39. Evolução do ARS 8P SP

Fonte: Autor

A evolução do desempenho individual de todas as avaliações de cada uma das rodadas de simulação computacional do ARS 8P SP, encontra-se melhor detalhada no Anexo desta tese.

5.5 Performance do ARS 4P Wp80 Lp4

Os gráficos da figura 40 apresentados de forma comparativa o erro de estimativa de taxa de dose por distância viajada pelo robô, do erro de estimativa de taxa de dose pelo tempo gasto durante o deslocamento e pelo erro de estimativa de taxa de dose por cada avaliação realizada.

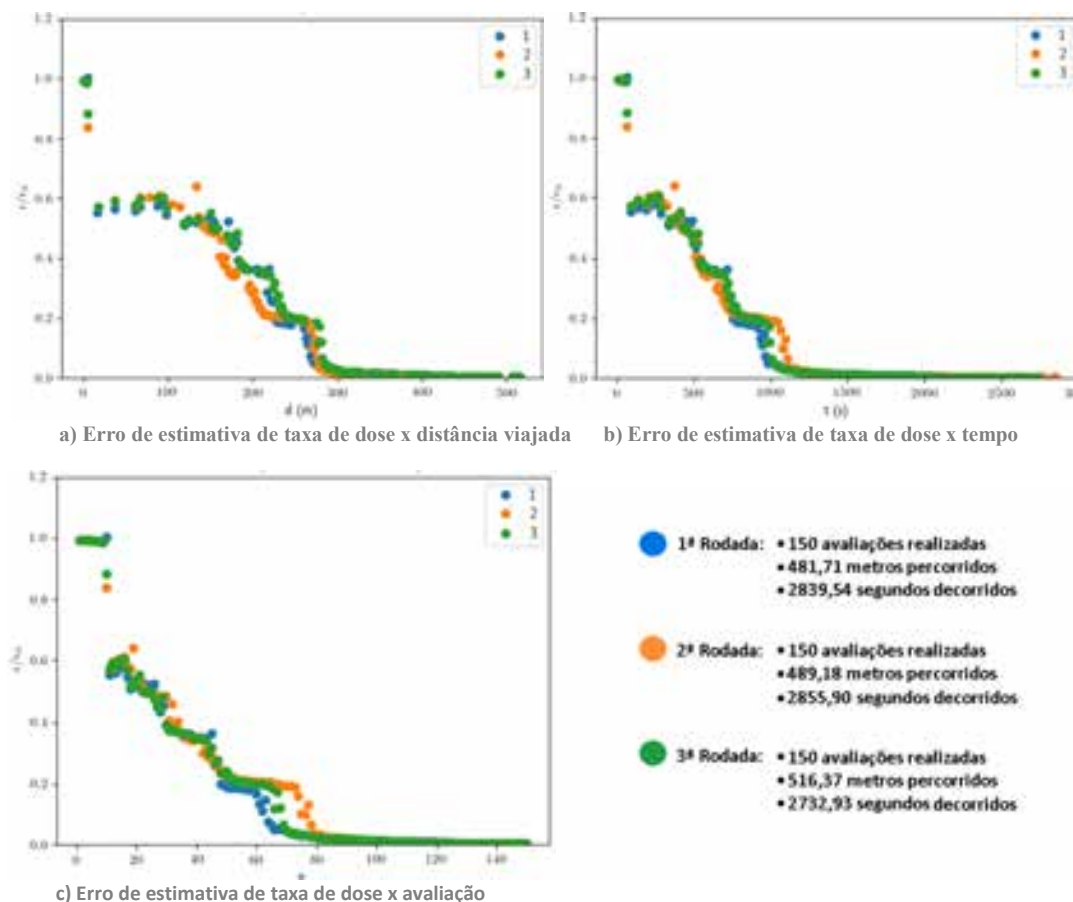


Figura 40. Comparativo do erro estimado de taxa de dose entre as rodadas do ARS 4P Wp80 Lp4

Fonte: Autor

A figura 41 apresenta de forma gráfica a reprodução do tempo decorrido através das 150 avaliações realizadas, exibindo as curvas correspondentes ao Tempo total da simulação, o Tempo de viagem do robô, Tempo de cálculo de rota e o Tempo de medida, bem como o delineamento do trajeto realizado pelo robô no decorrer da navegação.

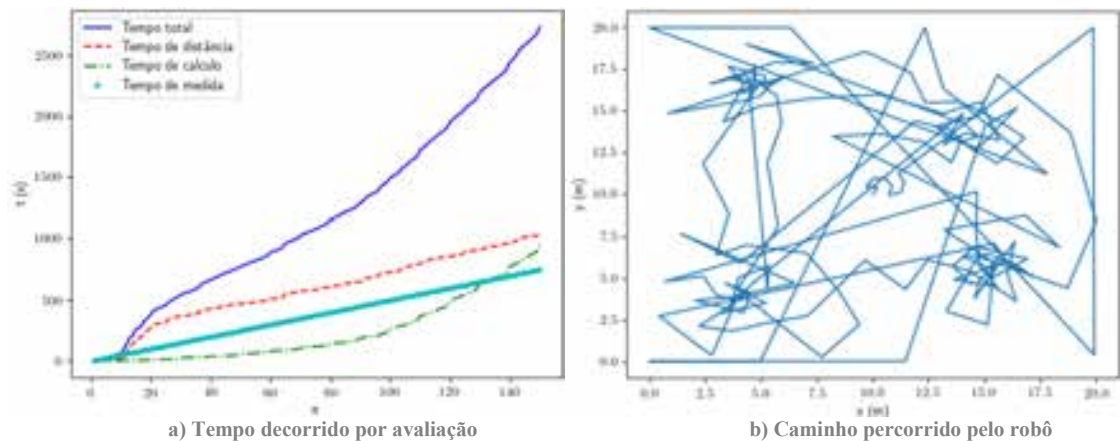


Figura 41. Comparativo de tempos e delineamento do trajeto do robô com o ARS 4P Wp80 Lp4

Fonte: Autor

A análise da evolução do algoritmo demonstra que após a septuagésima quinta avaliação de cada rodada o algoritmo consegue reconstruir o campo de radiação real com um erro estimado de taxa de dose consideravelmente baixo, conforme a evolução demonstrada nas figuras 42, 43 e 44.

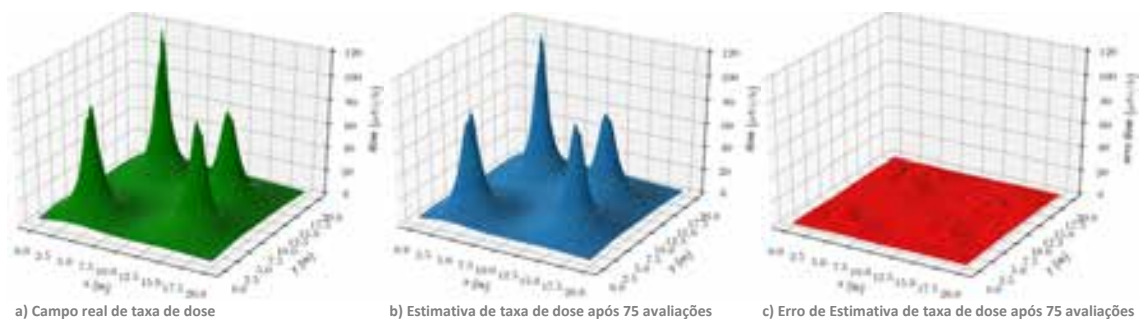


Figura 42. Resultado da 1ª rodada do ARS 4P Wp80 Lp4 após 75 avaliações

Fonte: Autor

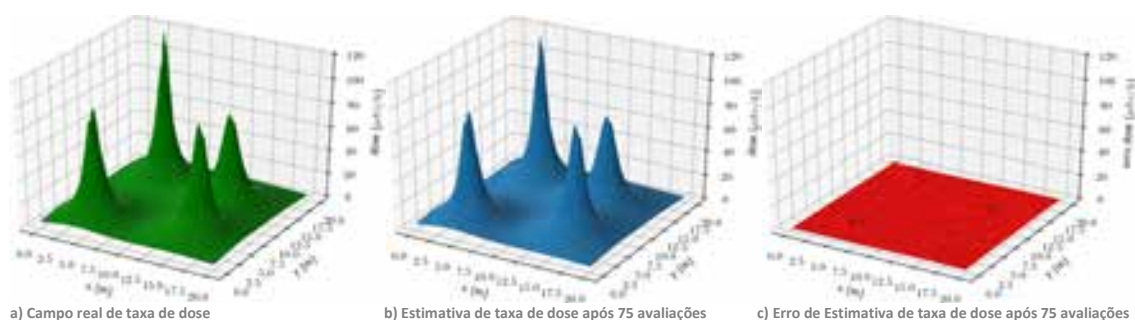


Figura 43. Resultado da 2ª rodada do ARS 4P Wp80 Lp4 após 75 avaliações

Fonte: Autor

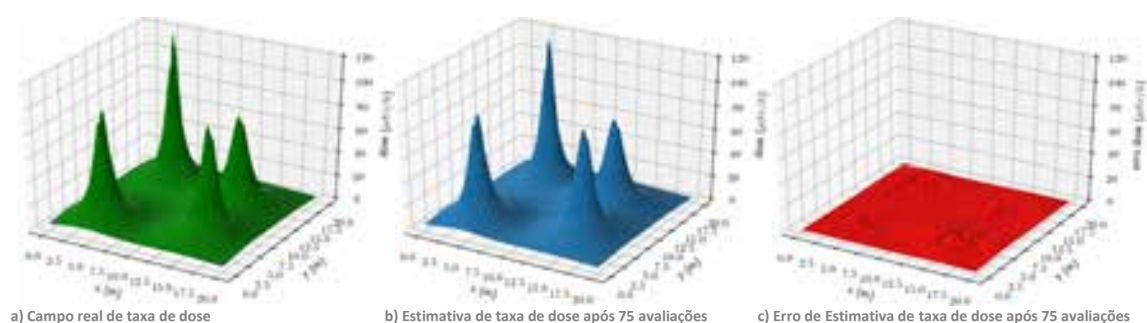


Figura 44. Resultado da 3ª rodada do ARS 4P Wp80 Lp4 após 75 avaliações

Fonte: Autor

Os resultados das performances da 1ª, 2ª e 3ª rodadas do ARS 4P Wp80 Lp4 alcançaram um erro médio de 5,42 conforme os valores apresentados na tabela 7:

Tabela 7. Média dos resultados do ARS 4P Wp80 Lp4 após 75 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	75	5,42	524,1s	138,2m
2ª	75	5,42	518,5s	135,1m
3ª	75	5,41	512,8s	136,5m
Média	75	5,42	518,4s	136,7m

A análise da evolução do algoritmo após a decorrência das 150 avaliações, mostra que o erro de estimativa de taxa de dose reduz em 94,8% em comparação a avaliação 75, enquanto que o tempo decorrido aumenta em 441,9% e há um acréscimo de 262,6% na distância percorrida, conforme os dados apresentados na tabela 8.

Tabela 8. Média dos resultados do ARS 4P Wp80 Lp4 após 150 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	t	d
1ª	150	0,28	2839,5s	481,7m
2ª	150	0,28	2855,9s	489,1m
3ª	150	0,28	2732,9s	516,3m
Média	150	0,28	2809,4s	495,7m

A Figura 45 mostra as capturas de tela do número intermediário de pontos visitados (5, 10, 25, 50, 75, 100, 125 e 150) e a melhoria seqüencial obtida com o decorrer da evolução do algoritmo.

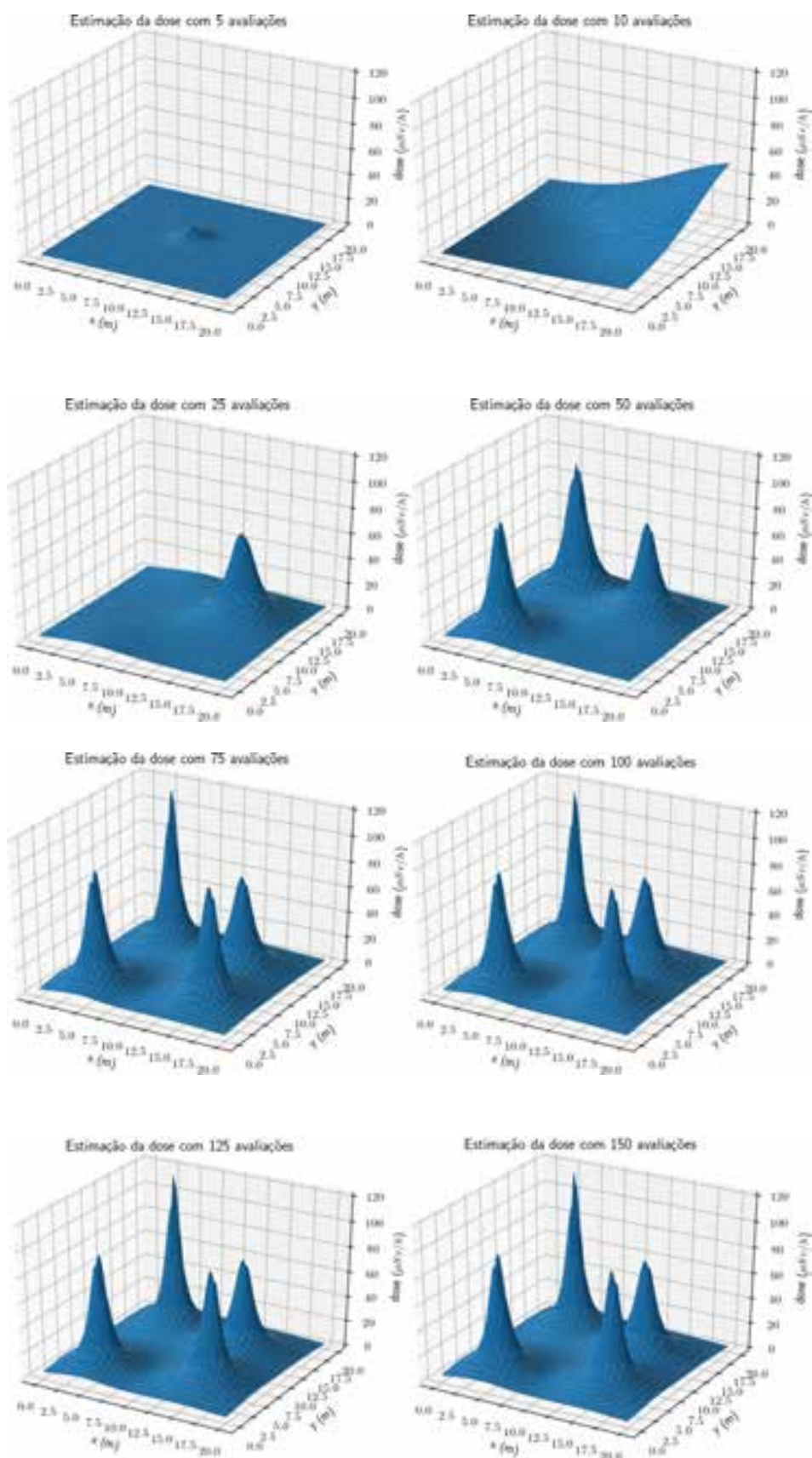


Figura 45. Evolução do ARS 4P Wp80 Lp4

Fonte: Autor

O evolução do desempenho de todas as 150 avaliações de cada uma das 3 rodadas de simulação computacional do ARS 4P Wp80 Lp4, encontra-se melhor detalhada no Anexo desta tese.

5.6 Performance do ARS 6P Wp40 Lp8

Os gráficos da figura 46 apresentados de forma comparativa o erro de estimativa de taxa de dose por distância viajada pelo robô, do erro de estimativa de taxa de dose pelo tempo gasto durante o deslocamento e pelo erro de estimativa de taxa de dose por cada avaliação realizada.

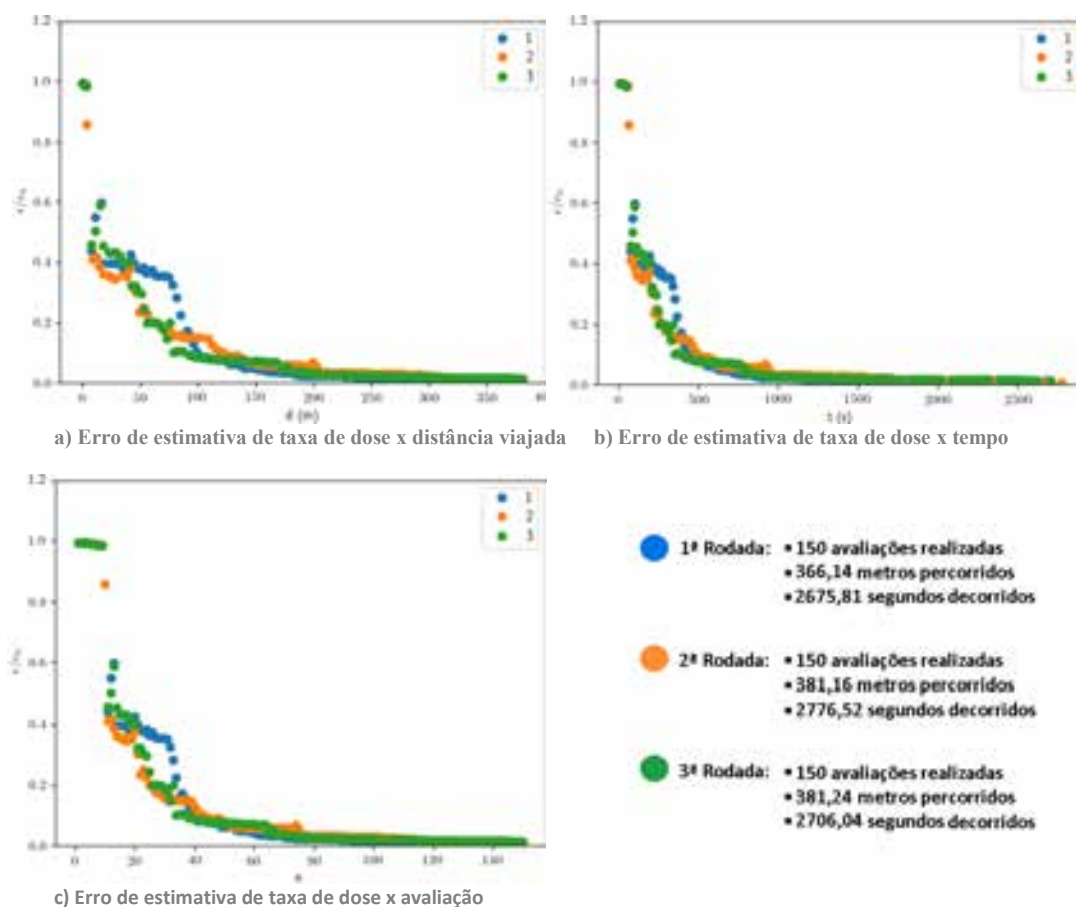


Figura 46. Comparativo do erro estimado de taxa de dose entre as rodadas do ARS 6P Wp40 Lp8

Fonte: Autor

A figura 47 apresenta de forma gráfica a reprodução do tempo decorrido através das 150 avaliações realizadas, exibindo as curvas correspondentes ao Tempo total da simulação, o Tempo de viagem do robô, Tempo de cálculo de rota e o Tempo de medida, bem como o delineamento do trajeto realizado pelo robô no decorrer da navegação.

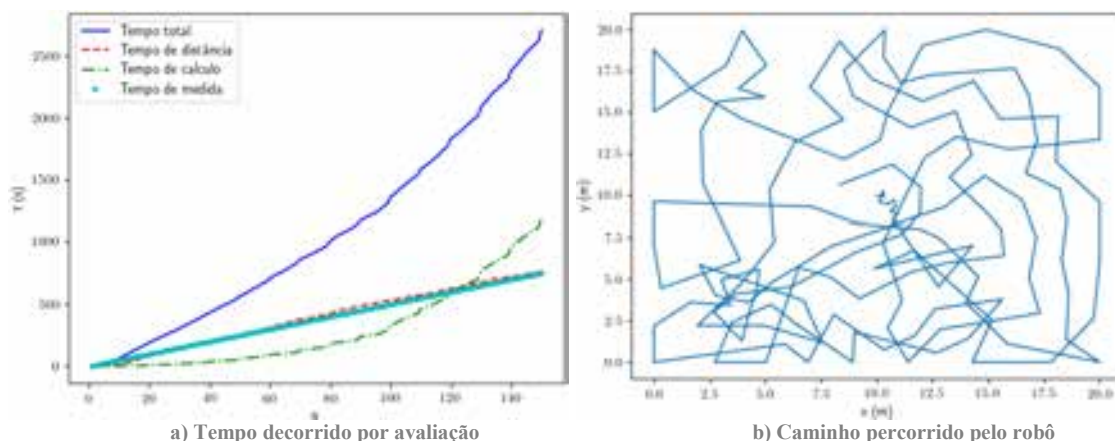


Figura 47. Comparativo de tempos e delineamento do trajeto do robô com o ARS 6P Wp40 Lp8
Fonte: Autor

A análise da evolução do algoritmo demonstra que após a septuagésima quinta avaliação de cada rodada o algoritmo consegue reconstruir o campo de radiação real com um erro estimado de taxa de dose consideravelmente baixo, conforme a evolução demonstrada nas figuras 48, 49 e 50.

