



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Interações Humanas em Semiautônomos de Sistemas de Aeronaves Remotamente Pilotadas (SARP)

Richard Siqueira Miranda

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientador
Prof. Dr. Edison Ishikawa

Brasília
2022

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

SM672ii Siqueira Miranda, Richard
Interações Humanas em Semiautônomos de Sistemas de
Aeronaves Remotamente Pilotadas (SARP) / Richard Siqueira
Miranda; orientador Edison Ishikawa; co-orientador Richard
Siqueira Miranda. -- Brasília, 2022.
102 p.

Dissertação (Mestrado - Mestrado Profissional em
Computação Aplicada) -- Universidade de Brasília, 2022.

1. Sistemas de Aeronaves Remotamente Pilotadas (SARP) .
2. Modelo de Software Crenças Desejos e Intenções. 3. Cadeias
de Markov. I. Ishikawa, Edison , orient. II. Siqueira
Miranda, Richard, co-orient. III. Título.



Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Interações Humanas em Semiautônomos de Sistemas de Aeronaves Remotamente Pilotadas (SARP)

Richard Siqueira Miranda

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof. Dr. Edison Ishikawa (Orientador)
PPCA/UnB

Paulo Fernando Ferreira Rosa Marcelo Antonio Marotta
IME PPCA/CIC

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 26 de Abril de 2022

Dedicatória

Um sentimento especial de gratidão a meus amorosos pais (in memoriam Alierth e Leny), cujas palavras de incentivo e insistência na tenacidade ainda ecoam em meus ouvidos.

Dedico este trabalho e agradeço especialmente à minha maravilhosa esposa Derlinéa e aos filhos Rafael e Leonardo por estarem comigo durante todo o programa de mestrado. Vocês foram meus melhores líderes de torcida.

Aos meus familiares que residem Espírito Santo, mesmo estando longe, nunca saíram do meu lado e são muito especiais.

Agradecimentos

À Deus, por tudo que nos tem proporcionado na vida. A minha esposa Derlinéa e os meus filhos Rafael e Leonardo pelos préstimos e compreensão nos momentos em que estive ausente. Aos meus pais, irmãos e sobrinhos, pelos incentivos.

Ao meu orientador, Prof. Dr. Edison Ishikawa, pela dedicação, confiança e por todo o apoio prestado ao longo deste trabalho, suas orientações foram fundamentais para o transcurso de todas as dificuldades encontradas. Aos Professores Doutores Marcelo Antonio Marotta, Paulo Fernando Ferreira Rosa e Edson Pignaton por valiosas sugestões oferecidas na evolução da Dissertação. A Prof. Dra. Aletéia Patrícia Favacho de Araújo por me proporcionar o primeiro contato com a UNB e também todo amparo prestado.

A Universidade de Brasília (UNB) e toda a equipe do Programa de Pós Graduação em Computação Aplicada (PPCA) pelas oportunidades oferecidas. Aos professores do PPCA, pelo conhecimento e experiência transmitidas que me ajudaram muito na adaptação e retorno a vida acadêmica. Aos colegas das turmas do PPCA 2018, 2019 e 2020, pela camaradagem, orientação, compreensão, ajuda e por todos os momentos de dificuldades que passamos juntos.

A Diretoria de Controle de Efetivos e Movimentações (DCEM) pelo auxílio e compreensão que me ajudaram a realizar esse sonho. Ao Estado Maior do Exército (EME), o Comando de Operações Terrestres (COTER) e a Academia Militar das Agulhas Negras (AMAN) pelas contribuições para a pesquisa. A Empresa XMobots Aeroespacial Drones para Defesa e Agricultura por fornecer informações vitais para a Pesquisa. A Universidade Federal do Espírito Santo (UFES) pelas valiosas orientações.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Os Sistemas Robóticos contemporâneos estão cada vez mais presentes na vida cotidiana. No cenário militar, são cada vez mais utilizados, pois por meio da operação remota é possível realizar diversos tipos de tarefas sem colocar em risco vidas humanas. Nesse contexto, a participação de sistemas semiautônomos tem se tornado expressivamente útil e conveniente, tendo em vista que os sistemas robóticos podem operar em condições adversas e com eventual interferência de sinais de comando. Uma forma amplamente utilizada de desenvolver agentes de software para decisões autônomas e semiautônomas são os modelos de arquitetura de Crenças, Desejos e Intenções (BDI). Esse tipo de agente possui os meios para analisar o cenário em que se encontra e, a partir das informações coletadas e dos desejos estabelecidos, é capaz de elaborar planos de tarefas para atingir seu objetivo. O que se percebe nos agentes baseados nesta arquitetura é a obediência total e irrestrita ao operador, o que pode interferir no cumprimento das tarefas e planos estabelecidos, o que se supõe afetar diretamente o desempenho geral do sistema robótico. A hipótese desta pesquisa é que os agentes de modelagem sob a estrutura do BDI podem trazer benefícios práticos se mapear a interação humana, aprimorando o desenvolvimento de novos sistemas. O agente de software semiautônomo, ao interpretar e avaliar os comandos do operador antes de obedecê-los, pode evitar ações equivocadas que apresentam o risco de comprometer a missão a que foi atribuído. Esta pesquisa tem como objetivo propor um modelo de agente baseado na estrutura do BDI que avalia os comandos do operador e permite que decida se deseja executá-los. Para justificar a proposta, foram realizadas simulações utilizando a Cadeia de Markov por meio da ferramenta PRISM, em agentes semiautônomos nos quais a Interação Humana (IH) é realizada sem restrições. Foram observados que o modelo BDI modificado proposto aumentou as chances do SARP sobreviver ou cumprir a missão. Como exemplo ilustrativo, foi considerado um cenário de reconhecimento aéreo por meio do SARP com alguma autonomia e recebendo comandos humanos remotos.

Palavras-chave: SARP, BDI e Cadeia de Markov.

Abstract

Contemporary Robotic Systems are increasingly present in everyday life. In the military scenario, they are increasingly used, because through remote operation it is possible to perform various types of tasks without putting human lives at risk. In this context, the participation of semi-autonomous systems has become significantly useful and convenient, considering that robotic systems can operate in adverse conditions and with possible interference from command signals. A widely used way of developing software agents for autonomous and semi-autonomous decisions is the Beliefs, Desires, and Intentions (BDI) architectural models. This type of agent has the means to analyze the scenario in which it finds itself and, based on the information collected and the established desires, it is capable of drawing up task plans to achieve its objective. What is perceived in agents based on this architecture is total and unrestricted obedience to the operator, which can interfere with the fulfillment of established tasks and plans, which is supposed to directly affect the overall performance of the robotic system. The hypothesis of this research is that modeling agents under the framework of BDI can bring practical benefits if they map human interaction, improving the development of new systems. The semi-autonomous software agent, by interpreting and evaluating the operator's commands before obeying them, can avoid misguided actions that present the risk of compromising the mission to which it was assigned. This research aims to propose an agent model based on the BDI structure that evaluates the operator's commands and allows it to decide whether to execute them. To justify the proposal, simulations were performed using the Markov Chain using the PRISM tool, in semi-autonomous agents in which human interaction is performed without restrictions. It was observed that the proposed modified BDI model increased the chances of UAV surviving or mission accomplishment. As an illustrative example, an aerial reconnaissance scenario was considered through UAV with some autonomy and receiving remote human commands.

Keywords: UAV, BDI and Markov Chain.

Sumário

1	Introdução	1
1.1	Definição do Problema	3
1.2	Justificativa	3
1.3	Objetivo	4
1.3.1	Objetivos Específicos	4
1.4	Metodologia	5
1.5	Estrutura do Trabalho	6
2	Trabalhos Relacionados	7
2.1	Modelagem BDI	7
2.1.1	BDI	7
2.1.2	Componentes do BDI	8
2.1.3	BDI Estendido	10
2.2	Sistemas Semiautônomos	13
2.3	Experimentos Estocásticos	14
3	Fundamentação Teórica	18
3.1	Sistemas Multiagentes	18
3.1.1	Comportamento do Sistema Multiagente	19
3.2	Agentes Inteligentes	20
3.2.1	Agentes Baseados em Metas	21
3.2.2	Agente Único e Multiagente	23
3.3	Raciocínio Prático	24
3.3.1	Ambiente em Mudança	25
3.4	Modelo de Software	26
3.5	Modelo de Interação Humana	27
3.5.1	<i>Human-In-The-Loop</i> (HITL)	27
3.5.2	<i>Human-On-The-Loop</i> (HOTL)	28
3.5.3	<i>Human-Out-Of-The-Loop</i> (HOOTL)	29

3.6	Ferramenta PRISM 4.7	30
3.6.1	Processo de Decisão de Markov	31
3.7	Ferramenta PRISM-GAMES 3.0	32
3.7.1	<i>Turn-Based Stochastic Multi-Player Games</i> (SMG)	34
3.8	Sistemas Não Tripulados	35
4	Proposta do Trabalho	38
4.1	Interações Humanas e BDI	38
4.1.1	A Ponderação do BDI e IH	39
4.1.2	Ações do BDI	39
4.2	Elaboração do Modelo Conceitual	40
4.3	Proposta	41
4.3.1	Comando Externo do Operador	42
4.3.2	Comando Externo Enumerado	42
4.3.3	Índice de Confiança ao Comando	42
4.3.4	Índice de Instinto ao Comando	44
4.3.5	Grau de Compatibilidade ao Comando Externo	44
4.3.6	Grau de Obediência	45
4.4	Funcionamento do BDI Proposto	46
5	Experimentos	47
5.1	Aderência dos Experimentos Estocásticos ao Modelo BDI	47
5.2	Parâmetros Utilizados	48
5.3	Equipamento Referenciado	49
5.4	Considerações Sobre os Cenários Utilizados	51
5.5	Experimentos Envolvendo Perda do SARP	52
5.5.1	Cenário Utilizado	52
5.5.2	Plano de Voo	53
5.5.3	Perda do SARP em Função do Risco	55
5.5.4	Distância Percorrida em Função do Risco e Perda do SARP	57
5.6	Experimentos Considerando a Interação Humana	57
5.6.1	Resultados Sob Interação Humana	60
5.7	Resultados Sob Interação Humana e Grau de Obediência	62
5.8	Distância e Risco	65
5.9	Discussão	66
6	Conclusão	68
6.1	Limitações	69

6.2 Trabalhos Futuros	70
Referências	71
Apêndice	76
A Código PRISM - Cenário Contendo Perda do SARP e Distância Per- corrida	77
B Código PRISM - Cenário Contendo Interações Humanas	80
C Código PRISM-GAMES - Cenário Contendo Grau de Obediência	84

Lista de Figuras

2.1	Modelo BDI Baseado em Agentes.	10
2.2	Modelo BDI proposto por Lee.	12
2.3	Mapa do Percurso Aéreo.	15
3.1	Característica do Agente no Ambiente.	19
3.2	Modelo de Agente Baseado em Metas.	22
3.3	<i>Printscreen</i> da Ferramenta PRISM 4.7.	31
3.4	<i>Printscreen</i> da Ferramenta PRISM-GAMES 3.0.	33
3.5	Localização de Rotor no SARP.	35
3.6	Componentes Internos de um SARP.	37
4.1	Proposta de BDI Contendo o Comando Externo do Operador.	41
4.2	Executor de Decisão Contendo o Grau de Obediência.	45
5.1	Aderência dos Experimentos ao Modelo BDI.	48
5.2	SARP NAURU 500C Realizando Manobra de Decolagem Vertical (VTOL).	50
5.3	Cenário Envolvendo Perda do SARP (<i>Sink</i>).	53
5.4	Taxa dos SARP que Chegaram ao Destino.	55
5.5	Distância Percorrida dos SARP que Chegaram ao Destino.	57
5.6	Distância Percorrida dos SARP que não Chegaram ao Destino.	58
5.7	Cenário do Percurso do SARP.	59
5.8	Distância Percorrida em Função da Interação Humana e Variação do Cenário.	61
5.9	Distância Percorrida em Função da Interação Humana e Grau de Obediência.	64
5.10	Parâmetros que Influenciam o Plano de Voo Normal.	65
5.11	Distâncias Percorridas Considerando a Probabilidade de Ocorrência de TO e GO.	66

Lista de Tabelas

3.1	Categorias de SARP Utilizados pelo EB.	18
3.2	Exemplos de Tamanho de SARP.	36
5.1	Descrição das Distâncias e Probabilidades Iniciais das Arestas.	54
5.2	Descrições e Distâncias das arestas.	60

Lista de Abreviaturas e Siglas

ANYLOGIC Ferramentas de software de modelagem de simulação.

AT Tipo de interação humana **atrasar**.

BBN Rede de Crenças Bayesianas.

BDI Modelo de Software Crenças, Desejos e Intenções.

BPM Gestão de Processos de Negócio.

BPMN Notação de Gestão de Processos de Negócio.

CPU Unidade Central de Processamento.

CSG Jogos Multijogador Simultâneos Estocásticos.

DE Tipo de interação humana **desviar**.

DFT Teoria de Campo da Decisão.

DSR Pesquisa Baseada em Design Science.

DTMC Cadeia de Markov em Tempo Discreto.

EB Exército Brasileiro.

GIS Sistema de Informações Geográficas.

GO Grau de Obediência.

HITL Humano no Circuito.

HOOTL Humano fora do Loop.

HOTL Humano no Loop.

IA Inteligência Artificial.

IC Índice de Confiança.

ICC Índice de Confiança ao Comando.

IH Interação Humana.

IHP Interação Humana Ponderada.

II Índice de Instinto.

IIC Índice de Instinto ao Comando.

MAS Sistemas Multiagentes.

MDP Processo de Decisão de Markov.

MMC Método de Monte Carlo.

NAURU 1000C Modelo do SARP da XMobots adquirido pelo EB em 2021.

NAURU 500C Modelo de SARP da XMobots.

PIH Probabilidade de Interação Humana.

PRISM Ferramenta Verificadora de Modelos Probabilístico.

PRISM-GAMES Extensão da ferramenta PRISM.

SARP Sistemas de Aeronaves Remotamente Pilotadas.

SE Seguir em Frente com o plano de voo ou Ausência de Interação Humana.

Sink Representa a perda do SARP.

SMG Jogos Multijogador Estocásticos Baseados em Turnos.

TIH Tipo de Interação Humana.

TO Tipo de interação humana AT, VO ou DE.

TPTG Jogos Cronometrados Probabilísticos Baseados em Turnos.

UAV Veículo Aéreo não Tripulado.

UD Unidade de Distância.

UGV Veículo Terrestre não Tripulado.

UML Linguagem de Modelagem Unificada.

VANT Veículo Aéreo Não Tripulado.

VO Tipo de interação humana **voltar**.

VTOL Decolagem e Aterrissagem Verticais.

XMobots Empresa Fabricante de SARP.

Xplanner Sistema de Elaboração de Planos de Voo da XMobots.

Capítulo 1

Introdução

Atualmente a tecnologia robótica tem desempenhado um papel cada vez mais importante na indústria bélica. O seu uso tem por objetivo alcançar a superioridade na capacidade de combate. Nos últimos anos, os robôs têm sido utilizados cada vez mais no lugar dos humanos e, em muitos casos, realizam missões de forma mais eficaz e segura, não expondo assim ao perigo as vidas humanas [1]. São inúmeras as vantagens oferecidas pelos robôs, pois é observado com frequência que as unidades militares estão optando pelo uso desta tecnologia.

Existem diversas variantes e aplicações dos sistemas robóticos, como é o caso dos dispositivos terrestres blindados [2], muitas vezes chamados de drones blindados. Há modelos que chegam a um peso de 10 toneladas e possuem a capacidade de levar lançadores de mísseis, canhões automáticos e metralhadoras pesadas. Estes dispositivos, por não serem tripulados, podem permanecer dias ou semanas em missão, além de serem imunes aos agentes químicos, biológicos, radiológicos e nucleares [3].

Os sistemas robóticos podem fazer a diferença no equilíbrio de forças. É possível citar o combate ocorrido no ano de 2021 entre Armênia e Azerbaijão, em que um número expressivo de carros de combate e outros veículos blindados armênios foram destruídos ou avariados por ataques de sistemas semelhantes aos SARP, como os modelos Bayraktar TB2 de fabricação Turca. Tais drones são relativamente baratos, pequenos e silenciosos. Existem também drones do tipo Kamikaze, a exemplo do IAI Harop Israelense, que, de forma autônoma, pode sobrevoar uma determinada área por horas em busca de um alvo [4].

Os Sistemas de Aeronaves Remotamente Pilotadas (SARP) são amplamente empregados em missões militares, assumindo missões de monitoramento do inimigo, de tráfego e controle, comunicações sem fio, dentre outras. Além disso, o uso dos SARP também têm como objetivo aumentar a superioridade na batalha através das capacidades que são, muitas vezes, exclusivas dos SARP, como é o caso da não exposição de vidas humanas,

autonomia de voo, capacidade de manobra, dentre outras.

A utilização de tecnologias que prometem trazer mais autonomia aos sistemas robóticos não tripulados tem sido uma opção interessante atualmente, pelas diversas vantagens oferecidas, como é o caso da redução da presença humana na operação. Além disso, há tomadas de decisões que, dada a dinâmica de uma missão, os seres humanos podem ser incapazes de tomar.

Na busca de tornar os SARP mais eficientes, tem havido um esforço para torná-los mais autônomos. Existem diversas tarefas que requerem precisão e alta velocidade de resposta e que, na maioria das vezes, não podem ser realizadas por humanos. Nestes casos, é altamente recomendável se esforçar para alcançar a independência preferencialmente completa quando uma operação não supervisionada é considerada essencial.

Exemplos de automação podem ser vistos em diferentes momentos de nossas vidas, como por exemplo, em sistemas de fabricação e produção automatizados, em carros semiautônomos, em sistemas de segurança e vigilância e em eletrodomésticos. Embora as máquinas possam executar algumas tarefas com alta eficiência e precisão, não é verdade que a autonomia total elimina a necessidade da supervisão humana, tendo em vista que vários sistemas autônomos não tripulados, os *Unmanned Aerial Vehicle* (UAV) e *Unmanned Ground Vehicles* (UGV), veículos subaquáticos e outros tipos de robôs, dependem da perícia humana em funções cruciais [5].

Os equipamentos SARP categoria 2 que foram adquiridos pelo EB no ano de 2021 possuem características semiautônomas que permitem manter o voo mesmo com a perda de contato com o operador. No contexto em que o SARP é considerado como um agente, e que possua características de sistema autônomo. Estes possuem a capacidade de perceber o ambiente por meio dos seus diversos sensores e atuam ao longo do tempo através de seus atuadores.

O modelo de software baseado em crenças, desejos e intenções (BDI) estabelece um paradigma que permite desenvolver agentes inteligentes baseados em Inteligência Artificial (IA), caracterizados pela implementação de virtudes humanas, tais como crenças, desejos e intenções conforme Bratman em [6]. Os sistemas de agentes baseados em BDI podem ser aplicáveis ao contexto de sistemas autônomos ou semiautônomos dos SARP para resolverem problemas complexos relacionados à operação autônoma. No contexto de Sistemas Multiagentes (MAS), os agentes de Software e Humanos podem agir em conjunto para atingir um objetivo em comum.

O desenvolvimento cada vez maior da IA, aumenta a possibilidade de que as Forças Militares passem a delegar a esses sistemas cada vez mais autonomia nas decisões. O problema é que a Inteligência Artificial (IA) pode ser iludida por meio de ataques cibernéticos, mecanismos para burlar os sensores desses sistemas com parâmetros errados, falhas

em seus sistemas de IA, falhas em seus sistemas de comunicações, defeitos de hardware, dentre outros [7]. O que motiva a necessidade da ação humana na cadeia de comando para assegurar o cumprimento da missão.

É neste contexto que surge a necessidade de estudos que lidam com o tratamento que deve ser dado à Interação Humana (IH). Este trabalho propõe um modelo de IH em sistemas multiagentes BDI. Os estudos iniciais realizados mostram que é possível compatibilizar a IH em sistemas semiautônomos aumentando a possibilidade de cumprimento da missão.

1.1 Definição do Problema

A utilização de dispositivos não tripulados está em constante crescimento. A frota é significativa em diversos países e tende a crescer ainda mais, inclusive no Brasil, motivado pelos inúmeros benefícios comprovados que este tipo de dispositivo apresentou nos últimos anos.

No entanto, os dispositivos não tripulados, por razões operacionais, podem necessitar da IH para o cumprimento de seus objetivos. Por isso a necessidade de estudos relacionados à área.

Esta pesquisa será desenvolvida para reforçar os estudos a respeito desses dispositivos. Com o intuito de abordar a autonomia relacionada à IH, principalmente no contexto BDI e a sua influência nos sistemas semiautônomos.

A utilização da abordagem estocástica realizada por esta pesquisa, poderá motivar novas formas de simulações. E o fato de haver a chance da IH afetar o desempenho da missão, já justifica a necessidade de estudar o assunto e buscar melhorias.

Em um sistema semiautônomo de IA, a IH pode influenciar o cumprimento de planos. Um modelo de processo de decisão BDI para este tipo de sistema, que permita considerar a IH, pode aprimorar as técnicas de desenvolvimento de sistemas para esta finalidade, ao permitir considerar a interferência da IH e em como ela impacta nos resultados esperados para determinados planos.

1.2 Justificativa

Estudos relacionados a sistemas de Inteligência Artificial (IA) estão em constante evolução, como por exemplo, sistemas baseados em aprendizado de máquina, redes neurais, modelos de software BDI, dentre outros. Tais sistemas estão cada vez mais presentes em diversas aplicações tanto no emprego civil quanto no militar. E com isso, o uso dos sis-

temas de IA em sistemas semiautônomos, estabelece a necessidade de estudos que levem em consideração a IH.

Em alguns casos, a IH pode colocar em risco os objetivos da missão. Pois a decisão pode estar prejudicada por fatores, como a interferências de comunicação, velocidade de resposta, falta da consciência situacional, proficiência do operador, dentre outros. No contexto de BDI, existe a necessidade da construção de um modelo que possibilita ponderar a IH, quando este comprometer a missão.

Foram realizadas diversas pesquisas em várias bases de dados onde foram obtidos diversos artigos correlatos. No entanto, não foram encontrados trabalhos que continham modelagens que considerasse a IH em modelos BDI. Com isso, esta abordagem pode ser uma importante lacuna de pesquisa.

A Hipótese deste trabalho é a elaboração da proposta de modelo de software baseado em BDI para sistemas SARP semiautônomos que considera a IH aprimora as técnicas de desenvolvimento de novos sistemas de IA.

1.3 Objetivo

Esta pesquisa propõe um modelo conceitual de agente BDI misto que inclui a IH e que permita analisá-la antes de executá-la, seguindo um Grau de Obediência (GO) pré-estabelecido pelo BDI. O GO refere-se a probabilidade de obediência do BDI ao comando externo do operador que está detalhado na proposta deste trabalho, no Capítulo 3.

Para calcular a influência da IH no BDI, foram desenvolvidos experimentos estocásticos com base na literatura, em cenários hipotéticos, contendo movimentos e parâmetros de SARP extraídos da literatura. Além da utilização do GO do BDI proposto e a aplicação de Tipos de Interações Humanas diferentes.

Os experimentos estocásticos consideraram alguns dos principais componentes da extensão de modelo BDI proposto e os parâmetros dos equipamentos SARP são dos equipamentos adquiridos recentemente pelo Exército Brasileiro (EB). Os experimentos estocásticos permitirão realizar levantamentos quantitativos da performance da missão através da distância percorrida e perda do SARP.

1.3.1 Objetivos Específicos

Para atingir o objetivo geral, o trabalho foi dividido nos seguintes objetivos específicos:

- Realizar investigação de trabalhos correlatos que contribuam para a elaboração de modelos baseados em IH em sistemas inteligentes;
- Desenvolver um modelo conceitual BDI que permita ponderar a IH através do GO;

- Modelar cenários para permitir a validação do modelo BDI;
- Realizar experimentos estocásticos em cenários que considere a perda do SARP;
- Realizar experimentos estocásticos para se obter o levantamento quantitativo do desempenho da missão com o uso da IH;
- Realizar experimentos estocásticos com o uso do GO do BDI.

1.4 Metodologia

A metodologia utilizada para atingir os objetivos foi baseada no Método *Design Science Research* (DSR). Conforme descrito por Lacerda *et al.* em [8], esta metodologia promove a criação de artefatos a fim de resolver problemas da vida real. Esta pesquisa seguiu as seguintes etapas:

- **Identificação e Conscientização do problema:** Encontram-se no Capítulo 1;
- **Trabalhos Relacionados:** Foram utilizados manuais, livros, dissertações e artigos científicos contendo as principais palavras-chaves, tais como: BDI, SARP, HITL, human interaction, plan e robot. As bases de dados utilizadas foram: IEEE, SCOPUS, ACM Digital Library, dentre outros. Objetivou-se buscar informações relevantes acerca do tema e mais recentes. Os trabalhos relacionados encontram-se Capítulos 3 e referenciada ao longo da Dissertação;
- **Desenvolvimento do Modelo:** Desenvolvido a proposta de modelo de software BDI no Capítulo 4 para a solução do problema de pesquisa. E nesta construção, foram utilizadas as ferramentas apropriadas, obedecendo aos critérios obtidos da literatura;
- **Avaliação do Modelo:** Realizadas implementações e verificação dos resultados quantitativos dos diversos experimentos, utilizando-se para isso ferramentas apropriadas e alguns dos componentes do modelo BDI proposto;
- **Explicitação das Aprendizagens:** Os resultados do presente trabalho foram apresentados nesta Dissertação;
- **Comunicação dos Resultados:** Os resultados do presente trabalho foram aceitos na Revista Militar Científica da AMAN, que é uma referência importante no contexto de mestrado profissional deste tema. O assunto também foi submetido às importantes conferências, tais como o HRI 2022 e RSS 2022, que forneceram *feedbacks* para melhoria desta pesquisa.

1.5 Estrutura do Trabalho

Este trabalho está estruturado em 6 capítulos, conforme descrito abaixo:

- **Capítulo 1:** Mostra o contexto, introdução, motivações, metodologia, problema da pesquisa, dentre outros;
- **Capítulo 2:** Expõe os trabalhos relacionados;
- **Capítulo 3:** Exibe a fundamentação teórica;
- **Capítulo 4:** Descreve a proposta BDI que possibilita considerar a IH, apresentando o modelo de software BDI além dos seus componentes;
- **Capítulo 5:** Contém a descrição dos experimentos e resultados desta pesquisa;
- **Capítulo 6:** Descreve a conclusão, as limitações e trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Este capítulo trata dos principais trabalhos que ajudaram a direcionar esta dissertação. Serão abordados a modelagem BDI, sistemas semiautônomos e experimentos estocásticos.

2.1 Modelagem BDI

Os trabalhos de Bratman em [6], e Wooldridge em [9] apresentaram os conceitos iniciais de BDI. Os trabalhos de Kim *et al.* em [10], Ujjwal *et al.* em [11], Rao e Georgeff *et al.* em [12], Archibald *et al.* em [13], Lee *et al.* em [14] e Noorunnisa *et al.* em [15] propuseram avanços significativos em relação aos trabalhos de Bratman e Wooldridge, sendo considerado como estado da arte.