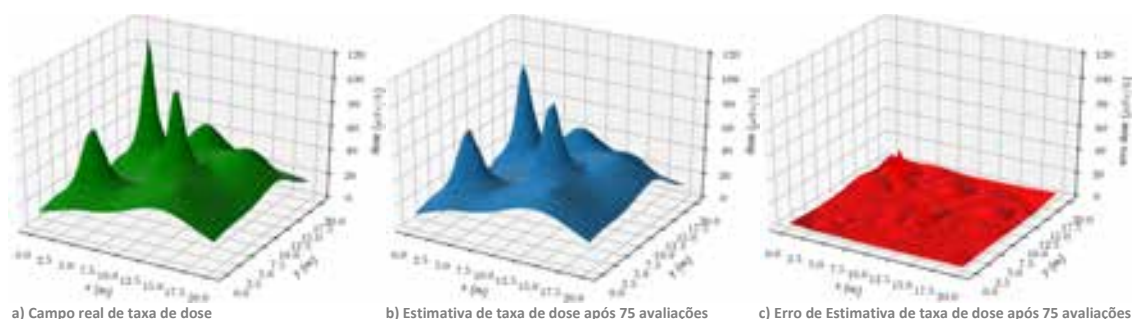


Figura 48. Resultado da 1ª rodada do ARS 6P Wp40 Lp8 após 75 avaliações
Fonte: Autor

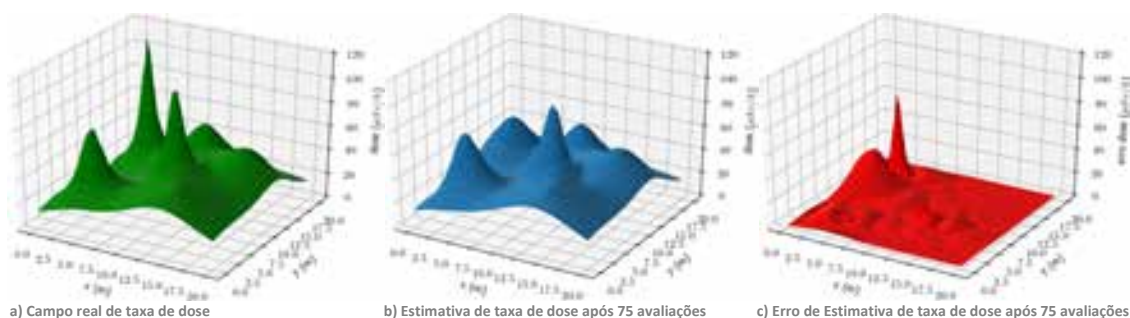


Figura 49. Resultado da 2ª rodada do ARS 6P Wp40 Lp8 após 75 avaliações

Fonte: Autor

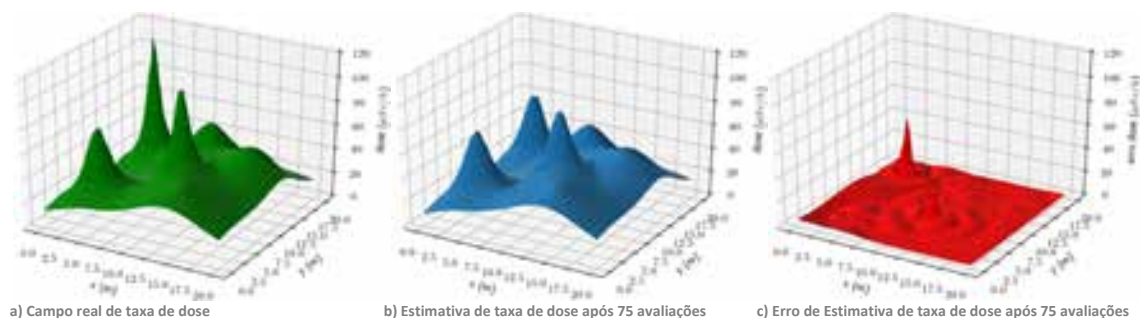


Figura 50. Resultado da 3ª rodada do ARS 6P Wp40 Lp8 após 75 avaliações

Fonte: Autor

Os resultados das performances da 1ª, 2ª e 3ª rodadas do ARS 6P Wp40 Lp8 alcançaram um erro médio de 3,78 conforme os valores apresentados na tabela 9:

Tabela 9. Média dos resultados do ARS 6P Wp40 Lp8 após 75 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	75	3,25	1046,7s	257,8m
2ª	75	4,12	825,7s	203,3m
3ª	75	3,98	854,7s	210,5m
Média	75	3,78	900,0s	221,6m

A análise da evolução do algoritmo após a decorrência das 150 avaliações, mostra que o erro de estimativa de taxa de dose reduz em 64,4% em comparação a avaliação 75, enquanto que o tempo decorrido aumenta em 202,1% e há um acréscimo de 69,7% na distância percorrida, conforme os dados apresentados na tabela 10.

Tabela 10. Média dos resultados do ARS 6P Wp40 Lp8 após 150 avaliações
 Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	150	1,16	2675,8s	366,1m
2ª	150	1,15	2776,5s	381,1m
3ª	150	1,20	2706,0s	381,2m
Média	150	1,17	2719,4s	376,1m

A Figura 51 mostra as capturas de tela do número intermediário de pontos visitados (5, 10, 25, 50, 75, 100, 125 e 150) e a melhoria seqüencial obtida com o decorrer da evolução do algoritmo.

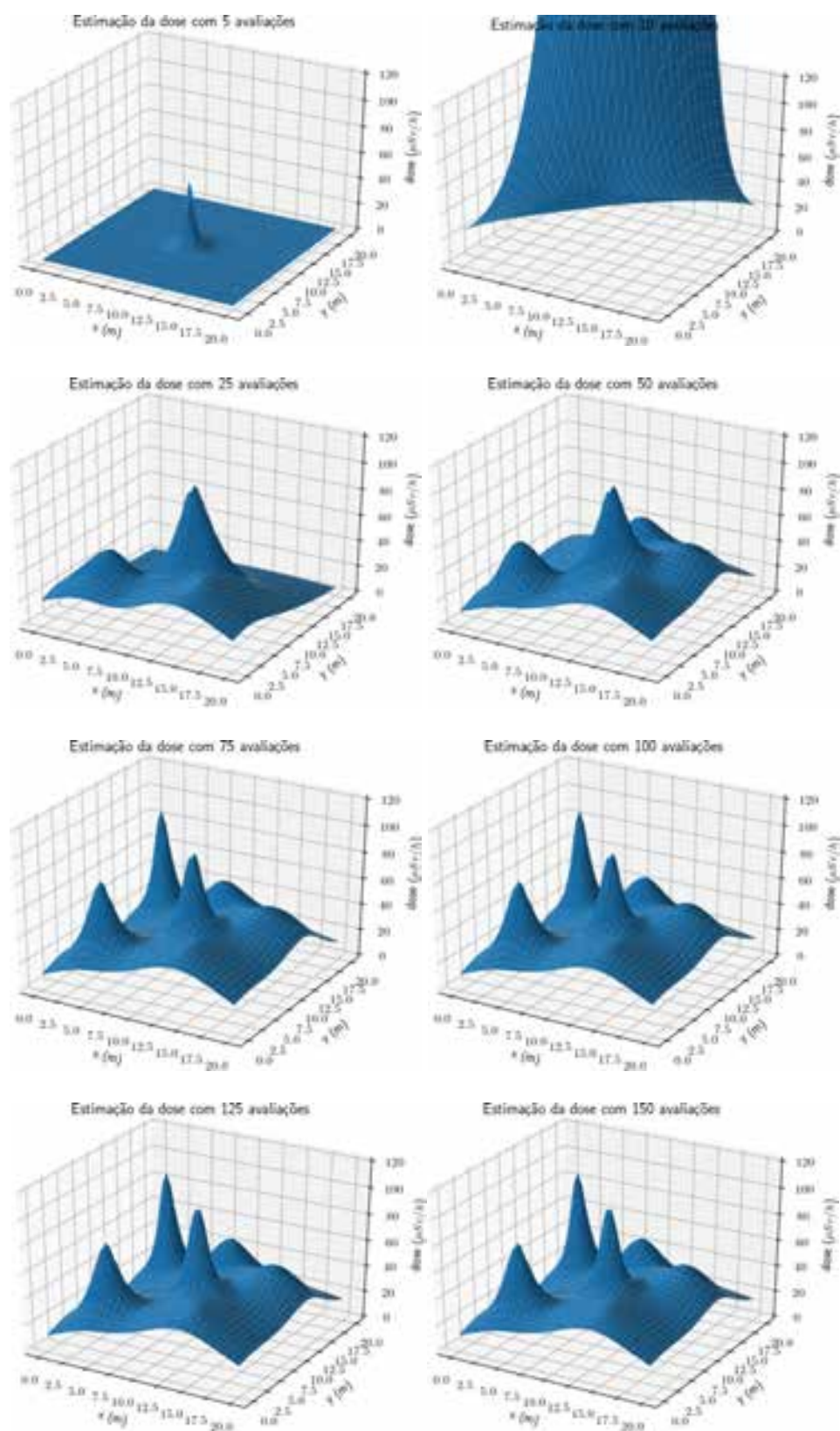


Figura 51. Evolução do ARS 6P Wp40 Lp8

Fonte: Autor

O evolução do desempenho de todas as 150 avaliações de cada uma das 3 rodadas de simulação computacional do ARS 6P Wp40 Lp8, encontra-se melhor detalhada no Anexo desta tese.

5.7 Performance do ARS 8P Wp40 Lp4

Os gráficos da figura 52 apresentados de forma comparativa o erro de estimativa de taxa de dose por distância viajada pelo robô, do erro de estimativa de taxa de dose pelo tempo gasto durante o deslocamento e pelo erro de estimativa de taxa de dose por cada avaliação realizada.

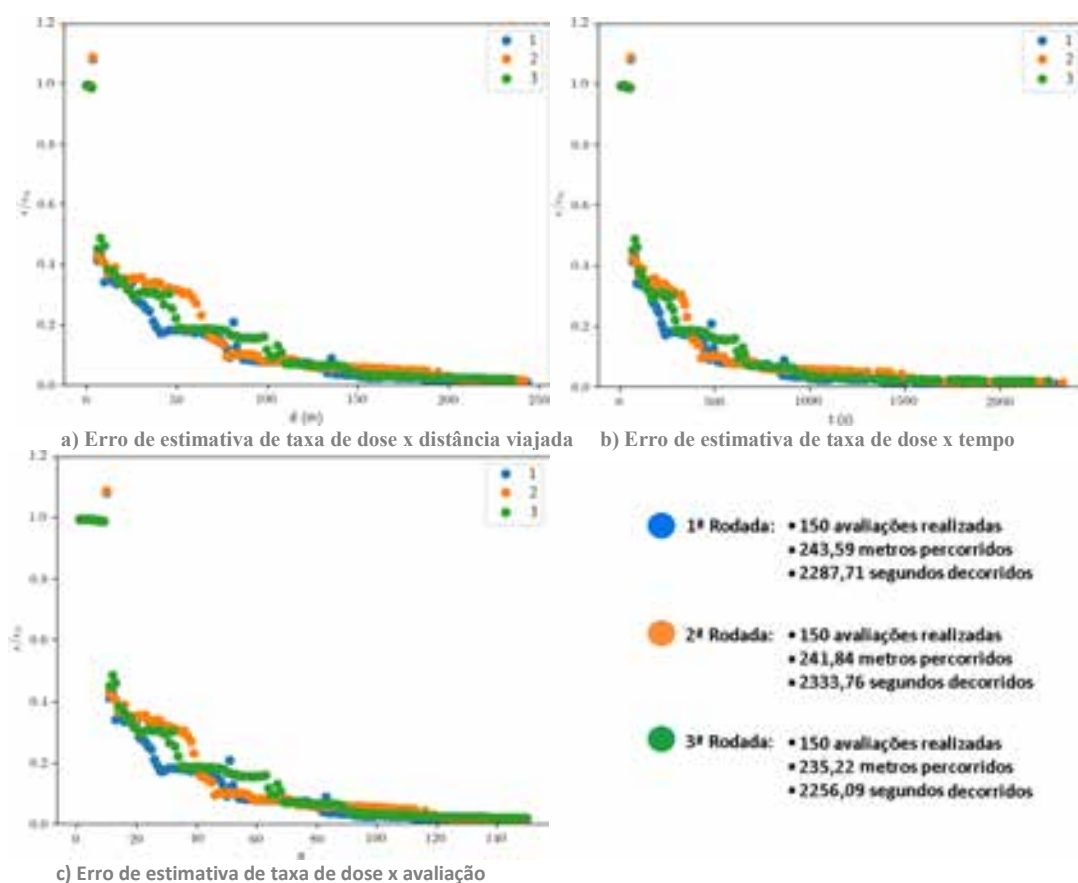


Figura 52. Comparativo do erro estimado de taxa de dose entre as rodadas do ARS 8P Wp40 Lp4

Fonte: Autor

A figura 53 apresenta de forma gráfica a reprodução do tempo decorrido através das 150 avaliações realizadas, exibindo as curvas correspondentes ao Tempo total da simulação, o Tempo de viagem do robô, Tempo de cálculo de rota e o Tempo de medida, bem como o delineamento do trajeto realizado pelo robô no decorrer da navegação.

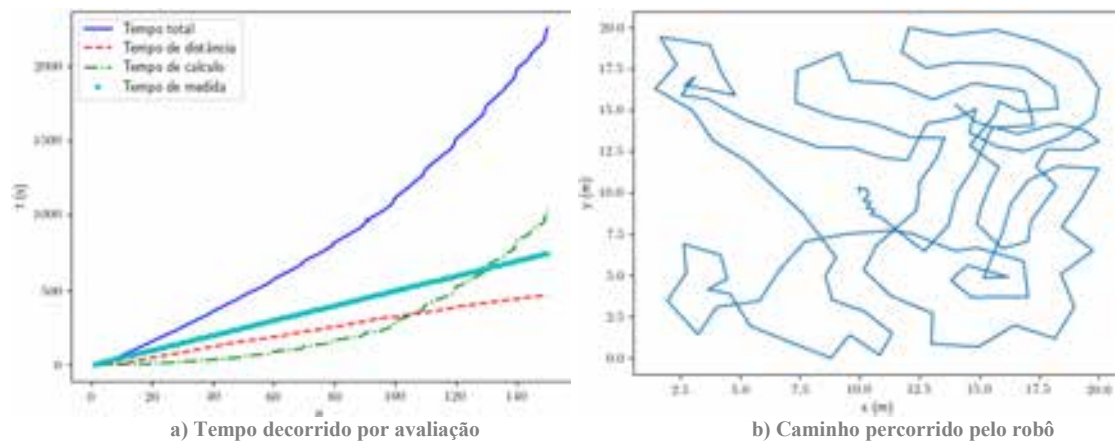


Figura 53. Comparativo de tempos e delineamento do trajeto do robô com o ARS 8P Wp40 Lp4
Fonte: Autor

A análise da evolução do algoritmo demonstra que a partir da centésima vigésima quinta avaliação de cada rodada, o algoritmo consegue reconstruir o campo de radiação real com um erro estimado de taxa de dose consideravelmente baixo, conforme a evolução demonstrada nas figuras 54, 55 e 56.

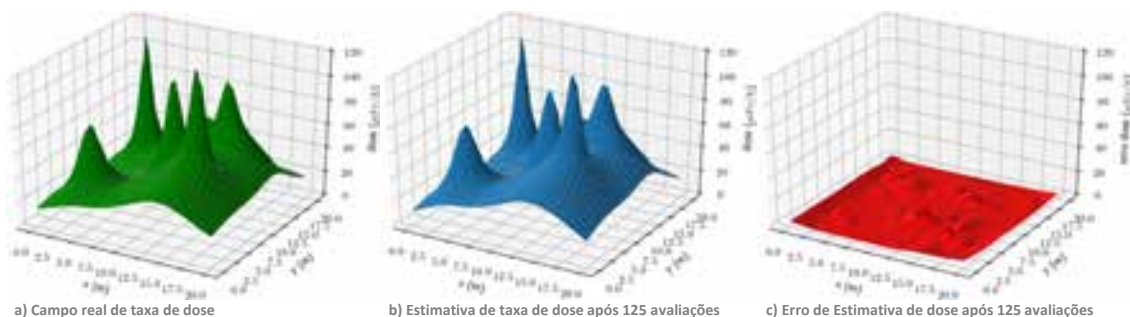


Figura 54. Resultado da 1ª rodada do ARS 8P Wp40 Lp4 após 125 avaliações
Fonte: Autor

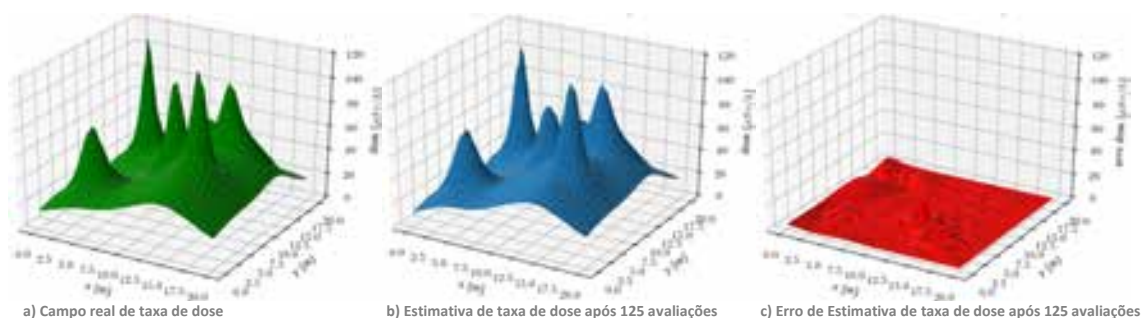


Figura 55. Resultado da 2ª rodada do ARS 8P Wp40 Lp4 após 125 avaliações

Fonte: Autor

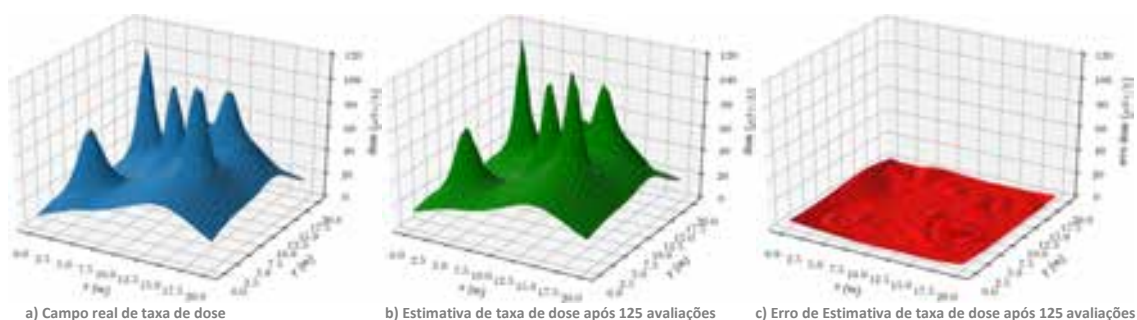


Figura 56. Resultado da 3ª rodada do ARS 8P Wp40 Lp4 após 125 avaliações

Fonte: Autor

Os resultados das performances da 1ª, 2ª e 3ª rodadas do ARS 8P Wp40 Lp4 alcançaram um erro médio de 1,88 conforme os valores apresentados na tabela 11:

Tabela 11. Média dos resultados do ARS 8P Wp40 Lp4 após 125 avaliações

Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	Tempo	Distância
1ª	125	1,84	1736,2s	251,0m
2ª	125	1,90	1686,2s	240,6m
3ª	125	1,89	1695,1s	244,4m
Média	125	1,88	1704,2s	245,7m

A análise da evolução do algoritmo após a decorrência das 150 avaliações, mostra que o erro de estimativa de taxa de dose reduz em 15,9% em comparação a avaliação 125, enquanto que o tempo decorrido aumenta em 34,5% e há uma redução de 2,2% na distância percorrida, conforme os dados apresentados na tabela 12.

Tabela 12. Média dos resultados do ARS 8P Wp40 Lp4 após 150 avaliações
Fonte: Autor

Rodada	Avaliação	$\varepsilon/\varepsilon_0$	t	d
1ª	150	1,55	2287,7s	243,5m
2ª	150	1,58	2333,7s	241,8m
3ª	150	1,60	2256,0s	235,2m
Média	150	1,58	2292,5s	240,2m

A Figura 57 mostra as capturas de tela do número intermediário de pontos visitados (5, 10, 25, 50, 75, 100, 125 e 150) e a melhoria seqüencial obtida com o decorrer da evolução do algoritmo.

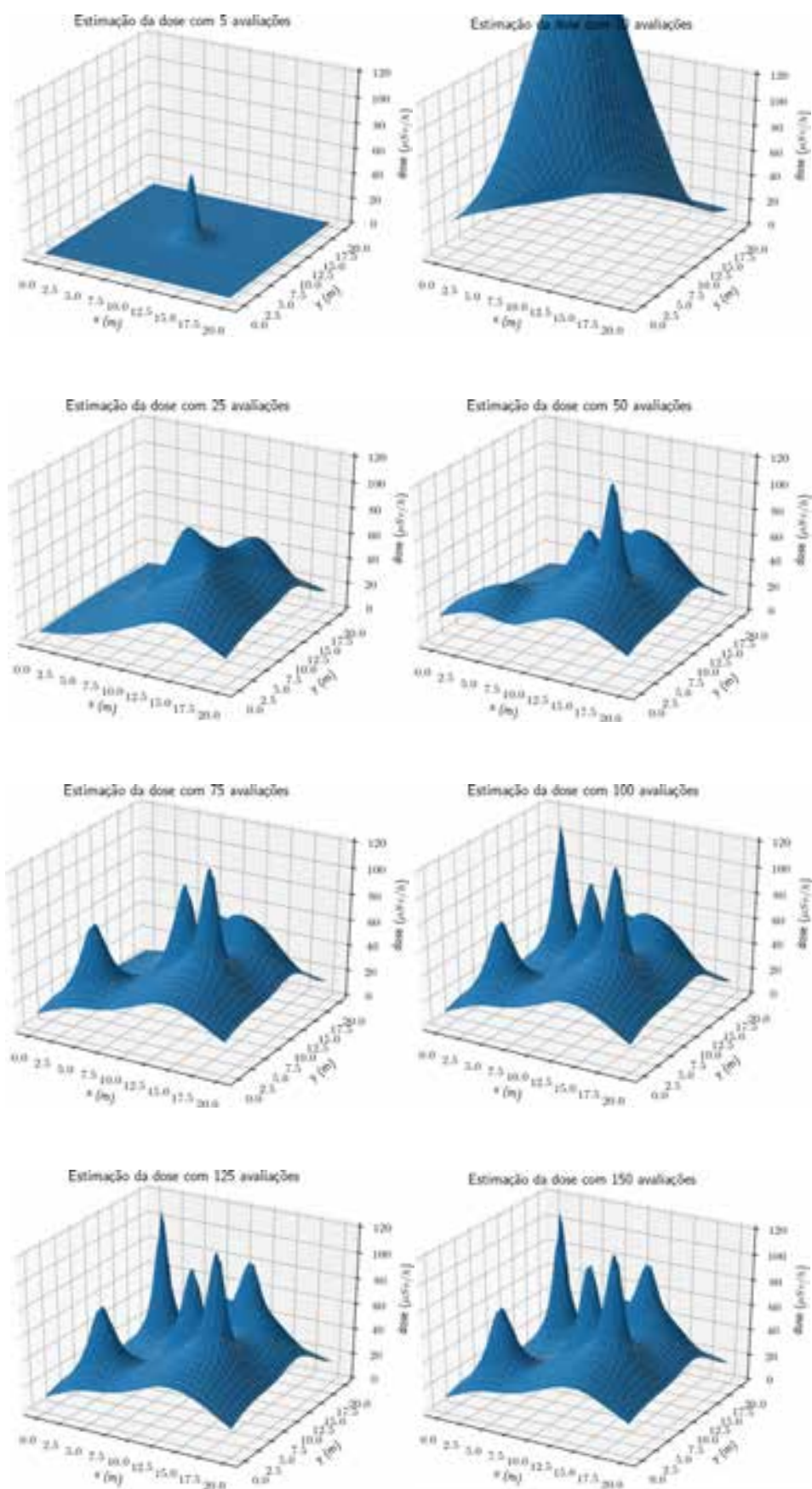


Figura 57. Evolução do ARS 8P Wp40 Lp4

Fonte: Autor

O evolução do desempenho de todas as 150 avaliações de cada uma das 3 rodadas de simulação computacional do ARS 8P Wp40 Lp4, encontra-se melhor detalhada no Anexo desta tese.