Apesar das contribuições de várias pesquisas acerca do BDI, não foram encontrados modelagens que considere a IH na construção do modelo BDI. A maioria das abordagens focaram na solução de determinados problemas específicos.

Estes trabalhos serviram de base para a construção da modelagem do BDI proposto. A proposta contida nesta dissertação, deu continuidade ao trabalho de Lee *et al.* em [14] onde foram incluídos componentes, fluxos e definições.

2.1.1 BDI

O modelo BDI, em inglês *Belief-Desire-Intention* ou Crenças-Desejos-Intenções, são conceitos inspirados na tradição filosófica relacionados a compreensão do raciocínio prático e que foram propostos inicialmente por Bratman em [6]. Esse tipo de raciocínio está relacionado em como o processo deve decidir e quais as ações a executar para atender um determinado objetivo. O BDI é um modelo que imita o comportamento humano e ganhou grande atenção nos últimos anos na academia e na indústria [10]. Segundo Ujjwal *et al.* em [11], o BDI é uma das principais abordagens para construir sistemas multiagentes in-

teligentes. Pois é um modelo inspirado no raciocínio humano baseado em três conceitos: crenças, Desejo e intenções.

O modelo tem sua origem na teoria do raciocínio prático humano de Bratman [6] *apud* [15]. As ideias de Bratman foram formalizadas pela primeira vez por Rao e Georgeff *et al.* em [12], que posteriormente propuseram uma arquitetura abstrata na qual as crenças, os desejos e intenções eram explicitamente representadas como estruturas de dados globais e o comportamento do agente era conduzido por eventos.

Para Simari *et al.* em [16], os modelos BDI foram inspirados pela tradição filosófica de compreender o raciocínio humano e, inicialmente, foram propostos por Michael Bratman em [6]. Esse tipo de raciocínio pode ser descrito como o processo de decisão de ações a realizar com o intuito de atingir determinado objetivo.

Conforme Wooldridge em [9], o modelo de arquitetura BDI é uma teoria bastante compreendida que está altamente alinhada com a ação racional dos seres humanos. Tal modelo pode ser implementado em uma série de aplicações complexas de campo e a teoria deve ser formalizada em uma família de lógicas BDI.

2.1.2 Componentes do BDI

O crescimento da utilização do modelo BDI avançou porque os modelos teóricos de decisão eram até então computacionalmente intratáveis para serem usados na prática [16]. Isso levou a uma percepção de que os modelos são de alguma forma lançados em complemento aos já existentes, de modo que a adoção do modelo BDI apenas se confirma caso os modelos teóricos de decisão sejam de alguma forma impraticáveis, tendo em vista que o desempenho dependerá das abordagens das implementações e heurísticas utilizadas.

Segundo Wooldridge em [9], o processo de raciocínio prático pode ser dividido nos seguintes principais componentes básicos:

- (B) um conjunto de crenças atuais que representam as informações que o agente tem em seu poder atualmente, incluindo seu estado interno e o ambiente;
- (D) um conjunto de Desejos que implicará nas possíveis direções das ações do agente;
- (I) um conjunto de intenções que representa o foco atual do agente, ou seja, aqueles estados com os quais ele está comprometido a realizar;

Conforme Wooldridge em [9], pode-se incluir as Funções:

- Função de Revisão de Crença: cuja a entrada é na percepção das crenças atuais do agente e, com base nisso, pode determinar um novo conjunto de crenças;

- A Função de Geração de Opções: que determina as alternativas possíveis disponíveis para o agente com base nas crenças atuais sobre o ambiente e suas intenções atuais. Esta Função representa uma parte do componente de Deliberação do agente e, além disso, mapeia um conjunto de crenças juntamente com o conjunto de intenções e desejos;
- A Função de Filtro: que está contido no componente de Deliberação do agente. Esta Deliberação determina um conjunto consistente de intenções com base em suas crenças, desejos e intenções atuais;
- A Função de Plano: que implementa o raciocínio meios-fins. Esta função mapeia o conjunto atual de crenças e intenções para a sequência de ações.

Essas conceituações procuram atender fielmente a teoria de Bratman. Para o trato computacional, Noorunnisa *et al.* em [15] propôs as seguintes mudanças em termos de representação:

- Apenas as crenças sobre o estado atual do mundo são representadas explicitamente;
- Os desejos são chamados de metas, que são representados como eventos. As metas têm apenas uma representação transitória, atuando como gatilhos para invocações de plano;
- As informações sobre os meios de alcançar certos estados (desejos) futuros são representadas processualmente como planos;
- As intenções são representadas implicitamente pela coleção de planos atualmente ativos.

Para Archibald *et al.* em [13], a crença é um dos componentes do BDI e corresponde ao que o agente acredita que existe no ambiente. Este, por sua vez, alimenta os desejos com um conjunto de eventos externos aos quais o agente pode responder. Para responder a esses eventos, o agente seleciona um plano apropriado (de acordo com suas crenças) da biblioteca de planos pré-definidos e se compromete com o plano selecionado, transformando-o em uma nova intenção.

Foi dito por Noorunnisa *et al.* em [15], que embora uma meta específica possa ser alcançada por meio de vários planos, um agente deve se comprometer a selecionar um único plano para sua realização. No entanto, se esse plano falhar, os meios para atingir a meta podem ser reconsiderados e um agente pode perseguir vários objetivos simultaneamente.

A Figura 2.1 apresenta o modelo BDI adaptado do artigo de Sameera *et al.* em [17] que contém quatro estruturas de dados. As crenças correspondem às informações que o agente tem sobre o mundo, os desejos correspondem intuitivamente às tarefas que lhe

são atribuídas, as intenções representam os desejos que ele se comprometeu a atuar e os planos que contém as ações que devem ser realizadas para atender as intenções. O componente Interpretador é responsável por atualizar as estruturas de dados e executar as ações contidas no plano.

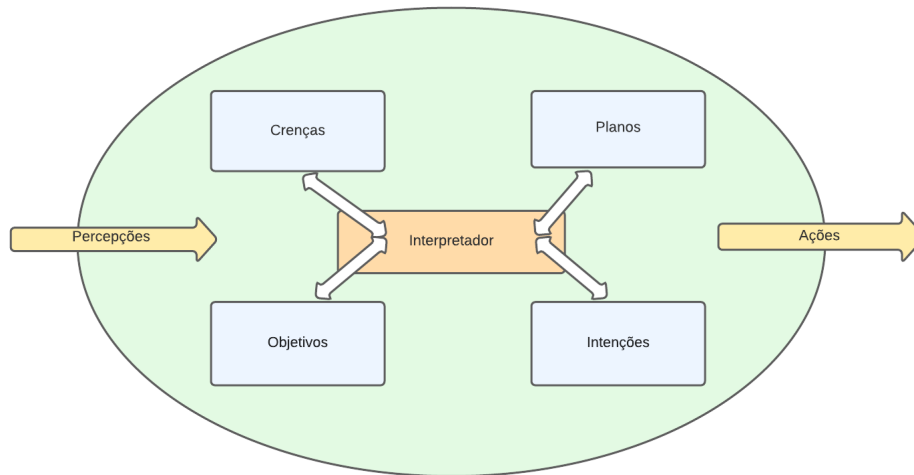


Figura 2.1: Modelo BDI Baseado em Agentes.

Segundo Kim *et al.* em [10], embora a estrutura estendida de BDI tenha sido aplicada com sucesso para imitar comportamentos humanos em vários cenários, as IH detalhadas não foram abordadas, apesar de que os comportamentos humanos podem ser alterados enquanto as pessoas interagem umas com as outras. Kim *et al.* disse também que o conceito central do paradigma BDI é originalmente baseado na psicologia popular, que permite o uso de uma linguagem de programação para descrever o raciocínio humano e as ações na vida cotidiana. Zhao *et al.* em [10] estenderam o modelo BDI original para incluir Submódulos detalhados, como Deliberador, Planejador Realtime e Executor de Decisão no módulo de Tomada de Decisão (intenção).

2.1.3 BDI Estendido

O artigo de Kim *et al.* em [18] mostrou duas extensões, a primeira envolve a evolução dinâmica ao longo do tempo de aprendizagem e esquecimento, e a segunda, as IH, competição, colaboração, compromisso, acomodação e evasão. Foram utilizados vários aplicativos para ilustrar o trabalho de cenários de evacuação de emergência durante um ataque a bomba, comportamentos de motoristas e pedestres e rede social cibernética.

Kim *et al.*, também apresentou detalhes da estrutura de BDI estendida integrando modelos e técnicas envolvendo as disciplinas engenharia, psicologia, IA e estatística, além

das principais funções cognitivas humanas e a discussão da estrutura de BDI estendida, que poderá lidar com o aprendizado de comportamentos e IH.

Dando continuidade ao trabalho anterior, Lee *et al.* em [14] propôs uma estrutura de modelagem de BDI semelhante à de Kim *et al.* em [18] integrada para a tomada de decisão e planejamento humano, e foram utilizados como conceitos, Rede de Crenças Bayesianas (BBN) e Teoria do Campo da Decisão (DFT). Para imitar os comportamentos humanos realistas, os atributos da estrutura do BDI e foram submetidos à engenharia reversa a partir dos experimentos HITL.

A modelagem proposta também utilizou comportamentos de evacuação de humanos em uma situação de ataque terrorista. O ambiente simulado foi implementado no software de simulação ANYLOGIC, em que cada agente executa sua função de percepção e seu plano em tempo real através do software, funções de gestão e de decisão. A simulação construída foi usada para testar o impacto dos dados demográficos de pessoas, número de policiais e no desempenho de evacuação: tempo médio de evacuação, porcentagem de vítimas [18].

A implementação permitiu que a arquitetura BDI estendida fosse usada para imitar os comportamentos humanos, mesmo em situações complexas. Desse modo, as técnicas propostas e o *framework* BDI estendido foram demonstrados por meio da simulação baseada em agente e em vários ambientes dinâmicos.

A simulação desenvolvida permitiu observar o comportamento da multidão nas diversas condições, tendo em vista o potencial de permitir a avaliação de diferentes políticas de evacuação e controle de danos, o que, por sua vez, permite a evacuação de multidão de modo mais eficaz [18].

No artigo [14], Lee incluiu no modelo BDI o módulo Tomada de Decisão e os Submódulos Deliberador, Planejador Realtime e Executor de Decisão. Além disso, o módulo Emocional contendo Índice de Confiança (IC) e Índice de Instinto (II) também foi anexado para representar naturezas humanas mais psicológicas. O Processador Perceptivo no módulo de crença traduz as informações sobre o meio ambiente e atualiza as suas crenças.

A Figura 2.2, apresenta a adaptação do modelo proposto por Lee *et al.* em [14] que por sua vez é uma extensão do modelo BDI proposto inicialmente por Bratman em [6]. Os retângulos representam os processos, as circunferências representam os elementos internos do BDI ou bases de conhecimento e as setas representam os fluxos de informações.

Com as crenças atuais, que são as memórias de curto prazo, atualiza o seu Índice de Instinto, pois quando o Índice de Instinto está abaixo de um limiar que é o modo normal, o BDI decide quais estados de coisas alcançar, chamados de desejos, por meio do Gerador de Desejos. O BDI então seleciona um desejo e gera intenções para realizar o desejo por meio do Deliberador [14].

A principal função do Planejador Realtime, que se encontra no Módulo de Tomada de Decisão, é gerar os planos alternativos com base nas crenças atuais, para alcançar as suas intenções. Um plano é uma sequência de ações, então, uma vez que um plano ótimo ou satisfatório é identificado, o Executor de Decisão no Módulo de Tomada de Decisão executa a série de tarefas especificadas no plano.

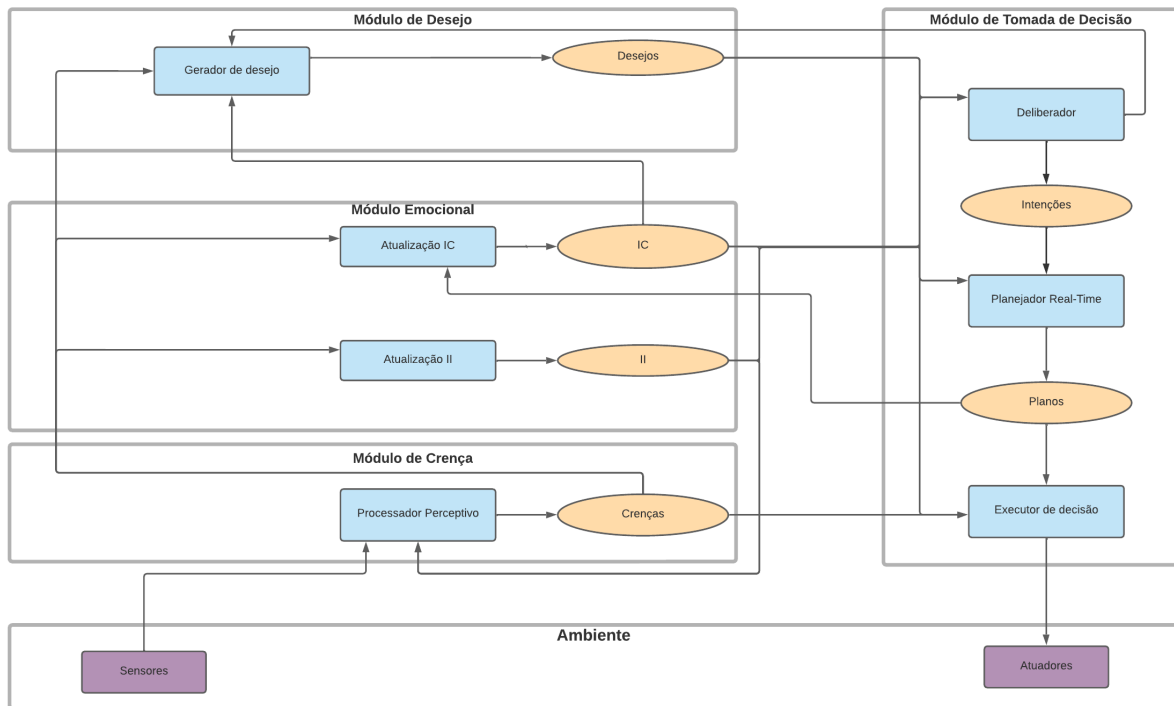


Figura 2.2: Modelo BDI proposto por Lee.

Por outro lado, se o II está acima de um limite, ou seja, no modo instintivo [14], o Executor de Decisão executa tarefas com base em seus Instintos recuperados de suas crenças, memória de longo prazo, sem envolver planejamento. Se o IC estiver acima de um limite, ou seja, no modo confiável, o executor de decisão executa todas as tarefas no plano.

O artigo de Chella *et al.* em [19] estuda a possibilidade de dar ao robô uma vida interior “humana”, como a capacidade de pensar sobre si mesmo e entender o que os outros membros da equipe estão fazendo. De acordo com o estudo, isto aumentaria a eficiência das interações confiáveis com os outros membros da equipe. O objetivo é fornecer ao robô um modelo computacional da vida interior, ajudando o robô a raciocinar sobre si mesmo, suas capacidades, seu ambiente e seus companheiros de equipe. A fala interna do robô faz parte do objetivo da pesquisa. Foram levantadas as direções desse assunto.

Foi apresentado no artigo de Chella uma arquitetura cognitiva onde os principais módulos de introspecção tratam de todos os elementos de um estado mental que podem desencadear o processo de decisão do robô. A criação de módulo que antecipa os resultados das tarefas de raciocinar sobre as ações que melhor se adequam ao objetivo. Foi criado um ambiente simulador implementado o modelo interno do robô, que roda em paralelo com o robô e permite que os participantes da equipe participem das atividades do robô. Foram estabelecidos o estabelecimento, manutenção e reparação da confiança na equipe humano-robô. E foram analisadas a eficiência do robô.

2.2 Sistemas Semiautônomos

Os trabalhos de Jingjing *et al.* em [20], Ozgur *et al.* em [21] e Katherine e Roy em [22] serviram de base para as modelagens dos cenários utilizados por este estudo no que se refere a sistemas semiautônomos.

Jingjing *et al.* em [20] considerou a utilização de uma rede sem fio habilitada para o SARP com um conjunto de dispositivos terrestres distribuídos aleatoriamente em uma determinada área e cada um possuindo uma certa quantidade de dados para transmissão. A missão do SARP é sobrevoar esta região desde um ponto de partida até um destino, o SARP se comunica com os dispositivos terrestres para maximizar os dados coletados cumulativos. Foi verificado que no contexto de otimização da trajetória do SARP, devido à incerteza relacionadas as localizações dos dispositivos de terra e da dinâmica de comunicação é considerado um sistema difícil de se manter. Foi apresentado, através de modelos estocásticos, um algoritmo de otimização de trajetória baseado em aprendizado por reforço.

No experimento do artigo de Jingjing, foi criado um cenário e este dividido em pequenas grades de espaço onde é realizada a ação. Os resultados da simulação demonstraram que a abordagem de aprendizado por reforço pode encontrar uma estratégia considerando as restrições de tempo de voo. O problema de otimização de trajetória foi transformado em um MDP com estados e ações finitos.

O artigo de Ozgur *et al.* em [21] apresentou uma abordagem para o planejamento de movimento para cenários que envolve interação humano e máquina. O objetivo é que haja a interação segura e eficiente, o movimento da máquina deve ser ágil e percebido como natural pelo humano. A máquina deve conhecer o seu ambiente e os movimentos humanos, para agir de forma proativa, garantindo o cumprimento das tarefas. A previsão do movimento humano é fundamental para o planejador de movimento. O sistema proposto (baseado em movimentos probabilísticos) gera uma previsão do movimento humano através da distribuição de trajetória aprendida. O algoritmo proposto de planejamento

baseado em otimização estocástica, encontra progressivamente parâmetros de otimização viáveis, para replanejar o movimento e garante a prevenção de colisões enquanto minimiza o custo da trajetória relacionada à tarefa. Os resultados do artigo de Ozgur demonstraram que a abordagem proposta pode produzir trajetórias livres de colisões enquanto atinge o objetivo com sucesso. Os resultados do teste sustentam que o algoritmo estocástico proposto pode exceder o desempenho do trabalho de otimização de trajetória. E os movimentos da máquina estão associados ao movimento humano. A interação é então proporcionada considerando um obstáculo a ser evitado. Essa combinação não leva em consideração a resposta humana ao movimento da máquina. As observações em cenários reais de interação humano e humano podem ser utilizadas para replicar uma interação semelhante ao humano e máquina.

No artigo de Katherine e Roy em [22], apresentou um método de otimização estocástica para sistemas HITL. Este método identifica um conjunto de estados mais precisos que um veículo dirigido por humanos pode encontrar através de um conjunto de trajetórias observadas. A proposta apresentou um modelo que encapsula as trajetórias prováveis, utilizando-se para isso os limites de probabilidades, rejeitando de forma otimizada os valores discrepantes do conjunto de dados.

Ainda no artigo de Katherine e Roy, essa ferramenta forneceu previsões definidas que consistem em trajetórias observadas a partir da dinâmica e comportamentos de veículos dirigidos por humanos e pode também explicar os modos de comportamento, como a intenção do motorista, permitindo prever o comportamento de direção com boa precisão.

2.3 Experimentos Estocásticos

O trabalho de Feng *et al.* em [23], que é uma continuidade do [24] teve papel importante nesta dissertação pois mostrou a forma que os experimentos deveriam ser implementados estocasticamente utilizando os conceitos relacionados a HITL.

Feng neste teve como objetivo sintetizar protocolos de sistemas autônomos. Foi utilizado um cenário criado através de um modelo hipotético para a verificação do comportamento e desempenho do operador humano. O cenário proposto foi o da vigilância de rede rodoviária com o uso de UAV, que neste trabalho é considerado como SARP com certo grau de autonomia e controlado por um operador humano que representou alguns dos efeitos das características do desempenho humano nas interações de automação.

Na elaboração do cenário foram utilizadas técnicas relacionadas a MDP e jogos estocásticos para dois jogadores com o uso das ferramentas PRISM e PRISM-GAMES no modo SMG. O objetivo foi realizar avaliações das características que afetam o operador

no desempenho da missão, tais como carga de trabalho, proficiência e fadiga. Foram apresentados gráficos com o intuito de extrair *feedbacks* informativos.

Apesar da abordagem estar relacionada ao contexto de SARP, é possível aplicar em outros projetos de sistemas semiautônomo que interagem com operadores humanos. Foi dito por Feng *et al.* em [23] que a literatura não trata de Interação Humana (IH) nos sistemas semiautônomos, preocupando-se com ações humanas somente em pós-projeto.

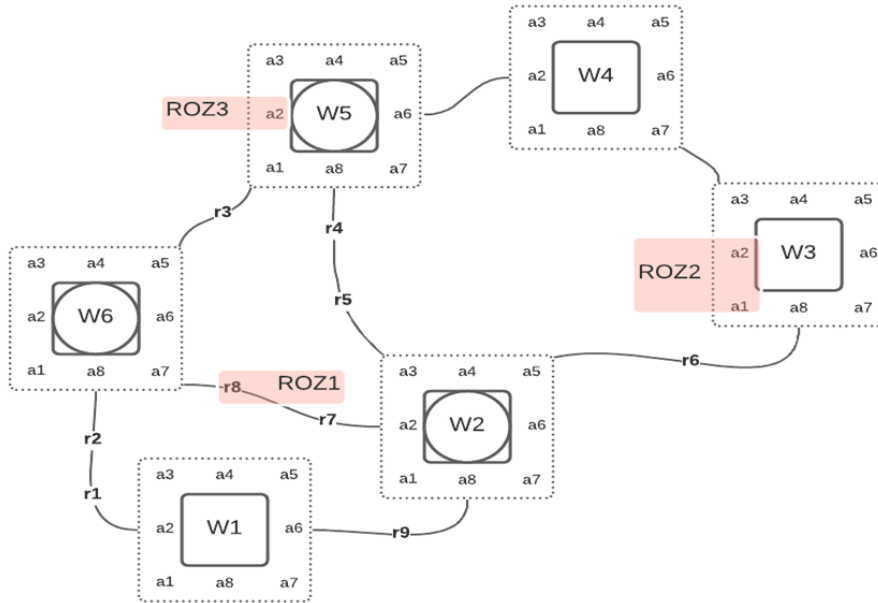


Figura 2.3: Mapa do Percurso Aéreo.

A Figura 2.3, adaptado do trabalho de Feng *et al.* em [23] apresenta o mapa contendo as vias aéreas utilizado na vigilância de uma determinada rodovia, onde:

- (w) identificados como Waypoint de passagem;
- (a) identificados como posicionamento para fotografia;
- (r) identificados como pontos da estrada e (ROZ) identificado como áreas restritas.

O trabalho de Muhammad em [5], publicado em 2021, que pode ser considerado uma continuação do trabalho do Feng, utilizou os principais conceitos de Feng *et al.* em [23], e propôs uma abordagem para sintetizar protocolos de controle para sistemas autônomos que respondem por incertezas e imperfeições nas interações com operadores humanos.

Foi também considerado um cenário envolvendo a vigilância da rede viária por SARP, controlado remotamente por um operador humano, com um certo grau de autonomia. Foi utilizado as ferramentas PRISM e PRISM-GAMES, essa abordagem permitiu a síntese

de planos de missões ideais dependentes do operador para o SARP, destacando os efeitos das características do operador, tais como carga de trabalho, capacidade de operação do SARP e fadiga no desempenho da missão do SARP.

Como resultado, estas abstrações são então integradas com modelos formais do sistema autônomo em consideração. E as técnicas de síntese reativa são aplicadas para gerar protocolos de controle autônomo dado uma especificação para o desempenho geral. Com o experimento estocástico do modelo do cenário e humano hipotéticos, os resultados experimentais mostram que estas abordagens podem ser viáveis para sintetizar protocolos de controle para sistemas autônomos que respondem por incertezas e imperfeições nas interações com operadores humanos.

Foram demonstradas, por meio de um estudo de caso detalhado de planejamento de missão de SARP, que os MDP e jogos estocásticos podem ser usados para modelar interações de automação humana e sintetizar protocolos de controle do tipo HITL. A ferramenta PRISM, foi utilizada e demonstrou ser mais do que suficiente para sintetizar estratégias para o problema citado, conforme Muhammad em [5].

O artigo de Chen *et al.* em [24] que é considerado como base para o trabalho de Feng *et al.* em [23]. Buscou como objetivo obter uma estratégia para jogos estocásticos de dois jogadores através da conjunção de Lógica Temporal Linear e metas de recompensa esperada. Estes procedimentos foram utilizados para elaboração de estratégias de vitória determinísticas dos jogos.

Os experimentos estocásticos do artigo de Chen foram implementados no PRISM-GAMES, com a técnica de verificador de modelo para jogos multijogador estocásticos, através de um estudo de caso para criar estratégias para controle de veículos autônomos cujo objetivo foi o de avaliar a aplicabilidade das técnicas. O Objetivo foi o desenvolvimento de um protótipo no PRISM-GAMES para criar e avaliar estratégias para a condução urbana autônoma usando dados de mapas reais. O PRISM-GAMES no modo SMG com 3 jogadores estocásticos foi utilizado nos experimentos desta pesquisa.

O Livro [16] de Simari *et al.* realizou diversas comparações e definições de BDI e MDP ajudando nas definições na aderência desses dois conceitos. Cujas implementações foram realizadas através das ferramentas PRISM e PRISM-GAMES, que são linguagens que possuem recursos de modelos estocásticos e MDP. O trabalho do Simari contribuiu na aderência dos experimentos MDP ao modelo BDI desta pesquisa.

Foi realizado também comparações entre a performance do BDI e MDP. Analisando dentre outros fatores, qual o melhor mecanismo para um agente tomar decisões para lidar com problemas de uma ampla variedade de maneiras, utilizados principalmente em sistemas autônomos.

O modelo BDI se enquadra numa classe de abordagens descritivas, ou seja, abordagens baseadas na análise da maneira como as pessoas ou animais tomam decisões. Já o MDP, pertence à classe das abordagens prescritivas, ou seja, abordagens estas que tentam identificar a decisão ótima, que são tipicamente baseadas na Teoria da Decisão [16].

Os resultados foram apresentados através de dados empíricos e demonstraram que em determinadas tarefas específicas, há casos em que o uso de um modelo MDP fornece a melhor solução, enquanto que existem casos em que abordagens BDI funcionam melhor. Foi verificado que à medida que a tarefa cresce em tamanho, além da gama de problemas que podem ser resolvidos de forma otimizada utilizando um MDP, nestes casos, a abordagem BDI supera [16].

O livro demonstrou que alguns aspectos da abordagem do BDI podem valer mais a pena. Os resultados mostraram que o modelo BDI não necessariamente está abaixo do ideal e que é possível construir um conjunto de intenções que podem tender para uma política ótima.