5.8 Resultados comparativos

Com o intuito de averiguar a eficiência do método, os resultados apresentados pelas variações do algoritmo em cada cenário hipotético de superfície foram agrupados de forma comparativa, conforme ilustram as tabelas 13, 14 e 15. Cabe ressaltar que todas as variações dos métodos experimentadas alcançaram o objetivo de reconstruir na íntegra a superfície ao qual foram testadas. No entanto, para o fim da aplicação em sistemas robóticos móveis é interessante que o algoritmo realize o levantamento radiométrico de um ambiente em um menor espaço de tempo possível e percorrendo a menor distância possível.

Tabela 13. Comparativo dos métodos experimentados para a topologia de quatro picos
Fonte: Autor

150 pontos avaliados	ARS 4P SP	ARS 4P Wp80 Lp4	ARS 4P Wp80 Lp8	ARS 4P Wp40 Lp4	ARS 4P Wp40 Lp8
1ª Rodada	d = 1268,9 m t = 4683,4 s	d = 481,7 m t = 2839,5 s	d = 525,3 m t = 3042,8 s	d = 303,8 m t = 2389,0 s	d = 315,8 m t = 3793,9 s
2ª Rodada	d = 1272,1 m t = 4624,4 s	d = 489,1 m t = 2855,9 s	d = 534,1 m t = 3026,7 s	d = 228,0 m t = 3121,8 s	d = 321,9 m t = 3497,5 s
3ª Rodada	d = 1254,1 m t = 4531,7 s	d = 516,3 m t = 2732,9 s	d = 519,1 m t = 3074,4 s	d = 252,9 m t = 3904,1 s	d = 350,6 m t = 2579,2 s
Média final	d = 1265,0 m t = 4613,2 s	d = 495,7 m t = 2809,4 s	d = 526,2 m t = 3047,9 s	d = 261,6m t = 3138,3s	d = 329,4 m t = 3290,2 s
$\varepsilon/\varepsilon_0$	0,23	0,28	0,24	0,96	0,46

As tabelas 13, 14 e 15 apresentam de forma detalhada o tempo e a distância percorrida em cada uma das rodadas de simulação, para cada um dos métodos experimentados, bem com o erro médio verificado após a realização das 150 avaliações.

Tabela 14. Comparativo dos métodos experimentados para a topologia de seis picos

Fonte: Autor

150 pontos avaliados	ARS 6P SP	ARS 6P Wp80 Lp4	ARS 6P Wp80 Lp8	ARS 6P Wp40 Lp4	ARS 6P Wp40 Lp8
1ª Rodada	d = 1456,0 m t = 5005,7 s	d = 551,9 m t = 3151,0 s	d = 575,3 m t = 3144,3 s	d = 272,8 m t = 2450,3 s	d = 366,1 m t = 2675,8 s
2ª Rodada	d = 1424,7 m t = 5019,2 s	d = 582,3 m t = 3108,4 s	d = 601,5 m t = 3168,9 s	d = 251,5 m t = 2266,2 s	d = 381,1 m t = 2776,5 s
3ª Rodada	d = 1542,4 m t = 5199,2 s	d = 538,4 m t = 3078,8 s	d = 631,0 m t = 3234,7 s	d = 265,7 m t = 2438,6 s	d = 381,2 m t = 2706,0 s
Média final	d = 1474,4 m t = 5074,7 s	d = 557,5 m t = 3112,8 s	d = 602,6 m t = 3182,6 s	d = 263,3 m t = 2385,0 s	d = 376,1 m t = 2719,4 s
$\varepsilon/\varepsilon_0$	0,62	0,91	0,88	1,84	1,17

Tabela 15. Comparativo dos métodos experimentados para a topologia de oito picos

Fonte: Autor

150 pontos avaliados	ARS 8P SP	ARS 8P Wp80 Lp4	ARS 8P Wp80 Lp8	ARS 8P Wp40 Lp4	ARS 8P Wp40 Lp8
1ª Rodada	d = 1408,8 m t = 4875,3 s	d = 493,3 m t = 2810,6 s	d = 570,7 m t = 3135,7 s	d = 243,5 m t = 2287,7 s	d = 363,6 m t = 2612,5 s
2ª Rodada	d = 1417,7 m t = 4832,2 s	d = 508,8 m t = 2931,8 s	d = 573,0 m t = 3095,2 s	d = 241,8 m t = 2333,7 s	d = 357,9 m t = 2532,5 s
3ª Rodada	d = 1446,9 m t = 4953,0 s	d = 539,5 m t = 2742,3 s	d = 613,6 m t = 3184,4 s	d = 235,2 m t = 2256,0 s	d = 358,7 m t = 2592,1 s
Média final	d = 1424,5 m t = 4886,8s	d = 513,90 m t = 2828,26 s	d = 585,8 m t = 3138,4 s	d = 240,2 m t = 2292,5 s	d = 360,1 m t = 2579,1 s
$\varepsilon/\varepsilon_0$	0,89	1,59	1,15	1,58	1,34

Após a comparação do desempenho dos diversos métodos, pode-se perceber que por diversas vezes os métodos já apresentavam, em um número de avaliações inferiores ao critério máximo estabelecido (150 avaliações), uma condição de desempenho

satisfatória, alcançando um grau de reconstrução de superfície muito próxima ao cenário real apresentado. Fazendo com que a melhoria obtida nas evoluções subsequentes seja considerada irrelevante em comparação ao custo benefício, em virtude do acréscimo de tempo e do trajeto percorrido pelo robô. As tabelas 16, 17 e 18 apresentam os valores de tempo decorrido, distância percorrida e o erro apresentado no momento em que cada método alcançou uma qualidade satisfatória de reconstrução do cenário hipotético experimentado.

Tabela 16. Apresentação do melhor desempenho para a topologia de quatro picos

Fonte: Autor

RCB *	ARS 4P SP	ARS 4P Wp80 Lp4	ARS 4P Wp80 Lp8	ARS 4P Wp40 Lp4	ARS 4P Wp40 Lp8
Tempo	1342,2 s	518,0 s	787,9 s	879,5 s	1169, s
Distância	501,7 m	136,7 m	450,6 m	484,0m	181,7 m
Avaliação	75	75	75	75	75
$\varepsilon/\varepsilon_0$	2,91	5,42	4,98	3,52	1,94

*Relação Custo Benefício - Relação do menor tempo de execução e do menor caminho percorrido pelo robô com a capacidade de reconstrução da superfície.

Ao analisar os dados dispostos na tabela 16, pode-se concluir que o menor desempenho na reconstrução da topologia de quatro picos é atribuído ao ARS 4P Wp80 Lp4 o qual alcançou um tempo de 518,0 s em um percurso de 136,7 m.

O melhor desempenho do algoritmo na reconstrução da topologia de seis picos pode ser observado nos dados apresentados pelo ARS 6P Wp40 Lp8, conforme mostrado na tabela 16, onde o algoritmo alcança um tempo de 900,00 s em um percurso de 221,69 m.

Tabela 17. Apresentação do melhor desempenho para a topologia de seis picos

Fonte: Autor

RCB [*]	ARS 6P SP	ARS 6P Wp80 Lp4	ARS 6P Wp80 Lp8	ARS 6P Wp40 Lp4	ARS 6P Wp40 Lp8
Tempo	2803,8 s	1767,0 s	5427,2 s	1130,37 s	900,00 s
Distância	1009,6 m	323,1 m	1306,8 m	267,66 m	221,69 m
Avaliação	100	100	150	150	75
$\varepsilon/\varepsilon_0$	2,08	1,64	0,88	1,84	3,78

*Relação Custo Benefício - Relação do menor tempo de execução e do menor caminho percorrido pelo robô com a capacidade de reconstrução da superfície.

Na tabela 18 observa-se o destaque no desempenho do ARS 8P Wp40 Lp4 o qual performou, na centésima vigésima quinta avaliação, um tempo de 1704,2 s ao percorrer uma distância de 245,7 m.

Tabela 18. Apresentação do melhor desempenho para a topologia de oito picos

Fonte: Autor

RCB [*]	ARS 8P SP	ARS 8P Wp80 Lp4	ARS 8P Wp80 Lp8	ARS 8P Wp40 Lp4	ARS 8P Wp40 Lp8
Tempo	7725,8 s	2267,3 s	1654,0 s	1704,2 s	2015,5 s
Distância	2797,7 m	613,5 m	393,8 m	245,7 m	357,4 m
Avaliação	150	125	100	125	125
$\varepsilon/\varepsilon_0$	0,89	1,99	2,59	1,88	1,67

*Relação Custo Benefício - Relação do menor tempo de execução e do menor caminho percorrido pelo robô com a capacidade de reconstrução da superfície.

Com base nos dados apresentados, pode-se concluir que nos diversos casos tratados os valores de desempenho apresentados encontram-se satisfatórios para a implementação em um sistema robótico real.

Capítulo 6

PROTOTIPAGEM E RESULTADOS

Este capítulo aborda a construção do protótipo do robô autônomo e suas peculiaridades na implementação do algoritmo de reconstrução de superfície, apresentando de forma geral a montagem do hardware, do software e do firmware de controle.

6.1 Visão geral do projeto

A construção do protótipo seguiu rigorosamente os parâmetros estabelecidos durante o desenvolvimento deste trabalho, no intuito de viabilizar em uma plataforma real o emprego do algoritmo desenvolvido, adequando variáveis de simulação computacional a um sistema robótico dedicado.

No intuito de facilitar o entendimento do projeto, a confecção do protótipo foi dividida em três etapas básicas:

- Estrutura mecânica;
- Eletrônica embarcada;
- Programação operacional.

6.2 Estrutura mecânica

A Construção da estrutura mecânica contou primeiramente com a confecção do conjunto de esteriras (lagartas) e seu suporte de fixação todo elaborado em acrílico transparente.

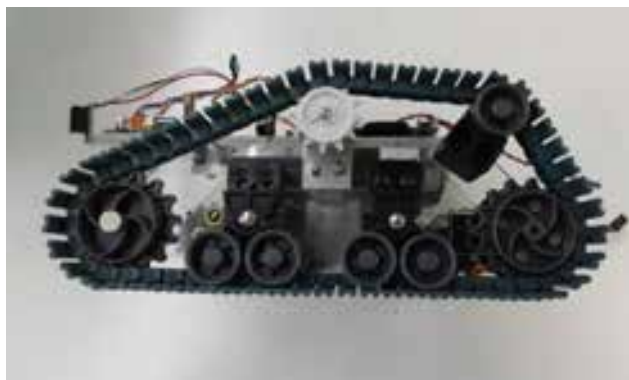


Figura 58. Estrutura do sistema de tracionamento

Fonte: Autor

O sistema de tracionamento do robô foi desenvolvido de modo a criar uma liberdade de movimentos compatível com a locomoção solicitada pelo ARS proposto. Para tal, faz-se necessário dois conjuntos simétricos de tração, sendo individualmente compostos de dois conjuntos de rodas de apoio, um rodete de apoio e duas engrenagens sincronizadas. Conforme apresentado na figura 58.

Cada conjunto de tração faz uso de um servo motor adaptado para trabalhar em regime de giro permanente e de um encoder incremental AB com resolução de 12 pulsos por revolução (PPR) responsável por medir o deslocamento linear de cada conjunto de tração. Nas figuras 59 e 60 são apresentados estes componentes.



Figura 59. Servomotor em visão explodida

Fonte: <http://amsfrancisco.planetaclix.pt>



Figura 60. Encoder incremental AB

Fonte: Pololu Robotics and Electronics

Foi desenvolvida uma plataforma para a acomodação das placa eletrônicas de controle e do conjunto de baterias de lítio polímero responsável pela alimentação elétrica do robô.

6.3 Eletrônica embarcada

O protótipo utiliza diversos tipos de hardwares e firmwares, tanto para controle quanto para processamento. O processamento principal é realizado através de uma placa Raspberry PI B, que é baseada em um *system on a chip* (SoC) que possui um circuito Integrado (CI) Broadcom BCM2835 capaz de fornecer um barramento *General Purpose*

Input/Output (GPIO) de dados. É dotada de um processador ARM1176JZF-S de 700 MHz, GPU VideoCore IV e 512 MB de memória RAM. Na figura 61 é apresentado o modelo da placa e as funções de seus conectores.

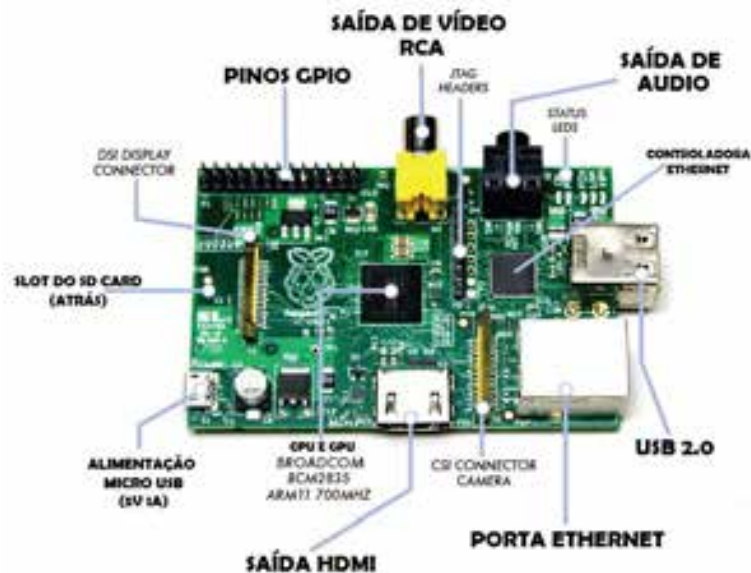


Figura 61. Raspberry PI B
Fonte: <https://www.raspberrypi.org>

Para a aquisição de sensoriamento e adequação de dados de controle, foi desenvolvido no LARO do LMP/PEN/UFRJ um circuito eletrônico capaz de atuar como uma interface de dados convertendo os sinais elétricos dos sensores embarcados em sinais compatíveis com os do barramento *GPIO* do Raspberry, conforme ilustra a figura 62.

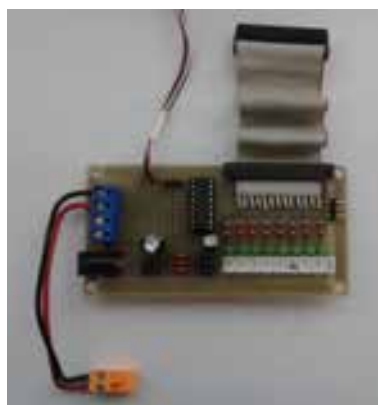


Figura 62. Placa de interface de sensoriamento
Fonte: Autor

A placa de interface de sensoriamento (figura 62) é responsável pela conversão de DC/DC de oito canais de controle digitais, conversão A/D de dez canais analógicos, controle e regulação da carga das baterias e indicação de acionamento de portas de controle.

Para o acionamento do conjunto de tração, foi desenvolvido um circuito eletrônico foto-acoplado que atua como *Driver* de controle dos motores, que possui a função de elevar os níveis de corrente e tensão produzidos pelas saídas digitais do Raspberry, conforme ilustrado na figura 63.



Figura 63. Placa de *driver* de motores

Fonte: Autor

O robô conta com a opção de utilizar dez sensores analógicos, no entanto para o desenvolvimento deste trabalho foram utilizados somente os sensores de Chama, IR e Gas Hidrogênio, os quais se encontram localizados na parte superior do robô em uma estrutura denominada “*POD*”.

Sensores de proximidade foram instalados na frente da estrutura do robô no intuito de detectar possíveis obstáculos. Ao todo são três sensores retroreflexivos com detecção por infravermelho, um localizado na posição central, uma na lateral direita e

outro na lateral esquerda, de modo realizar uma varredura de 90° de cobertura durante seu deslocamento. A figura 64 apresenta o sensor.



Figura 64. Sensor de proximidade
Fonte: Pololu Robotics and Electronics

O sensor de radiação utilizado nesta aplicação é um Contador Geiger Muller desenvolvido pelo IEN/CNEN e projetado de forma dedicada para este tipo de aplicação robótica de detecção móvel, conforme mostra a figura 65.

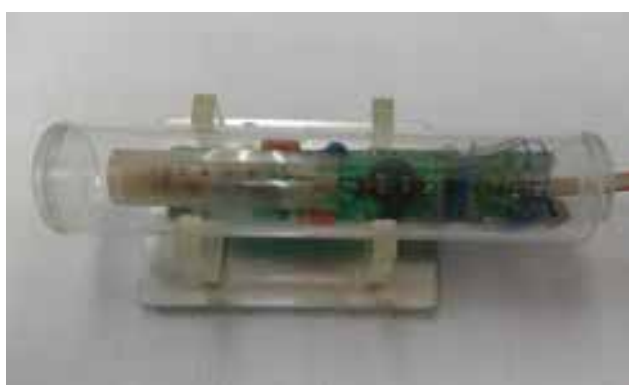


Figura 65. Sensor de radiação
Fonte: Autor

Este sensor detecta uma intensidade de radiação através da geração de pulsos elétrico que são interpretados e convertidos em sinais de controle inteligíveis, e em seguida adequados para serem transmitidos através de um protocolo de comunicação serial *modbus*. Este sinal é transferido para um circuito conversor DC/DC (figura 66) o qual realiza a adequação da intensidade de sinal do sensor de radiação para os níveis de sinais de controle da porta serial (Tx e Rx) do Raspberry.

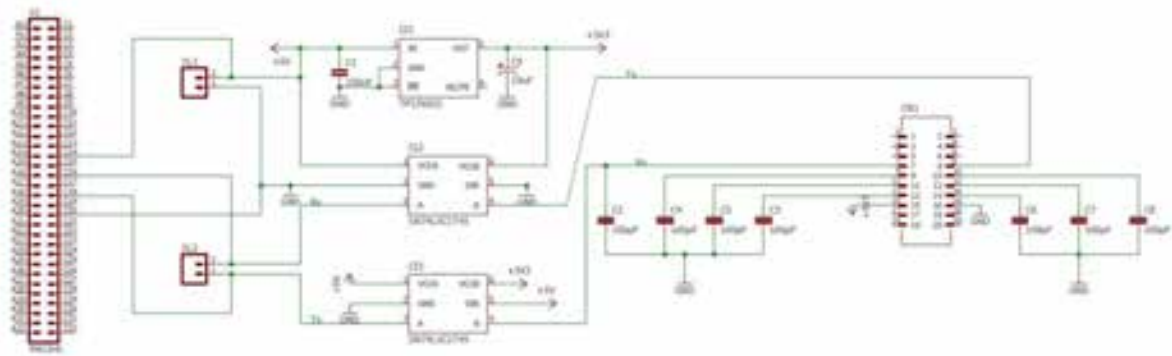


Figura 66. Diagrama elétrico do circuito adequador de sinais para o sensor de radiação

Fonte: Autor

O sistema do robô ainda conta com uma placa reguladora de tensão, responsável por manter a alimentação do sistema constantemente em 5vcc, conforme apresentada na figura 67.



Figura 67. Placa reguladora de tensão

Fonte: Pololu Robotics and Electronics

A tabela 19 apresenta as especificações técnicas de todos os sensores utilizados neste projeto, tratada de forma resumida os dados dos *datasheets* de cada fabricante.

Tabela 19. Dados técnicos dos sensores

Fonte: Autor

Sensor de Temperatura	
Tipo:	Termômetro infravermelho
Alcance:	-32 a 350°C
Precisão:	±1,5°C
Resolução:	0,1°C
Tempo de resposta:	500 mseg
Sensor de Chama	
Tipo:	Fototransistor infravermelho de silício plástico
Alcance:	760 nm to 1100 nm
Ângulo de detecção:	60 graus
Dissipação linear de energia (25 ° C):	1.33 mW/°C
Comprimento de onda de sensibilidade máxima:	880 nm
Temperatura de operação:	-40 to +100 °C
Dissipação de energia:	100 mW
Sensor de luminosidade	
Tipo:	LDR - Resistor dependente da luz
Resistência à luz em 10Lux (25 ° C):	8~20KΩ
Resistência escura a 0 Lux:	1.0MΩ(min)
Valor de gama em 100-10 Lux:	0.7
Dissipação de energia (25 ° C):	100mW
Tensão máxima (25 ° C):	150V
Pico de resposta espectral (25 ° C):	540nm
Faixa de temperatura ambiente:	-30~+70°C
Sensor gas	
Tipo:	Sensor de semiconductor
Gás de detecção:	CO e gás combustível
Concentração:	10-1000ppm CO / 100-10000 ppm gás combustível
Sensibilidade:	Rs(in air)/Rs(100ppm CO)≥5
Declive:	≤0.6(R300ppm/R100ppm CO)
Tem. Umidade:	20°C±2°C; 65%±5%RH
Sensor de proximidade	
Tipo:	Retroreflexivo
Tensão min / máx:	2.7V - 6.2V
Alcance:	2-10cm
Saída:	Digital
Tamanho:	21.6 mm × 10.4 mm × 8.9 mm
Peso:	1.5 g
Sensor de rotação	
Tipo:	Encoder incremental
Tensão operacional:	4.5V - 5.5V
Saídas digitais:	2 canais (quadratura)
Consumo atual a 5,0V	14 mA
Resolução:	48 contagem por revolução
Peso:	1.6 g
Sensor de radiação	
Tipo:	Contador Geiger Muller
Tensão de operação (discriminador / contador)	5.0V - 15V
Tensão de operação (fotomultiplicador)	1000V
Saídas digitais:	onda quadrada
Consumo atual a 5,0V	250mA
Faixa de operação:	μSv/hr - 0.01 to 1000 CPS - 0 to 5000
Peso:	110g

6.4 Programação operacional

A implementação do ARS em um sistema robótico real, diferentemente de uma simulação computacional, requer o desenvolvimento de certas ferramentas de controle capazes de controlar o hardware desenvolvido para o funcionamento do robô.

Foram desenvolvidas bibliotecas com funções básicas de controle e posicionamento de robô, leitura de sensores de proximidade, IR, radiação, temperatura e gases. No intuito de adequar a leitura de dados do ARS e também propiciar um ambiente amigável para um futuro desenvolvimento de novas aplicações, tornando este projeto uma *facility* para pesquisadores interessados em desenvolver aplicações robóticas para a área nuclear. A tabela 20 apresenta um descritivo das funções desenvolvidas.

Tabela 20. Biblioteca de controle do robô

Fonte: Autor

Biblioteca de controle do robô	
move_robot.py	Função que possui a finalidade de mover os motores do robô.
read_temperature.py	Função que possui a finalidade ler o valor do sensor de temperatura.
read_ir.py	Função que possui a finalidade ler o valor do sensor de infravermelho.
read_encoder.py	Função que possui a finalidade ler o valor do encoder.
read_radiation.py	Função que possui a finalidade ler o valor do sensor de radiação.
read_proximity.py	Função que possui a finalidade ler os valores dos sensores de proximidade.
rx_video.py	Função que possui a finalidade de receber a imagem do vídeo.