De acordo com Simari *et al.* em [16], o BDI tem flexibilidade também para construir intenções não necessariamente na busca de políticas ótimas, mas apenas de conhecimento de domínio. Os resultados lançam luz sobre a relação entre os dois modelos, pois mostra como os planos de BDI podem derivar das políticas prescritivas de MDP, sejam eles ótimos ou não. O trabalho mostrou que nos últimos anos essa abordagem é viável e pode levar a planejadores com a utilização de domínios estocásticos complexos.

Capítulo 3

Fundamentação Teórica

Neste capítulo serão apresentados os assuntos relacionados a sistemas multiagentes, agentes inteligentes, raciocínio prático, modelo de software, modelo de interação humana, ferramenta PRISM 4.7 e ferramenta PRISM-GAMES 3.0 e sistemas não tripulados.

Muitos conceitos apresentados por esta dissertação podem ser amplos. Neste Capítulo foram reunidos alguns deles, de forma resumida, e que estão relacionados com as abordagens pretendidas na dissertação.

3.1 Sistemas Multiagentes

Para Wooldridge em [9], os Sistemas Multiagentes (MAS) pertencem à área da IA distribuída que estudam agentes autônomos em um ambiente multiagente. O termo MAS está relacionado ao fato de que os agentes possuem existência própria e que podem existir sem a dependência de outros agentes.

É possível verificar na Figura 3.1 que cada agente (A1, A2, ..., A6) possui a sua esfera de influência no ambiente. Neste contexto, cada agente é uma entidade capaz de perceber seu ambiente, por meio de sensores, e de agir sobre esse ambiente ao longo do tempo, por meio de seus atuadores. Cada agente autônomo possui características que se baseiam em

Tabela 3.1: Categorias de SARP Utilizados pelo EB.

Categoria	Altitude	Raio de ação	Autonomia
0	até 0,9 Km	9 Km	1 hora
1	até 1,5 Km	27 Km	2 horas
2	até 3,3 Km	63 Km	15 horas
3	até 5,5 Km	270 Km	20 - 25 horas
4	até 9,0 Km	270 a 1.110 Km	25 - 40 horas
5	até 19,8 Km	5.550 Km	> 40 horas
6	até 19,8 Km	5.550 Km	> 40 horas

suas percepções e aprendizado ao invés de se basear apenas nos seus sistemas básicos de projeto [9].

Os MAS são compostos por múltiplos agentes que se interagem. E cada agente atua sobre o seu ambiente possuindo a sua esfera de influência. As esferas de influências, ao se sobreporem, podem gerar relações entre os diferentes agentes [9].

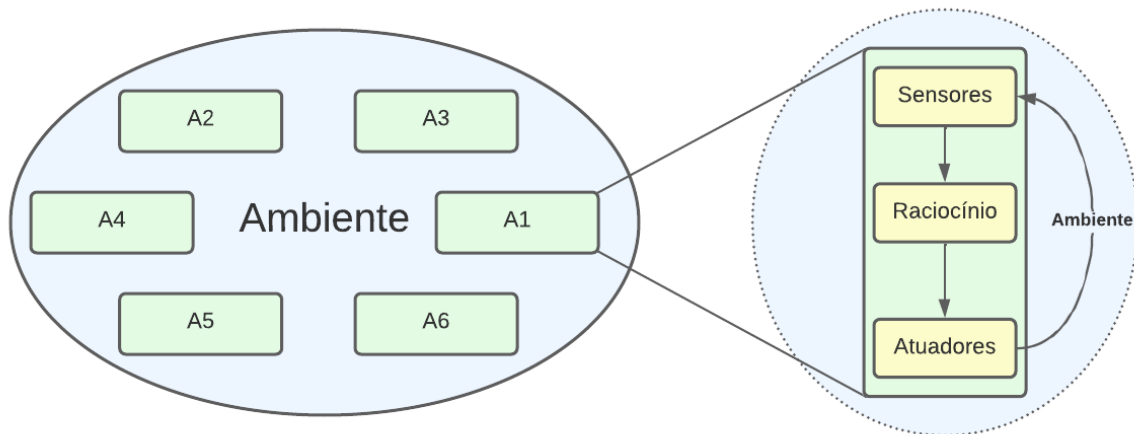


Figura 3.1: Característica do Agente no Ambiente.

Muitas vezes o aprendizado por reforço ajuda o agente a aprender por meio da interação com o ambiente. Essas interações permitem criar inferência de relações no que diz respeito a causa e efeito sobre as ações do agente e o que deve ser feito para atingir os seus objetivos. Através desse aprendizado, o agente pode mapear ações para diversas situações, de modo a maximizar a sua recompensa. Nesta situação, o agente pode lidar com o problema de decisão sequencial podendo afetar não só apenas a recompensa imediata, mas as recompensas seguintes [9].

3.1.1 Comportamento do Sistema Multiagente

Para Chen *et al.* em [24], os agentes possuem um conjunto de capacidades que estão relacionados com a sua competência, seus objetivos, e sua autonomia para alcançar os seus objetivos. Cada agente pode ser uma unidade computacional que possui comportamento autônomo, permitindo assim decidir as suas ações. As escolhas dos agentes levam em consideração as mudanças na esfera em que atuam, de modo que os desejos dos agentes estão diretamente relacionados aos seus objetivos.

Uma perspectiva diferente é que o comportamento geral de um sistema multiagente pode ser definido através do comportamento dos agentes. Segundo Chen *et al.* em [24],

em um sistema multiagente não é necessário que todos os agentes possuam a capacidade de ser inteligente para que possa alcançar um comportamento global.

Segundo Simari *et al.* em [16], o modelo de software BDI é uma das principais abordagens utilizadas para construção de sistemas multiagente. O BDI nos propicia mecanismos para separar as atividades de escolha de uma ação a partir da execução de planos ativos, onde se cria uma definição para esses agentes. Esses modelos também consideram os recursos vinculados para que se crie uma intenção após o raciocínio do agente.

Os robôs possuem sensores que constroem conjuntos de crenças que criam significância sobre o ambiente em torno dos agentes utilizando diferentes valores e diferentes parâmetros. Um conjunto de crenças descreve a situação específica em que o robô se encontra em um determinado momento.

Com base na localização, um objetivo pode ser definido para o robô que poderá se tornar desejo. Foi dito por Ujjwal *et al.* em [11] que um interpretador seleciona uma determinada ação de uma biblioteca de planos com base na situação em que se encontra.

3.2 Agentes Inteligentes

Conforme apresentado anteriormente, um agente pode ser qualquer coisa que pode ser vista percebendo seu ambiente através de sensores e age sobre esse ambiente por meio de seus atuadores [25].

No caso de um agente humano, estes possuem olhos, ouvidos e diversos outros órgãos como sensores e mãos, pernas, trato vocal e assim por diante como os atuadores de robô. No caso de um agente robótico, podem possuir câmeras e telômetros infravermelhos para sensores e vários motores como atuadores.

Para Russel *et al.* em [25], os agentes de software recebem, por exemplo, os pressionamentos de teclas, conteúdos de arquivos e pacotes de rede como entradas sensoriais e atua no ambiente proposto exibindo em telas, gravando arquivos e enviando novos pacotes de rede.

No nosso caso, o termo percepção se refere às entradas perceptivas do agente em qualquer momento. A sequência de percepção de um agente é de criar um histórico de tudo que o agente já percebeu. Em função disso, a escolha de uma determinada ação de um agente em um determinado momento pode depender de uma sequência de percepção observada até o presente momento.

Ao caracterizar a escolha da ação do agente para cada percepção possível, é exequível especificar tudo que o agente poderá fazer. De forma precisa, pode-se dizer que o comportamento de um agente depende de sua função, onde está mapeado todas as sequência de percepção dada às possíveis ações [25].

Suponha a necessidade de tabular todas as funções de um determinado agente. Na maioria dos casos, será uma tabela muito grande, que dependendo do tamanho pode se tornar intratável. E nesta situação, a estratégia é que seja definido um limite na percepção priorizando o foco a ser considerado. Dado um agente, pode-se em princípio construir uma tabela utilizando todas as percepções possíveis e registros de quais ações o agente deve realizar em sua resposta.

De acordo com Amato em [26], a organização do funcionamento do agente é muito conveniente. Infelizmente, os agentes em quase todas as situações não têm acesso a todas as informações necessárias sobre seu ambiente em que se encontram. Então, os agentes frequentemente podem agir sob condições de incerteza segundo Jayavel *et al.* em [27]. Este fato reforça a necessidade de estudar simulação multiagente e interação dos agentes como forma de promover maior autonomia aos agentes, possibilitando-os individualmente tirar conclusões lógicas e tomar decisões.

Pode-se dizer então que a tabela é uma caracterização externa de um agente. A parte interna de um agente artificial será implementada por um programa ou software, é importante ressaltar que essas duas ideias são distintas. A função do agente é uma característica de descrição matemática abstrata. Enquanto o programa do agente é uma característica de implementação concreta e rodando dentro de algum sistema físico. A IA opera os artefatos e podem necessitar de recursos computacionais significativos e o ambiente das tarefas que requerem uma tomada de decisão pode não ser trivial, segundo Russell e Peter em [25].

3.2.1 Agentes Baseados em Metas

Estes tipos de agentes geralmente possuem a característica de serem mais lentos. Porém possuem grande flexibilidade, pois eles possuem uma meta a ser seguida e os priorizam de acordo com esse objetivo e incluindo as suas percepções conforme Russell e Peter em [25]. Imagine a necessidade de saber algo sobre a situação atual de um determinado ambiente, isso nem sempre é o suficiente para determinar o que fazer.

Em uma situação de um cruzamento aéreo, o Sistemas de Aeronaves Remotamente Pilotadas (SARP) pode virar à esquerda ou direita ou seguir em frente. Esta decisão depende para onde o SARP quer chegar, ou seja, além da situação atual, o agente precisa de informações que atendam os seus desejos [25].

Então o software do agente pode juntar isso com o modelo para a escolha de suas ações para atingir os seus objetivos. A Figura 3.2 apresenta a estrutura de agentes baseada em metas adaptado do Artigo [25]. Ele monitora o estado do mundo, bem como um conjunto de objetivos que está tentando alcançar, e escolhe uma ação que eventualmente levará à realização de seus objetivos.

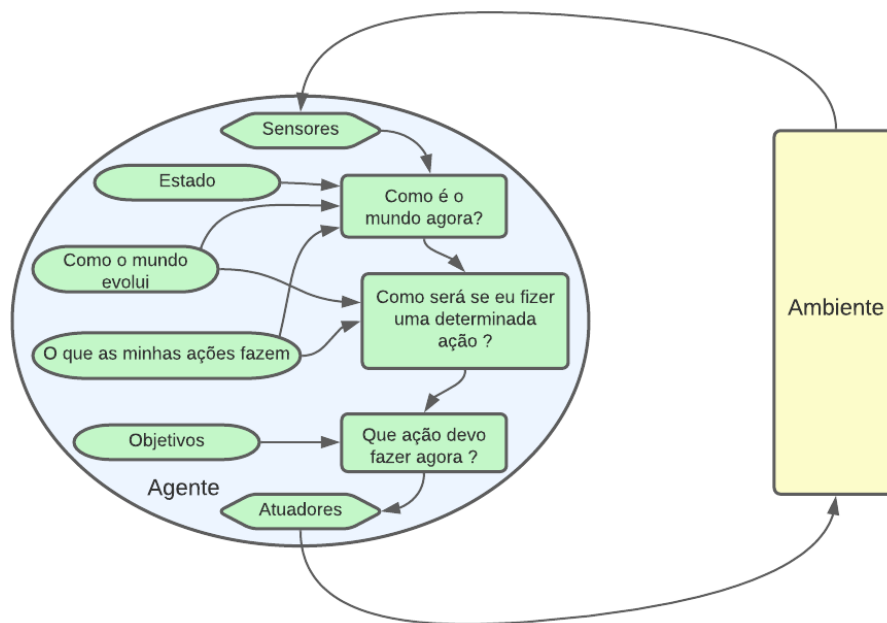


Figura 3.2: Modelo de Agente Baseado em Metas.

Em algumas situações, a seleção das ações com base em objetivos é direta, quando a resultante é uma única ação. Em outras situações, por exemplo, o agente terá que considerar longas sequências de ações para atingir um objetivo segundo Russell e Peter em [25].

Existem pesquisas relacionadas a IA dedicados a encontrar sequências de ação que atinjam as características do agente. As decisões desse tipo são diferentes das regras das condições e ações já descritas, pois envolvem considerações futuras. Como por exemplo, para atingir os objetivos devo fazer isso ou aquilo, conforme Russell e Peter em [25].

Na construção dos agentes, tais informações não são representadas explicitamente, pois as regras mapeiam diretamente às percepções e às ações. O agente do reflexo vai frear o carro quando encontrar as luzes de freio. No caso de um agente baseado em metas, a princípio se pode raciocinar que caso o carro da frente esteja com as luzes de freio acesas, ele poderá apenas reduzir a velocidade.

Dada a forma como o mundo evolui, a única ação que atingirá plenamente o objetivo de não bater nos carros da frente é frear [25]. Embora aparentemente o agente baseado em metas pareça ser menos eficiente, ele é mais flexível porque o conhecimento que apoia suas decisões é representado explicitamente e poderá ser modificado. Se começar a chover, por exemplo, o agente poderá atualizar os seus conhecimentos sobre a eficiência como os seus freios funcionarão. E tais ações fará com que os comportamentos relevantes sejam

alterados para se adequarem às novas condições.

Segundo Russell e Peter em [25], em relação ao instinto, tem que reescrever muitas regras para atender à condição-ação. O comportamento do agente baseado em metas pode ser facilmente alterado para o objetivo de ir para um destino diferente, apenas especificando o novo destino como a meta. As regras do agente no caso de virar e ir direto, funcionarão apenas para um único destino. E todos eles deverão ser substituídos para ir a algum lugar novo.

3.2.2 Agente Único e Multiagente

O sistema multiagente é composto de muitos agentes inteligentes capazes de interagir com diferentes tipos de ambientes. Os agentes podem ser compostos por robôs ou seres humanos. O ambiente pode possuir obstáculos e pode ser do tipo virtual ou real. Um exemplo é a interação dos seres humanos em jogos. Pode-se considerar que os humanos são os agentes e os aplicativos de jogos são os ambientes virtuais [27].

Os agentes inteligentes podem se comunicar por meio de seus sensores e atuadores, dos quais recebem as informações do ambiente por meio dos sensores e respondem por meio dos atuadores ao ambiente em que se encontram. Os MAS têm o potencial de resolver diversos problemas complexos que não podem ser feitos por apenas um agente individual. Podem ser compostos por combinações de sistemas heterogêneos e homogêneos. Alguns agentes podem ser independentes, tomando assim decisões de forma independente e outros agentes possuem apenas visualização local dentro do sistema, não sendo agentes de controle [27].

A natureza dos ambientes pode aparentar ser simples, porém, como exemplo, um agente que resolve um quebra-cabeças sozinho está claramente inserido em um ambiente de agente único, enquanto um agente jogando xadrez está em no ambiente de dois agentes.

Foi descrito por Russell *et al.* em [25] que uma entidade poderá ser vista como agente. Por exemplo, o agente operador do SARP-A tem que tratar o objeto de outro operador do SARP-B como um agente, ou pode ser tratado apenas como o objeto que se comporta de acordo com as suas leis da física, de forma análoga às ondas na praia ou nas folhas soprando no vento.

Observa-se que a principal diferença do comportamento do SARP-B é melhor descrito como aumento de uma medida de desempenho cujo valor depende apenas do comportamento do agente A. No xadrez por exemplo, a entidade do seu oponente SARP-B está tentando aumentar a sua medida de desempenho, que por sua vez, pelas regras do xadrez, minimiza a medida de desempenho do agente SARP-A.

Pode-se tratar que o xadrez é considerado como um ambiente competitivo para o multiagente. Já o ambiente do SARP, evitar as colisões podem maximizar como a medida de

desempenho de todos os agentes, por isso é um ambiente multiagente deve ser considerado parcialmente cooperativo. E também é parcialmente competitivo porque, por exemplo, apenas um SARP poderá ocupar um único local de pouso. Os problemas de projeto de agente em ambientes multiagentes costumam ser bem diferentes daqueles em ambientes de agente único [25].

Foi dito por Oliveira *et al.* em [28], que os MAS permitem abordar problemas complexos através das entidades computacionais autônomas e capazes de trabalhar em ambientes dinâmicos. Nos MAS, cada agente lida com um determinado subproblema e compartilha informações com seus outros pares, a fim de possibilitar atingir a meta geral do sistema, trabalhando de forma coordenada, podendo ser cooperativa ou competitiva.

Nos MAS, os agentes podem não trabalhar de forma isolada, porque fazem parte de uma sociedade e podem compartilhar os mesmos objetivos ou ter conflitos [29]. Portanto, nestas sociedades, é necessário introduzir normas e conceitos normativos em sistemas MAS. Estas normas têm por objetivo manter com precisão o comportamento do sistema, no intuito de prevenir conflitos, garantindo a coordenação e cooperação. Desta forma promovendo a convivência harmoniosa entre os agentes.

Os agentes inteligentes são entidades conscientes de seu ambiente circundante, o que direciona a forma como executam suas tarefas. Eles precisam interagir com outros agentes para realizar atos sociais em uma sociedade controlada e, portanto, estão sujeitos a normas em um contexto normativo do sistema. A compreensão do ciclo de vida das normas e sua atualização é crucial para aqueles ambientes multiagentes modernos, complexos, diversos e altamente interativos [29].

Foi apresentado por Oliveira *et al.* em [28], a necessidade de análise dos processos através de normas nas sociedades naturais e artificiais, apontando diversas diferenças entre elas. Observou-se a necessidade de uma melhor sistematização das atividades e das etapas de geração e evolução em relação às normas. Foi observado também um certo grau de ambiguidade sobre a taxonomia dos diferentes ciclos de vida, além das técnicas utilizadas.

3.3 Raciocínio Prático

Segundo Simari *et al.* em [16], o raciocínio prático envolve basicamente dois processos importantes:

- Quais objetivos tentar alcançar: conhecido como um conjunto de intenções, que são coisas que o deliberador compromete seus recursos para a sua realização;

- Como alcançá-los: é considerado como um processo relacionado ao raciocínio envolvendo meios e finalidades. Ou seja, elaborar uma sequência de ações que alcançará as intenções.

Segundo Bratman em [6], para alcançá-los, deve-se incluir um raciocínio prático, um processo direto, pelo menos para os humanos, e que na maioria das vezes poderá ser executado em um breve momento e sem muito esforço. No entanto, o conceito possui uma série de complicações não triviais do ponto de vista de um agente autônomo arbitrário.

Segundo Wooldridge em [9], tais limites mostram as seguintes implicações:

- A computação é um recurso importante para agentes que se encontram localizados em ambientes de tempo real. O uso eficiente desses recursos computacionais terá um efeito importante na capacidade do agente encontrar as ações adequadas, ou seja, o raciocínio estará vinculado a sua capacidade computacional.
- Os agentes não podem deliberar indefinidamente, o processo deve terminar em algum momento. Pode acontecer que, depois de parados, os compromissos selecionados não sejam os melhores. Paralelamente, o raciocínio teórico está relacionado às crenças e o raciocínio prático está relacionado à ação. Por exemplo, decidir se joga ou não a bola é um raciocínio prático, pois está relacionado às ações de quem o irá realizar.

Basicamente o raciocínio prático envolve dois processos importantes: decidir quais as metas que deverão ser alcançadas e como alcançá-las. O primeiro processo a ser utilizado envolve a Deliberação, que é um conjunto de intenções, definindo-se quais os recursos o deliberador vai comprometer para tentar alcançar a meta. O segundo processo é considerado um processo de raciocínio final, que elabora uma sequência de ações para alcançar as intenções. Bratman em [9] incluiu a seguinte definição: "O raciocínio prático é uma questão de ponderar considerações conflitantes a favor e contra as ações concorrentes, onde as considerações pertinentes são fornecidas pelo agente".

A Deliberação e o raciocínio de meios são processos computacionais, o que significa que eles estão sujeitos aos limites de recursos (como espaço e tempo) capazes de obter do sistema. Esses limites estabelecem a quantidade de cálculo que um agente será capaz de executar em um determinado período de tempo [9].

3.3.1 Ambiente em Mudança

Espera-se que o agente BDI atue em um ambiente dinâmico, de modo que o raciocínio do agente leve em consideração as mudanças do ambiente para realizar uma ação [15]. O modelo BDI deve se preocupar com a forma de como um agente toma as decisões racionais, através das ações que realiza em seu ambiente.

Para Mark *et al.* em [30], um dos maiores problemas encontrados nos projetos de agentes de raciocínio envolve o questionamento de como obter o equilíbrio entre deliberação, raciocínio, meios-fins e ação. Especificamente, um agente pode em algum momento abandonar determinadas intenções, por diversos motivos, que valerá a pena o agente parar e reconsiderar as suas intenções existentes. No entanto, esta reconsideração tem os seus custos que são frequentemente relacionados ao tempo e aos recursos computacionais.

Essa situação apresenta o impasse de não parar para reconsiderar ou continuar tentando alcançar intenções que já estão ultrapassadas. E se o agente reconsiderar constantemente as suas intenções, ele pode gastar muito tempo deliberando e nessas circunstâncias, poderá não fazer nada.

Segundo Mark *et al.* em [30], em uma série de experimentos realizados com uma implementação do modelo BDI foi verificado que é possível que o agente nunca atinja as suas intenções.

Para ilustrar, considere o parâmetro taxa de mudança do mundo (γ). Os principais resultados encontrados foram:

- Se γ for baixo, ou seja, o ambiente não muda rapidamente, então os agentes ousados funcionam bem em relação aos que são cautelosos. Isso ocorre porque os agentes cautelosos perdem seu tempo reconsiderando seus compromissos, enquanto os agentes ousados trabalham para alcançar seus objetivos.
- Se γ for alto, ou seja, o ambiente muda com frequência, então os agentes cautelosos tendem a ter um desempenho melhor do que os ousados, em virtude de que eles podem reconhecer quando suas intenções não são mais válidas e podem aproveitar novas oportunidades quando surgem.

Ambientes diferentes requerem diferentes estratégias de decisão. Em ambientes estáticos, o comportamento proativo é adequado, já em ambientes mais dinâmicos a capacidade de reagir às mudanças modifica as intenções e torna-se mais importante. Com isso, não existe a "melhor forma" para resolver o problema levantado [30].

Cada aplicativo deverá ser ajustado ao seu ambiente e o melhor equilíbrio dependerá desses aplicativos. A melhor maneira de resolver esse conflito pode ser implementar algum tipo de mecanismo de adaptação com o qual os agentes reconheçam a frequência que devem parar para reconsiderar suas intenções [30].

3.4 Modelo de Software

As modelagens de software são as atividades que visam construir modelos que expliquem as características e os comportamentos de um determinado software ou de vários softwares

que compõem um sistema [31]. Nesta construção, os modelos são identificados por meio de sua análise de requisitos e suas destinações.

Frequentemente as modelagens utilizam algum tipo de notação gráfica e também são apoiadas pelo uso de ferramentas gráficas, como por exemplo, Linguagem de Modelagem Unificada (UML) para a modelagem de programas orientados a objeto e a Modelagem dos Processos de Negócios (BPM), cujas funções são chamadas *Business Process Modeling Notation* (BPMN). A modelagem pode ser definida como a arte de criar moldes de algo [31].

3.5 Modelo de Interação Humana

Segundo Nahavandi em [32], os níveis de autonomia variam de operações remotas a sistemas totalmente autônomo. Em ordem crescente de autonomia, a mesma pode ser categorizada em três classes amplas: *Human-In-The-Loop* (HITL), *Human-On-The-Loop* (HOTL) e *Human-Out-Of-The-Loop* (HOOTL).

3.5.1 *Human-In-The-Loop* (HITL)

Neste modelo, as máquinas realizam uma tarefa por um período de tempo, então elas param e esperam por comandos humanos para continuar. Estes modelos são conhecidos como sistemas HITL [32]. É definido como um modelo que requer obrigatoriamente a IH. Pode servir para simulação ao vivo, virtual e construtiva.

Na simulação, um humano sempre faz parte da simulação e conseqüentemente influencia o resultado de tal forma que é difícil reproduzir exatamente a mesma simulação. O HITL também permite prontamente a identificação de problemas e requisitos que podem não ser facilmente identificados por outros tipos de simulações que não envolvem HITL.

O HITL é usado em simulações que incluem operadores humanos, como em um simulador de voo ou um simulador de condução. As simulações virtuais podem injetar o HITL, no qual exercem habilidades como por exemplo controle de aeronaves e habilidades de tomada de decisão. O HITL permite que o usuário altere o resultado de um evento ou processo. É utilizado para fins de treinamento e para contribuir para uma transferência positiva de habilidades adquiridas [33]. Isso pode ser demonstrado por exemplo, simuladores de voo na preparação de pilotos.

Para Tomaszewski em [34], os modelos HITL requerem sempre a IH. Os sistemas HITL permitem que os humanos mudem a produção dos sistemas convencionais de aprendizagem. E os simuladores HITL possuem interface humana como parte da simulação. Os humanos influenciam nos resultados da simulação de tal forma que os mesmos podem não

ser reproduzíveis. As simulações HITL, ajudam na identificação de falhas de modelo que podem não ser aparentes antes do teste em um ambiente real.

3.5.2 *Human-On-The-Loop* (HOTL)

Como exemplo de HOTL, as máquinas que podem executar uma tarefa completa e independente, mas têm um humano em uma função de monitoramento ou supervisão, com capacidade de interferir [32]. Os sistemas HOTL podem ser totalmente autônomos se os supervisores humanos permitirem que eles desempenhem uma função inteiramente por conta própria. Todos os experimentos realizados neste trabalho se basearam em HOTL.

Foi dito por Nahavandi em [32] que HOTL pode ser totalmente autônomo caso o supervisor permita que eles realizem uma tarefa completa e por conta própria. Manter o humano no Loop considera adicionar interação necessária através de uma interface homem-máquina. O grau de autonomia será determinado com base na relação do HOTL com o humano. HOTL recebe, avalia, decide e inicia a execução de uma operação, porém o supervisor poderá interromper a qualquer momento. Dispositivos dotados de HOTL são sistemas que podem usar a capacidade de autonomia para engajar os objetivos.

O principal questionamento é se esses sistemas HOTL são confiáveis. Ou seja, é importante possuir a capacidade de distinguir inimigos e amigos no campo de batalha ou outras questões éticas. Para se ter um HOTL utilizável e completo, a autonomia confiável torna-se valiosa.

Segundo Deborah *et al.* em [35], um sistema autônomo requer confiabilidade e confiança em sua tecnologia. No HOTL, a confiança pode ser definida como o nível de confiança que um ser humano tem em um sistema autônomo com base nas observações, percepções e expectativas sobre o desempenho do sistema e outras informações consideradas como evidência de competência.

Para John *et al.* em [36], a confiança no HOTL também é definida como sua capacidade de realizar uma atividade com sucesso, em um momento específico e sob condições caracterizadas por vulnerabilidade e incerteza. Se as ações de HOTL levar a consequências prejudiciais aos seres humanos ou equipamentos, o humano precisa restabelecer e manter a confiança nas operações de HOTL.