O algoritmo de reconstrução de superfície ARS utilizado durante as simulações computacionais, recebeu algumas alterações para ser capaz de acessar as informações provenientes da biblioteca de controle robô. Outra modificação necessária foi a inclusão de um algoritmo de navegação com o objetivo de proporcionar uma condição de reação

eficiente para o desvio de obstáculos. Este sistema tem seu funcionamento firmado na premissa dos sistemas especialistas, onde o processamento de soluções viáveis para o problema da transposição de obstáculos detectáveis é realizado através da aplicação de um motor de inferência sobre uma base de conhecimento. A inicialização do algoritmo ocorre com a atribuição de parâmetros computacionais às variáveis aquisitadas através dos sensores de obstáculo, como mostrado na tabela 21.

Tabela 21. Atribuição de parâmetros computacionais

Fonte: Autor

Atribuição de parâmetros computacionais	
SEAT	Sensor Esquerdo Atuado
SDAT	Sensor Direito Atuado
SFAT	Sensor Frontal Atuado
SENA	Sensor Esquerdo Não Atuado
SDNA	Sensor Direito Não Atuado
SFNA	Sensor Frontal Não Atuado
MFCE	Movimento Frontal Circular a Esquerda
MFCD	Movimento Frontal Circular a Direita
MFDR	Movimento Frontal com Deslocamento Reto
MTDR	Movimento para Trás com Deslocamento Reto

Para viabilizar o desenvolvimento deste algoritmo foi gerado um conjunto de declarações dependentes do domínio do problema, conforme apresentado na tabela 22. Estas declarações formam a base de conhecimento necessária para oferta das possíveis soluções para o problema. Todo o conhecimento prévio sobre o problema a ser resolvido encontra-se representado nas definições destas declarações.

Tabela 22. Declarações do domínio do problema

Fonte: Autor

Declarações do domínio do problema	
Se SEAT e SFAT e SDNA	→ MFCD
Se SENA e SFAT e SDAT	→ MFCE
Se SEAT e SFNA e SDAT	→ MFDR
Se SENA e SFAT e SDNA	→ MFCE ou MFCD
Se SEAT e SFAT e SDAT	→ MTDR

Durante a execução do algoritmo, a opção de selecionar a quarta declaração por exemplo: Se SENA e SFAT e SDNA → MFCE ou MFCD ou seja, se os sensores esquerdo e direito não atuados e o sensor frontal atuado resulta na escolha entre movimento frontal circular à esquerda ou à direita. Para tal, o algoritmo fará uso de uma rotina de seleção aleatória com igualdade de probabilidade.

Outro recurso utilizado durante a execução do algoritmo é a não permissão de seleção da quinta declaração Se SEAT e SFAT e SDAT → MTDR, uma vez que ela tenha sido utilizada por duas vezes consecutivas. Isto evita que o robô fique preso a escolha dos movimentos de ir para frente e para trás sem sair do lugar.

A inserção dos parâmetros iniciais no motor de inferência desencadeará uma combinação de procedimentos de “raciocínio” progressivo, aplicado à base de conhecimento, que resultará no encontro de uma solução ótima. A condição de parada do algoritmo, esta relacionada diretamente a um número predeterminado de ciclos de execução da rotina. A figura 68 apresenta um pseudocódigo que expõem em detalhes as etapas de execução do algoritmo proposto.

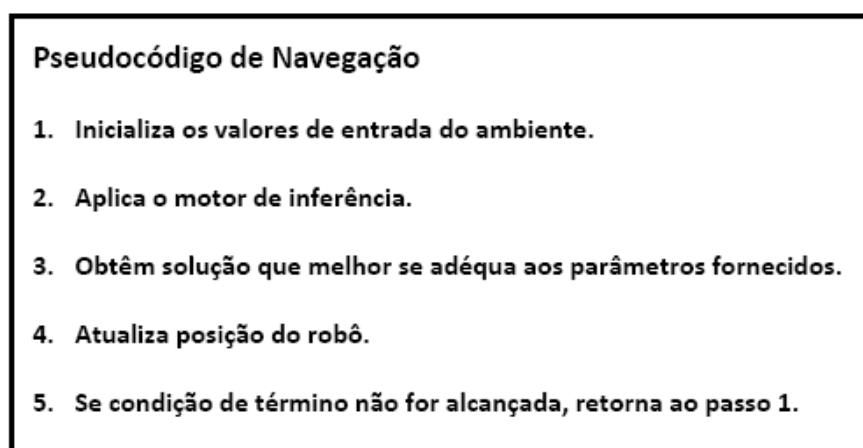


Figura 68. Pseudocódigo do algoritmo de navegação

Fonte: Autor

CONCLUSÃO

O presente trabalho é decorrente da necessidade crescente de monitoração das usinas nucleares, em virtude principalmente das exigências de segurança envolvidas nas situações de detecções e inspeções radiológicas rotineiras ou emergenciais, sejam elas durante operação, manutenção ou evacuação, bem como, da carência na área nuclear de modelos robóticos que auxiliem na tomada de decisão e no planejamento estratégico da empresa.

Neste trabalho foi desenvolvido um veículo robótico autônomo com capacidade de realizar levantamentos dosimétricos em ambientes radioativo, com um sistema baseado em processamento ARM, cujo funcionamento geral e a aplicabilidade do projeto foram detalhados no Capítulo 3. O Capítulo 4 apresenta elaboração do algoritmo computacional desenvolvido com a finalidade de reconstruir uma superfície de radiação, bem como detalha as metodologias utilizadas, o descritivo operacional e a modelagem do sistema. O Capítulo 5 comprova a eficiência do projeto através de simulações computacionais realizadas no intuito de avaliar o desempenho e a robustez do sistema desenvolvido. O Capítulo 6 apresenta a prototipagem do robô.

Os resultados obtidos através das simulações computacionais dos métodos testados mostraram que nos ARS conseguiram reconstruir as superfícies simuladas com poucas medições e com um erro bem reduzido. Todos os métodos foram testados com uma condição de parada baseado na leitura de 150 pontos de medição, no entanto os ARS 4P SP, ARS 4P Wp80 Lp4, ARS 4P Wp80 Lp8, ARS 4P Wp40 Lp4 e ARS 4P Wp40 Lp8 alcançaram a reconstrução da topologia de quatro picos de máximo em 75 pontos medidos e com um erro de 2,91; 5,42; 4,98; 3,52 e 1,94 respectivamente.

Os ARS 6P SP, ARS 6P Wp80 Lp4, ARS 6P Wp80 Lp8, ARS 6P Wp40 Lp4 e ARS 6P Wp40 Lp8 alcançaram a reconstrução da topologia de seis picos de máximo após 100, 100, 150, 150 e 75 pontos medidos respectivamente e obtendo um erro de 2,08; 1,64; 0,88; 1,84 e 3,78 respectivamente.

Os ARS 8P SP, ARS 8P Wp80 Lp4, ARS 8P Wp80 Lp8, ARS 8P Wp40 Lp4 e ARS 8P Wp40 Lp8 alcançaram a reconstrução da topologia de oito picos de máximo após 150, 125, 100, 125 e 125 pontos medidos respectivamente e obtendo um erro de 0,89; 1,99; 2,59; 1,88 e 1,67 respectivamente.

Pode-se perceber que nos diversos casos tratados os valores de desempenho apresentados pelos ARS que utilizam o método de penalização abordado nesta tese apresentaram resultados muito superiores aos dos ARS sem penalização por distância percorrida. No ARS 4P SP o robô apresentou uma distância média percorrida de 1265,0 metros em um tempo de 4613,2 segundos em quanto os ARS utilizando a penalização por distância obtiveram uma média geral na distância média percorrida de 403,2 metros em um tempo médio total de 3071,45 segundos.

No ARS 6P SP o robô apresentou uma distância média percorrida de 1474,4 metros em um tempo de 5074,7 segundos em quanto os ARS utilizando a penalização por distância obtiveram uma média geral na distância média percorrida de 449,8 metros em um tempo médio total de 2849,9 segundos.

No ARS 8P SP o robô apresentou uma distância média percorrida de 1424,5 metros em um tempo de 4886,8 segundos em quanto os ARS utilizando a penalização por distância obtiveram uma média geral na distância média percorrida de 425,0 metros em um tempo médio total de 2709,6 segundos.

Os resultados satisfatórios alcançados pelos testes auferidos devem-se em grande parte ao modelo de *fitness* desenvolvido nesta tese, o qual faz uso da entropia e da distribuição log normal para nortear a busca por regiões com menos informações mapeadas e com probabilidade de possuir valores mais elevados na medição de pontos. Isso somado a penalização de distância proporcionam uma ótima métrica para a escolha de novos pontos.

No que tange ao emprego e a aplicação operacional do sistema, todos os testes e simulações realizadas demonstraram concordância com os objetivos almejados neste projeto, o que permite às seguintes conclusões:

(1) – É viável o desenvolvimento e a utilização de um sistema robótico autônomo, para operações de detecção radiológica e levantamento dosimétrico tanto em ambientes internos como em ambientes externos, durante operações rotineiras ou emergenciais.

(2) – A combinação de utilização de uma regressão por processo Gaussiano (*GP*) com um sistema de aprendizado *Active Learning*, com uma otimização auxiliar baseada em Evolução Diferencial (ED) proporciona um fator de importância neste tipo de controle autônomo, pois permite uma otimização de tempo x deslocamento mitigando as limitações do emprego de robótica em tarefas de longa duração.

Como melhoria para este trabalho, sugere-se:

- Desenvolver um critério de parada do algoritmo que não se baseie em números de medições de pontos;

- Estabelecer critérios de monitoramento da radiação absorvida pelos circuitos eletrônicos e mecânicos do sistema robótico, de forma a estimar a vida útil de seus componentes, proporcionar as melhorias necessárias nos circuitos empregados e a otimização de seu emprego funcional. A elaboração destes critérios é um fator de grande importância para futuros desenvolvimentos de aplicativos e acessórios para este sistema robótico, bem como para o desenvolvimento de uma nova geração de robôs;

- Implementar um sistema de posicionamento baseado em triangulação de sinal, com o intuito de aumentar a precisão do posicionamento do robô em ambientes fechados sem acesso a sinal GPS. Realizado através de transmissores moveis lançados individualmente pelo próprio robô durante seu deslocamento pelo terreno.

- Projetar um enxame de robôs se comunicando em rede *ad hoc* para agilizar e aumentar a eficiência no processo de reconstrução dos perfis de taxa de dose.

- Desenvolver protótipos de drones para ambientes abertos.

REFERÊNCIA BIBLIOGRÁFICA

- AHMED, A., *Eletrônica de Potência*. São Paulo, Pearson Prentice Hall, 2008.
- BARES, L.C., *Rosie: Remote Work System for Decontamination and Dismantlement*, Waste Management 1996.
- BARROS, V. T.O., *Avaliação da interface de um aplicativo computacional através de teste de usabilidade, questionário ergonômico e análise gráfica do design*, Tese de M.Sc., Universidade Federal de Santa Catarina, Florianópolis, 2003.
- BATISTA, C.B., “Qualidade visual em interfaces interativas”, *Congresso Nacional de Ambientes Hipermídia de Aprendizagem*, Florianópolis, 2004.
- BATISTA, C.B., *Desenvolvimento de interface para ambiente hipermídia voltado ao ensino de geometria sob a ótica da ergonomia e do design gráfico*, Tese de M.Sc., Universidade Federal de Santa Catarina Florianópolis, 2003.
- BENEDETTO, J.M., *et al.*, "High total dose response of the UTMC gate array fabricated at Lockheed Martin Federal Systems," IEEE Radiation Effects Data Workshop 1998. pp. 86-90, 1998.
- BENNETT, P., *RHOBOT: Radiation Hardened Robotics: Experiment*, Sandia National Laboratories, SAND97-2405 I UC-51O, 1997.
- BISHOP, R. H., “The mechatronics handbook”, CRC Press LLC, 2002.
- BOLTON, W., *Engenharia de controle*. São Paulo: Makron, McGraw-Hill, 1995.
- BORENSTEIN, J. *et al.*, *Where am I? Sensor and Methods for Mobile Robot Positioning*, New Edition, The University of Michigan, 1996.
- BORENSTEIN, J., EVANS, J., “The omnimate mobile robot - design, implementation, and experimental results”, IEEE International Conference on Robotics and Automation, pp. 3505–3510, 1997.
- BRAUNL, T., “Mobile Robot Design and Application with Embedded Systems”, Springer Berlin Heidelberg New York, 2003.
- BREGA, R., TOMATIS, N., ARRAS, K.O., *et al.*, “The need for autonomy and real-time in mobile robotics: a case study of XO/2 and Pygmalion”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu, Japan, 2000.

CALIFE, J. L., Diário do Vale: Uma tecnologia para enfrentar desastres nucleares, 2011. Disponível em:
<http://diariodovale.uol.com.br/noticias/0,37831,Uma%20tecnologia%20para%20enfrentar%20desastres%20nucleares.html#ixzz1pfOe5Gx3>. Acesso em: 20 Set 2011, 17:21:32

CAMPION, G., BASTIN, G., D'ANDRÉA-NOVEL, B, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots”, IEEE Transactions on Robotics and Automation v.12, n.1, pp. 47– 62, 1996.

CAMPION, G., *et al.*, “Structural properties and a classification of kinematic and dynamic models of wheeled mobile robots”, IEEE Trans. robotic and automation, v. 12, n.1, p. 47-62, 1996.

COMRADA, N., *Karel Capek – Vida e Obra*, EBooks, 2010

CAPELLI, A., *Mecatrônica Industrial*. São Paulo: Editora Saber Ltda., 2002.

CARDOSO, E., “Programa de integração CNEN, Modulo Informação Tecnica”, 2005.

CARDOSO, E., *et al* “Programa de integração CNEN-Perguntas Frequentes”, 2005.

CARDOSO, E., *et al.*, “Programa de integração CNEN-Aplicações da energia nuclear”, 2005.

CARDOSO, E., *et al.*, “Programa de integração CNEN-Energia nuclear”, 2005.

CARDOSO, E., *et al.*, “Programa de integração CNEN-Radioatividade”, 2005.

CHILDRESS, D. H., *As Invenções Fantásticas de Nicola Tesla*, Lost Science-Adventures Unlimited Press, 1993

CHONG, K. S. AND KLEEMAN, L., “Accurate odometry and error modelling for a mobile robot”, IEEE International Conference on Robotics and Automation, pp. 2783–2788, 1997.

CORDES, J., “The creation of an autonomous multi-terrain mechatron”, 2004.

COSTA,V.A.S.,” Dimensionamento e calibração de suspensão tipo duplo A para veículo mini baja”, 2004.

CRAIG, J., *Introduction to Robotics: Mechanics and Control*, 2nd Edition, Addison-Wesley, Publishing Company Inc., Reading, Massachusetts, 1989.

CRAIG, J., *Introduction to robotics: mechanics and control*, U.S.A: Addison-Weslwy Publishing Company, 450 p., 1986.

DASGUPTA, S., LANGFORD, J., *Active Learning Tutorial*, ICML 2009.

DECRETON, M., “High gamma dose tolerant glasses for vision systems in nuclear applications”, SKC/CEN Nuclear Research Centre 2400 Mol, Belgium, 0-7803- 1793-9, 1994IEEE.

DRAPER, J. V., *Remote maintenance design guide for compact processing units*: Experiment, ORNL/TM-2000/124, August 2000.

DUDEK, G., and JENKIN, M., “Computational Principles of Mobile Robotics”, Cambridge University Press: New York, NY, 2000.

ELETRONUCLEAR. Estudo de Impactos Ambientais - EIA da Unidade 3 da Central Nuclear Almirante Álvaro Alberto. 2006. Disponível em: <<http://www.eletronuclear.gov.br/hotsites/eia/>>. Acesso em: 18 jun. 2014.

FENTON, J. *Handbook of Automotive Powertrains and Chassis Design*, London, 1998.

FONG, T., NOURBAKHSH, I., DAUTENHAHN, K., “A survey of socially interactive robots”, *Robotics and Autonomous Systems*, v. 42, n. (3-4), pp. 143-166, 2003.

FREITAS, J.C.A. and SANTOS, P.B., *Robótica*. São Paulo: SENAI-SP, 2000.

FU, K., GONZALEZ, R., LEE, C., *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill, 1987.

GILLESPIE, T. D., “Fundamentals of Vehicle Dynamics”, United States of America, 1992.

GOMIDE, F. A. C., GUDWIN, R. R., TANSCHKEIT, R., “Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações”, 2003.

GORIS, K., “Autonomous Mobile Robot Mechanical Design”, M.Sc. Dissertation, Vrije Universiteit Brussels, Belgium, 2005.

GRAF, B., SCHRAFT, R.D., *et al.*, *A mobile robot platform for assistance and entertainment*, In: Proceedings of the International Symposium on Robotics, Montreal, Quebec, Canada, 2000.

GROOVER, M. P., WEISS, M., NAGEL, R. N. *et al.*, *Industrial Robotics: Technology, Programming and Applications*. Singapore, McGraw Hill Book Company, 1986.

HAYKIN, S., *Redes Neurais - Princípios e Prática*, 2ª edição, editora Bookman, 2008.

HOELZEL, C. G. M., *Design ergonômico de interfaces gráficas humano-computador: um modelo de processo*, Tese de M.Sc., Universidade Federal de Santa Catarina, Florianópolis, 2004.

HOUSSAY, L., “Robotics and radiation hardening in the nuclear industry”, 2000.

HSIU, T., RICHARDS, S., BHAVE, A. *et al.*, *Designing a Low-cost, Expressive Educational Robot: Experiment*, 2002.

HUANG, L., *et al.*: “Design and Analysis of a Four-wheel Omnidirectional Mobile Robot”, Proc. of the 2nd Int. Conf. on Autonomous Robots and Agents, 2004.

KAPOOR, A. *et al.*: Active learning with Gaussian processes for object categorization. In Proc. International Conference on Computer Vision (ICCV), 2007.

Link: http://people.csail.mit.edu/rurtasun/publications/iccv_kapoor_et_al.pdf

KENJO, T. *Power Electronics for the Microprocessor Age*, Oxford, NY: University Press Inc., 1995.

KERWIN, D.B., BENEDETTO, J.M., “Total dose and single event effects testing of UTM Commercial rad-hard gate arrays.” IEEE Radiation Effects Data Workshop 1998. pp. 80-85, 1998.

KISSEL, Thomas E.. *Industrial Electronics*, New Jersey: Prentice Hall. 2000.

KORDIC, V., LAZINICA, A., MERDAN, M., “Cutting Edge Robotics”, Advanced Robotic Systems International ISBN 3-86611-038-3, 2005.

KOREN, Yoram. *Robotics for engineers*, New York: McGraw-Hill. 1985.

KRAUSE, A., GUESTIN, C., Nonmyopic active learning of gaussian processes: an exploration/exploitation approach. In ICML '07: Proc. of the 24th int. conf. on Machine learning, pages 449–456. ACM, 2007.

Link: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.7132&rep=rep1&type=pdf>.

KRAUSE, A., SINGH, A., GUESTIN, C., Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. J. Machine Learning Research, 9:235–284, 2008.

Link: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.449.6516&rep=rep1&type=pdf>

LAURIDSEN, K., CHRISTENSEN, P., KONGSO, H., “Assessment of the reliability of robotic systems for use in radiation environments”, Reliability Engineering and System Safety v.53, pp. 265-276, 1996.

LEBOUTHILLIER, A., “W. Grey Walter and his Turtle Robots”, *The Robot Builder (TRB)*, v.11, n. 5, pp.1-3, May 1999.

LIU, Y., *et al.*, “Omni-Directional Mobile Robot Controller Design by Trajectory Linearization.” Proc. of the 2003 American Control Conference, 2003.

MACKAY, D. J. C., Introduction to Gaussian Processes. In Bishop, C. M., editor, Neural Networks and Machine Learning. Springer-Verlag, 1998.

MARIAN, F., *The Cost-Benefit of Robotic Devices in Nuclear Power Plants*, Public Service Electric and Gas Company, 1987.

MARIAN, F.A, ROWAN, H.T., *The Cost-Benefit of Robotic Devices in Nuclear Power Plants*, Public Service Electric and Gas Company, 1987.

MARTORELL, S., SANCHEZ, A., et al, “*The Use of Maintenance Indicators to evaluate the Effects of Maintenance Programs on NPP Performance and Safety*”, Reliability Engineering and System Safety 65 (1999) 85-94, 1998.

MCDONALD, M.J., *et al.*, ” Virtual collaborative control to improve intelligent robotic system efficiency and quality”, International Conference on Robotics and Automation, Albuquerque, NM, EUA, 1997.

MORTIER, F., “Application of sensory control in nuclear environment, 2nd Int. Conf. on Intelligent”, Autonomous Systems, Amsterdam, v.11, n.14, December 1989.
NETO, A.C. *Dinâmica Veicular*, SAE Brasil, 2006.

NAIFF, D. ., SILVEIRA, P. C. R., PEREIRA, C. M. N. A., “*Field Dose Radiation Determination by Active Learning with Gaussian Process for Autonomous Robot Guiding*”, *International Nuclear Atlantic Conference - INAC 2017*

NOF, Shimon Y. *Handbook of industrial robotics*, New York: John Wiley. 1999.

NORTMAN, S., “Pneuman: A Humanoid Robot Implementation”, Florida Conference on Recent Advances in Robotics (FCRAR), Florida, May 23-24, 2002.

NORTMAN, S., PEACH, J., NECHYBA, M. *et al.*, “Construction and Kinematic Analysis of an Anthropomorphic Mobile Robot”, Proceedings of the 2001 Florida Conference on Recent Advances in Robotics, Volume 7, Pages 32-42.

NORTON, R. L., *Projeto de máquinas*, 2ª ED., PORTO ALEGRE, 2004.

NOUAILHETAS, Y., “Radiações ionizantes e a vida”, 2005.

OLDHAM, T., *et al.*, "An overview of radiation-induced interface traps in MOS structures," Semicond. Sci. Technol, v. 4, pp. 986-999, 1989.

OLIVEIRA, E. R., *Avaliação ergonômica de interfaces da SciELO – Scientific Electronic Library Online*, Tese de M.Sc., Universidade Federal de Santa Catarina, Florianópolis, 2001.

PAUL, R., *Robot Manipulators: Mathematics, Programming, and Control*, MIT Press, 1981.

PAYNE, A., “Design and Construction of a Pair of Cooperating Autonomous Mobile Robot”, 2004.

PAZOS, Fernando. *Automação de Sistemas & Robótica*, Rio de Janeiro: Axcel Books. 2002.

PODNAR, G., DOWLING, K., BLACKWELL, M., *A Functional Vehicle for Autonomous Mobile Robot Research: Experiment*, CMU-RI-TR-84-28, 1984

POLONSKII, M. *Introdução à robótica e mecatrônica*, Caxias do Sul: Editora UCS, 1996.

RASMUSSEN, C. E. and WILLIAMS, C. K. I., *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

RATNER, D., “Using LabVIEW to prototype an industrial-quality real-time solution for the Titan outdoor 4WD mobile robot controller”, *International Conference on Intelligent Robots and Systems*, RSJ, 2000.

RATNER, D., MCKERROW, P.J., *Advancing mobile robot research by using global engineering software tools with technical support over the Internet*, FSR'99, August, Pittsburgh, pp 332-338, 1999.

REE, *Revista Eletrônica de Energia*, v. 4, n. 1, p. 38 -52, jan./dez. 2014

REIMPELL, J. and STOLL, H., *The Automotive Chassis: Engineering Principles*, Great Britain, 1996.

RICH, E. and KNIGHT, K., “*Inteligência Artificial*”, 2 ed., São Paulo, SP, 1993.

RODRIGUES D. W. L., *Uma avaliação comparativa de interfaces homem-computador em programas de geometria dinâmica*, Tese de M.Sc., Universidade Federal de Santa Catarina, Florianópolis, 2002.

ROMAN, H.T., “Robotic applications in PSE &G’s nuclear and fossil power plants,” *IEEE Trans. Energy Conversion*, v.8, pp. 584–592, Sept. 1993.

ROMANO, Victor Ferreira. *Robótica Industrial: Aplicação na Indústria de Manufatura e de Processos*, São Paulo: Edgard Blücher Ltda. 2002.

ROSS, T. J., *Fuzzy Logic With Engineering Applications*, 2ª edição, editor Wiley, 2015.

RUMBAUGH, J., BIAHA, M., PREMERLANI, W. et al, “*Modelagem e Projetos em Objetos*”. 8 ed., Rio de Janeiro, RJ, 1994.

RUSSELL, S. and NORVIG, P., “*Artificial Intelligence – A Modern Approach*”, USA, 1995.

SALANT, Michael A. *Introdução à robótica*, São Paulo: Makron, McGraw-Hill, 1991.

SANDIN, P., *Robot Mechanisms and Mechanical Devices Illustrated: Experiment*, DOI: 10.1036/007142928X, 2003.

SAVALL, J., “Two Compact Robots for Remote Inspection of Hazardous Areas in Nuclear Power Plants”, *Proceedings of the 1999 IEEE International Conference on Robotics & Automation* Detroit, Michigan May 1999.

SETTLES, B., *Active learning literature survey*. Computer Science Technical Report 1648, University of Wisconsin-Madison, January 2010. (um survey de active learning.

link: <http://burrsettles.com/pub/settles.activelearning.pdf>

SCHILLING, R. J., *Fundamentals of Robotics: Analysis & Control*, Prentice-Hall, 1990.

SHARP, R., “Radiation tolerance of components and materials in nuclear robot applications”, *Engineering and System Safety*, 1996, pp. 291-299.

SHOHAM, Moshe. *Fundamentals of Robotics*, Tel Aviv: Eshed Robotec Ltd. 1982.

SHOHAM, Moshe. *Robotics Structure*, Tel Aviv: Eshed Robotec Ltd. 1982.

SIEDLE, H., *et al*, “The management of radiation-induced faults in teleoperation used in nuclear plants.”, *ANS 6th Topical Meeting on Robotics and Remote Systems*, Monterey, CA, v.5 n.10, pp.169-176, February 1995.

SIEGWART, R., *et al.*, “Robox at Expo.02: A large-scale installation of personal robots”, *Robotics and Autonomous Systems* v. 42, pp. 203–222, 2003.

SIEGWART, R., NOURBAKHSH I. R., *Introduction To Autonomous Mobile Robots*, The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004.

SIKKING, L., “The Development of an Indoor Navigation Algorithm for an Autonomous Mobile Robot”, 2004.

SILVEIRA, P. C. R., *Robô Baseado em Tecnologia Celular Android e Lógica Nebulosa para Inspeção e Monitoração em Usinas Nucleares*, Dissertação M.Sc., PEN/COPPE/UFRJ, 2012.

SILVEIRA, P. C. R., NAIFF, D. F., PEREIRA, C. M. N. A., SCHIRRU, R., “Reconstruction of radiation dose rate profiles by autonomous robot with active learning and Gaussian process regression”, *Annals of Nuclear Energy*, Volume 112, February 2018, Pag. 876–886

SNELSON, E., *et al.*, Warped Gaussian processes. In *Advances in Neural Information Processing Systems*, 2003.

(link aqui: <http://mlg.eng.cam.ac.uk/pub/pdf/SneRasGha04.pdf>)

SOMMERVILLE, I., *Engenharia de Software*, Pearson Education, 9ª edição, 2011

SOUZA, J.A.M.F., “Robôs não Industriais”, 2007

TAUHATA, L., “Radioproteção e Dosimetria: Fundamentos”, 2003.

VANDERGRIFF, K., *Designing equipment for use in gamma environments: Experiment*, ONRL/TM-11175, 1990.

VAUCANSON, J., *Le Mecanisme Du Fluteur Automate: Presente A Monsieur De L'Academie Royale Des Science*, Frenh Edition, 2009.

VAZ, A. I. F., “Métodos Numéricos”- Notas de aula, Departamento de Produção e Sistemas, Escola de Engenharia, Universidade do Minho, 2007.

VIEIRA, R. S., *Protótipo de um sistema de monitoramento remoto inteligente*, Tese de M.Sc., Universidade Federal de Santa Catarina, Florianópolis 1999.

VIERCK, R. K. “Vibration Analysis”, 1974.

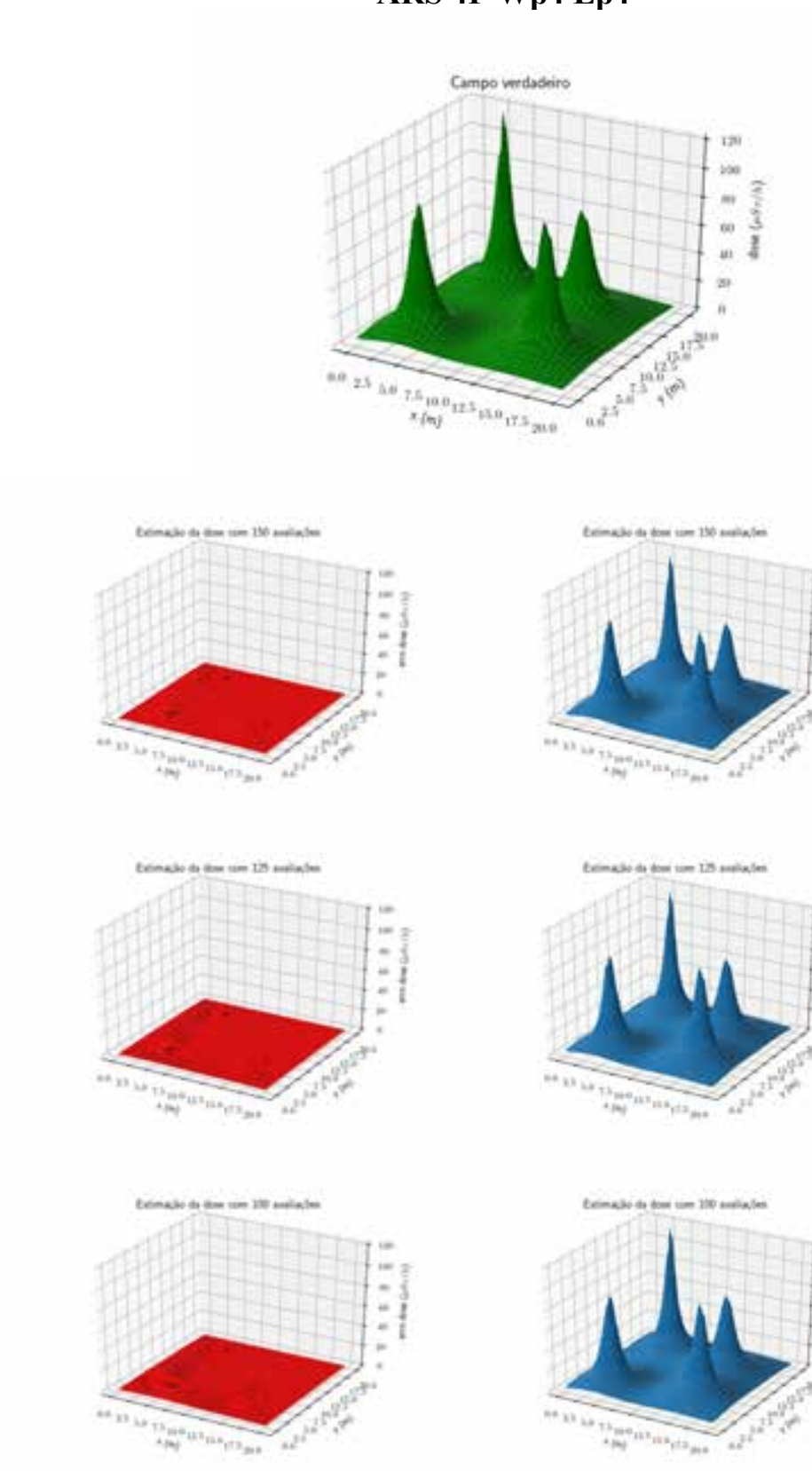
WERGER, B., *People Are Robots, Too. Almost*, 2003. Disponível em:
http://www.nasa.gov/vision/universe/roboticexplorers/robots_like_people.html. Acesso em: 10 Out. 2011, 21:30:14

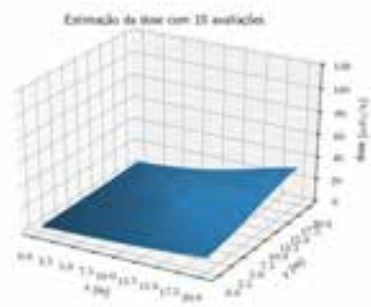
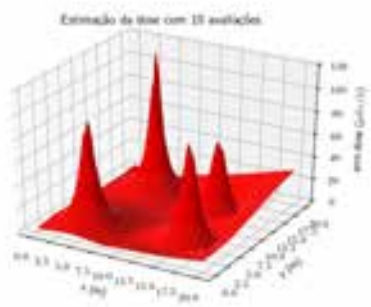
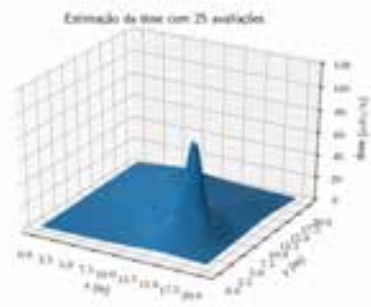
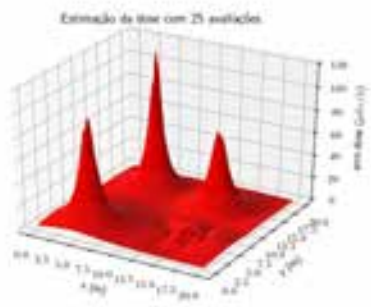
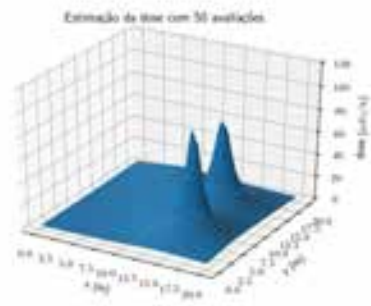
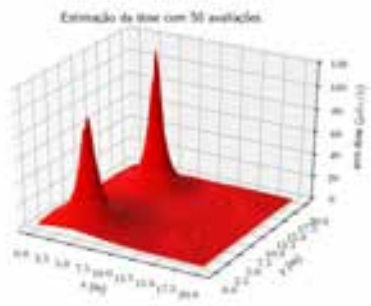
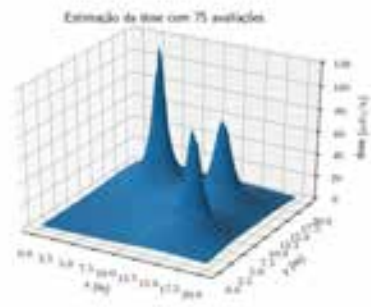
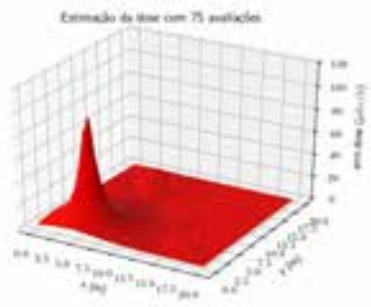
WILFONG, G.T., “Autonomous Mobile Vehicles”, NEC Research Institute QT&T Bell Laboratories, Princeton, Murray Hill, USA, 1990.

YOUK, G., *et al*, “Technology Development for the Radiation Hardening of Robots”, IEEE RSJ International Conference on Intelligent Robots and Systems, 1999.

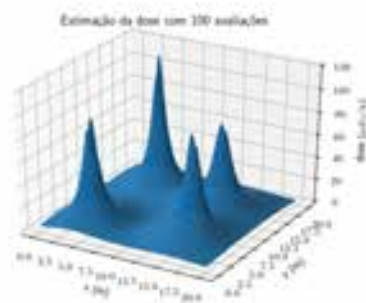
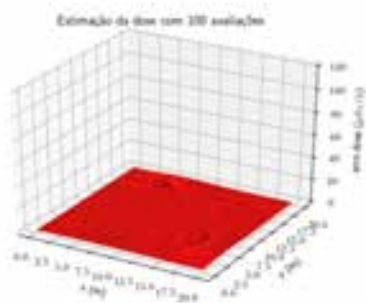
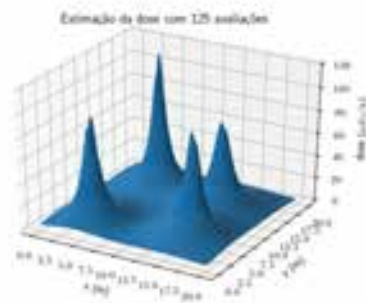
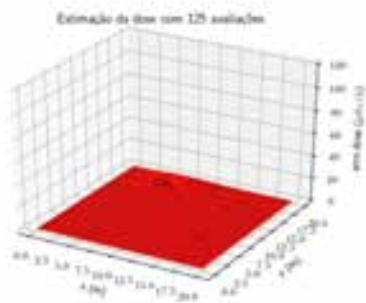
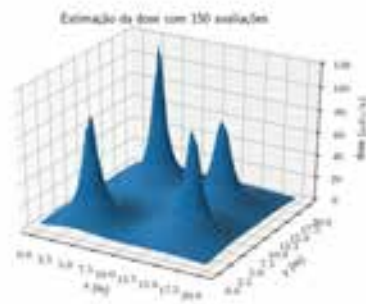
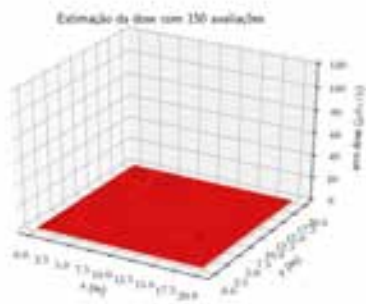
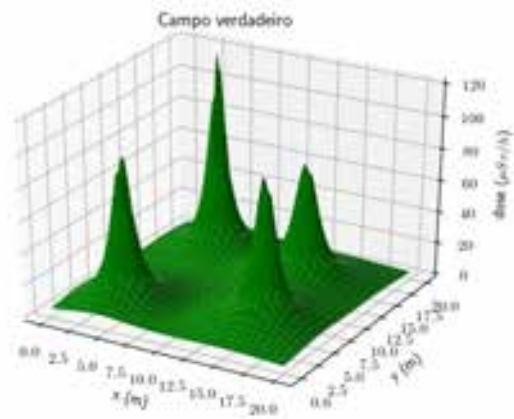
ANEXOS

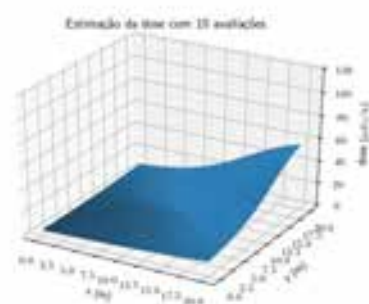
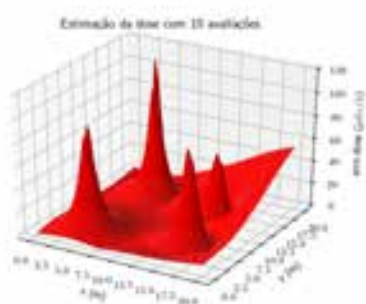
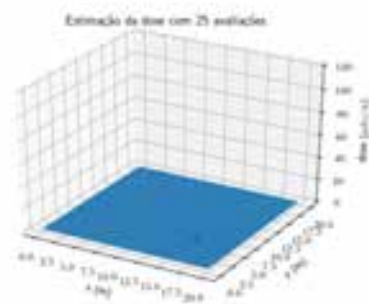
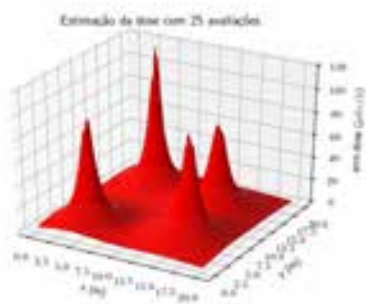
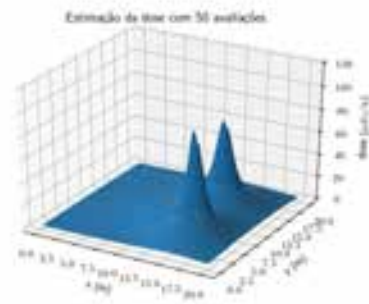
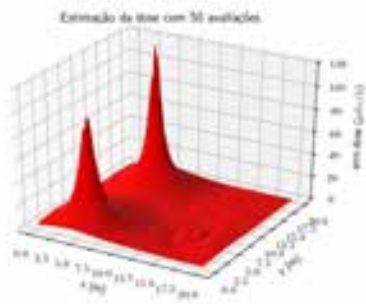
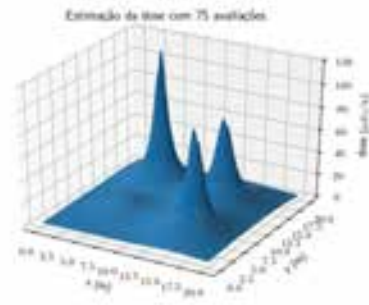
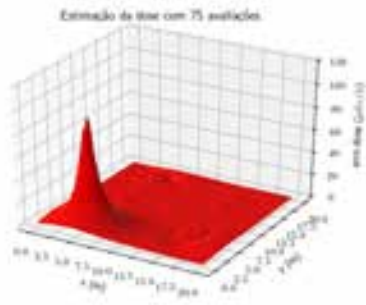
ARS 4P Wp4 Lp4



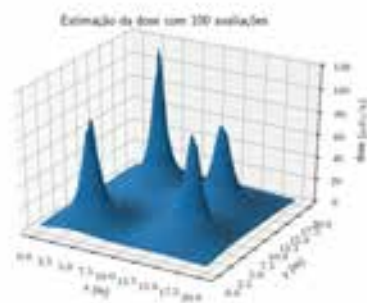
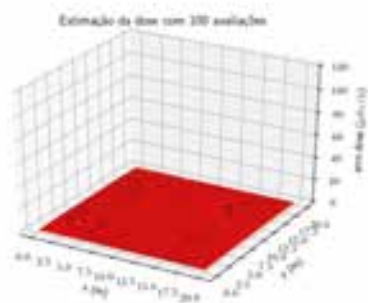
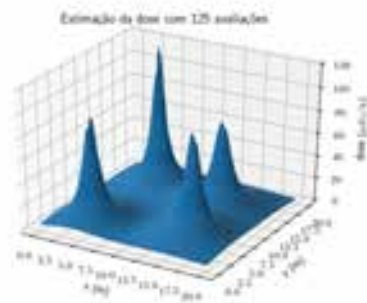
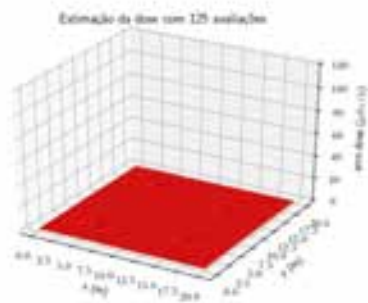
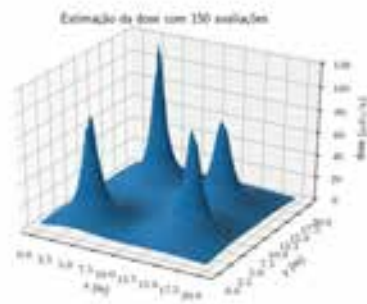
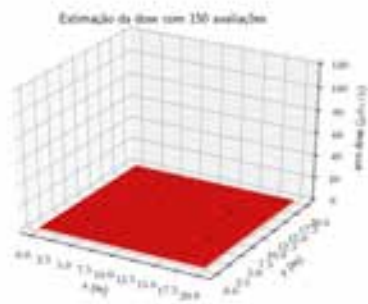
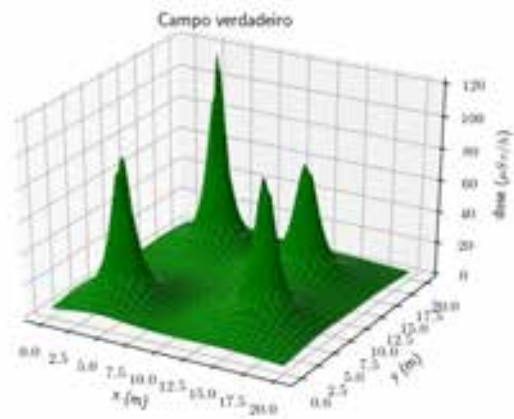