Os humanos precisam estar confiantes de que o HOTL responderá as condições de forma adequada em todas as situações e irá tomar as decisões corretas para realizar suas tarefas com precisão e eficiência.

Foi dito por John *et al.* em [36], que no momento da construção da confiança do HOTL, os sistemas precisam estar certificados. A certificação é um meio formal pelo qual um agente regulador confirma a eficiência e o desempenho esperados dos vários componentes de um sistema autônomo.

A confiança também pode ser estabelecida por meio de uma forma eficiente de comunicação de informações. O supervisor humano precisa de informações corretas do sistema para estabelecer uma confiança adequada e evitar o uso indevido e o mau funcionamento do sistema. Compreender os motivos das falhas do sistema pode aumentar a confiança, além de fornecer informações essenciais para tornar o sistema mais autônomo [32].

Um HOTL típico, possui uma série de componentes, como por exemplo:

- Detecção e percepção, que envolve as habilidades de sentir, interpretar, detectar e avaliar objetos em diferentes ambientes;
- Controle e tomada de decisão, que envolve a capacidade de tomar decisões precisas em um ambiente incerto e imprevisível;
- Execução, que envolve a capacidade de executar tarefas fornecidas pela unidade de controle e tomada de decisão.

Os *cockpits* de aviões são ambientes complexos e relativamente estressantes. O piloto investido de autoridade de controle, precisará de soluções desenvolvidas pela Inteligência Artificial sob sua supervisão. Neste caso, o piloto terá a prerrogativa de selecionar, recusar ou vetar uma sequência específica. Esse modo autônomo supervisionado é denominado de Humano no Loop (HOTL). Os autores veem este último modo como a base para o melhor subsídio [37].

No HOTL é necessário alcançar o melhor ajuste para compensar as fraquezas humanas e as limitações da máquina durante o processo de tomada de decisão [37].

3.5.3 *Human-Out-Of-The-Loop* (HOOTL)

No contexto HITL, deve ser considerada também a existência do modo (HOOTL), o qual a IA executa soluções sem IH. Embora este modo possa parecer inapropriado em termos de controle de uma operação onde o julgamento ético e legal humano é considerado crucial. Há situações críticas, tais como o excesso de informações, velocidade de resposta ou perda de comunicação, onde os humanos são incapazes de aplicar um julgamento sólido, nestes casos apenas uma IA baseada em regras será capaz de exercer uma tomada de decisão bem-sucedida [37].

O HOOTL é também uma possibilidade de compensar as fraquezas humanas e as limitações da máquina quando esta recebe a IH durante o processo de tomada de decisão. Na maioria dos casos, uma combinação dos modos HITL, HOTL e HOOTL permitirá o processo mais rápido e seguro [37].

3.6 Ferramenta PRISM 4.7

Segundo Kwiatkowska *et al.* em [38], o PRISM é uma ferramenta de verificação que suporta a construção e a análise quantitativa formal de vários modelos probabilísticos, incluindo cadeias de Markov de tempo discreto, tempo contínuo e processos de decisão. A verificação formal é constituída de aplicações de técnicas rigorosas baseadas na matemática para estabelecer com mais exatidão os sistemas computadorizados [39].

O modelo probabilístico contido no PRISM a partir da versão 4.0 possui verificação quantitativa de autômatos temporais probabilísticos, foco no suporte a cadeias de Markov de tempo contínuo e modelos de recompensa, e podem ser usados para analisar propriedades de performance [40].

Além disso, o PRISM pode ser utilizado também em sistemas que exibem características probabilísticas, não determinísticas e em tempo real. Em muitos domínios de aplicação estão, por exemplo, a simulação de controladores incorporados em sistemas automotivos ou aviônicos [40].

Kwiatkowska *et al.* em [38] disse que o PRISM é de código aberto e contém vários componentes de uso independente, como por exemplo: um kit de ferramentas extensível para construir, verificar e refinar abstrações de modelos probabilísticos, uma biblioteca de verificação de modelo probabilístico de estado explícito, um mecanismo de simulação de eventos discretos para verificação de modelo estatístico, além de suporte para geração de adversários ideais.

O PRISM pode ser utilizado em vários domínios de aplicação, oferecendo suporte a uma variedade de modelos probabilísticos, como o Processo de Decisão de Markov (MDP) para capturar o comportamento do sistema, segundo Parker *et al.* em [41]. É necessário ressaltar que esses modelos são descritos em uma linguagem própria, em que o sistema é dividido em diversos módulos que podem ou não interagir uns com os outros.

A Figura 3.3 apresenta um *printscreen* do PRISM. A tabela que se encontra na metade inferior da Figura mostra, neste exemplo que será amplamente debatido no Capítulo 4 e 5, mostra o posicionamento do SARP nos vértices representados pelas variáveis (g, w, r e f). Quando essas variáveis possuem o valor 1 representa o estado presente do SARP, e quando possui o valor 0, representa o estado ausente do SARP.

Os módulos do PRISM possuem o seu funcionamento com composições paralelas, as propriedades que precisam ser verificadas são aprisionadas usando a linguagem que incorpora a função probabilística da lógica. O PRISM é frequentemente usado para o cálculo das probabilidades com a estrutura de recompensa em seus cálculos [39].

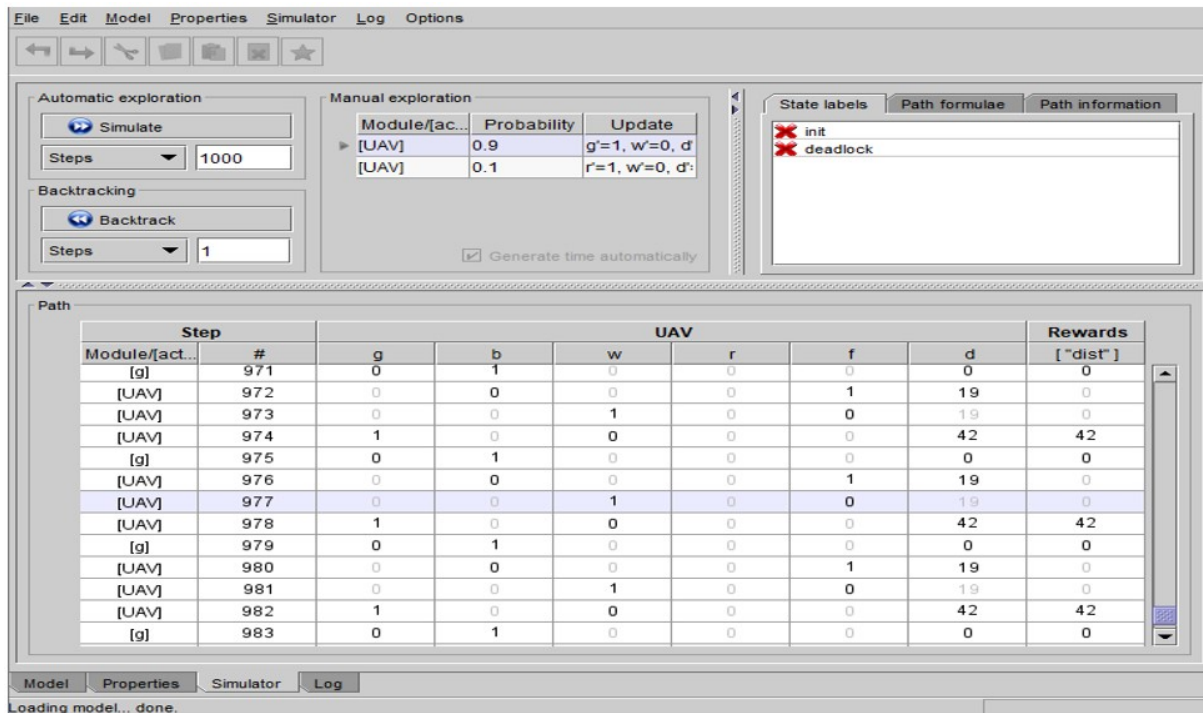


Figura 3.3: *Printscreen* da Ferramenta PRISM 4.7.

3.6.1 Processo de Decisão de Markov

Para Rosenschein em [42], o MDP é um modelo para a tomada de decisão sequencial, é assumido que os agentes possuem sensores para que saibam sempre exatamente em que estado se encontram, mesmo que haja incerteza sobre o efeito de suas ações. E o agente não precisa reter nenhuma informação sobre o histórico de suas ações anteriores para tomar as melhores decisões.

O problema a ser solucionado é encontrar a melhor maneira de se comportar diante de um modelo completo e correto do ambiente e também de seu objetivo. São problemas de Inteligência Artificial (IA) como os de planejamento, mas a consideração de estocástico nos força a partir do modelo tradicional a computar soluções na forma de políticas no lugar de planos [42].

Uma política existente é o mapeamento completo de estados para ações. Nesse contexto, uma vez que uma política é calculada a partir do modelo de transição e da função de utilidade, é trivial decidir o que fazer, de modo que é a função que dirá ao agente o que deve ser feito que está representada explicitamente pela política.

Segundo Muhammad em [5], o MDP é um dos modelos amplamente conhecidos que é usado para capturar o comportamento dos sistemas reativos. Em um MDP, os parâmetros do sistema são representados pelos estados do sistema e a evolução do sistema, desencadeada por uma ação escolhida não deterministicamente a partir de um conjunto de

ações disponíveis em um estado, é capturada em tempo discreto por meio de transições de estado. Se cada estado do sistema evolui inteiramente com base em alguma distribuição de probabilidade, ou seja, há no máximo uma ação disponível em cada estado, o modelo de transição de estado é denominado Cadeia de Markov de Tempo Discreto (DTMC).

Ao contrário, em um MDP, um estado pode ter mais de uma ação disponível. Nesse caso, as transições probabilísticas com base na distribuição de probabilidade de uma ação podem ser feitas apenas depois que a ação é escolhida de forma não determinística a partir de um conjunto de ações disponíveis. O não-determinismo é frequentemente resolvido com a ajuda de um planejador (ou controlador) que escolhe uma ação entre as disponíveis [42].

O problema para o cálculo de uma política ótima em um ambiente estocástico acessível baseado em um modelo de transição conhecido é chamado de Problema de Decisão de Markov. Em um problema de decisão, a propriedade de Markov é válida se as probabilidades de transição de qualquer estado determinado dependerem apenas do estado e não do histórico anterior [42].

3.7 Ferramenta PRISM-GAMES 3.0

O verificador de modelos do PRISM-GAMES possui as seguintes características adicionais em relação ao PRISM [39]:

- Jogos Estocásticos Simultâneos trazem modelagem mais realista de agentes interagindo de forma simultânea;
- As propriedades baseadas em equilíbrio fornecem um meio de analisar jogos nos quais jogadores concorrentes ou colaboradores são impulsionados por objetivos distintos;
- Uma extensão em tempo real de jogos estocásticos (baseados em turnos) facilita a verificação e a síntese de estratégia para sistemas em que o tempo é um aspecto importante.

Kwiatkowska *et al.* em [43] fala sobre o desempenho e a escalabilidade da ferramenta, sendo aplicado em diversos estudos de caso que vão desde protocolos de segurança até coordenação de robôs. Esses desafios inspiraram o desenvolvimento de uma ferramenta de verificação de modelo para jogos estocásticos.

À medida que os sistemas de computação envolvem cada vez mais agentes autônomos em ação simultânea, as abordagens teóricas do jogo estão se tornando generalizadas na ciência da computação como uma abstração de modelagem fiel. Essas técnicas podem ser utilizadas para raciocinar sobre o comportamento competitivo ou colaborativo de múltiplos agentes Racionais ou entidades com objetivos semelhantes ou distintos [43].

A Figura 3.4 apresenta o *printscreen* do PRISM-GAMES no modo Simulador. Note que a coluna *Step* contém os módulos que estão sendo utilizados em turnos distintos de Jogos Multijogador Estocásticos Baseados em Turnos (SMG): BDI, HITL e planos.

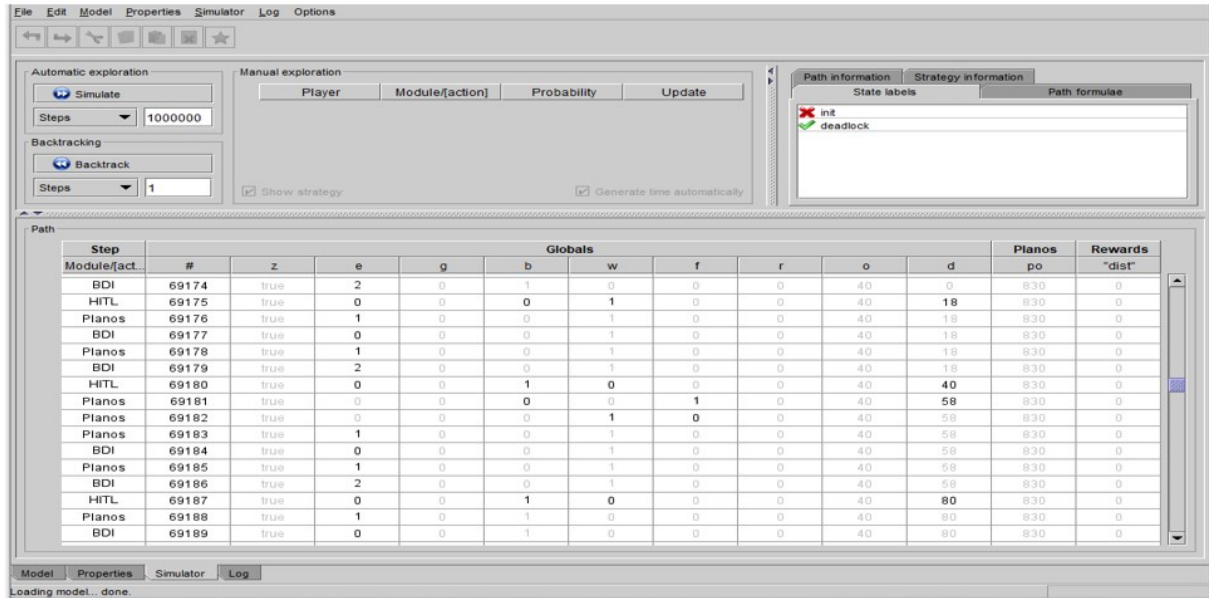


Figura 3.4: *Printscreen* da Ferramenta PRISM-GAMES 3.0.

Para Kwiatkowska *et al.* em [44], projetar sistemas teóricos corretamente ainda é um desafio, em virtude da complexidade dos comportamentos decorrentes das interações entre autonomia, concorrência e recompensas. Além disso, muitas dessas aplicações exigem raciocínio. Os protocolos podem empregar robôs autônomos que operam em ambientes incertos e podem usar componentes de hardware e sensores não confiáveis. sistemas baseados em dados, como sistemas de navegação, podem depender de modelos probabilísticos aprendidos para sua execução. O PRISM-GAMES permite:

- O suporte a modelagem e análise de jogos multijogadores estocásticos simultâneos (CSG). Versões anteriores da ferramenta suportavam CSG, nos quais se presume que cada estado do jogo é controlado por um jogador específico. Os CSG permitem que os jogadores tomem decisões simultaneamente, sem conhecimento das escolhas uns dos outros, fornecendo um modelo mais realista de execução simultânea e tomada de decisão. Para isso, é compreensível permitir que o usuário especifique a concorrência e a sincronização entre os agentes, bem como associe recompensas a ações conjuntas ou únicas [44];
- O PRISM-GAMES adiciona suporte para jogos multijogadores cronometrados probabilísticos (TPTG). Estes estendem jogos multijogadores estocásticos com relógios,

no estilo de autômatos cronometrados (probabilísticos). Isso permite que aspectos em tempo real de um sistema sejam modelados com mais precisão usando a abordagem dos relógios digitais, para que os modelos cronometrados sejam traduzidos automaticamente para modelos discretos que deverão ser verificados [44].

A verificação quantitativa e o desenvolvimento de estratégia são técnicas poderosas para a modelagem e análise de sistemas informatizados que requerem raciocínio sobre aspectos quantitativos, como probabilidade, tempo ou uso de recursos. Exemplos de aplicações em que essas técnicas têm sido utilizadas incluem controladores de energia, veículos aéreos não tripulados, protocolos de condução autônoma e comunicação [44].

3.7.1 *Turn-Based Stochastic Multi-Player Games (SMG)*

O PRISM-GAMES configurado no modo SMG funciona como uma ferramenta para jogos multijogadores estocásticos baseados em turnos, em que cada estado é controlado por um jogador e esse jogador seleciona uma ação a ser executada [43].

O comportamento resultante evolui para probabilidade, permitindo que o modelo capture aspectos do jogo. O ponto principal para raciocinar sobre jogos multijogadores estocásticos são as definições de estratégias, que são representadas por escolhas feitas por um determinado jogador, com base na execução do modelo até o presente momento. Um jogo estocástico composto por apenas um jogador configura-se como um Processo de Decisão de Markov (MDP). Neste caso, pode-se escolher o comportamento do jogador como adversário, um exemplo é a representação de um código malicioso de um protocolo de segurança [43].

Para Kwiatkowska *et al.* em [43], com o PRISM-GAMES é possível verificar se o modelo exibe certas propriedades formalmente especificadas, independentemente do comportamento do adversário. De forma alternativa, pode-se supor que são capazes de controlar as escolhas do único jogador neste modelo, como por exemplo uma representação de um sistema de controle de navegação de um veículo autônomo.

Nesse cenário, pode-se criar uma estratégia sobre a qual o comportamento do jogo satisfaz uma propriedade que foi formalmente especificada. No caso em que participem vários jogadores, nos permitirá modelar situações em que existam jogadores com objetivos opostos. O PRISM-GAMES fornece possibilidades para o levantamento estratégico para garantir a satisfação de uma estratégia, independente das utilizadas pelos outros jogadores [43].

Por exemplo, em um veículo semiutônomo SARP, é possível gerar uma estratégia para que possa garantir que a probabilidade de completar uma viagem com sucesso seja estabe-

lecida. Independentemente do comportamento de outros aspectos que não são controláveis [43].

3.8 Sistemas Não Tripulados

O Artigo [45], Pimentel aborda que tem-se notado nos últimos anos um grande crescimento de sistemas robóticos, dos quais pode-se caracterizar a capacidade de realizar movimentos e ao mesmo tempo possuir autonomia e capacidade de interagir com o meio ambiente, possibilitando assim um vasto campo de aplicações e conseqüentemente trazendo muitos desafios.

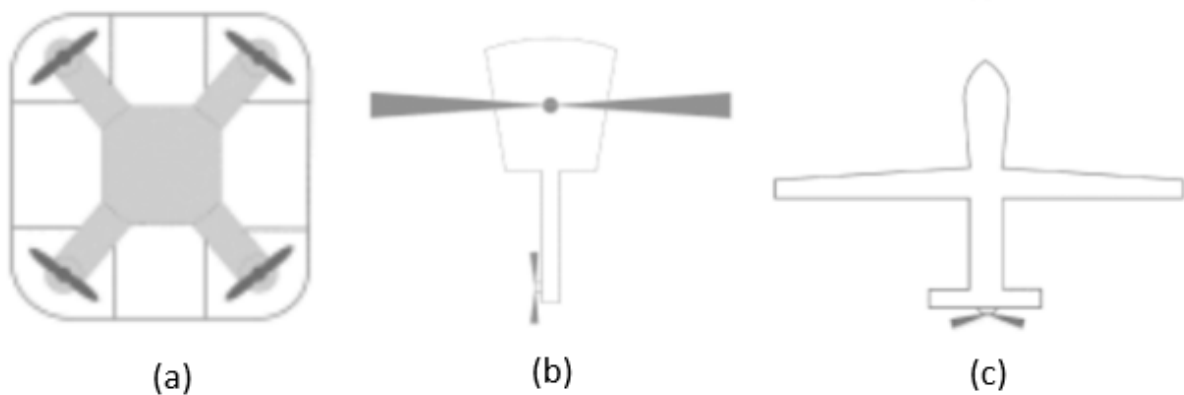


Figura 3.5: Localização de Rotor no SARP.

Os UAV, também conhecidos como drones ou Veículo Aéreo Não Tripulado (VANT), e no EB conhecidos como Sistemas de Aeronaves Remotamente Pilotadas (SARP), estão disponíveis nos mais diferentes tamanhos e especificações conforme pode ser visto nos modelos classificados pela localização do rotor na Figura 3.5, adaptada de [46], em que há os formatos (a) quadrimotor, (b) helicóptero e (c) avião. Tais equipamentos possuem a característica de serem facilmente empregados na maioria das condições.

A Tabela 3.2, adaptado do artigo de Fotouhi *et al.* em [46] apresenta as especificações dos modelos de SARP. Dependendo de seus mecanismos de voo, os drones são classificados em três tipos: drones multi-rotor (também conhecidos como drones de asas rotativas) possibilitam decolagem e pouso verticais, além de poder pairar sobre um local fixo podendo fornecer vigilância e comunicação, estes equipamentos possuem alta manobrabilidade e precisão, no entanto possuem mobilidade limitada pois tem alto consumo de energia no deslocamento em relação a outros tipos de rotores.

A Figura 3.6, cuja fonte é do próprio autor, apresenta as principais partes que compõem um quadricóptero. Este modelo possui o módulo de comunicação, para se comunicar

Tabela 3.2: Exemplos de Tamanho de SARP.

	Pequeno(a)	Médio(b)	Grande(c)
Peso	2 a 25 Kg	25 a 50 Kg	Acima de 150 Kg
Modelo	S900	Scout B-330 UAV	Predador
Peso	3.3 Kg	90 Kg	2.223 Kg
Carga Útil	4.9 Kg	50 Kg	1.700 Kg
Mecanismo de Voo	Multirotor	Multirotor	Asas fixas
Alcance	N/A	N/A	1.852 Km
Altitude	N/A	3 Km	15 Km
Tempo de Voo	18 min	180 min	1.800 min
Velocidade	57.6 Km/h	100 Km/h	482 Km/h
Energia	Bateria	Gasolina	Querosene de Aviação
Motor	Motor Elétrico	Motor a Pistão	Turbina

com o operador e envio de imagens. O módulo de controle para controlar a velocidade dos motores, permitindo que o SARP realize as manobras. O módulo Unidade Central de Processamento (CPU), que neste trabalho é considerado como o sistema de auto pilotagem, que serve para prover a capacidade semiautônoma ao SARP, através do plano de voo armazenado, sensores, antena GPS e comandos via rádio.

Existem dispositivos que condensam a maioria dos seus módulos eletrônicos em uma única placa de circuito eletrônico, com a finalidade de reduzir os custos de fabricação e também para deixar mais leve e compacto.

Caso o SARP perca a comunicação com o operador, o módulo CPU assume o controle e pode realizar manobras de retorno a base ou até o restabelecimento da comunicação. O módulo da CPU também é capaz de realizar manobras independentes através do plano de voo.

Segundo Fotouhi *et al.* em [46], os drones de asa fixa praticamente deslizam sobre o ar, tornando-os significativamente mais eficientes em termos de energia em relação ao multi-rotor. Além de serem capazes de transportar mais carga, as características da asa fixa ajudam os drones a viajar em uma velocidade mais rápida. O alcance pode variar de algumas dezenas de metros para os pequenos a centenas de quilômetros para os drones grandes.

Entre as principais desvantagens dos drones de asa fixa está a necessidade de uma pista para decolagem e pouso para os modelos maiores, além de não possuírem a capacidade de pairar sobre um local fixo. Mas atualmente surgiram os drones híbridos, que possuem as características de drones de asa fixa e asa rotativa que dispensam o uso de pistas para pouso e decolagem.

A fonte de energia determina significativamente a sua resistência, autonomia, tempo de uso e distância percorrida. Podem ser de baterias recarregáveis ou alimentados por com-

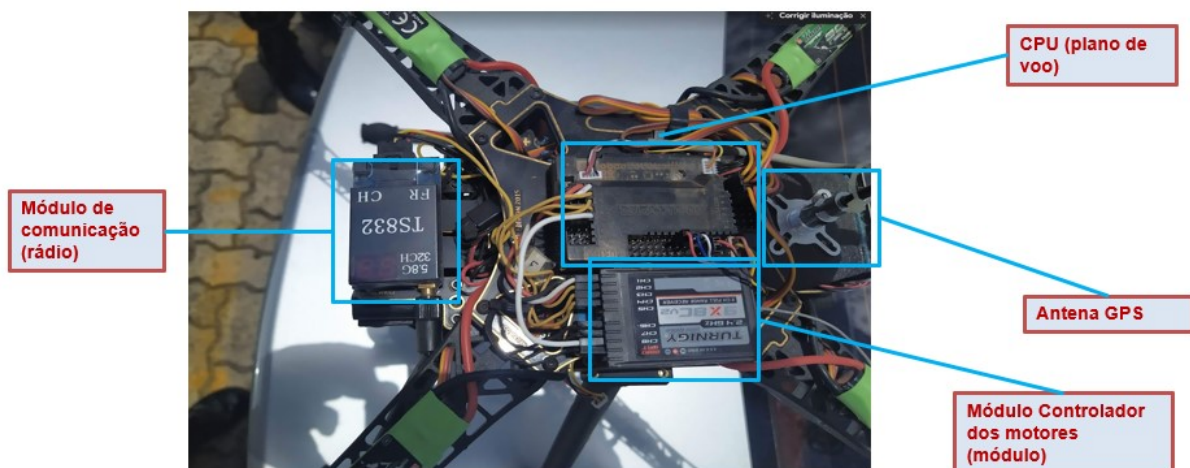


Figura 3.6: Componentes Internos de um SARP.

bustíveis fósseis, este último normalmente utilizado em drones maiores, obtendo tempos de voo mais longos. Os equipamentos a bordo internos dos drones, precisam de fonte de alimentação (energia elétrica) necessitando o transporte de bateria ou geradores conforme Fotouhi *et al.* em [46].

A Tabela 3.1, adaptado do manual de campanha EB70-MC-10.214 [47] apresenta a classificação e categorias dos SARP utilizados no EB. Os UGV na explicação de Pedro *et al.* em [48] são também conhecidos como drones terrestres, cuja tração empregada é determinada pelo ambiente em que será utilizado, podendo ser tracionado por esteiras, que é indicado para ambientes em que o terreno é irregular ou com lama. Estes, porém, possuem a desvantagem da baixa velocidade de locomoção.

A tração por pneus apresenta maior velocidade de locomoção que o de esteiras e pode ser utilizada em situações mais silenciosas. A tração é obtida por um ou mais eixos ou até mesmo ser independente em cada pneu. Já a tração por patas apresenta semelhanças com animais. Existem UGV quadrúpedes que são capazes de transportar cargas em lugares muito acidentados [46].

Capítulo 4

Proposta do Trabalho

Neste Capítulo serão tratados as interações humanas e BDI, elaboração do modelo conceitual, proposta e funcionamento do BDI proposto.

A partir das contribuições dos trabalhos relacionados, constantes no Capítulo 2, esta pesquisa realizou a Construção do BDI proposto que considera a (IH).