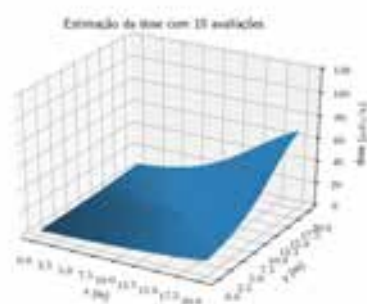
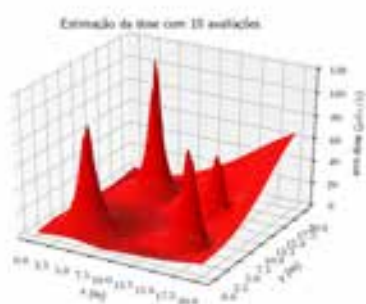
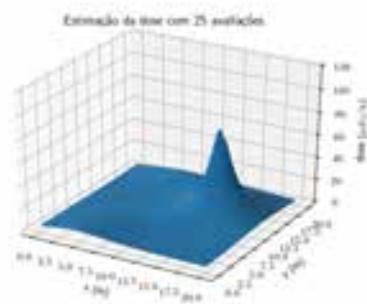
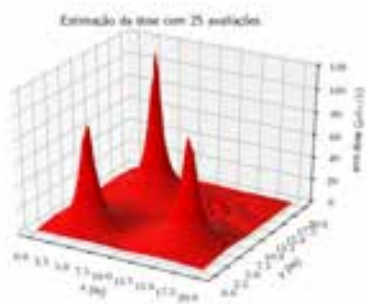
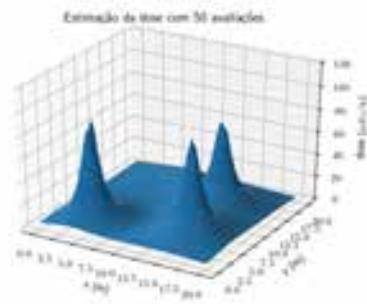
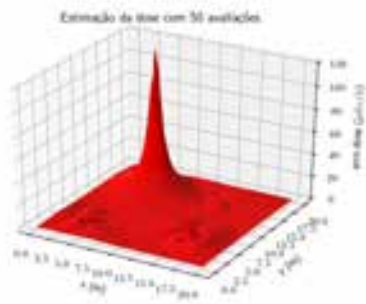
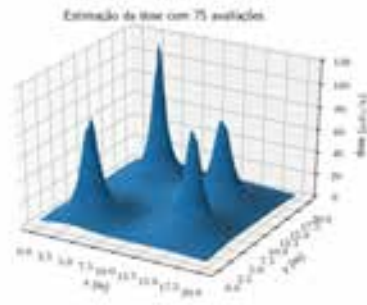
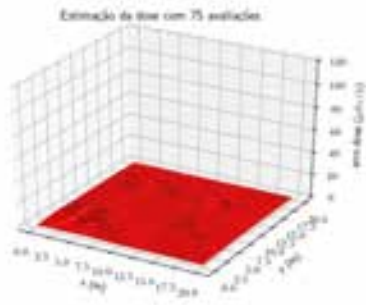
ARS 4P Wp4 Lp8



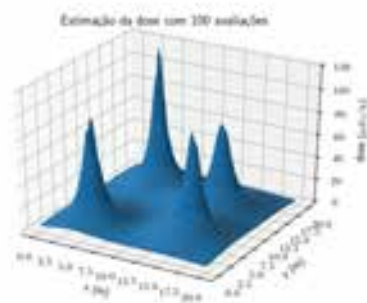
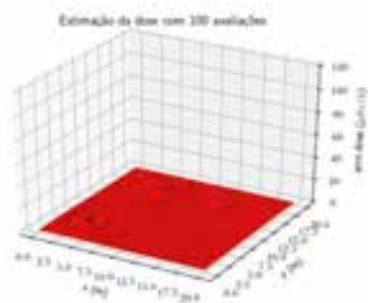
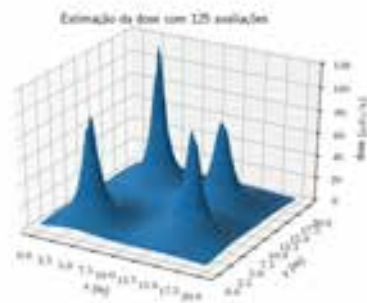
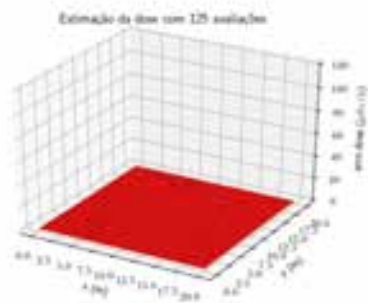
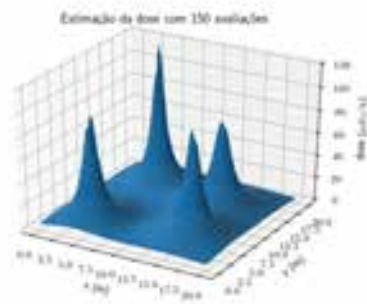
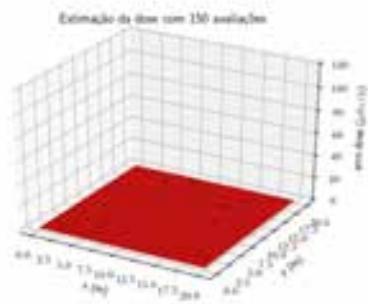
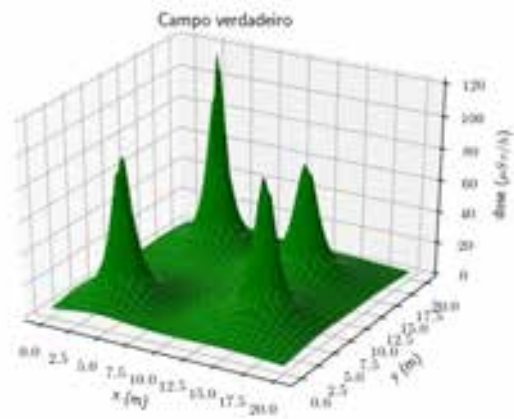


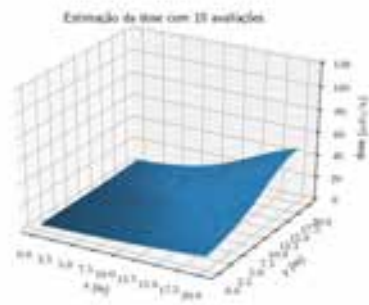
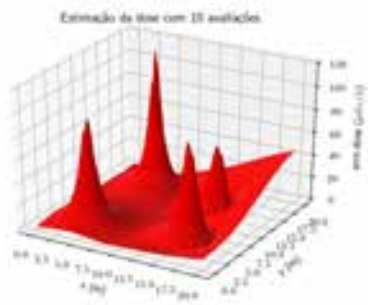
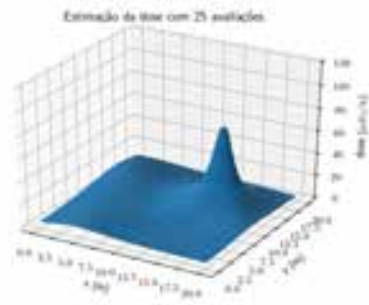
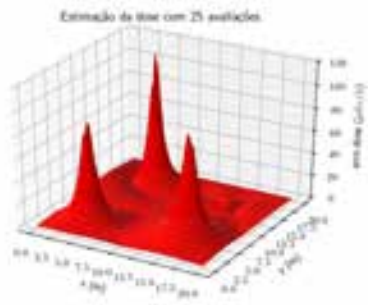
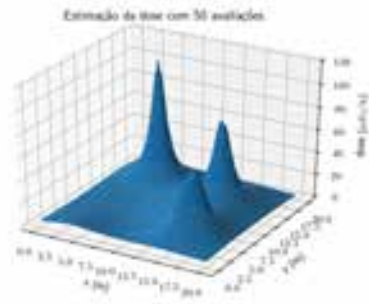
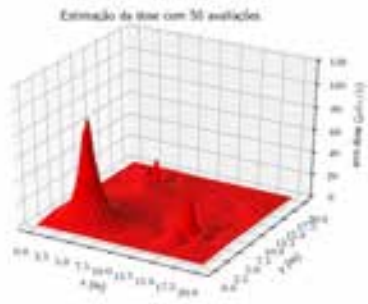
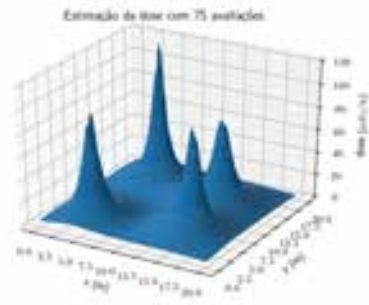
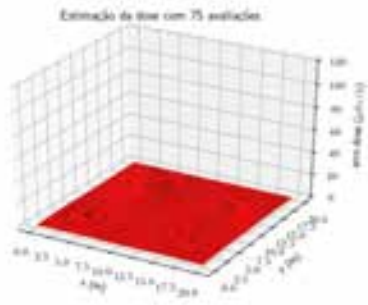
ARS 4P Wp8 Lp4



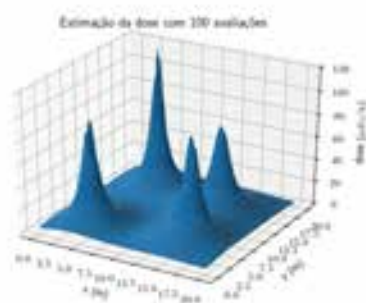
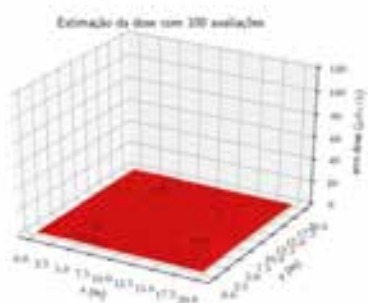
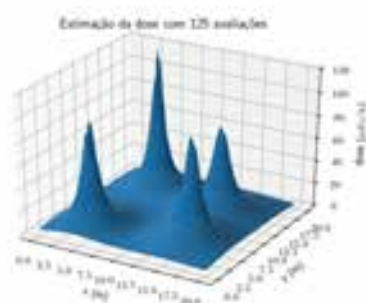
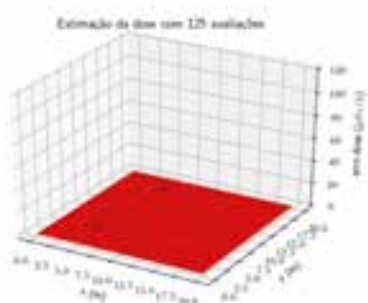
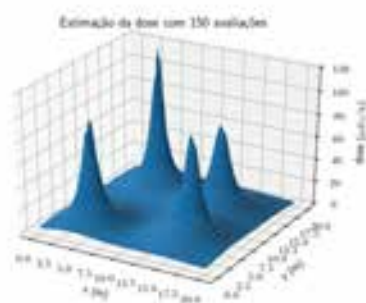
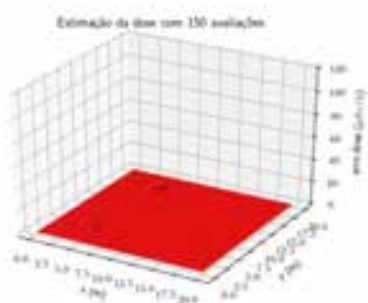
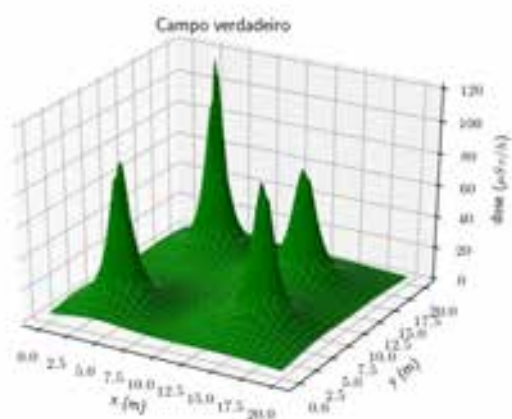


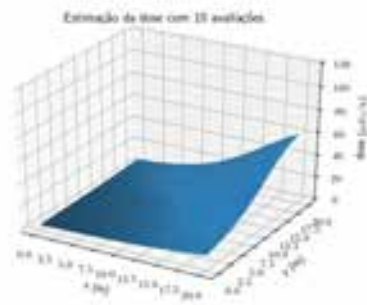
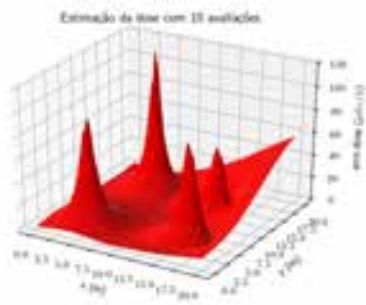
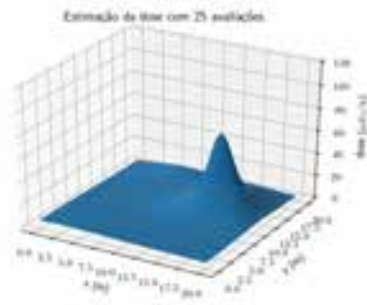
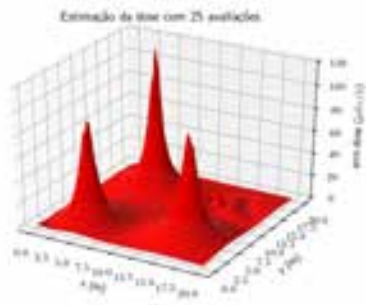
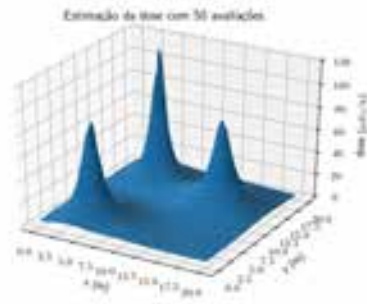
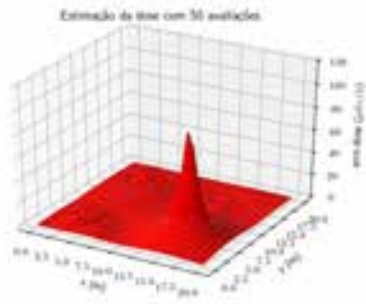
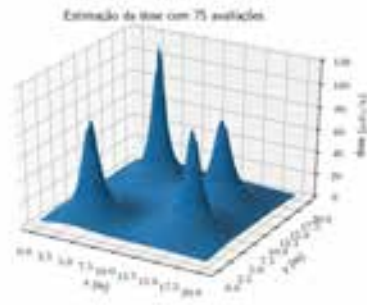
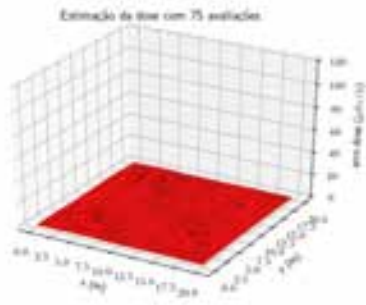
ARS 4P Wp8 Lp8



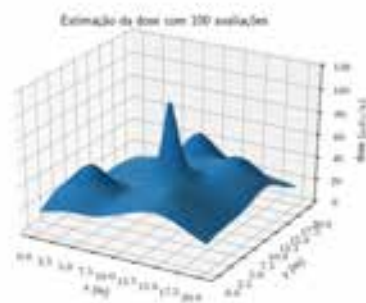
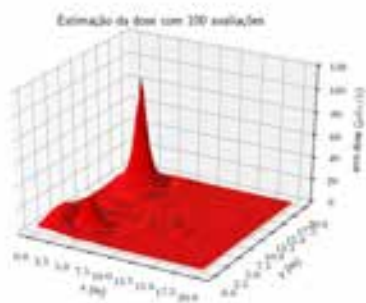
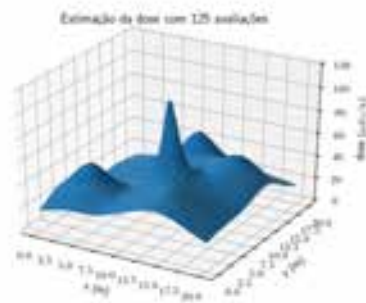
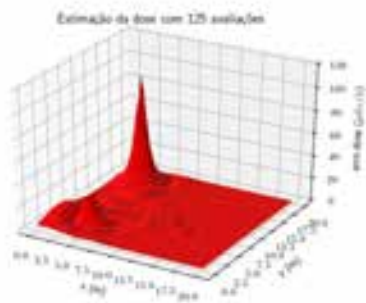
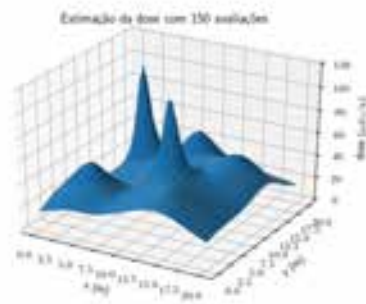
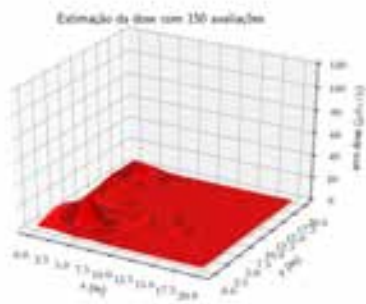
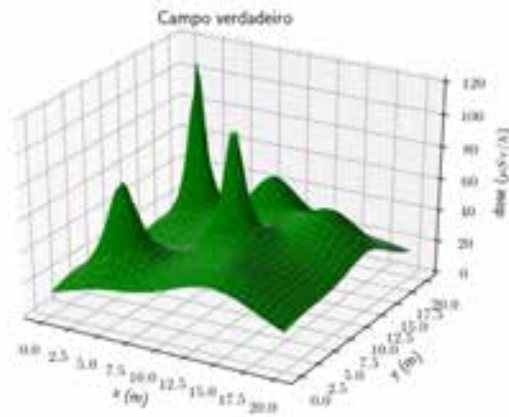


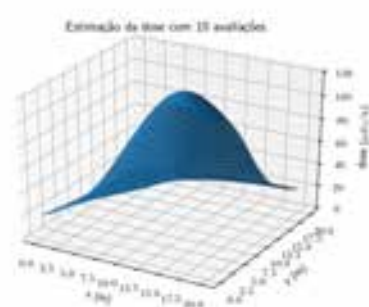
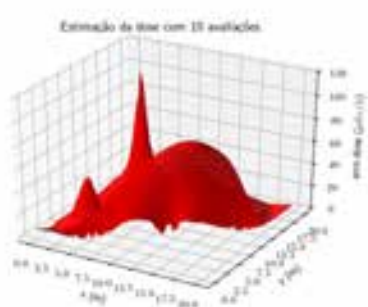
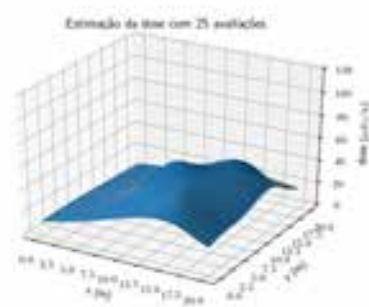
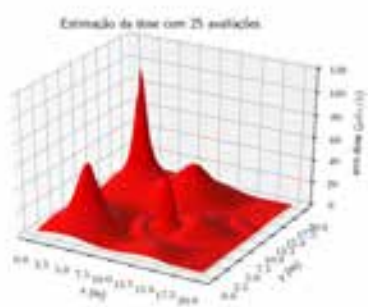
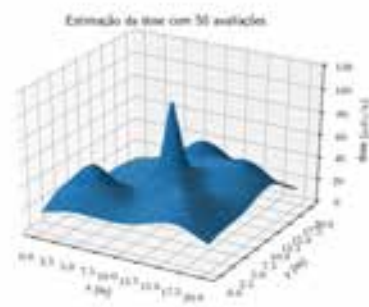
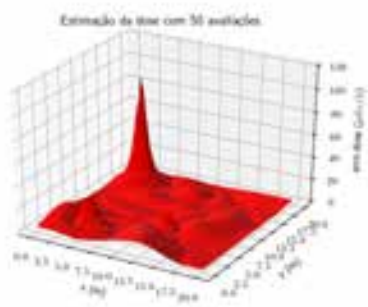
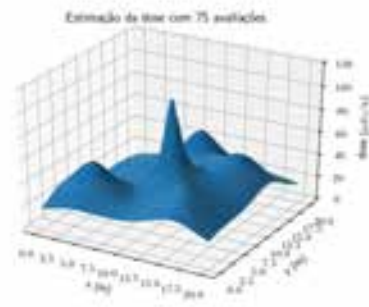
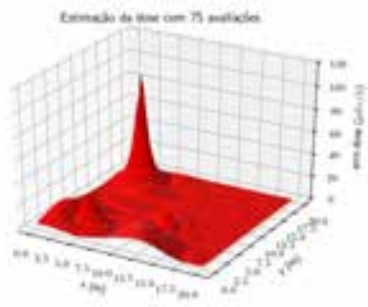
ARS 4P SP



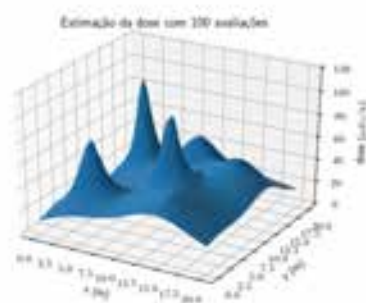
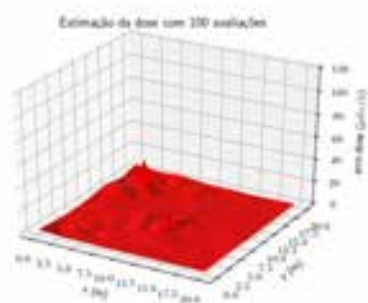
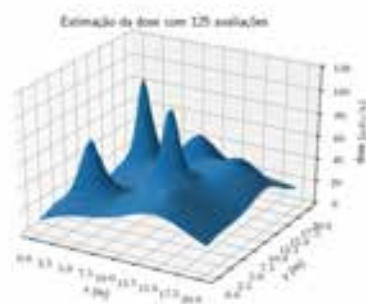
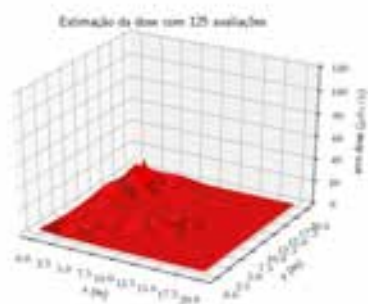
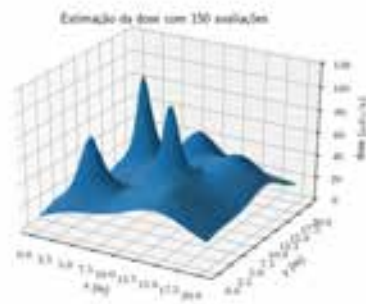
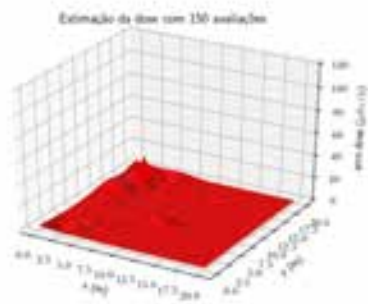
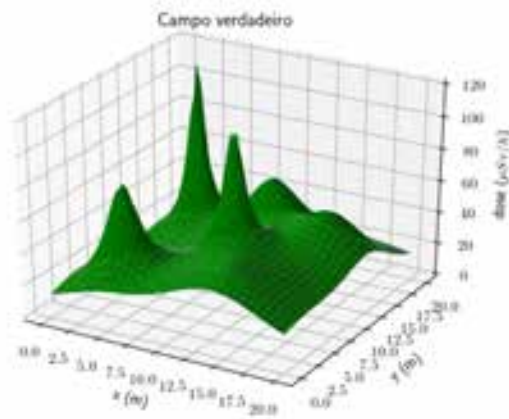


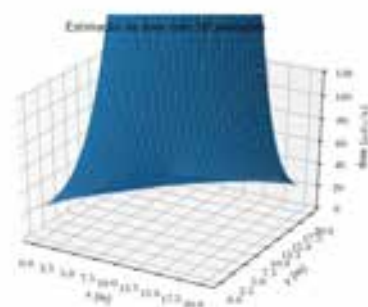
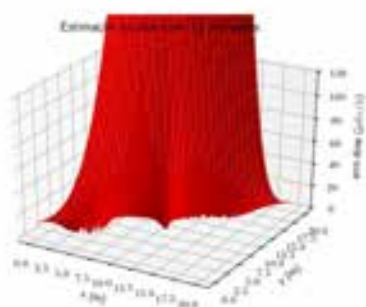
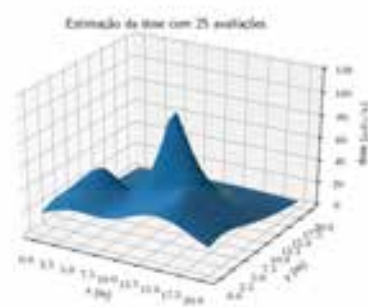
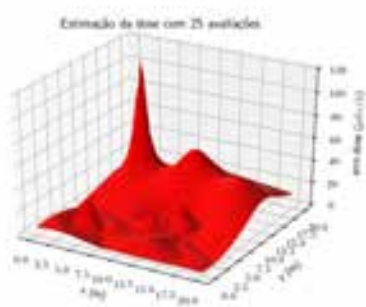
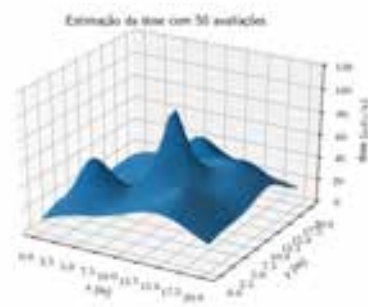
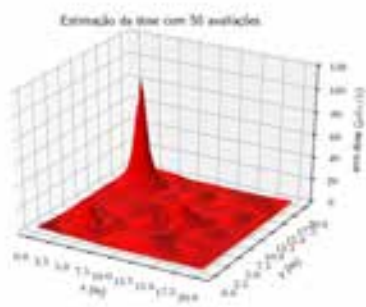
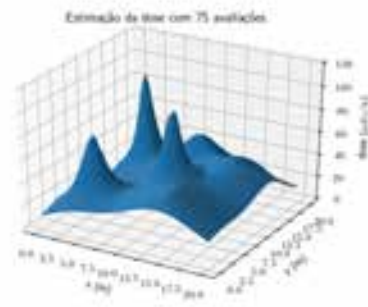
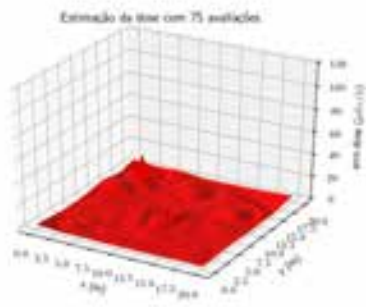
ARS 6P Wp4 Lp4



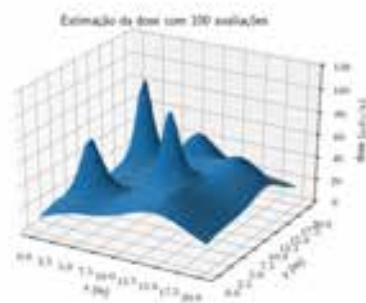
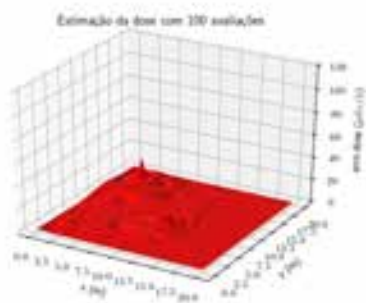
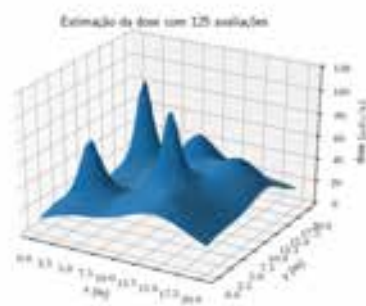
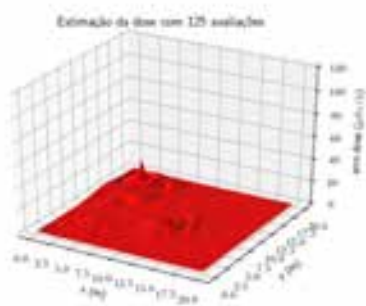
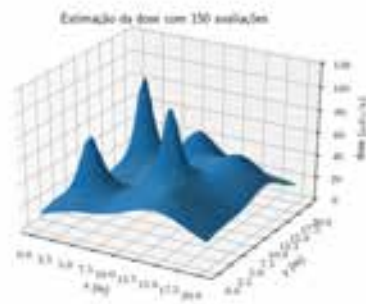
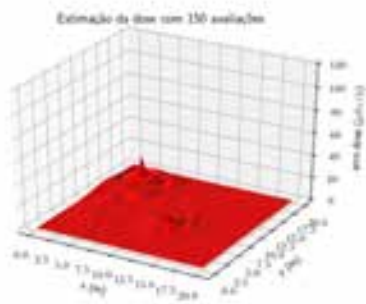
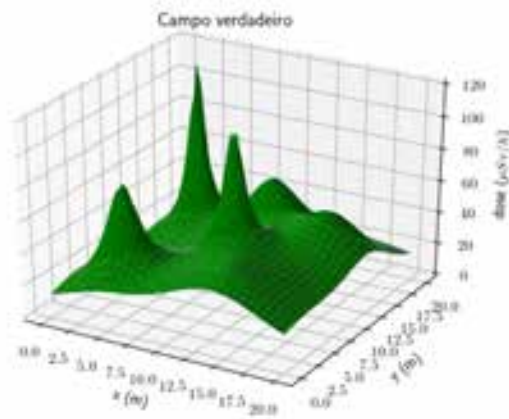


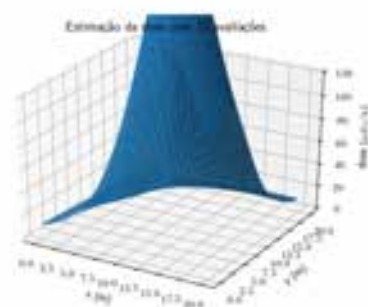
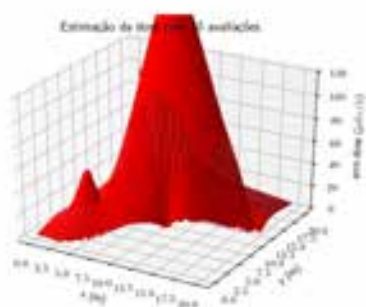
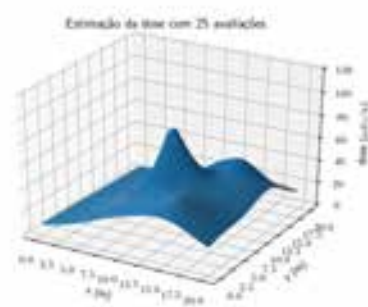
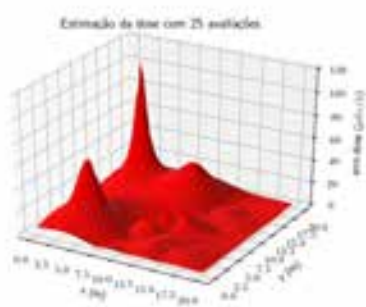
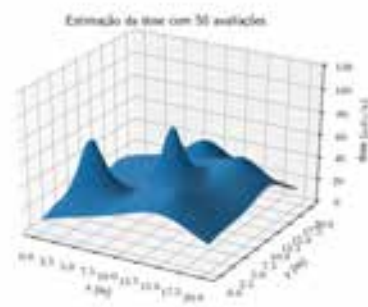
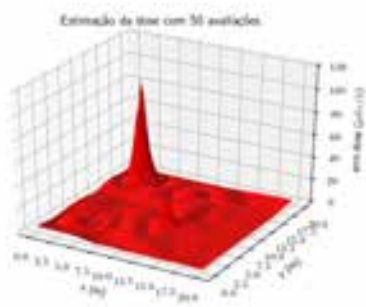
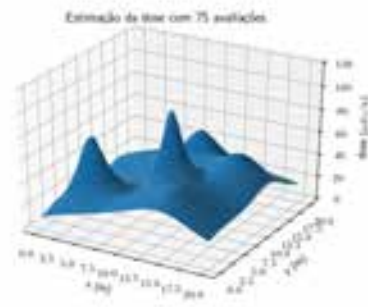
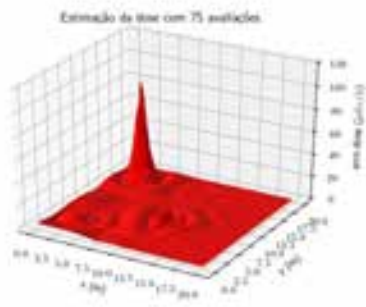
ARS 6P Wp4 Lp8



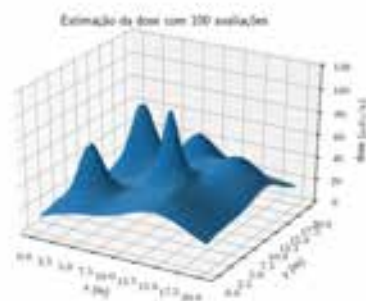
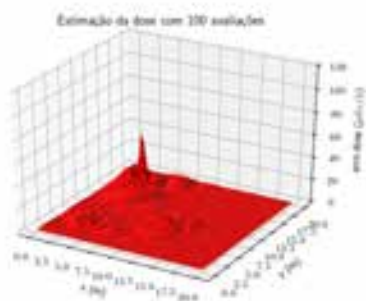
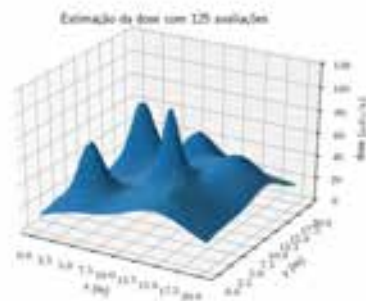
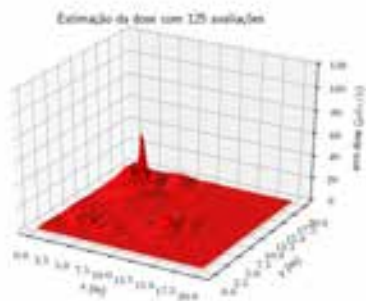
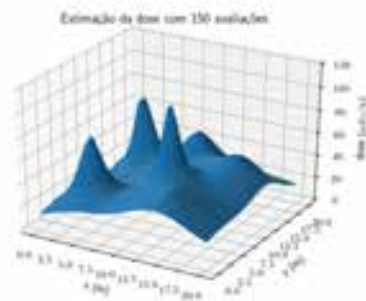
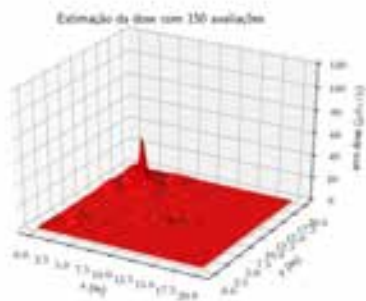
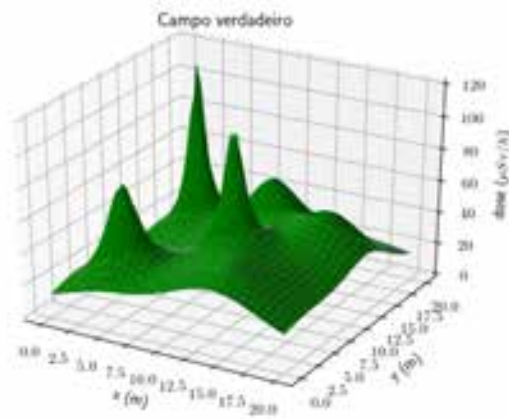


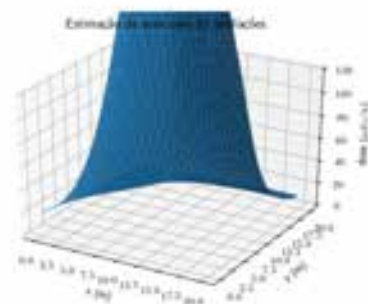
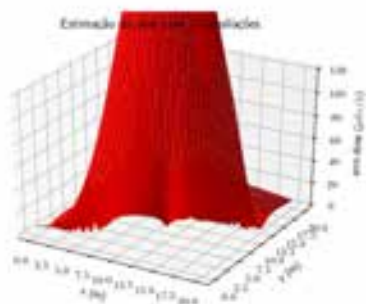
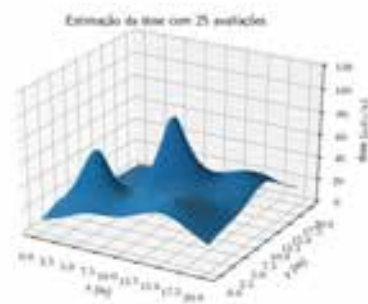
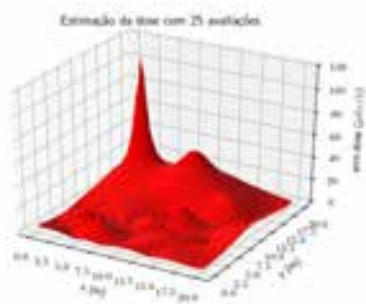
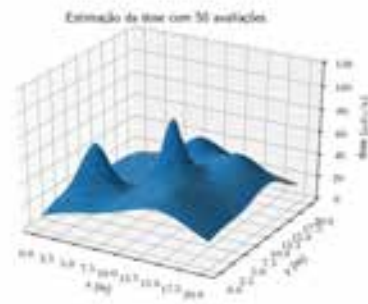
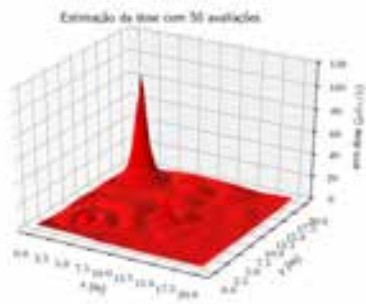
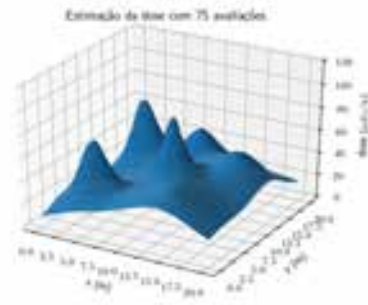
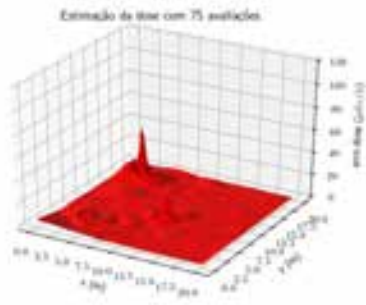
ARS 6P Wp8 Lp4



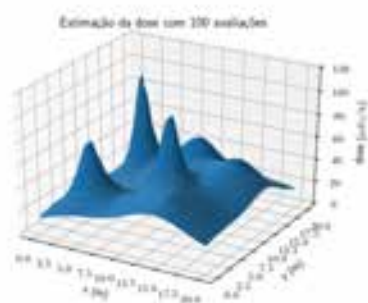
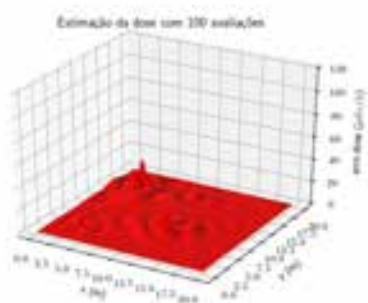
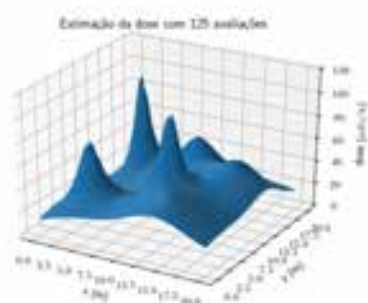
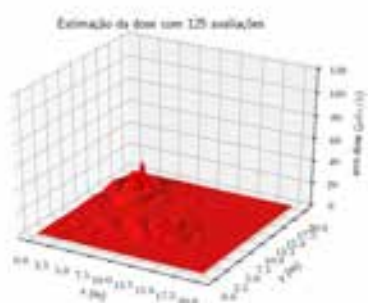
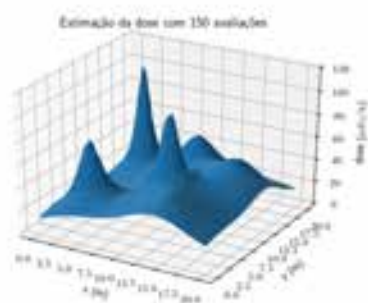
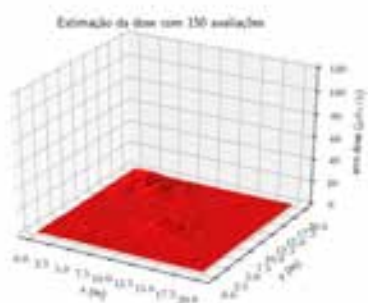
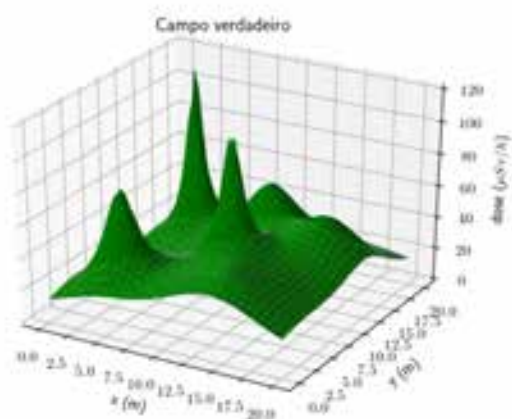


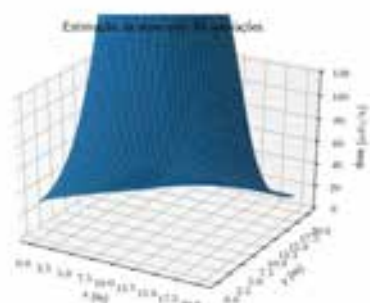
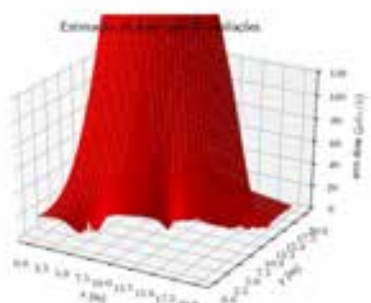
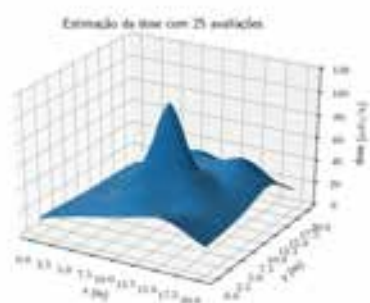
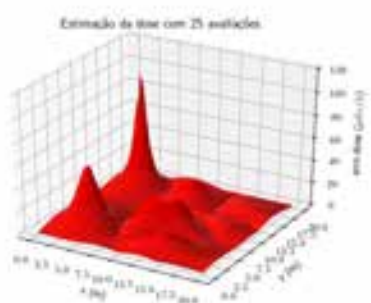
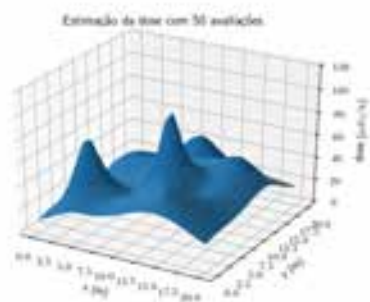
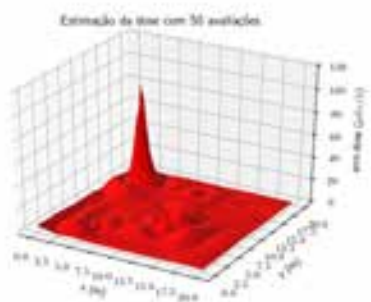
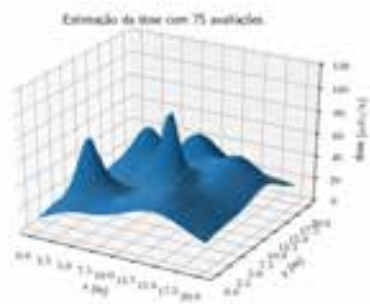
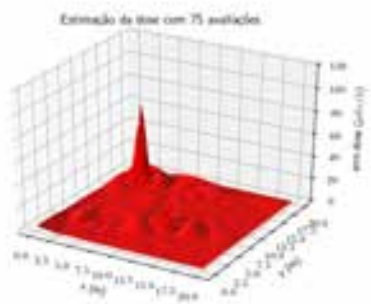
ARS 6P Wp8 Lp8