4.1 Interações Humanas e BDI

Conforme visto anteriormente, embora a modelagem de agentes de software baseado em BDI esteja essencialmente voltada ao comportamentos de máquina, na literatura não foi encontrada discussão acerca da IH influenciando as decisões do agente BDI. Pelo contrário, o que se vê é um comportamento padrão de obediência inequívoca da máquina, de tal sorte que após a IH ocorrer, basta que o agente de software recalcule seus parâmetros operacionais e defina um novo plano de ações.

Esta pesquisa procurou trazer um novo conceito para a construção do agente de software BDI, que passa a avaliar o operador como se fosse uma fonte externa com a qual deva lidar. Ao estabelecer um critério geral chamado de GO, o agente de software passa a ter um limiar de aceitação nas IH, com vistas a cumprir seus Desejos pré-programados.

Dessa forma, o decisor que está contido no mecanismo de BDI poderá em algum momento escolher se obedece o operador ou se continua executando sua missão planejada. O que se espera analisar é o conflito entre a ação da ordem do operador e a ação esperada pelo robô. Nas próximas seções serão apresentadas a proposta da pesquisa assim como a explicitação dos parâmetros mais importantes utilizados.

No Capítulo 2 foram abordados os conceitos de HITL e HOTL e de acordo com a literatura entende-se que o HITL sempre incluirá o humano e que o HOTL não necessariamente incluirá, este podendo executar a missão sozinho sem a interferência humana. Conforme visto anteriormente, o conceito mais aplicável a este trabalho é o HOTL. Porém,

é importante frisar que o HOTL é considerado uma particularidade do HITL de acordo com a sua extensa literatura disponível.

4.1.1 A Ponderação do BDI e IH

O modelo estendido BDI de Zhao *et al.* em [49] possui diversos conceitos relacionados a elementos que fazem parte da IA, que desconsideram a IH, pois a IH é prioridade diante das decisões do BDI. Contudo, espera-se que com a modelagem proposta, a mesma venha a considerar a IH no processo de decisão.

Dessa forma, o SARP tem condições de realizar a ponderação da IH através do GO do BDI, que dependerá de fatores que contribuem em não pôr em risco a missão ou o próprio SARP, e assim decidir se é o caso de obedecer ou não a IH.

Uma justificativa para essa abordagem, por exemplo, é que as informações disponíveis em relação ao ambiente podem não ser tão reais quanto às do sistema BDI ou vice-versa. Um exemplo está na IH do tipo HOTL, cuja decisão foi prejudicada pelas interferências na comunicação do SARP, como os sinais de monitoramento precários, a ponto de prejudicar a operação remota.

Outra justificativa é que devido à velocidade com que as ameaças ao SARP aparecem e desaparecem, a velocidade da resposta humana a estas ameaças pode ser lenta demais. Ou seja, nestes casos, o próprio BDI pode ser o mais apto a tomar a decisão a tempo de evitar tais ameaças. Dentro deste contexto, o operador pode ser induzido a tomar atitudes equivocadas com base em informações sensoriais transmitidas, podendo gerar diversos tipos de imprecisões.

O BDI, tendo conhecimento do risco associado da ação humana, pode corrigir a conduta, considerando o objetivo principal da missão e suas metas pré-definidas em seu modelo de software. Se for considerado que a IH é mais uma entrada entre os sensores percebidos pelo sistema BDI estendido, é possível avaliar a IH como um elemento de confiabilidade ou de assertividade em relação ao objetivo da missão. Isso permite tomar a IH como correta caso ela esteja dentro dos parâmetros operacionais da missão ou plano.

O foco deste trabalho não é simplesmente modelar situações em que a missão vai ser alterada pela IH. A contribuição do trabalho está em mostrar que é possível usar uma modelagem BDI para parametrizar a GO à IH no cumprimento de missões do sistema autônomo.

4.1.2 Ações do BDI

Foi detalhado no Capítulo 1 o contexto do emprego de SARP, no qual possui sistemas semiautônomos de controle. Uma das opções viáveis é a utilização de agente de soft-

ware implementados sob uma arquitetura BDI, sendo que o resultado esperado é que o SARP consiga realizar tarefas complexas descritas em seus planos, respeitando os limites estabelecidos em suas crenças e intenções.

Contudo, é prevista a participação de um operador que pode e provavelmente vai comandar o robô, por vezes até contrariando o plano previsto constante no BDI. Ele pode executar ações não previstas no plano, podendo alterar o parâmetro da distância percorrida no cumprimento dos objetivos determinados para o SARP, o parâmetro de distância percorrida foi utilizado como uma das métricas de desempenho da missão nesta pesquisa.

Nesses casos, entende-se como conveniente permitir que o SARP tome o controle e retorne aos planos originais, com vistas a concluir os seus objetivos estabelecidos sob condições de desempenho aceitável.

Segundo Feng *et al.* em [23], a implementação típica de agentes de software BDI possui mapeado Crenças, Desejos, Intenções, planos, um conjunto de estados possíveis e ações que transitam entre os estados estabelecidos. A participação humana em sistemas semiautônomos, por sua vez, é tratada como um evento externo e serve de entrada para novas tomadas de decisão.

A capacidade de decisão de um operador não pode ser considerada nem melhor nem pior que a do sistema autônomo, porém diferente aos olhos do BDI, pois o ser humano como operador ativo, pode vir a ter acesso a informações que o robô não tem, como monitoramentos externos, sensores de ambiente, comunicação de rádio, dados de inteligência, dados meteorológicos dentre outros.

Conforme visto anteriormente, as IH não são consideradas nos atuais modelos de software de sistemas BDI. Desse modo, não há controle ou acompanhamento da evolução de seus efeitos durante a execução do agente BDI, o que pode representar uma lacuna de pesquisa.

Dessa forma, enxerga-se constituir relevância a modelagem dessa nova arquitetura de software que possibilita ponderar a IH durante o cumprimento da missão, aumentando a resiliência do agente em situações imprevisíveis por meio das informações disponíveis mas que seja coerente por parte do operador, que possivelmente tem mais informações à disposição.

4.2 Elaboração do Modelo Conceitual

Para constituir o modelo conceitual que também possa utilizar IH como entradas para o processo de decisão, sugere-se aprimorar o modelo de agente de software BDI estendido, conforme proposto por Lee *et al.* em [14]. Tal modelo infere que o operador do sistema

semiautônomo pode enviar comandos a qualquer tempo, e este ser obedecido prontamente, sendo este aspecto o que se propõe alterar neste trabalho.

4.3 Proposta

Em extensão ao modelo BDI proposto por Lee *et al.* em [14], apresentado na Figura 2.2, a Figura 4.1 apresenta a proposta contendo o modelo de agente de software BDI que permite avaliar a IH antes de obedecê-lo, ou seja, somente obedecerá se isto não for comprometer o cumprimento de seus planos e Desejos, utilizando-se para isso um levantamento do GO.

Neste contexto, a entrada do comando externo de operador deixa de ser aplicada diretamente aos sistemas de controle das superfícies aerodinâmicas do SARP e passa a ser um elemento de entrada no processador perceptivo.

Os componentes que foram incluídos no BDI desta pesquisa estão destacados na cor verde. Os retângulos representam os processos e as circunferências são bases de informações dos principais componentes do BDI, as setas indicam os fluxos de informações.

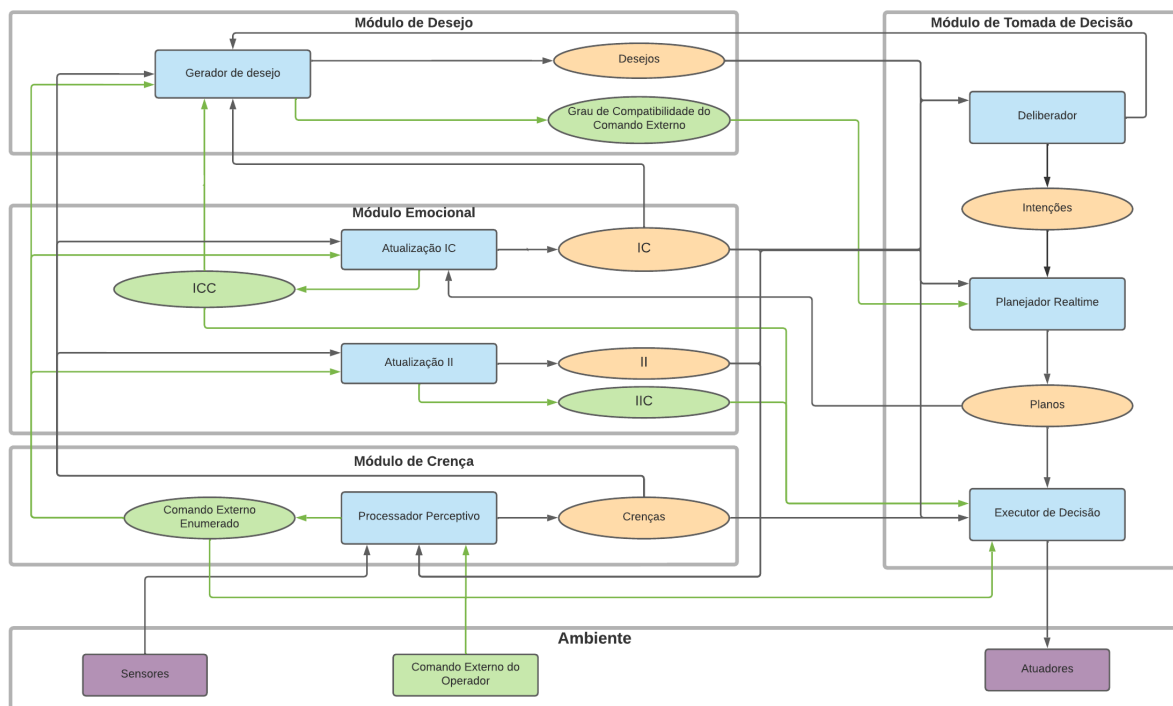


Figura 4.1: Proposta de BDI Contendo o Comando Externo do Operador.

4.3.1 Comando Externo do Operador

O operador faz parte do ambiente e para executar a IH, envia sinais elétricos que são repassados via rádio ao sistema BDI do SARP através de, por exemplo, um *Joystick* ou botões de controle. Estas interações podem ser do tipo que devem ser executadas imediatamente ou não. Esta ação faz com que o Comando do Operador alimente o Processador Perceptivo que está localizado no Módulo de Crenças.

A IH proveniente do Comando Externo do Operador pode ser desde um comando único, ou até mesmo a combinação de vários comandos compondo assim um comando complexo. Desde que seja exequível pelo sistema semiautônomo. No caso de um comando complexo, este pode não estar mapeado no sistema BDI, haja vista a quantidade possível de combinações de comandos.

Ressalta-se que o comando emitido pelo operador pode estar incorreto. Devido a interpretação oriunda de comunicação com falha ou incapacidade de responder a tempo. Comandos incorretos podem trazer riscos à missão.

4.3.2 Comando Externo Enumerado

Após a IH ser executada pelo Processador Perceptivo, este alimenta o Comando Externo Enumerado, que é a representação do agente de software para IH, de modo que possa ser analisado no modelo BDI. Tal processo funciona como Crença ao Comando do sistema BDI e possui uma gama de comandos mapeados sobre as intenções do operador.

As sequências de comandos do operador são armazenadas por este componente onde é atribuída a finalidade a que se destina para que sejam processados posteriormente. Neste momento, os movimentos que comprometem a navegabilidade do SARP serão ignorados.

O Comando Externo Enumerado alimenta o Executor de Decisão e também os módulos de Desejo e Emocional. Tais módulos possibilitam a execução da IH, de forma imediata ou não.

Nas situações em que haja a necessidade de atendimento urgente da IH, como por exemplo, a certeza do sucesso na execução do comando ou a iminência de um ataque, o Índice de Instinto ao Comando (IIC) ficará elevado. O que acarretará no aumento das chances do Executor de Decisão executar a IH proveniente do Comando Externo Enumerado.

4.3.3 Índice de Confiança ao Comando

As informações contidas no Comando Externo Enumerado são enviadas ao componente de Atualização do Índice de Confiança que por sua vez alimentará o Índice de Confiança ao Comando (ICC). O ICC alto é um indicativo para as IH que não precisam ser executadas

imediatamente. No caso do ICC baixo, todo o Módulo Emocional relacionado a IH pode ser opcional na constituição do agente de software BDI.

Caso o ICC esteja alto, indicará ao Gerador de Desejos que a IH deve ser executada a longo prazo. O que acarretará na necessidade da IH estar compatível com os planos originais provenientes do BDI. Tal situação pode ocasionar a possibilidade de conflito da IH com os planos originais.

O ICC, alimentará o Módulo de Tomada de Decisão e o Planejador Realtime usará os parâmetros gerados pelo Índice de Instinto ao Comando (IIC) e ICC para definir as ações da elaboração dos planos. Que posteriormente contribuirá para o cálculo do GO utilizado pelo Executor de Decisão.

Obtenção do Índice de Confiança ao Comando

Para se obter o ICC em um determinado momento (t), onde deve-se considerar a existência dos modos de execução confiável (ICC alto) ou suspeito (ICC baixo) em uma estrutura BDI que foi adaptado por Lee *et al.* em [14] para o calculo do ICC.

O ICC é obtido através da Equação 4.1 atendendo as condições de 4.2 e 4.3.

$$ICC_t = \alpha \cdot e^{-d_t} + (1 - \alpha)ICC_{t-1} \quad (4.1)$$

$$d_t > 0, 0 \leq \alpha \leq 1 \quad (4.2)$$

$$0 \leq ICC_0 \leq 1 \quad (4.3)$$

A variável d_t que basicamente define o valor de ICC é obtida através do desvio do que é previsto (durante a fase de planejamento) e o ocorrido no ambiente real (durante a fase de execução).

Para obter d_t , considera-se icc_t^e é a avaliação de ocorrência de ICC , e icc_t^e é a avaliação estimada da quantidade de passos no percurso planejado em um determinado tempo t . E icc_t^a e icc_p^a são os valores atuais avaliados de ocorrência de ICC e quantidade de passos executados respectivamente em um determinado tempo, então temos a definição de d_t na Equação 4.4.

$$d_t = (icc_t^e - icc_t^a) + (icc_t^e - icc_t^a) \quad (4.4)$$

O parâmetro α ajusta o efeito da confiança ao comando anterior à confiança atual, que varia dependendo do tipo de operador humano.

4.3.4 Índice de Instinto ao Comando

O Comando Externo Enumerado ao alimentar a Atualização do Índice de Instinto, este por sua vez alimenta o IIC. E conforme visto anteriormente, o IIC elevado, é o indicativo da necessidade de execução mais imediata da IH.

Para isso, o IIC alimenta o Executor de Decisão no módulo de Tomada de Decisão e quando estiver baixo, ocorrerá maior chance da IH ser executada mais tarde. Tais IH são as que devem ser executadas com mais cautela e serão processadas pelo Módulo de Desejo.

Uma outra atribuição do Índice de Instinto ao Comando (IIC) é o de servir de parâmetro para a obtenção do GO, que é um submódulo que se encontra no Executor de Decisão.

De forma semelhante a obtenção do ICC, o IIC é obtido através das interações humanas que obtiveram sucesso, nas condições $d_t > 0$, $0 \leq \alpha \leq 1$ e $0 \leq IIC_0 \leq 1$ na Equação 4.5.

$$IIC_t = \alpha \cdot e^{-d_t} + (1 - \alpha)IIC_{t-1} \quad (4.5)$$

Onde a variável d_t que é importante no levantamento do IIC é definida por 4.6.

$$d_t = (iic_t^e - iic_t^a) + (iic_t^e - iic_t^a) \quad (4.6)$$

4.3.5 Grau de Compatibilidade ao Comando Externo

O Gerador de Desejos ao ser alimentado pelo Comando Externo Enumerado e Índice de Confiança ao Comando (ICC), alimentará o Grau de Compatibilidade ao Comando Externo. Que cria uma sequência de Comandos que são executados mais tarde e que minimizem os conflitos da IH com os planos originais do BDI.

Dessa forma, o Grau de Compatibilidade ao Comando Externo alimenta o Planejador Real-Time que além de criar os planos através das intenções, procurará compatibilizar com a IH. O Grau de Compatibilidade do Comando Externo é obtido segundo uma regra pré-estabelecida em relação às ações da IH e das ações possíveis contidas no plano do BDI.

O Grau de Compatibilidade ao Comando Externo, alimenta o Planejador Real-Time que inclui os planos que contêm a informação do quanto a IH é compatível com os planos atuais. Tais parâmetros, serão úteis posteriormente na definição do GO, o que permitirá o Módulo Executor de Decisão, avaliar se deve obedecer a IH ou executar o plano original.

O Grau de Compatibilidade ao Comando Externo funcionará quando o ICC estiver alto. Tais situações são provenientes das IH que não necessitam ser realizadas com urgência.

4.3.6 Grau de Obediência

O GO deve ser estabelecido durante a constituição do agente de software, que é totalmente dependente da arquitetura do sistema autônomo. Cada caso certamente realizará ajustes específicos a este valor, em virtude das peculiaridades de operação dos robôs modelados.

A Figura 4.2 apresenta as dependências utilizadas no levantamento do GO que estão considerados no funcionamento do BDI proposto. Para se calcular o GO, utilizou-se a Equação 4.7.

$$GO_t = (II_t - IIC_t) + (IC_t - IIC_t) \quad (4.7)$$

O GO influi diretamente no comportamento do sistema autônomo, e por causa disso a sua influência deve ser avaliada com cautela, observando o desempenho desejado. Se o objetivo é que o sistema tenha alto desempenho e fluidez autônoma, o GO deve ser o mais estrito possível, ou seja, obedecer pouco. Por outro lado, se o sistema autônomo tiver que ser responsivo e atento à interação do operador, o GO deve ser mais relaxado, ou seja, obedecer muito.

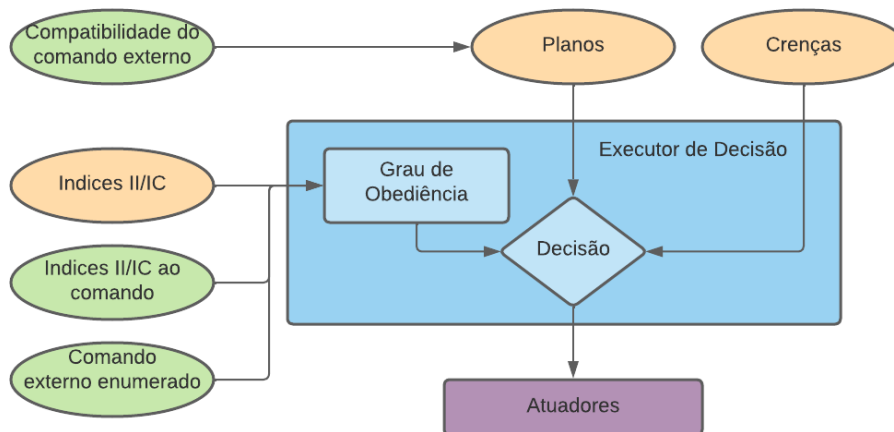


Figura 4.2: Executor de Decisão Contendo o Grau de Obediência.

Para validar o modelo proposto, serão realizados nos Capítulos 5 experimentos utilizando a ferramenta PRISM contendo os componentes Comando Externo do Operador, o Módulo Planos e o Submódulo Grau de Obediência (que se encontra no Módulo Executor de Decisão). Estes Módulos e Submódulos do BDI proposto foram escolhidos, pois estão relacionados diretamente no desempenho do BDI.

4.4 Funcionamento do BDI Proposto

Nos modelos de BDI sem ponderação, quando o operador envia comandos ao sistema semiautônomo, este obedece sem questionar, pois o comando pode atuar somente nos sistemas de controle do SARP, este comando não sendo ponderado pelo BDI. O que aqui se propõe é que o BDI do SARP possa avaliar e decidir se obedece ao comando externo, utilizando-se para isso o levantamento do GO.

Inicialmente, o Comando Externo do Operador apresentado expressa-se em um comando dentre quaisquer comandos possíveis que o BDI possa receber, tendo sido um conjunto finito de comandos mapeados e conhecidos pelo agente de software.

Ao ser recebido pelo Processador Perceptivo, este será convertido em um Comando Externo Enumerado, que por sua vez é uma representação conhecida pelo agente de software para o referido comando.

Com base no Comando Externo Enumerado, o Processador Emocional estabelece dois novos valores para o ICC e IIC, que servem como parâmetros do Módulo Emocional para elementos decisores do processo. O IIC retroalimenta o Executor de Decisão nas funções mais urgentes.

Este índice pode indicar, por exemplo, quantas vezes esse comando foi executado, sua periodicidade e grau de sucesso, mantendo uma memória de curta duração. O ICC vai retroalimentar o Gerador de Desejos.

Caso um Tipo de Interação fora do contexto da missão faça com que o Índice de Confiança fique baixo, este poderá influenciar na criação de novos planos ou alteração de planos existentes através do Grau de Compatibilidade ao Comando. Ainda com base no comando, o Módulo de Desejo deve estabelecer o Grau de Compatibilidade do Comando Externo em relação aos Desejos atuais. Através dessa relação e outros parâmetros, juntos decidirão se o comando externo deve ser obedecido ou não.

O Planejador Realtime, localizado no Módulo de Tomada de Decisão, utiliza os parâmetros gerados para definir as próximas opções de decisão do agente de software. O Executor de Decisão por sua vez, decide qual comando executar. O GO é importante, pois é com base neste valor que o executor decidirá se deve obedecer o comando externo ou se deve executar o plano oriundo do Planejador Realtime.

Capítulo 5

Experimentos

Este Capítulo abordará a aderência dos experimentos estocásticos ao modelo BDI, parâmetros utilizados, equipamento referenciado, experimento envolvendo a perda do SARP, experimentos considerando a IH, resultados sob IH e GO, distância e risco e por fim a discussão.

5.1 Aderência dos Experimentos Estocásticos ao Modelo BDI

A Figura 5.1 apresenta a aderência do modelo BDI proposto ao modelo estocástico utilizado nos experimentos. Onde é possível ver a comparação entre o funcionamento do BDI estendido e o aplicado em um SARP tradicional, onde a subfigura (a) contém o modelo que desconsidera a IH onde o operador atua diretamente no SARP. E a subfigura (b) contém o modelo que considera a IH, passando pela ponderação do BDI, que é justamente o escopo da proposta desta dissertação.

Do lado esquerdo de cada subfigura, há a modelagem do ambiente real onde o operador atua no SARP ou no BDI. E do lado direito, a modelagem do ambiente simulado, onde possui as Cadeias de Markov e a ferramenta de simulação estocástica PRISM. No canto superior de cada subfigura, há o propósito em comum que está compartilhado com a Cadeia de Markov e alguns itens do BDI, que são as intenções, os planos e objetivos.

A extremidade inferior de cada subfigura apresenta a interação do operador ou IH. Fazendo um paralelo entre as duas situações, em que real é o SARP operando sob as diretrizes do BDI e executando as funções de mapeamento de intenções, planos e objetivos, tem-se a simulação, em que o PRISM simula o funcionamento do SARP e o BDI que está mapeado diretamente em suas Cadeias de Markov.

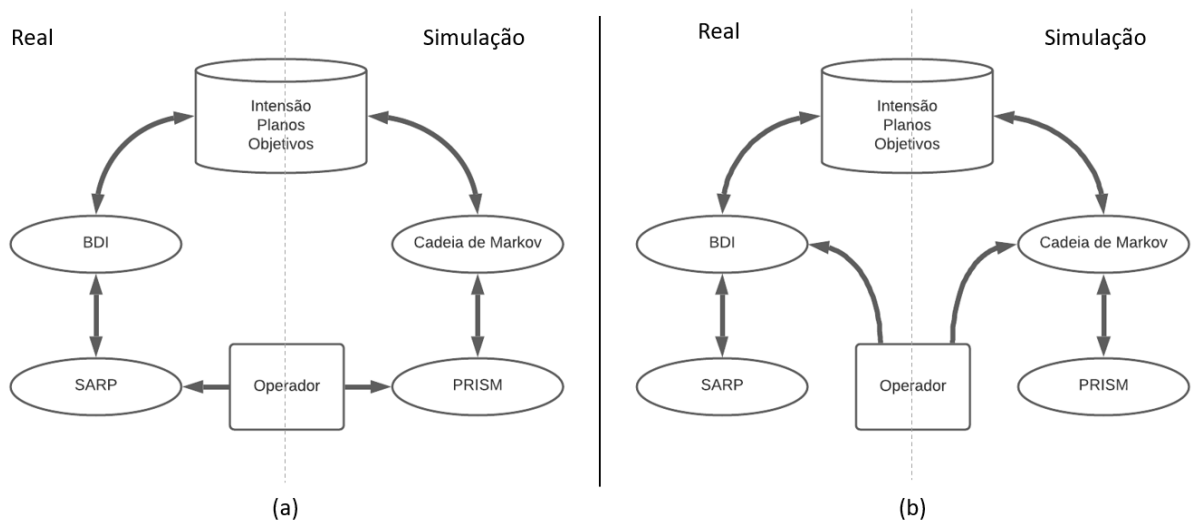


Figura 5.1: Aderência dos Experimentos ao Modelo BDI.

O GO refere-se a probabilidade de obediência que a IH pode assumir. Nos experimentos foi considerado o GO do BDI a IH e procurou-se utilizar todas as possíveis probabilidades (0 a 100 %) que o IH e o GO podem assumir.

5.2 Parâmetros Utilizados

Esta pesquisa realizou a construção dos cenários, que foram apresentados em forma de estudo de caso para demonstrar a sua validade do BDI proposto. E foram utilizados experimentos estocásticos visando identificar o quanto os parâmetros de IH, risco, GO, dentre outros aspectos, influenciam na execução dos planos do SARP. É importante ressaltar que foi considerado que a IH poderá ser ponderada durante a execução dos planos nas situações em que o SARP seja colocado em perigo.

Usou-se Unidade de Distância (UD) como métrica de distância. A UD foi baseada nos parâmetros do SARP considerado, onde 1 UD corresponde à aproximadamente 1 Km. É possível adaptar esta métrica a outros tipos de veículos inteligentes não tripulados. Como por exemplo, 1 UD pode representar 10 ou 100 metros, nos veículos inteligentes marítimos e terrestres respectivamente.

Optou-se por utilizar cenários com menor quantidade de vértices por dois motivos:

- Modelagem de cenários contendo mais vértices podem levar a situações, implementações e interpretações complexas, podendo inclusive perder o escopo dos objetivos desta pesquisa;

- É mais viável adaptar cenários que contenham poucos vértices em outros cenários correlatos. Uma vez que tal medida evitaria as peculiaridades que podem existir nos cenários com mais vértices, possibilitando o seu reuso.

Os cenários encontram-se em forma de grafo e procurou-se representar o mundo real. Como por exemplo a utilização dos principais pontos contidos em uma carta topográfica. Onde é possível extrair informações como distância, relevo, trajetos, dentre outros. O SARP pode percorrer os principais pontos de interesse e que podem ser representadas em forma de grafo.

Os códigos fontes, dados, resultados dos experimentos e o arquivo *readme* encontram-se disponíveis em <https://github.com/riovitoria/Intera-o-Humana>.

O objetivo dos experimentos é ajudar a elucidar e quantificar as consequências das ações da IH e do BDI. Cujos resultados, podem contribuir na administração dos recursos disponíveis de futuras missões SARP ou outras abordagens que utilizam IA relacionadas ao modelo BDI.

5.3 Equipamento Referenciado

Os parâmetros utilizados nos experimentos das Seções a seguir foram obtidos através de [50], estes parâmetros referem-se ao SARP modelo NAURU 500C. Tal equipamento também é comercializado para uso civil, porém é o modelo que foi empregado no treinamento dos futuros operadores do SARP NAURU 1000C adquirido pelo EB em 2021. Estes modelos foram desenvolvidos pela Empresa Brasileira XMobots. A Figura 5.2, cuja fonte foi cedida ao autor, mostra o NAURU 500C realizando a manobra de decolagem do tipo VTOL.

A seguir algumas das principais características do NAURU 500C:

- Plataforma Aérea: híbrido (asa rotativa e asa fixa);
- Envergadura: 3 metros;
- Carga Útil: 7 Kg (sensores e câmeras);
- Autonomia: 4 horas;
- Peso: 25 Kg;
- Alcance: Até 60 Km (alcance máximo de comunicação);
- Velocidade de Cruzeiro: 90 Km/h;
- Estrutura: Poliamida (*kevlar*);

- Lançamento e Pouso: *Vertical Take-Off and Landing* (VTOL);
- Fonte de Energia: Baterias (usado para comunicação e VTOL) e gasolina de aviação;
- Motorização: combustão e elétrico (VTOL);
- Gerador de Planos: Xplanner.



Figura 5.2: SARP NAURU 500C Realizando Manobra de Decolagem Vertical (VTOL).

O Xplanner é um software de código fechado desenvolvido pela empresa XMobots utilizado na operação do NAURU 500C e NAURU 1000C. Na fase de preparação para o voo, são fornecidos muitos parâmetros ao software Xplanner, dentre os quais, podemos destacar, os dados do Sistema de Informações Geográficas (GIS), tipo de missão, configurações da aeronave, tipo de câmera utilizada e área de interesse. É possível também realizar o sobrevoo de diversos pontos (waypoints) em um único plano de voo.

Todas as configurações do Xplanner são realizadas através de uma interface visual, que possibilita a criação de planos de forma rápida, além de possibilitar o acompanhamento dos movimentos do SARP, dados relacionados a comunicações, verificação do relevo, análise meteorológica, direção do vento, dentre outros [51]. Com base nessas informações, o Xplanner cria os planos de voo, que se traduz na capacidade semiautônoma do SARP. Neste trabalho foi considerado que o software Xplanner representa o componente planos do modelo BDI proposto. Considera-se que futuramente, a existência de sistemas mais aprimorados de geração de planos, com a inclusão de mais componentes no contexto do BDI, aumentará a capacidade semiautônoma do SARP.

Considera-se que a IH, que tem a sua origem no componente Comando Externo do Operador do BDI proposto, contrariam os planos de voo do SARP, ou seja, a aeronave receberá as IH através do comando do operador que por sua vez contraria o plano de voo. Em consequência, altera a distância percorrida prevista no percurso do SARP, e essa alteração de distância refletirá no desempenho da missão.

Nos experimentos, foram considerados apenas os parâmetros essenciais para o atendimento dos objetivos propostos desta pesquisa. Abstraindo-se assim de quaisquer outros parâmetros, tais como condições climáticas, direção do vento, velocidade, duração da missão, dentre outros. Com isso, a execução da implementação ficou mais leve e permitiu a ferramenta PRISM a obtenção dos resultados de forma relativamente rápida, possibilitando assim a execução de milhares de voos.

Os motores elétricos do SARP foram projetados apenas para manobras de pouso e decolagem (VTOL), não sendo adequados para pairar por longos períodos de tempo, sob o risco de esgotar a energia das baterias e não conseguir pousar e cair. Com isto, estes motores não poderão ser empregados para realizar manobras do tipo atrasar, para isso, o SARP deverá executar um círculo completo considerando o seu raio limite.

A manobra de atrasar utilizado no modelo de SARP considerado, tem um custo de aproximadamente 1 UD. Em virtude do limite mínimo de curva (raio limite) que o SARP modelo NAURU 500C pode realizar sem o uso de seus motores elétricos e sem perder sustentação conforme demonstrado por Medeiros em [52]. Por outro lado, considerou-se que o raio limite não pode ser excessivamente elevado, para não haver exposição e consumo de combustível exagerados.

5.4 Considerações Sobre os Cenários Utilizados

Na execução dos experimentos, optou-se pelo uso das ferramentas PRISM e PRISM-GAMES para a modelagem dos cenários dos experimentos, pois estas ferramentas são indicadas para modelagens e análises de sistemas que exibem comportamento probabilístico [53]. Nos experimentos, estas ferramentas realizaram as simulações dos movimentos do SARP, utilizando com base na implementação contida nos artigos que foram citados na Seção experimentos estocásticos contido no Capítulo 2.

Considerou-se um cenário hipotético representado em forma de grafo, que possua características militares, extraído de uma carta topográfica de uma zona rural. Desse modo, a furtividade no percurso do SARP é relevante.

Neste percurso, existe a possibilidade de ocorrer eventos adversos que podem impedir, restringir ou dificultar a locomoção do SARP e que precisam ser levados em conta. Assim, o operador terá que realizar interações que poderão influenciar na execução dos planos dos

voos contidos no BDI do SARP. E neste contexto, é importante ressaltar que determinadas informações, obtidas durante a execução da missão, podem estar disponíveis ao operador e não ao BDI. Tais como alertas de segurança, dados dos instrumentos de navegação, condições do equipamento, mudança de objetivos, dentre outros. Neste caso, no modelo proposto, a IH poderá ser plenamente justificada. O BDI através do seu GO poderá ponderar as IH, caso estas interações coloquem em risco a missão.

Para aumentar a furtividade nos deslocamentos, optou-se por utilizar percursos diferentes tanto para ida como para volta. Tal manobras, tem como objetivo reduzir a exposição do SARP ao inimigo e como consequência, é esperado que os deslocamentos do SARP possuam distâncias diferentes para ida e para volta. Foi considerado que o limite de distância percorrida (alcance) do SARP é de 360 UD. Este valor foi obtido através dos parâmetros do SARP de velocidade de 90 Km/h multiplicado pela autonomia que é de 4 horas. Procurou-se também respeitar os limites de comunicação do SARP que é de 60 UD.

5.5 Experimentos Envolvendo Perda do SARP

Esta seção realizará experimentos com o uso da ferramenta PRISM através de cenários que consideram a perda do SARP e o levantamento das distâncias percorridas em relação aos riscos e perdas existentes.

5.5.1 Cenário Utilizado

O cenário utilizado neste experimento foi o da Figura 5.3, que tem como objetivo o SARP se deslocar do vértice A para o vértice B. Este cenário possui no caminho o vértice P que representa o perigo e o vértice S que constitui o Sink (perda do SARP). Os vértices A1, A2, A3 e A4 representam respectivamente o aumento do risco do SARP de ir para o vértice P (perigo).

Nos experimentos foram utilizadas as distâncias e probabilidades iniciais constantes na Tabela 5.1. As distâncias consideradas são apenas para efeito ilustrativos, porém, procurou-se respeitar os limites do SARP que já citados anteriormente.

As probabilidades são consideradas iniciais na condição de risco inexistente no cenário, ou seja, quando se encontra a 0 %. Nos experimentos, este risco aumentará gradualmente, e como consequência aumentará as chances do SARP seguir para o vértice P (perigo).

Percebe-se que a distância percorrida é relativamente alta na aresta pa ($p_a=43$ UD), pois é o local de atuação do IHP. Considerou-se que a atuação do IHP realiza manobras de salvamento custosas com o objetivo do SARP sair do perigo P e se posicionando em seguida no vértice A.

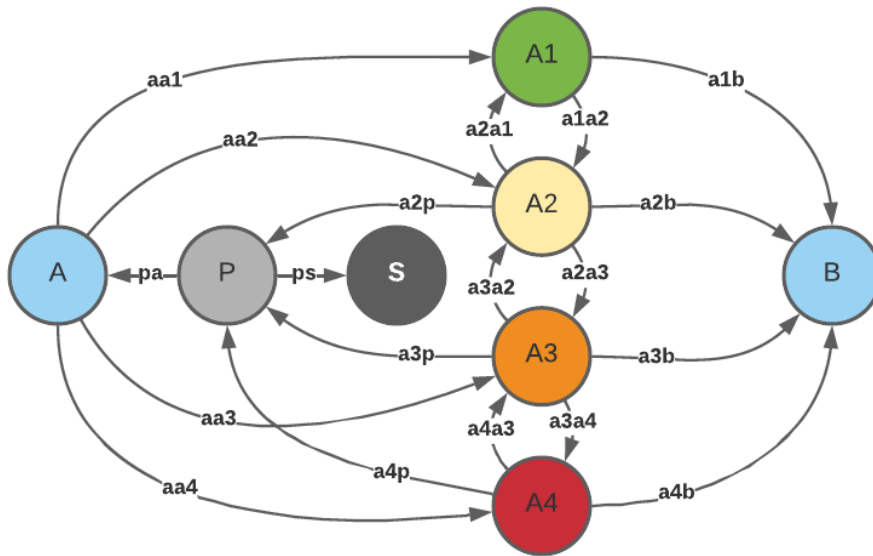


Figura 5.3: Cenário Envolvendo Perda do SARP (*Sink*).

Percebe-se também que a distância percorrida é 0 UD na aresta ps. Pois caso siga este caminho, isto indica que o SARP foi derrubado ou perdido (*Sink*). Neste caso, a atuação do IHP não foi suficiente para o salvamento do SARP.

5.5.2 Plano de Voo

Como dito anteriormente, objetivo do SARP é levantar voo do vértice A, passar pelos vértices que representam o risco e pousar no vértice B. Nos vértices que representam a mudança de perigo, o SARP poderá se deslocar entre eles. Como por exemplo (vértices e arestas): **A -> aa3 -> A3 -> a3a4 -> A4 -> a4B -> B** ou **A -> aa2 -> A2 -> a2a1 -> A1 -> a1b -> B**.

O código fonte que foi implementado na ferramenta PRISM encontra-se no Apêndice A. Em seguida, uma breve descrição dos blocos de programação da referida implementação:

- Bloco A: Inicialmente o SARP ao sair de A encontrará no caminho 4 waypoints A1, A2, A3 e A4. Onde terá inicialmente 25% de chances de seguir para cada um desses;
- Bloco F: O SARP estando no vértice A1, terá 50 % de chances de ir para A2 e 50 % de chance de ir para B. Note que estando em A1, não poderá ir diretamente para S;

Tabela 5.1: Descrição das Distâncias e Probabilidades Iniciais das Arestas.

Aresta	Distância (UD)	Probabilidade Inicial	Partindo do Vértice
aa1	28	25%	A
aa2	32	25%	A
aa3	29	25%	A
aa4	35	25%	A
a1a2	12	50%	A1
a1b	12	50%	A1
a2a1	10	25%	A2
a2a3	15	25%	A2
a2b	14	25%	A2
a2p	12	25%	A2
a3a2	20	20%	A3
a3b	15	20%	A3
a3p	25	20%	A3
a3a4	18	20%	A3
a3a2	20	20%	A3
a4a3	14	33%	A4
a4b	12	33%	A4
a4p	22	33%	A4
pa	43	50%	P
ps	0	50%	P

- Bloco B: Estando em A2, terá 25% de probabilidade de ir para cada um dos vértices: S, A1, A3 e B. É possível observar que A2 representa maior perigo que A1 pois tem mais chance de ir para S (25%) do que para A1;
- Bloco C: Caso esteja em A3 terá 25% de chances de ir para os vértices A2, A4, S e B. É possível ir para A1 sem passar pelo A2. Conclui-se que A3 representa maior perigo que A2;
- Bloco D: No caso de estar em A4, terá 33,3% de chances de ir para S, A3 e B. Representando assim maior perigo de ir para S em relação ao A3.

Com base nessas informações foi considerado como plano de voo normal o trajeto com a menor distância e menor risco do SARP encontrado (vértices e arestas): **A -> aa1 -> A1 -> a1b -> B**, cuja a soma das distâncias contidas nas arestas deste exemplo foi de 40 UD. Esta é a distância percorrida que está estabelecida nos planos do BDI e que será utilizada como base das comparações de desempenho.

5.5.3 Perda do SARP em Função do Risco

Existem diversas formas de perder o SARP, seja pelo esgotamento do combustível, seja por problemas técnicos, seja abatido, dentre outros. Através de dados, como alcance do SARP, é possível estabelecer limites para evitar que o SARP seja perdido por falta de combustível.

As IH podem não ser oportunas, devido a comandos errados ou falhas de comunicação. Conforme visto anteriormente, estes erros podem ser corrigidos pelo BDI através da ponderação da IH.

Neste experimento, chamou-se Interação Humana Ponderada (IHP) a IH que passou pela ponderação do BDI. Com isso, os experimentos realizados nesta Seção foram utilizados como parâmetros: o risco e IHP. Com isso, considera-se o modelo de aderência da Figura 5.1 (b).

A IHP atuará nos casos em que o SARP esteja no vértice P (perigo), e o resultado desta ação definirá se o SARP vai para Sink ou para o vértice A (no início do percurso). Em caso de salvamento, não será contado um novo voo, ou seja, a continuação do voo atual.

Pode-se considera-se como um único voo (vértices e arestas): **A -> aa2 -> A2 -> a2a3 -> A3 -> a3p -> P -> pa -> A -> aa4 -> A4 -> a4b -> B**. Note que o SARP esteve 2 vezes no vértice A (início do percurso), porém contou-se como um único voo.

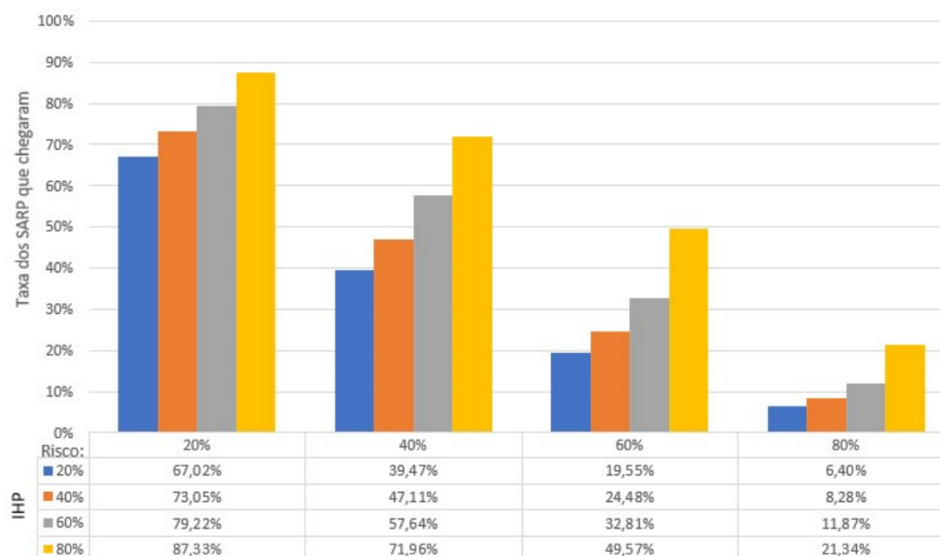


Figura 5.4: Taxa dos SARP que Chegaram ao Destino.

O trecho do código fonte abaixo, extraído da implementação PRISM que se encontra no Apêndice A, mostra a atuação do IHP no vértice P:

```
[UAV] (si=1) & (v<10000) -> (ihp) : (si'=0) & (a'=1) & (d'=d+cihp) +
(1-ihp) : (a'=1) & (si'=0) & (v'=v+1) & (cs'=cs+1) & (ts'=ts+d) & (d'=0);
```

Quando o SARP estiver em perigo ($si=1$), executará as probabilidades de IHP, que no referido código fonte, são os comandos que se encontram depois de ($- >$), na probabilidade de (ihp) reduzirá as chances do SARP de ir para S, enquanto ($1-ihp$) aumentará as chances do SARP ir para S.

Nestes dois casos, o SARP sairá do estado Sink ($si'=0$) e ingressará no vértice inicial ($a'=1$). A variável ($cihp$) contém a distância percorrida em UD, que é o custo do salvamento do IHP. A variável (v), contém a quantidade de voos totais realizados. O vértice P está intrínseco nas execuções das probabilidades da variável (ihp).

Caso o SARP seja salvo, a variável (d) que representa a distância percorrida em UD do voo, não é zerada e a variável (v) não é atualizada. Condições estas que não finalizam o voo atual.

E em caso de Sink, a variável (d) é zerada. E as variáveis de contagem de voo (v), a variável (cs) que representa a contagem geral dos voos que se perderam e a variável (ts) que contém a distância total dos SARP que se perderam são atualizadas. O termo [UAV] no início da linha refere-se ao módulo UAV utilizado na implementação PRISM.

A Figura 5.4 apresenta os resultados das simulações contendo a taxa de SARPs que conseguiram chegar a B (destino). Nos experimento foram realizados 10.000 voos em cada IHP considerado.

Neste experimento foram utilizadas as probabilidades de 20%, 40%, 60% e 80% para o IHP e risco. Os resultados das probabilidades de 0 % e 100 % foram omitidas pois resultaram em excessos de *outliers*. Utilizou-se o nível de confiança a 95% e $N = 10.000$, com estes parâmetros obteve-se o intervalo de confiança menor do que 2,33 % da média das distâncias percorridas pelo SARP.

Conforme esperado nos resultados, a atuação do IHP, que é uma combinação da IH e a ponderação do BDI reduziu o número de perda do SARP, obtendo maior atuação nos riscos maiores. Foi percebido que o risco obteve maior influência que o IHP nos resultados. Isto se deu pelo fato de que nos casos de salvamento do SARP pelo IHP, o SARP é posicionado no vértice A, que fica no início do percurso. Obrigando assim o SARP a passar novamente por todos os riscos existentes no mesmo voo.

Conforme apresentado até o momento e com base nos resultados, pode-se estimar a perda do SARP em função do risco. Tais perdas podem ser reduzidas, bastando ajustar os parâmetros de IHP.

5.5.4 Distância Percorrida em Função do Risco e Perda do SARP

Nesta Seção serão tratadas as distâncias percorridas pelos SARP que conseguiram chegar ao destino e também dos que não conseguiram chegar ao destino.

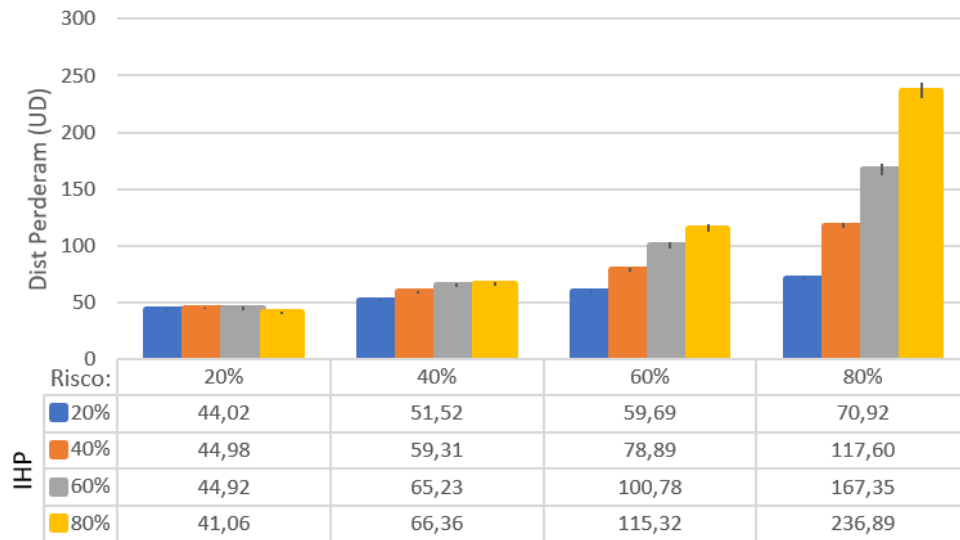


Figura 5.5: Distância Percorrida dos SARP que Chegaram ao Destino.

Com base no IHP e risco, a Figura 5.5 apresenta as distâncias percorridas dos SARP que chegaram ao destino (Vértice B). E a Figura 5.6 contém as distâncias percorrida dos SARP que não chegaram (Vértice S).

Percebe-se que foram maiores os deslocamentos dos SARP que conseguiram chegar ao destino. Tal fato ocorre em virtude da necessidade maior de movimentos dos SARP para completar a missão. As distâncias percorridas pelos SARP que não chegaram ao destino foram menores, demonstrando que quanto maior o risco, mais precocemente o SARP se perderá.

Tanto nos SARP que se salvaram, quanto nos SARP que se perderam, o deslocamentos foram maiores nos riscos maiores. Demonstrando assim, que nos riscos maiores, existe maior probabilidade do SARP seguir para o vértice P. Que por sua vez ocasiona maior quantidade de movimentos custosos de salvamento realizados pelo IHP na aresta PS.

5.6 Experimentos Considerando a Interação Humana

Em complemento aos experimentos da Seção anterior, esta Seção realizou experimentos considerando o Interação Humana Ponderada (IHP) do BDI proposto de forma separada (componentes separados): Comando Externo do Operador e GO.

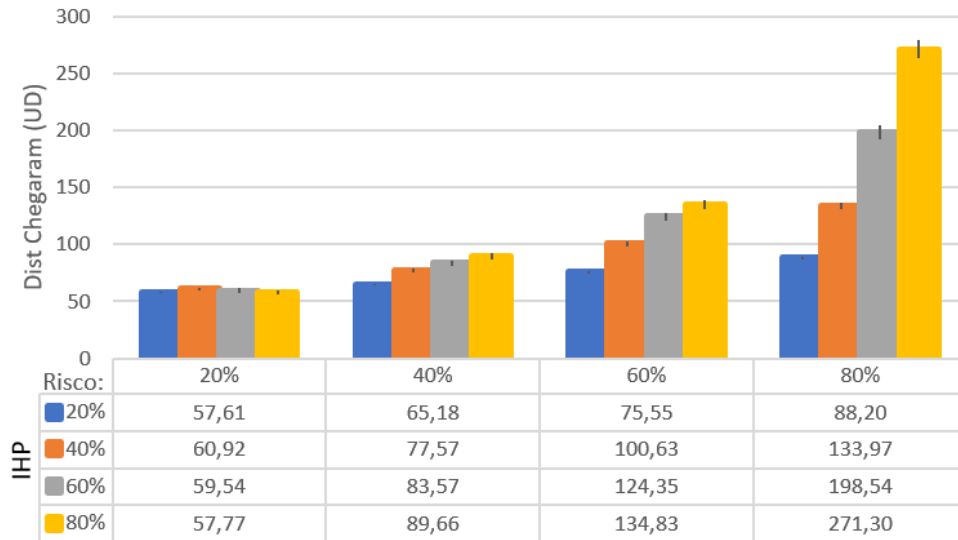


Figura 5.6: Distância Percorrida dos SARP que não Chegaram ao Destino.

Além disso foram utilizados 4 Tipos de Interações Humanas (TIH). Semelhante ao da Seção anterior, objetivo é o de quantificar a distância percorrida utilizando os componentes do BDI citado realizados de forma separada.

O cenário utilizado está modelado em forma de grafo na Figura 5.7, onde os vértices são os principais pontos de interesse e as arestas possuem os custos dos percursos em UD.

Os principais pontos de interesse são: base de apoio (B): onde se encontra a logística de operações; waypoint (W): local determinado onde se encontra o Objetivo; rota alternativa (R): é uma segunda opção de se chegar a W; fotografia (F): é o local de reconhecimento em que será fotografado que se encontra dentro de W; pouso e decolagem (G) do SARP que fica em B. As distâncias utilizadas nas arestas encontram-se na Tabela 5.2. E os TIH utilizados são:

- Voltar (VO): Retornar ao vértice anterior. Por exemplo, o SARP pode não fotografar e desistir da missão seguindo o trajeto (ação da IH em negrito): G -> B -> **W** -> **B** -> G ou o SARP ou fotografar 2 vezes: G -> B -> W -> **F** -> **W** -> **F** -> W -> B -> G;
- Desviar (DE): Seguir para o vértice diferente do que estava previsto em seu plano de voo. Por exemplo, o trajeto pode estar em perigo e o SARP ter que seguir por outro caminho G -> **B** -> **R** -> W -> F -> W -> B -> G;
- Atrasar (AT): Realizar manobra para aguardar um determinado tempo em um local (realizar uma volta de atraso com a aeronave). Por exemplo, a aeronave pode aguardar até que o caminho bg (aresta) esteja seguro: G -> **B** -> **B** -> G;

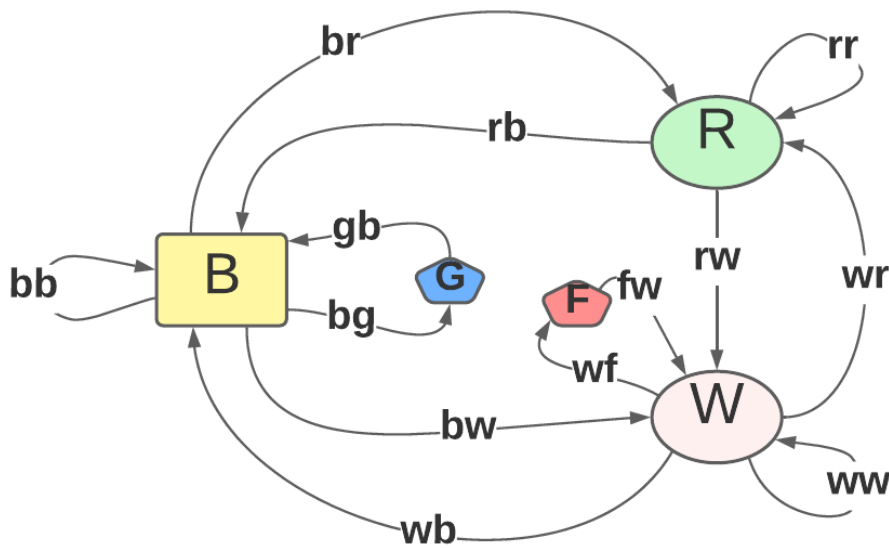


Figura 5.7: Cenário do Percurso do SARP.

- Todos (TO): Qualquer um dos 3 TIH descritos acima. Caso exista a probabilidade de 30 % de ocorrência do TO. Então a probabilidade VO, DE e AT será de 10% cada.

Ressalta-se a existência do evento SE que significa seguir em frente. Ou seja, seguir com os planos, e este evento ocorre quando temos ausência de IH. A probabilidade de ocorrência do evento SE pode ser calculada pelo desvio de 100 e probabilidade de interação. Ou seja, caso a interação AT seja de 10%, a probabilidade SE será de 90 % (100 - 10).