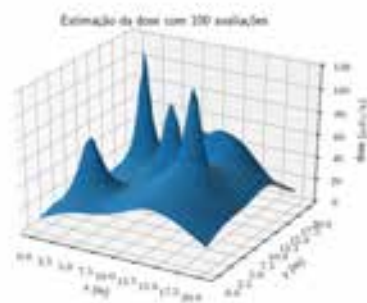
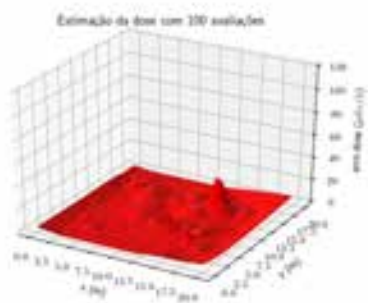
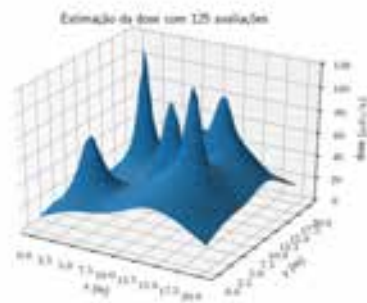
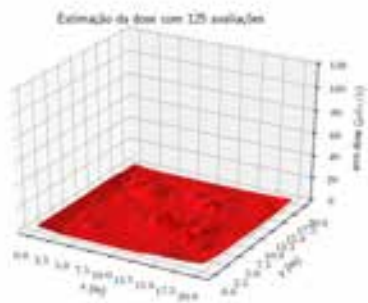
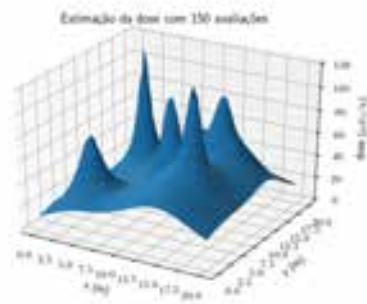
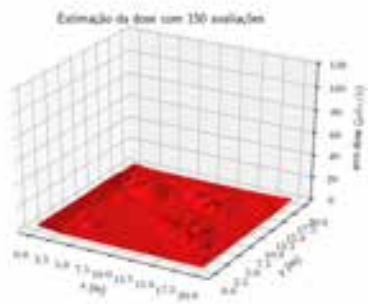
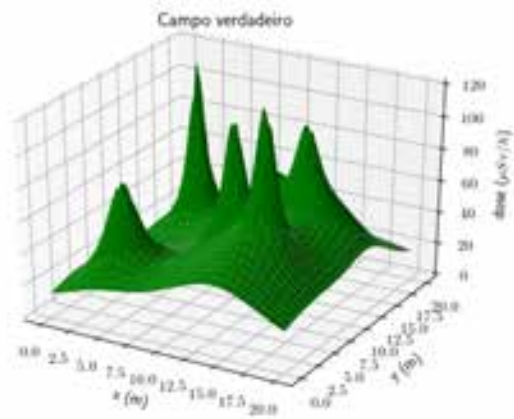


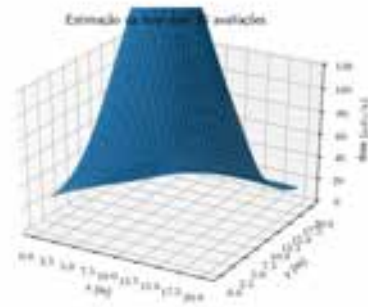
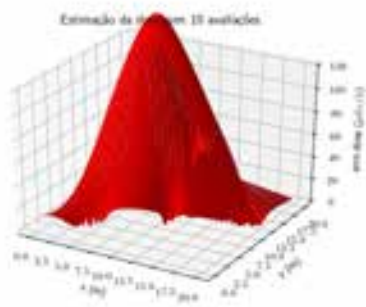
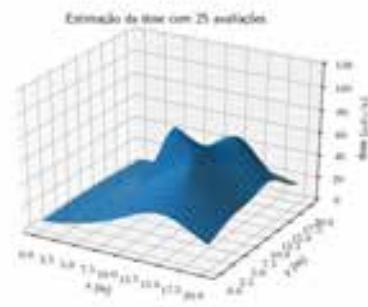
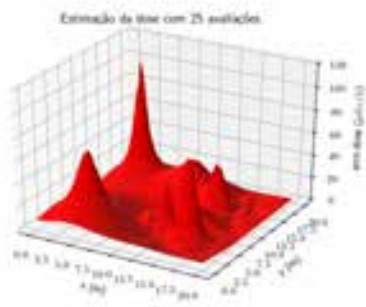
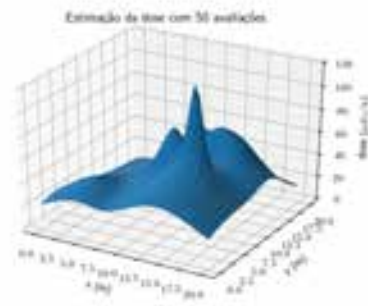
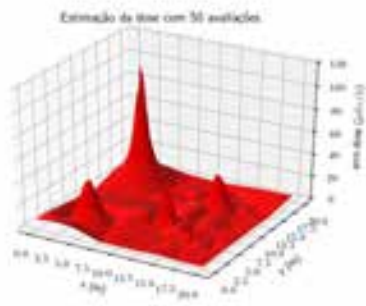
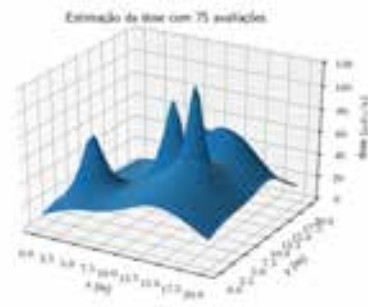
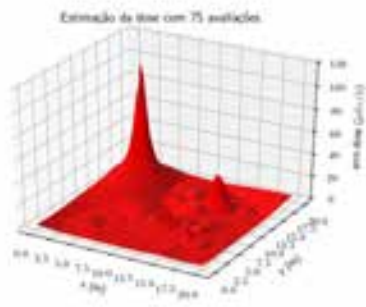
ARS 6P SP



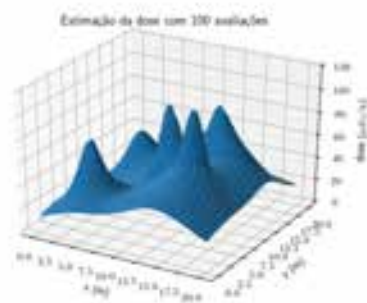
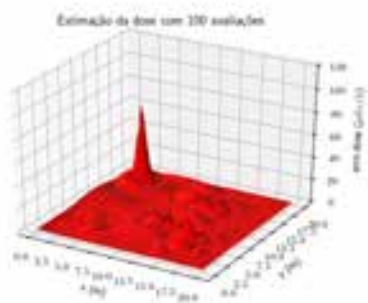
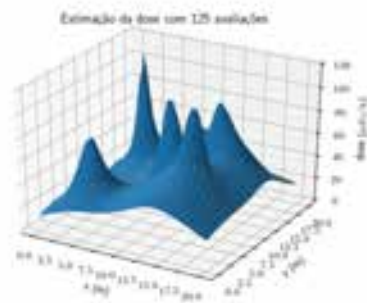
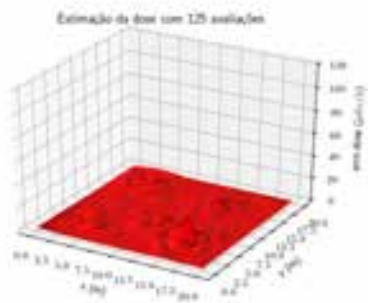
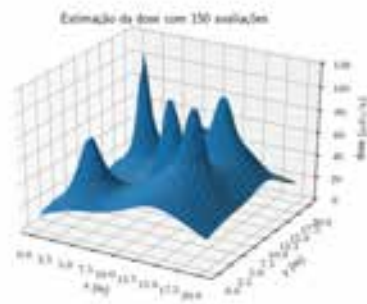
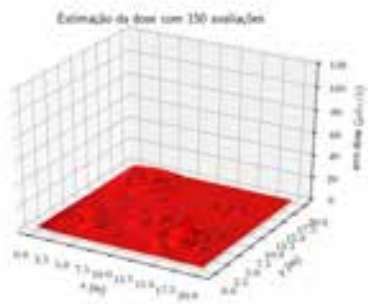
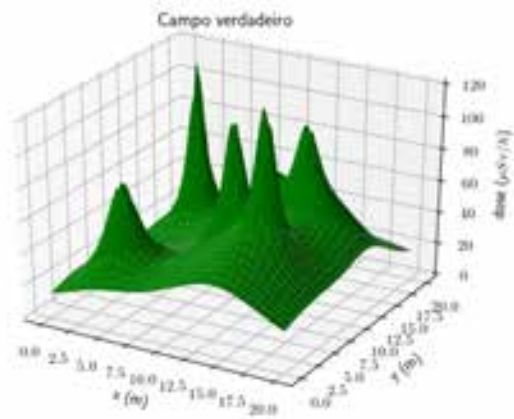


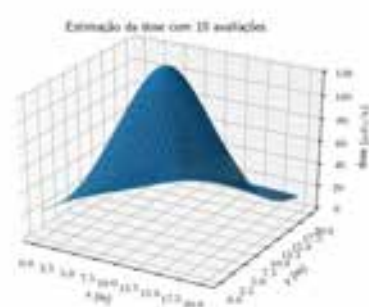
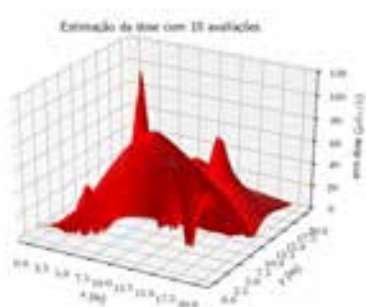
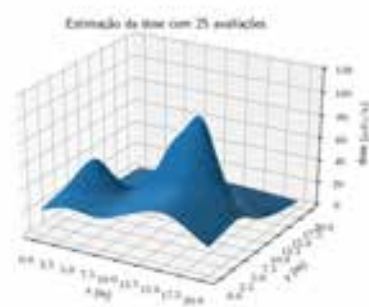
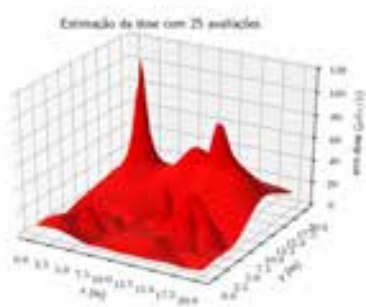
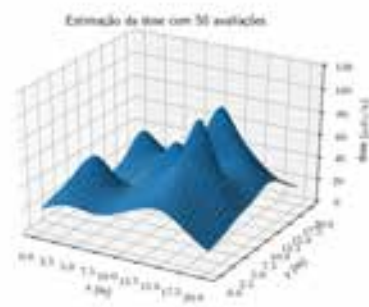
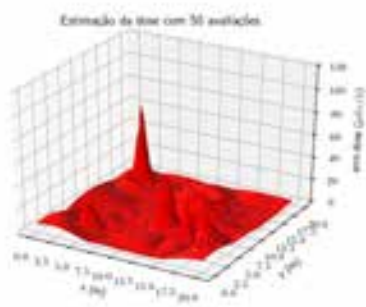
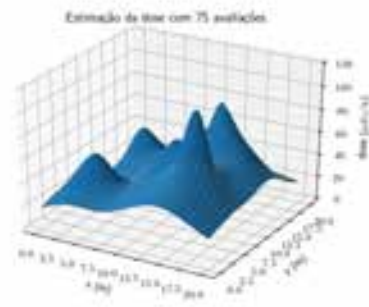
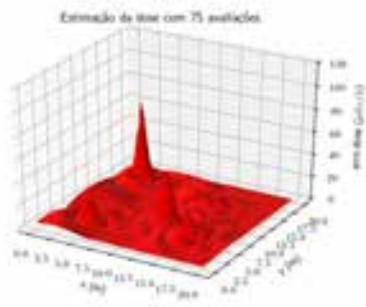
ARS 8P Wp4 Lp4



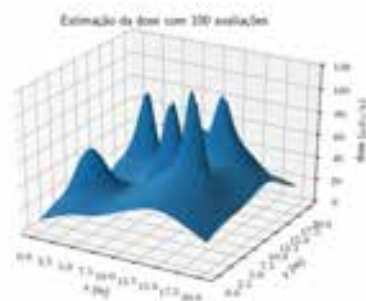
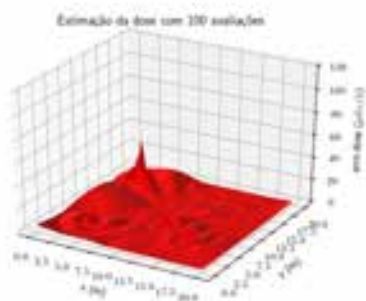
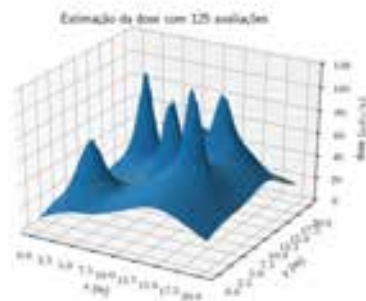
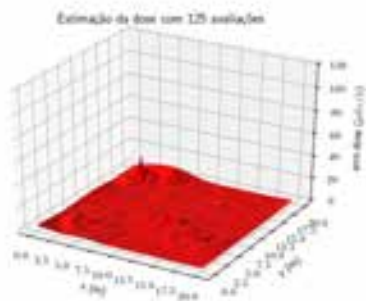
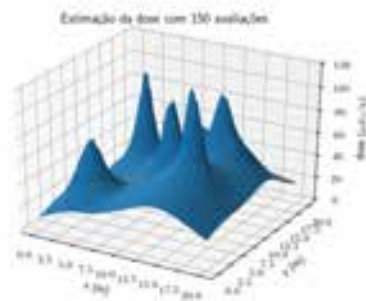
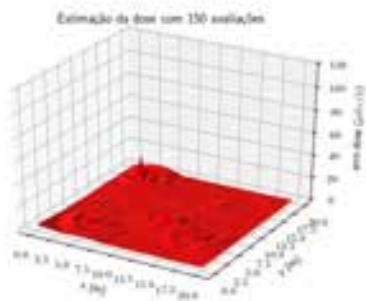
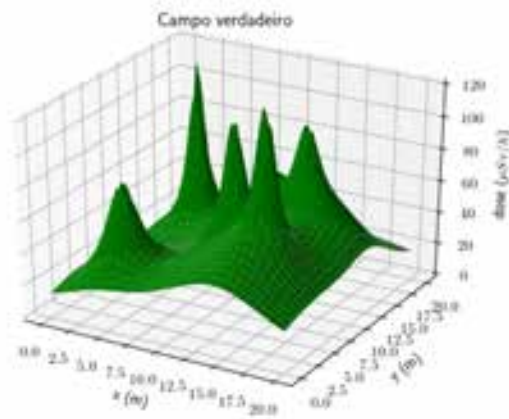


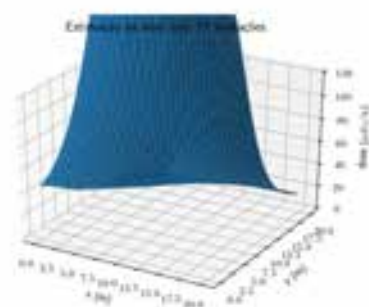
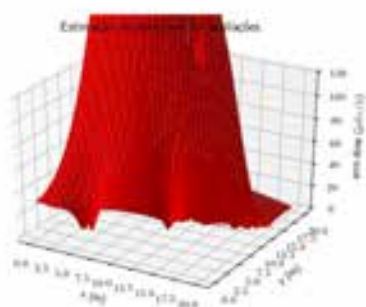
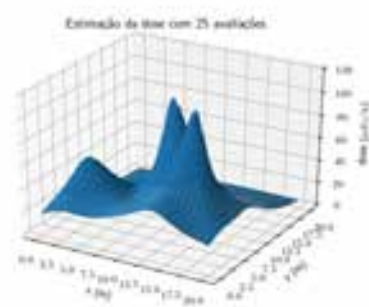
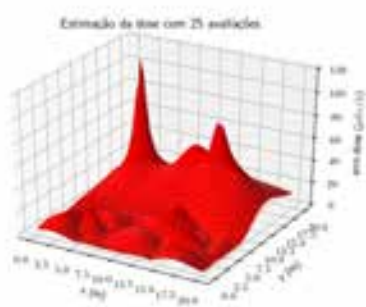
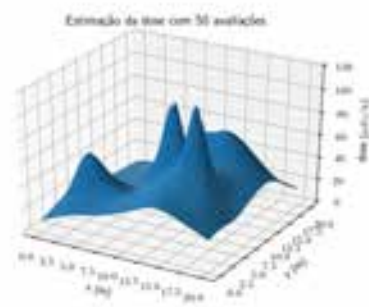
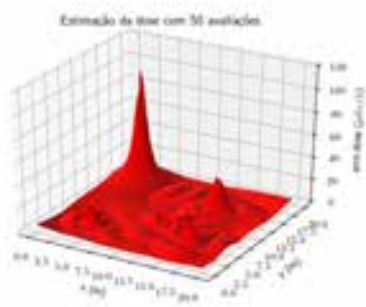
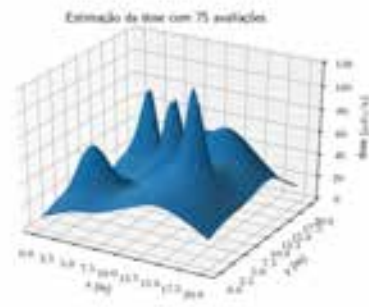
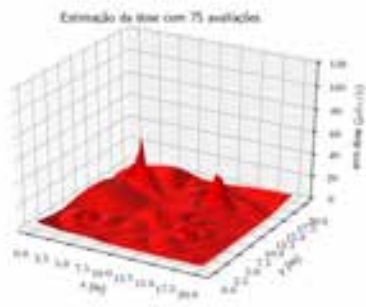
ARS 8P Wp4 Lp8



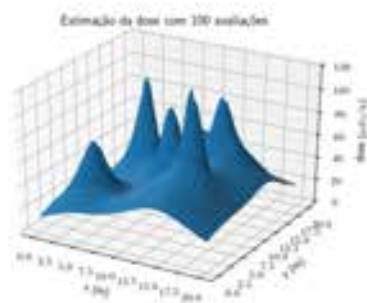
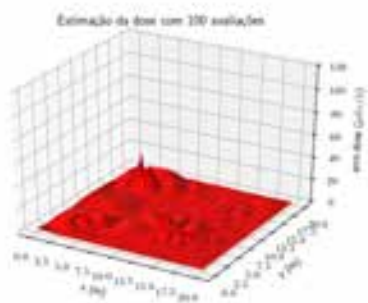
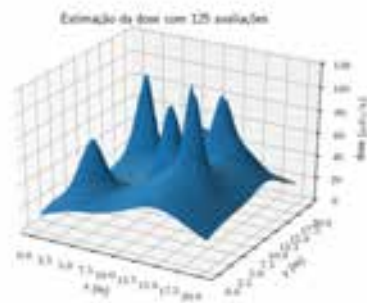
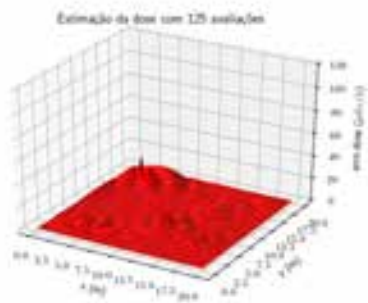
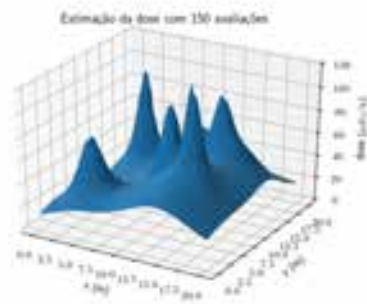
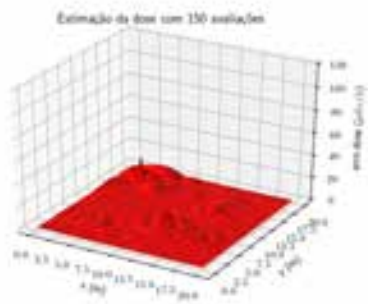
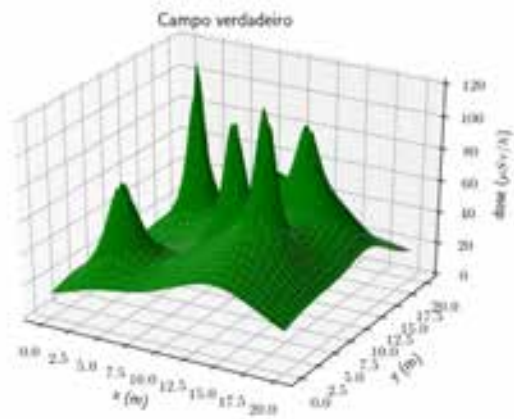


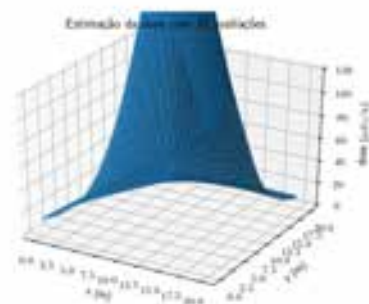
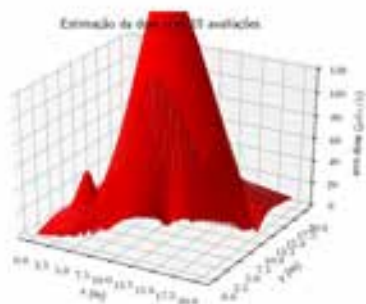
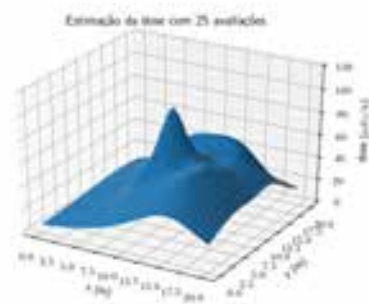
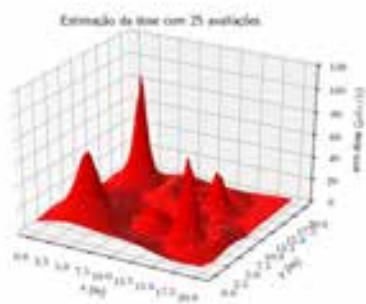
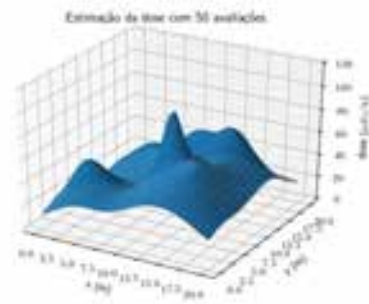
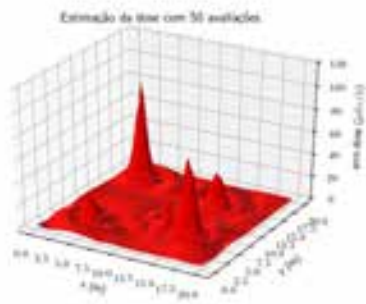
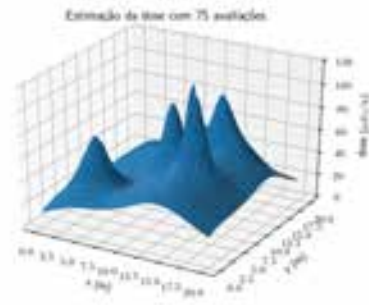
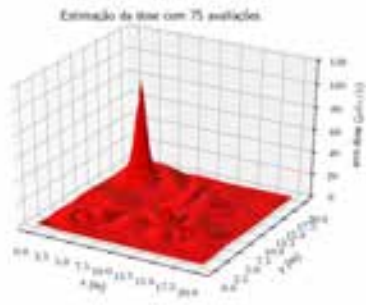
ARS 8P Wp8 Lp4





ARS 8P Wp8 Lp8





ARS 8P SP