O TO é a probabilidade de ocorrer os tipos de interações humanas VO, DE e AT na mesma proporção de probabilidade. Considerou-se que TO é o TIH mais próximo do real, uma vez que é mais provável que o operador utilize TIH diferentes compatíveis com o tipo de situação encontrada.

Com o intuito de simplificar, considerou-se nos experimentos que a IH só ocorre nos vértices (B, W e R), não ocorrendo nas arestas (durante o percurso do SARP). Por exemplo, a IH não ocorre na aresta bw, e sim quando o SARP chegar ao vértice W. Não haverá IH nos vértices F e G, em virtude de suas características peculiares em relação aos outros vértices. Nestes vértices ocorre apenas o evento SE, pois uma vez estando no vértice F (Fotografia), o SARP só poderá ir para o vértice W. E estando no vértice G (pousado), o mesmo, só poderá ir para o vértice B.

Poderão ocorrer mais de uma IH em um único voo, por exemplo, o SARP ao chegar no vértice B pode receber IH para voltar para o vértice W e receber outra IH e não pousar

Tabela 5.2: Descrições e Distâncias das arestas.

aresta	UD	Descrição
bw	18	Base até Waypoint.
br	25	Base até Rota Alternativa.
rb	20	Rota Alternativa até a Base
rw	07	Rota Alternativa até Waypoint.
wr	10	Waypoint até Rota Alternativa.
wb	22	Waypoint até a Base.
bb, ww e rr	01	Atrasos na Base, Waypoint e Rota Alternativa.
gb, bg, fw e wf	00	Pouso e Decolagem, Ida e Volta da Fotografia.

no vértice B. Em seguida ir receber outra IH e ir para o vértice R (rota alternativa).

Considerou-se que o objetivo da missão é o SARP levantar voo estando no vértice G, seguir para B, seguir para W e fotografar em F. Após fotografar, seguir o caminho contrário e pousar em G. Com base nessas informações o caminho contido no plano normal que se encontra no BDI do SARP, chamado de plano normal, é o seguinte percurso (vértices e arestas): **G -> gb -> B -> bw -> W -> wf -> F -> fb -> B -> bg -> G**.

A soma das distâncias contidas nas arestas deste exemplo foi de 40 UD. Esta é a distância percorrida que está estabelecida nos planos do BDI e que foi utilizada como base das comparações de desempenho.

5.6.1 Resultados Sob Interação Humana

Esta Seção realizou levantamentos quantitativos das distâncias percorridas com os parâmetros de IH em relação ao plano de voo preestabelecido. Neste primeiro experimento não foi utilizada a ponderação do BDI. Foi utilizado o evento SE e os TIH: VO, DE, AT e TO. As Probabilidade de Interação Humana (PIH) utilizadas foram de 10 %, 20 % e 30 %. O modelo de aderência utilizado nesta Seção foi o da Figura 5.1 (a).

Para que os resultados não sofram influência do cenário, as distâncias de 4 arestas foram variadas, utilizando 11 valores diferentes (5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5), incrementada nas arestas bw e wb e decrementadas nas arestas br e wr. Dos quais resultaram em 11 cenários com distâncias diferentes. Espera-se que os resultados dos experimentos acompanhem a variação do cenário, decrementado 2 UD para cada aumento de 1 UD na variação.

O código fonte deste experimento encontra-se no Apêndice B. A seguir estão descritos os principais Blocos de programação da linguagem PRISM:

- Bloco A: Definição das distâncias constantes na Tabela 5.2;
- Bloco B: Cálculos das distância de cada voo através do uso da função *rewards*;

- Bloco C: Definições dos TIH que podem ocorrer nos vértices. Tudo que estiver do lado esquerdo da seta (\rightarrow), representa a condição que deve ser atendida, como por exemplo: ($g=1$) o SARP está pousado, ($w=1$) o SARP encontra-se no vértice W ou ($r=1$) o SARP está na rota secundária R. Do lado direito da seta (\rightarrow), temos a representação das probabilidades dos TIH que poderão ocorrer, como por exemplo: seguir em frente ($1-p-v-t$), a condição de ir para a base e pousar ($g=1$), e desviar (p) que faz com que o SARP siga para a rota secundária ($r=1$), voltar (v) que faz o SARP volte para a base (b) e atrasar (a) que faz o SARP permanecer um certo tempo no mesmo lugar.

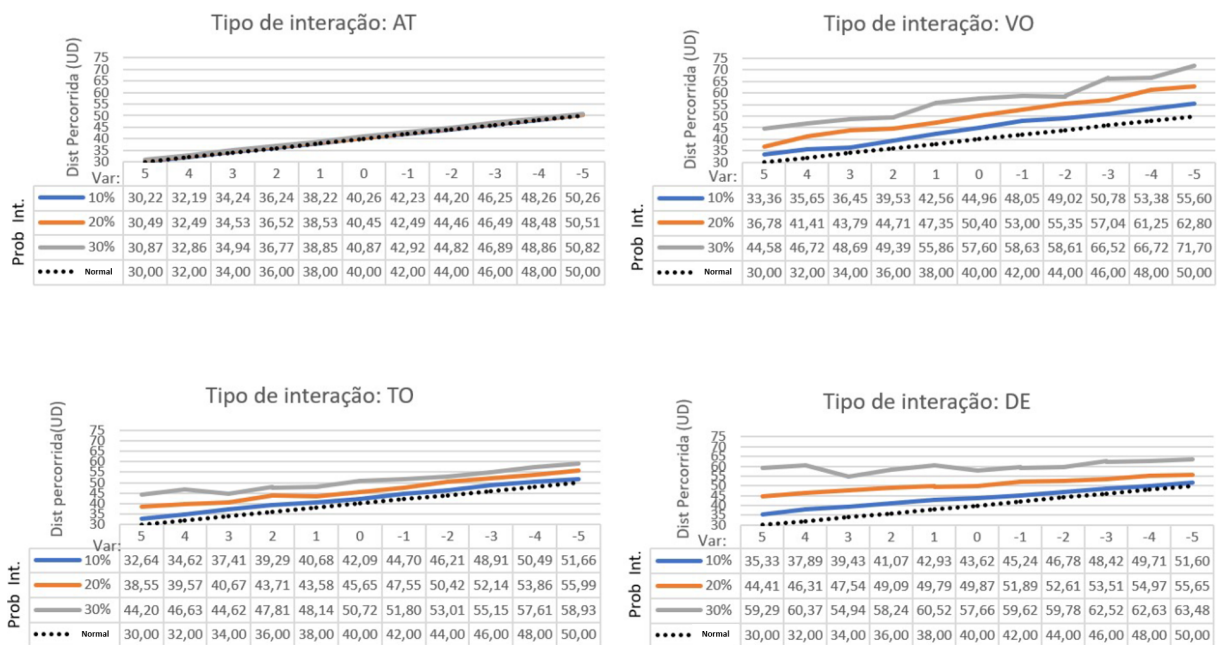


Figura 5.8: Distância Percorrida em Função da Interação Humana e Variação do Cenário.

Neste experimento foram utilizadas amostras contendo 500 voos cada, que totalizam mais de 350 mil estados no PRISM. Cada estado representa um movimento do SARP. A Figura 5.8 mostra os resultados separados por TIH (AT, VO, DE e TO) em cada subfigura. As legendas contidas em cada subfigura são as PIH de 10 %, 20 % e 30 %. A linha pontilhada representa a distância contida no plano de voo normal. O eixo X mostra a variação do cenário de 5 a -5 UD. O eixo Y apresenta as médias das distâncias percorridas em UD. Note que o eixo X estando em 0 (zero), a linha pontilhada apresenta a distância de 40 UD, que representa a distância percorrida no plano de voo normal. Neste experimento utilizou-se o nível de confiança a 95% e $N = 500$, e obteve-se um intervalo de confiança menor do que 2,38 % da média das distâncias percorrida.

Observa-se nos resultados, que a existência da PIH alterou as distâncias percorridas em relação ao plano de voo normal. E quanto maior a PIH, maior será a distância percorrida. Uma vez que o plano de voo normal, trata-se da trajetória ótima, ou seja, a de menor percurso.

Os VO e TO apresentaram resultados semelhantes e AT obteve as menores distâncias. O DE obteve as maiores distâncias percorridas. Pois o DE exigiu mais movimentos do SARP. Observou-se também, conforme esperado, que a evolução dos resultados acompanhou a variação do eixo X (- 5 a + 5). Verificou-se que a variação do cenário não afetou significativamente os resultados esperados, que acompanharam a tendência da linha pontilhada (plano de voo normal).

5.7 Resultados Sob Interação Humana e Grau de Obediência

Esta Seção utilizou os mesmos TIH e o cenário da seção anterior. E foram incluídos: Grau de Obediência (GO) do BDI do SARP, já explicitado no Capítulo 4, que tem a função de ponderar a IH; Maiores faixas de valores de PIH e GO, procurou-se utilizar de 0 a 100% (escala de 10%); Uso da ferramenta PRISM-GAMES no modo Jogos Multijogador Estocásticos Baseados em Turnos (SMG).

A ferramenta utilizada neste experimento foi o PRISM-GAMES, que é uma extensão da ferramenta PRISM com a capacidade de multijogador. A ferramenta foi configurada no modo *Turn-Based Stochastic Multi-Player Games* (SMGs) que possibilitou a implementação simultânea dos componentes do BDI proposto: planos, IH e ponderação do BDI com 3 módulos distintos.

Com o intuito de simplificar o experimento, a decisão do BDI do SARP é de sempre manter o plano de voo normal caso não obedeça o ser humano. Ou seja, quando o SARP não obedecer o ser humano, não se interrompe a sequência de execução do plano contido em seu BDI.

O modelo de aderência da simulação ao BDI que foi utilizado nesta Seção foi o da Figura 5.1 (b). Ou seja, a IH foi ponderada pelo BDI, que atende aos objetivos propostos por esta pesquisa. O código fonte da implementação encontra-se no Apêndice C e abaixo uma breve descrição de seus principais blocos de programação da linguagem PRISM-GAMES:

- Bloco A: Os *players* no modo de operação SMG, contendo os 3 módulos de jogadores estocásticos: P1 (planos), P2 (IH) e P3 (GO);

- Bloco B: Contém o plano de voo no módulo P1 do PRISM-GAMES. Com as variáveis de controle e as definições dos planos de voo quando o SARP encontra-se e posicionado em um determinado vértice;
- Bloco C: Contém a ponderação do BDI do módulo P3, onde foram mapeadas as decisões nos vértices B, W e R. A sua principal função é desconsiderar a IH quando necessário;
- Bloco D: Contém a IH que se encontra no módulo P2 do PRISM. O TIH ocorre nos vértices e são representadas pelas variáveis (b), (w) e (r). Após a execução do TIH, esta interação poderá ser ponderada pelo BDI, cuja comunicação é realizada pela variável (e).

Neste experimento foram utilizadas amostras contendo mil voos cada, dos quais resultaram em mais de 8 milhões de estados nas cadeias de Markov no PRISM-GAMES. O número de estados foi substancialmente maior do que o experimento da Seção anterior em virtude das maiores faixas de probabilidade utilizadas. A Figura 5.9 apresenta os resultados dos gráficos 3D em cada TIH. Este gráfico possui no Eixo Z as distâncias percorridas e nos Eixos X e Y as GO e IH, respectivamente. A visualização por faixas de cores ajuda no estabelecimento dos limites aceitáveis das GO e IH. Por exemplo, a cor azul utilizada no referido gráfico, representa os limites inferiores e superiores da distância percorrida que é de 40 a 90 UD.

É possível observar na Figura 5.9 que quando o GO, está próximo a 0 %, o SARP tende a percorrer a distância contido no seu plano de voo normal, que é de 40 UD. Ou seja, caso o BDI não obedeça a IH, esta fará com que o SARP percorra o que está estabelecido em seu plano de voo normal, que é de 40 UD. Frequentemente o SARP percorreu distâncias acima do seu alcance que é de 360 UD no DE, e PIH acima de 70 %. Com isso, a PIH foi limitada a 90 %. Neste experimento utilizou-se o nível de confiança a 95% e $N = 1000$, e obteve-se um intervalo de confiança menor do que 6,05 % da média das distâncias percorridas.

O uso da PIH até 70 % com o DE e até 80 % no VO e TO e AT apresentaram resultados que não ultrapassam o alcance do SARP em praticamente todas as GO. Permanecendo estáveis em praticamente todos os experimentos. Com base nos TIH utilizados, foram obtidos os seguintes resultados: O DE, obteve o maior aumento da distância percorrida; Enquanto o TO apresentou resultados próximos ao de VO; O AT foi o que menos aumentou a distância percorrida; O DE fez o SARP percorrer mais de 20 UD em cada voo nas IH maiores que 70%. Com isso, o DE causou aumento significativo nas distâncias percorridas. Frequentemente ultrapassando até mesmo o limite de autonomia do SARP, que é de 360 UD.

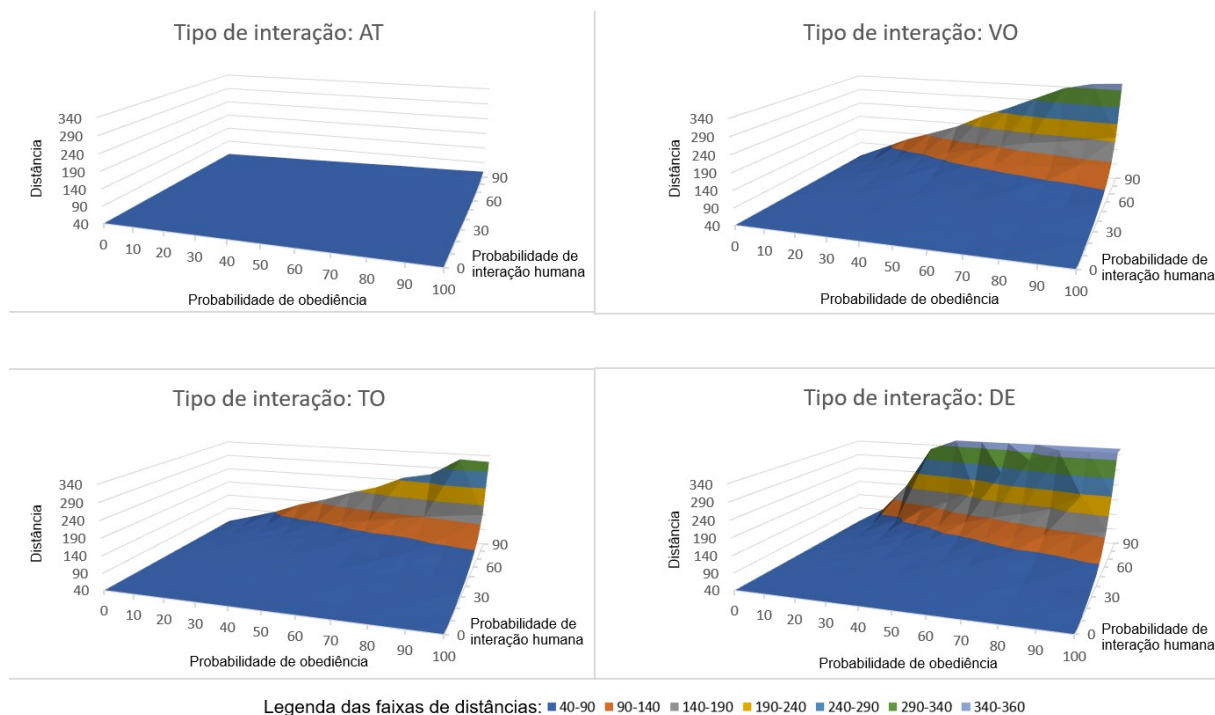


Figura 5.9: Distância Percorrida em Função da Interação Humana e Grau de Obediência.

Com os resultados desta Seção, o operador pode selecionar uma combinação de valores envolvendo os riscos encontrados e o objetivo da missão para a criação de um plano de voo ideal. Pode-se observar que diferentes TIH têm diferentes curvas no gráfico. Com base nestes parâmetros, pode-se inclusive prever o desempenho da missão comparando as curvas dos gráficos para ajustar o BDI do SARP com o intuito de garantir a viabilidade do cumprimento da missão.

É possível observar na Figura 5.9 que quando o GO está próximo a 0 %, o SARP tende a percorrer a distância pré-estabelecida contido no seu plano voo normal, que é de 40 UD. Ou seja, caso o BDI não obedeça a ação humana, este fará com que o SARP percorra o que está estabelecido em seu plano de voo normal.

Verifica-se que o DE demonstra a mudança mais dramática das curvas: parece que, a escolha do DE é mais custosa em praticamente todos os GO e IH utilizados. Quanto mais a curva se torna íngreme para o canto superior direito, maiores as chances do aumento na distância percorrida e, conseqüentemente, os custos associados. e Verificou-se que os resultados não foram tão dramáticos no AT, indicando que esta IH foi menos custosa em relação aos outros TIH.

5.8 Distância e Risco

Nas situações em que existe um conhecimento prévio dos riscos da missão associados ao deslocamento do SARP, é possível estimar a distância acrescida utilizando os resultados dos experimentos abordados.

Parte-se da premissa que toda ação humana racional é uma resposta ao risco encontrado, com isto, pode-se relacionar a PIH ao risco encontrado no percurso. Como resultado, haverá aumento da distância percorrida, proporcionando maiores chances de sobrevivência do SARP ou anulando os efeitos dos riscos encontrados no percurso.

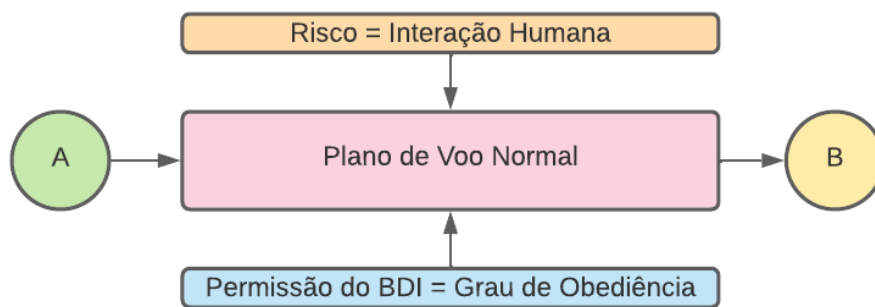


Figura 5.10: Parâmetros que Influenciam o Plano de Voo Normal.

Na modelagem da Figura 5.10 é mostrado que o deslocamento do SARP do ponto A para o ponto B é afetado pelo risco e pela obediência do BDI do SARP. Cujos objetivos são realizar estimativas das distâncias acrescidas em função do risco. Tais estimativas podem ajudar no planejamento no levantamento dos parâmetros e recursos necessários para o cumprimento da missão.

Através desses resultados é possível estimar, por exemplo, o combustível necessário para realizar uma determinada missão, o tempo de utilização do SARP, o desgaste de seus componentes internos, dentre outros. Tais parâmetros são importantes na organização de missões e elaboração dos planos. Com essas informações, é possível realizar customizações no GO do BDI ou até mesmo selecionar as características do SARP mais adequadas para determinados tipos de missões.

De posse dos resultados apresentados na Seção anterior, realizando um estudo de caso que considere que um determinado SARP precise percorrer a distância de 40 UD para cumprir a missão.

E que no trajeto, exista o risco de 80 % de ser abatido. Em resposta a este risco, o operador utilize o TO. De posse desses parâmetros e da tabela citada, pode-se concluir que o SARP deverá percorrer 112,04 UD para atender o risco contido no cenário proposto.

Ou seja, o SARP deve possuir autonomia de 280 % em relação ao planejado inicialmente que era de 40 UD para atender ao risco do percurso.

		Distâncias Percorridas (UD)										
		Grau de Obediência										
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Interação Humana	0%	40,00	40,00	40,00	40,00	40,00	40,00	40,00	40,00	40,00	40,00	40,00
	10%	40,00	40,33	40,55	40,79	41,24	41,37	41,72	41,74	42,25	42,22	42,65
	20%	40,00	40,57	41,10	42,04	42,18	43,46	44,11	44,02	44,54	46,11	45,15
	30%	40,00	40,77	42,18	42,98	44,61	44,86	46,25	47,47	47,77	49,82	51,63
	40%	40,00	41,63	43,38	45,82	46,89	49,27	50,04	51,51	52,10	55,06	57,02
	50%	40,00	42,60	44,77	47,55	50,40	52,49	54,82	60,21	62,04	63,27	68,08
	60%	40,00	44,37	48,29	50,65	54,67	60,50	63,62	67,36	75,63	79,45	85,30
	70%	40,00	46,19	52,10	57,90	64,46	71,83	80,55	86,13	92,97	103,01	111,42
	80%	40,00	49,91	60,69	72,99	86,63	97,56	112,04	126,47	139,93	151,22	161,87
	90%	40,00	65,36	91,73	125,84	151,05	182,17	208,54	248,68	266,71	322,99	319,73

Figura 5.11: Distâncias Percorridas Considerando a Probabilidade de Ocorrência de TO e GO.

Ressalta-se que os dados contido na Tabela da Figura 5.11 foram extraídos dos resultados da Figura 5.9 no TO. Os dados dos experimentos deste trabalho encontram-se no *gitHub* mencionado.

5.9 Discussão

O esperado foi trazer um modelo estocástico através da ferramenta PRISM que possua opções de configurações. Em que um determinado plano e objetivo final seja mantido, em detrimento das ações do operador ocorridas. Podendo se utilizar das configurações de GO para estabelecer os limites do SARP.

Um fator percebido é que a quantidade do número de estados da simulação está relacionado ao aumento da PIH, que aumenta, na maioria dos TIH utilizados, de forma exponencial quando o PIH encontra-se próximo a 100 %. Fazendo com que o SARP frequentemente percorra distâncias maiores que o limite do seu alcance. Além disso, a implementação dos experimentos na ferramenta PRISM se tornou mais complexa com o aumento do número de variáveis consideradas.

O PRISM devidamente configurado pode realizar milhares de voos em um tempo relativamente curto, podendo inclusive fazer uso das vantagens oferecida pelo Método de Monte Carlo (MMC). Que se baseia em amostragens aleatórias massivas para obter resultados mais precisos.

De acordo com os resultados, é possível afirmar que, independente do GO, é mais compensatório em termos de custos utilizar probabilidades de até 70% no DE e até 80 %

no VO ou TO. As distâncias percorridas no AT obtiveram valores mais estáveis em todas os GO e PIH, em virtude de seu baixo custo 1 UD considerado no experimento.

Com base nos resultados, foi possível observar que a IH e o GO podem alterar a distância percorrida pelo SARP, sendo possível mensurá-la. Apesar de não poder garantir que o efeito será sempre negativo, visto que o operador ou o BDI pode tomar uma atitude que favoreça o cumprimento da missão, um em complemento ao outro.

Conforme relatado neste capítulo, no contexto em que o operador está razoavelmente treinado, e só irá interferir no sistema autônomo se realmente for necessário. Nesta situação, considera-se que as ações do operador são benéficas, então pode-se relacionar as IH aos riscos encontrados no trajeto do SARP.

Foi possível mensurar as perdas e os custos através das distâncias percorridas envolvidos nos TIH utilizados, diversas probabilidades e uso do GO envolvendo experimentos contendo parâmetros dos SARP utilizados pelo EB. Tais parâmetros foram tratados individualmente, e os resultados podem ajudar no planejamento de futuras missões envolvendo SARP.

Capítulo 6

Conclusão

Com base nos resultados dos experimentos, foi constatado que a IH altera as distâncias percorridas constante nos planos contidos no BDI e fica evidente que em todas as interações do operador, o desempenho relacionado à missão é afetado, o que por si só justifica esta pesquisa.

A combinação de tarefas críticas e específicas, cronograma, restrições de tempo, dentre outros, aumenta o risco de falha da missão (particularidades que ocorrem sob a combinação de fatores negativos). Numa operação com o uso de SARP podem existir questões como distância, interferência nos canais de comunicação, mudanças bruscas nas decisões, obstáculos extraordinários. Além de outras limitações já citadas, que podem afetar a decisão humana. Em suma, ao permitir que o agente de software utilize a sua autonomia de decisão considerando as crenças e desejos preestabelecidos, o risco de falha por comandos inadequados ou mesmo interferências hostis podem ser minimizados pelo modelo BDI proposto.

É importante frisar que foram utilizadas modelagens sem considerar fatores como terreno, condições climáticas, combustível, dentre outros. Considerou-se apenas o deslocamento do SARP para simplificar e tornar exequível a simulação estocástica. Outros fatores podem ser inseridos nos experimentos de acordo com a necessidade, porém não sendo o foco desta pesquisa.

É comum nos simuladores convencionais exigir diversos detalhes na construção de cenários. Tais como, consumo de combustível, elevações do relevo, direção do vento, dentre outros. Assim, necessitam de muito tempo e pessoal especializado na sua construção. Neste trabalho, as implementações PRISM utilizaram apenas os parâmetros estritamente necessários e resultaram em implementações que possibilitaram a sua verificação quanto à sua correção.

Foi utilizado a ferramenta PRISM-GAMES no modo *Turn-Basead Stochastic Multi-Player Games* (SMG) contendo três jogadores. Nesta configuração pode ser usado para

simular planos de voo, IH e GO do modelo de agente de software BDI proposto, que resultaram no levantamento quantitativo de perda do SARP e distâncias percorridas. Tais experimentos podem ser úteis para futuros planejamentos de missões SARP.

Levando-se em conta o GO, constatou-se que nas probabilidades de 100% as simulações assemelhavam-se às características de HITL. Nas probabilidades entre 0 % a 100 % às características de HOTL. E o GO a 0% às características de HOOTL.

Embora não se possa garantir que o efeito seja sempre negativo, pois o operador também pode tomar uma ação que favoreça o cumprimento da missão. O fato de haver consequências no desempenho já justifica estudar o assunto em busca de melhorias.

Esta pesquisa trouxe um modelo de agente de software baseado em BDI que tivesse opções de configurações, ajustes e sintonia fina, em que o objetivo final seja mantido em detrimento das ações do operador externo. A IA pode possuir autonomia ou algum nível de autonomia para decidir aquilo que considera ideal para a missão ou o objetivo que ela tem que cumprir.

Nos experimentos que envolveram a perda do SARP e risco do ambiente, foi verificado que a ponderação da IH utilizando o Grau de Obediência (GO) do Modelo de Software Crenças, Desejos e Intenções (BDI) proposto, aumentou as chances de sobrevivência do SARP ou do cumprimento da missão.

As sugestões apresentadas por este trabalho em relação à abordagem da construção de uma proposta de modelo de BDI que considera a ação do operador, que por si só é novidade, contribui para o aprimoramento das técnicas de desenvolvimento de novos sistemas. Os experimentos apresentados nesta pesquisa, podem ajudar nos levantamentos quantitativos da influência da IH.

6.1 Limitações

O objetivo central desta pesquisa foi a construção de um modelo conceitual a fim de se verificar a sua viabilidade e implementação. Buscou-se atender a caracterização do cenário real utilizando diversos parâmetros de operação de SARP. Porém não foi possível garantir completamente a abrangência da proposta em virtude dos cenários limitados e a complexidade de implementação que envolveram a simulação BDI.

Diversos atributos que podem alterar os resultados não foram considerados, como velocidade do vento, obstáculos, condições meteorológicas, dentre outros. Tais atributos fornecem um ambiente mais próximo do real à simulação.

6.2 Trabalhos Futuros

Este trabalho pode servir de base para uma série de extensões e direções interessantes. Abaixo serão destacados alguns assuntos futuros que são considerados particularmente relevantes e que são semelhantes a este tema.

Pode-se utilizar parâmetros e cenários mais expressivos com simulações contendo rotas mais detalhadas. Como por exemplo, incertezas relacionadas ao tempo decorrido, utilização de outros parâmetros do SARP, defeitos, condições climáticas, dentre outros.

Além disso, pode-se estudar modelagens do BDI considerando as características peculiares dos sistemas multiagentes relacionados ao comportamento de agentes individuais em busca de um objetivo em comum, além do uso de outros TIH não citados neste trabalho.

Com o intuito de preencher lacunas relacionadas às simulações de BDI, experimentos utilizando outros componentes e subcomponentes do BDI que não foram abordados por este trabalho são bem vindos. Sugerem-se pesquisas utilizando outros Módulos do BDI, tais como o Planejador Realtimer, Crenças e Executor de Decisão.

Referências

- [1] Ismail, Rakshana Mohamed, Senthil Muthukumaraswamy e A. Sasikala: *Military Support and Rescue Robot*. Em *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, páginas 156–162. Institute of Electrical and Electronics Engineers Inc., maio 2020, ISBN 9781728148762. 1
- [2] *Army Robots: Two Contracts Forward, One Contract Back* « *Breaking Defense - Defense industry news, analysis and commentary*. <https://breakingdefense.com/2020/01/army-robots-two-contracts-forward-one-contract-back/>, acessado em 03/11/2022. 1
- [3] *Army's Robot Combat Vehicles: See the Family of Armed Robot Tanks*. <https://www.popularmechanics.com/military/weapons/a34659199/army-family-robotic-tanks/>, acessado em 01/11/2022. 1
- [4] *DefesaNet - Tank - Frederico Aranha - Drones vs. Blindados não é um simples jogo*. <https://www.defesanel.com.br/tank/noticia/38736/Frederico-Aranha----Drones-vs--Blindados-nao-e-um-simples-jogo/>, acessado em 12/08/2021. 1
- [5] Muhammad, Shahabuddin: *Modeling operator performance in human-in-the-loop autonomous systems*. *IEEE Access*, 9:102715–102731, 2021. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9490234>. 2, 15, 16, 31
- [6] Bratman, M.: *Intention, plans, and practical reason*. Harvard University Press, Cambridge, MA, 1987, ISBN 9780674458185. <http://books.google.de/books?id=I0nuAAAAMAAJ>. 2, 7, 8, 11, 25
- [7] 'Robôs assassinos': o perigo das armas autônomas letais | *Nexo Jornal*. <https://www.nexojornal.com.br/ensaio/2021/%E2%80%98Rob%C3%B4s-assassinos%E2%80%99-o-perigo-das-armas-aut%C3%B4nomas-letais>, Acessado em 05/03/2022. 3
- [8] Lacerda, Daniel, Aline Dresch, Adriano Proença e José Antonio Valle Antunes Júnior: *Design Science Research: A research method to production engineering*. *Gestão e Produção*, 20:741–761, dezembro 2012. 5
- [9] Wooldridge, Michael: *Intelligent Agents: The Key Concepts*. LNCS, páginas 3–43, 2001. https://link.springer.com/chapter/10.1007/3-540-45982-0_1. 7, 8, 18, 19, 25

- [10] Kim, Sojung, Hui Xi, Santosh Mungle e Young Jun Son: *Modeling human interactions with learning under the extended belief-desire-intention framework*. Em *62nd IIE Annual Conference and Expo 2012 ; Conference date: 19-05-2012 Through 23-05-2012*, páginas 2861–2870, 2012. 7, 10
- [11] K. C., Ujjwal e Jacques Chodorowski: *A Case Study of Adding Proactivity in Indoor Social Robots Using Belief–Desire–Intention (BDI) Model*. *Biomimetics*, 4(4), 2019, ISSN 2313-7673. <https://www.mdpi.com/2313-7673/4/4/74>. 7, 20
- [12] Rao, Anand S e Michael P George: *Bdi agents: From theory to practice*. Em *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, páginas 312–319, 1995. <http://www.agent.ai/doc/upload/200302/rao95.pdf>. 7, 8
- [13] Archibald, Blair, Muffy Calder, Michele Sevegnani e Mengwei Xu: *Modelling and verifying bdi agents with bigraphs*, 2021. 7, 9
- [14] Lee, Seungho e Young Jun Son: *Integrated human decision making model under belief-desire-intention framework for crowd simulation*. Em *2008 Winter Simulation Conference*, páginas 886–894, 2008. 7, 11, 12, 40, 41, 43
- [15] Noorunnisa, Salma, Dennis Jarvis, Jacqueline Jarvis e Marcus Watson: *Application of the GORITE BDI framework to human-autonomy teaming: A case study*. *Journal of computing and information technology*, 27(1):13–24, 2019. 7, 8, 9, 25
- [16] Simari, Gerardo I. e Simon D. Parsons: *Markov Decision Processes and the Belief-Desire-Intention Model: Bridging the Gap for Autonomous Agents*. Springer Publishing Company, Incorporated, 2011, ISBN 1461414717. 8, 16, 17, 20, 24
- [17] Abar, Sameera, Georgios K. Theodoropoulos, Pierre Lemarinier e Gregory M.P. O’Hare: *Agent based modelling and simulation tools: A review of the state-of-art software*. *Computer Science Review*, 24:13–33, 2017, ISSN 1574-0137. <https://www.sciencedirect.com/science/article/pii/S1574013716301198>. 9
- [18] Kim, Sojung, Santosh Mungle e Young Jun Son: *An agent-based simulation approach for dual toll pricing of hazardous material transportation*. Em *2013 Winter Simulations Conference (WSC)*, páginas 2520–2531, 2013. 10, 11
- [19] Chella, Antonio, Francesco Lanza, Arianna Pipitone e Valeria Seidita: *The Inner Life of a Robot in Human-Robot Teaming*. páginas 1–4, setembro 2020. 12
- [20] Cui, Jingjing, Zhiguo Ding, Yansha Deng e Arumugam Nallanathan: *Model-free based automated trajectory optimization for uavs toward data transmission*. Institute of Electrical and Electronics Engineers Inc., dezembro 2019. 13
- [21] Oguz, Ozgur S., Omer C. Sari, Khoi H. Dinh e Dirk Wollherr: *Progressive stochastic motion planning for human-robot interaction*. RO-MAN 2017 - 26th IEEE International Symposium on Robot and Human Interactive Communication, 2017-January:1194–1201, dezembro 2017. 13

- [22] Driggs-Campbell, Katherine, Roy Dong e Ruzena Bajcsy: *Robust, informative human-in-the-loop predictions via empirical reachable sets*. IEEE Transactions on Intelligent Vehicles, 3:300–309, setembro 2018. 13, 14
- [23] Feng, Lu, Clemens Wiltsche, Laura Humphrey e Ufuk Topcu: *Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems*. IEEE Transactions on Automation Science and Engineering, 13(2):450–462, 2016. 14, 15, 16, 40
- [24] Chen, Taolue, Marta Kwiatkowska, Aistis Simaitis e Clemens Wiltsche: *Synthesis for Multi-objective Stochastic Games: An Application to Autonomous Urban Driving*. Em *Quantitative Evaluation of Systems*, páginas 322–337, agosto 2013, ISBN 978-3-642-40195-4. 14, 16, 19
- [25] Russell, Stuart e Peter Norvig: *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, USA, 3rd edição, 2009, ISBN 0136042597. 20, 21, 22, 23, 24
- [26] Amato, Christopher: *Decision-Making Under Uncertainty in Multi-Agent and Multi-Robot Systems: Planning and Learning*. Em *IJCAI*, páginas 5662–5666, julho 2018. 21
- [27] Jayavel, Amudhavel, Giri Sruthy, D. Padmashree, N. Pazhaniraja, Saleem Basha e B. Bhuvaneshwari: *A Comprehensive Analysis on Multi Agent Decision Making Systems*. Indian Journal of Science and Technology, 9, março 2016. 21, 23
- [28] Oliveira, Alana e Rosario Girardi: *An Analysis of Norm Processes in Normative Multi-agent Systems*. Em *2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW)*, páginas 68–71, 2016. 24
- [29] Meyer, John Jules Ch. e Roel Wieringa: *Deontic logic in computer science : normative system specification*. International Workshop on Deontic Logic in Computer Science, página 317, 1994. 24
- [30] D’Inverno, Mark, Michael Luck, Michael Georgeff, David Kinny e Michael Wooldridge: *The dMARS Architecture: A Specification of the Distributed Multi-Agent Reasoning System*. Autonomous Agents and Multi-Agent Systems 2004 9:1, 9:5–53, julho 2004, ISSN 1573-7454. <https://link.springer.com/article/10.1023/B:AGNT.0000019688.11109.19>. 26
- [31] *Modelagem de Software Prof. Flávio de Oliveira Silva, Ph.D. - PDF Download grátis.* <https://docplayer.com.br/9818175-Modelagem-de-software-prof-flavio-de-oliveira-silva-ph-d.html>, acessado em 12/11/2021. 27
- [32] Nahavandi, Saeid: *Trusted Autonomy Between Humans and Robots: Toward Human-on-the-Loop in Robotics and Autonomous Systems*. IEEE Systems, Man, and Cybernetics Magazine, 3(1):10–17, 2017. 27, 28, 29
- [33] Haas, Michael, Robert Mills e Michael Grimaila: *Aiding Understanding of Contested Information Environment Affect on Operations*, páginas 175–202. Springer-Verlag, janeiro 2011, ISBN 978-0-85729-882-9. 27

- [34] Tomaszewski, John E.: *Chapter 11 - Overview of the role of artificial intelligence in pathology: the computer as a pathology digital assistant*. Em Cohen, Stanley (editor): *Artificial Intelligence and Deep Learning in Pathology*, páginas 237–262. Elsevier, 2021, ISBN 978-0-323-67538-3. <https://www.sciencedirect.com/science/article/pii/B9780323675383000117>. 27
- [35] Johnson, Deborah G. e Merel Noorman: *Responsibility Practices in Robotic Warfare*. *Military review*, 94:12, 2014. <https://www.law.upenn.edu/live/files/3770-johnson-d-and-noorman-m-responsibility-practices>. 28
- [36] Lee, John D. e Katrina A. See: *Trust in Automation: Designing for Appropriate Reliance*. *Human Factors*, 46(1):50–80, 2004. https://doi.org/10.1518/hfes.46.1.50_30392, PMID: 15151155. 28
- [37] *Human-On-the-Loop | Joint Air Power Competence Centre*. <https://www.japcc.org/human-on-the-loop/>, acessado em 15/06/2021. 29
- [38] Kwiatkowska, Marta, Gethin Norman e David Parker: *Verification of probabilistic real-time systems*. Em *PRISM 4.0*, volume 6806, páginas 585–591, julho 2011, ISBN 978-3-642-22109-5. 30
- [39] *Prism - probabilistic symbolic model checker*. <http://www.prismmodelchecker.org/>, PRISM is a probabilistic model checker, a tool for formal modelling and analysis of systems that exhibit random or probabilistic behaviour. It has been used to analyse systems from many different application domains, including communication and multimedia protocols, randomised distributed algorithms, security protocols, biological systems and many others- Acessado em 10/03/2022. 30, 32
- [40] Kwiatkowska, Marta, Gethin Norman e David Parker: *Prism: Probabilistic model checking for performance and reliability analysis*. *ACM SIGMETRICS Performance Evaluation Review*, 36, março 2009. 30
- [41] Kwiatkowska, Marta, Gethin Norman e David Parker: *Prism 4.0: Verification of probabilistic real-time systems*. Em Gopalakrishnan, Ganesh e Shaz Qadeer (editores): *Computer Aided Verification*, páginas 585–591, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg, ISBN 978-3-642-22110-1. 30
- [42] Rosenschein, Stanley J.: *Formal theories of knowledge in ai and robotics*. *New Generation Computing*, 3:345–357, dezembro 1985. <https://dl.acm.org/doi/10.1007/BF03037076>. 31, 32
- [43] Kwiatkowska, Marta, David Parker e Clemens Wiltsche: *Prism-games: verification and strategy synthesis for stochastic multi-player games with multiple objectives*. *International Journal on Software Tools for Technology Transfer*, 20:1–16, abril 2018. 32, 34, 35
- [44] Kwiatkowska, Marta, Gethin Norman, David Parker e Gabriel Santos: *Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time*. Em Lahiri,

- Shuvendu K. e Chao Wang (editores): *Computer Aided Verification*, páginas 475–487, Cham, 2020. Springer International Publishing, ISBN 978-3-030-53291-8. 33, 34
- [45] Pimentel, Juliano: *Avaliação e correção do modelo cinemático de robôs móveis visando a redução de erros no seguimento de trajetórias*, outubro 2015. Mestrado em Automação Industrial (PGAI); Universidade do Estado de Santa Catarina (UDESC). 35
- [46] Fotouhi, Azade, Haoran Qiang, Ming Ding, Mahbub Hassan, Lorenzo Galati Giordano, Adrian Garcia-Rodriguez e Jinhong Yuan: *Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges*. IEEE Communications Surveys and Tutorials, 21(4):3417–3442, outubro 2019, ISSN 1553877X. 35, 36, 37
- [47] EB70-MC-10.214: *Vetores Aéreos da Força Terrestre*. Manual De Campanha, 2020. <https://bdex.eb.mil.br/jspui/bitstream/123456789/6703/1/EB70-MC-10.214%20Vetores%20A%C3%A9reos%20da%20For%C3%A7a%20Terrestre.pdf>, acessado em 17/09/2021. 37
- [48] Pedro, Manoel e Medeiros Neto: *Universidade Federal Do Rio Grande do Norte Centro De Ciências Exatas E Da Terra Departamento de Informática E Matemática Aplicada Programa*. Relatório Técnico, UFRN, fevereiro 2016. <https://repositorio.ufrn.br/handle/123456789/21459>. 37
- [49] Zhao, X. e Y.J. Son: *Extended BDI Framework for Modelling Human Decision-Making in Complex Automated Manufacturing Systems*. International Journal of Modelling and Simulation, 28(3):347–356, 2008. <https://doi.org/10.1080/02286203.2008.11442487>. 39
- [50] *Nauru 500C Vtol – XMobots*. <https://xmobots.com.br/language/en/nauru-500c-vtol-en/>, Desenvolvido para operações BVLOS (Beyond Visual Line of Sight), o Nauru possui uma autonomia de 4 horas de voo. Capacidade de transporte de 7Kg de payload capacidade para sensor de mapeamento e sistema ISR (para aplicações de vigilância), sendo indicado em missões voltadas à segurança e defesa. Acessado em 15/10/2021. 49
- [51] *XPlanner – XMobots*. <https://xmobots.com.br/software/xplanner/>, Software de planejamento de missão para aeronaves da XMobots. A plataforma consiste em um sistema de informações geográficas (GIS) no qual o usuário especifica o tipo de missão, modelo de aeronave, câmera, o tipo de operação e área de interesse. Acessado em 17/09/2021. 50
- [52] Medeiros Neto, Manoel Pedro de: *Veículos aéreos não tripulados e sistema de entrega: estudo, desenvolvimento e testes*. Dissertação de Mestrado, 2016. Data de Defesa: 2016; Autor: Medeiros Neto, Manoel Pedro de; Orientador: Miranda, Leonardo Cunha de; Programa: Programa de Pós-Graduação em Sistemas e Computação; Universidade Federal do Rio Grande do Norte (UFRN). 51

- [53] Forejt, V., M. Kwiatkowska, G. Norman e D. Parker: *Automated Verification Techniques for Probabilistic Systems*. Em Bernardo, M. e V. Issarny (editores): *Formal Methods for Eternal Networked Software Systems (SFM'11)*, volume 6659 de *LNCS*, páginas 53–113. Springer, 2011. 51

Apêndice A

Código PRISM - Cenário Contendo Perda do SARP e Distância Percorrida

```

1 // PRISM 4.7
2 // Perda do SARP e distância percorrida
3 // Autor Richard Siqueira Miranda
4
5 mdp // MDP
6
7 const double ihp; // interação humana Ponderada
8 const double r; // risco
9
10 //definição das distâncias
11
12 // INICIO BLOCO E
13
14 const int aa1 = 28; const int alb = 12;
15 const int a2a1 = 10; const int a2a3 = 15;
16 const int a2s = 12; const int a3a2 = 20;
17 const int a3s = 25; const int a3a4 = 18;
18 const int a4a3 = 14; const int a4s = 22;
19 const int ala2 = 12; const int as = 18;
20 const int aa4 = 35; const int aa3 = 29;
21 const int aa2 = 32; const int a2b = 14;
22 const int a3b = 15; const int a4b = 12;
23 const int cihp = 43;
24
25 // FIM BLOCO E
26
27 module risco
28
29 cs:[0..10000] init 0;
30 cb:[0..10000] init 0;
31 ts:[0..1000000] init 0;
32 tb:[0..1000000] init 0;
33
34 a:[0..1] init 1; //posição inicial
35 t1:[0..1] init 0;
36 t2:[0..1] init 0;
37 t3:[0..1] init 0;
38 t4:[0..1] init 0;
39 b:[0..1] init 0;
40 si:[0..1] init 0;
41 v:[0..1000] init 0;
42 d:[0..1000] init 0;
43
44
45 // INICIO BLOCO A
46 //Posição A
47 [UAV] a=1 & v<10000 -> r:(si'=1)&(a'=0)&(d'=d+as)
48         + (1-r)/5:(t1'=1)&(a'=0)&(d'=d+aa1)
49         + (1-r)/5:(t2'=1)&(a'=0)&(d'=d+aa2)
50         + (1-r)/5:(t2'=1)&(a'=0)&(d'=d+aa2)
51         + (1-r)/5:(t3'=1)&(a'=0)&(d'=d+aa3)
52         + (1-r)/5:(t4'=1)&(a'=0)&(d'=d+aa4);
53 // FIM do BLOCO A
54
55 // INICIO BLOCO F
56
57 // Posição A1
58 [UAV] t1=1 & v<10000 -> r:(t2'=1)&(t1'=0)&(d'=d+a1a2)
59         + (1-r):(b'=1)&(t1'=0)&(d'=d+alb);
60
61 // FIM BLOCO F
62
63 // INICIO BLOCO B
64 // Posição A2
65 [UAV] t2=1 & v<10000 -> r:(si'=1)&(t2'=0)&(d'=d+a2s)
66         + (1-r)/4:(t1'=1)&(t2'=0)&(d'=d+a2a1)
67         + (1-r)/4:(t3'=1)&(t2'=0)&(d'=d+a2a3)
68         + (1-r)/4:(b'=1)&(t2'=0)&(d'=d+a2b)
69         + (1-r)/4:(b'=1)&(t2'=0)&(d'=d+a2b);
70
71 // FIM BLOCO B
72
73 // INICIO BLOCO C

```

```

74
75 // Posição A3
76 [UAV] t3=1 & v<10000 -> r:(si'=1)&(t3'=0)&(d'=d+a3s)
77     + (1-r)/4:(t2'=1)&(t3'=0)&(d'=d+a3a2)
78     + (1-r)/4:(t4'=1)&(t3'=0)&(d'=d+a3a4)
79     + (1-r)/4:(b'=1)&(t3'=0)&(d'=d+a3b)
80     + (1-r)/4:(b'=1)&(t3'=0)&(d'=d+a3b);
81
82 // FIM BLOCO C
83
84 // INICIO BLOCO D
85
86 // Posição A4
87 [UAV] t4=1 & v<10000 -> r:(si'=1)&(a'=0)&(d'=d+a4s)
88     + (1-r)/3:(t3'=1)&(t4'=0)&(d'=d+a4a3)
89     + (1-r)/3:(b'=1)&(t4'=0)&(d'=d+a4b)
90     + (1-r)/3:(b'=1)&(t4'=0)&(d'=d+a4b);
91
92 // FIM BLOCO D
93
94 // Posição Sink
95 [UAV] si=1 & v<10000 -> ihp:(si'=0)&(a'=1)&(d'=d+cihp)
96     + (1-ihp):(a'=1)&(si'=0)&(v'=v+1)&(cs'=cs+1)&(ts'=ts+d)&(d'=0);
97
98 // Posição Chegada
99 [UAV] b=1 & v<10000 -> (a'=1)&(b'=0)&(v'=v+1)&(cb'=cb+1)&(tb'=tb+d)&(d'=0);
100
101 endmodule

```

Apêndice B

Código PRISM - Cenário Contendo Interações Humanas

```

1 // Calculo das distancias envolvidas utilizando a variaçao das distâncias
2 // e Interações humanas na linguagem PRISM Model Checker
3 // Autor Richard Siqueira Miranda
4
5 mdp
6
7 // interação desviar
8 const double p;
9
10 // interação voltar
11 const double v;
12
13 // interação atrasar
14 const double t;
15
16 // OBS:
17 // (p+v+t) < 1
18 // todos: p/3 v/3 t/3
19
20 // distâncias em UD
21
22 // INICIO BLOCO A
23
24 //base
25 const int bw=18;
26 const int bb=1;
27 const int br=25;
28
29 //rota secundária
30 const int rb=20;
31 const int rr=1;
32 const int rw=7;
33
34 //waypoint
35 const int wr=10;
36 const int ww=1;
37 const int wb=22;
38
39 //FIM BLOCO A
40
41 //Variações UD
42 const l=-1;
43
44 //br+ wr+ bw- wb-
45 //Início módulo UAV
46 module UAV
47
48 //Variáveis de localização
49
50 //Pouso e decolagem
51 g:[0..1] init 1;
52
53 //Base
54 b:[0..1] init 0;
55
56 //Waypoint
57 w:[0..1] init 0;
58
59 //Rota secundária
60 r:[0..1] init 0;
61
62 //Ida e retorno da fotografia
63 f:[0..1] init 0;
64
65 //calculo da distância
66 d:[0..1000] init 0;
67
68 // INICIO BLOCO C
69

```



```

70 //Montagem do cenário e interações
71
72 //Pousado
73 [g]g=1 ->
74
75 // Seguir plano Pousado
76 (g'=0) & (b'=1) & (d'=0);
77
78 //Fotografia
79 [UAV]f=1 ->
80
81 //Seguir plano fotografia
82 (w'=1) & (f'=0);
83
84 //Base
85 [UAV]b=1 ->
86
87 //Seguir o plano Base
88 (1-p-v-t) : (f'=1) & (b'=0) &
89 (d'=d+bw-1)
90
91 //Interação Desviar Base
92 + p: (r'=1) & (b'=0) &
93 (d'=d+br+1)
94
95 //Interação voltar Base
96 + v: (w'=1) & (b'=0) &
97 (d'=d+bw-1)
98
99 //Interação Atrasar Base
100 + t: (b'=b) &
101 (d'=d+bb);
102
103 //Waypoint
104 [UAV]w=1 ->
105
106 //Seguir o plano Waypoint
107 (1-p-v-t) : (g'=1) & (w'=0) &
108 (d'=d+wb-1)
109
110 //Interação Desviar Waypoint
111 + p: (r'=1) & (w'=0) &
112 (d'=d+wr+1)
113
114 //Interação voltar Waypoint
115 + v: (b'=1) & (w'=0) &
116 (d'=d+wb-1)
117
118 //Interação Atrasar Waypoint
119 + t: (w'=w) &
120 (d'=d+ww);
121
122 //Rota secundária
123 [UAV]r=1 ->
124
125 //Seguir o plano Rota secundária
126 (1-p-v-t) : (w'=1) & (r'=0) &
127 (d'=d+rw+1)
128
129 //Interação Desviar Rota secundária
130 + p: (b'=1) & (r'=0) &
131 (d'=d+rb)
132
133 //Interação voltar Rota secundária
134 + v: (r'=0) & (w'=1) &
135 (d'=d+rw)
136
137 //Interação Atrasar Rota secundária
138 + t: (r'=r) &

```

```
139 (d'=d+rr);
140
141 endmodule
142 // FIM BLOCO C
143
144
145 // INICIO BLOCO B
146 // Distâncias UD
147 rewards "dist"
148
149 [g] true: d;
150
151 endrewards
152 // FIM BLOCO B
```

Apêndice C

Código PRISM-GAMES - Cenário Contendo Grau de Obediência

```

1
2 // Calculo das distancias envolvidas utilizando o Grau de obediencia do BDI e
3 // Interações humanas em planos pre definidos utilizando games estocásticos por turnos
na
4 // linguagem PRISM-GAMES Model Checker.
5 // Autor Richard Siqueira Miranda
6
7 smg //Turn-based stochastic multi-player games
8
9 // INICIO BLOCO A
10
11 //Jogos
12 player P1 // Jogador 1 - Planos
13 Planos
14 endplayer
15
16 player P2 // Jogador 2 - Interação Humana
17 HITL
18 endplayer
19
20 player P3 // Jogador 3 - BDI
21 BDI
22 endplayer
23
24 // FIM BLOCO A
25
26 // Probabilidades de ocorrencia de Interacao
27 // OBS: p+v+t < 1
28 // OBS: Probabilidades Todos: (p/3 + v/3 + t/3)
29 const double p=0.0; //Probabilidade de Desviar
30 const double v=0.7; //Probabilidade de Voltar
31 const double t=0.0; //Probabilidade de Atrasar
32 const double a=p+v+t; //Soma das probabilidades
33
34
35 // Distancias do mapa utilizado na modelagem
36 const int bw=18; // Distancia da Base até Waypoint
37 const int bb=1; // Distancia Base até Base (Interacao Atrasar)
38 const int br=25; // Distancia Base a Rota Alternativa
39 const int rb=20; // Distancia Rota Alternativa até a Base
40 const int rr=1; // Distancia Rota Alternativa até Rota Alternativa (Interacao Atrasar)
41 const int rw=7; // Distancia Rota Alternativa até o Waypoint
42 const int wr=10; // Distancia Waypoint até Rota Alternativa
43 const int ww=1; // Distancia Waypoint até Waypoint (Interacao Atrasar)
44 const int wb=22; // Distancia Eaypoint até a Base
45
46 // Variaveis de apoio
47 global z:bool init true; // Termina do processo
48 global e:[0..2] init 0; // Ligações entre os Modulos
49 global g:[0..1] init 1; // Posicao Pouso
50 global b:[0..1] init 0; // Posicao Base
51 global w:[0..1] init 0; // Posicao Waypoint
52 global f:[0..1] init 0; // Posicao Fotografia
53 global r:[0..1] init 0; // Posicao Rota Alternativa
54 global o:[0..100] init 0; // Grau de Obediencia
55 global d:[0..1000] init 0; // Distancia Percorrida
56
57 // Modulos utilizados
58
59 // INICIO BLOCO B
60
61 module Planos // Plano original de voo
62
63 po:[0..5000]; // contagem 5000 movimentos
64
65 [] (z & po > (1010)) // contagem de 1010 voos
66 -> (po'=0) & (o'=o+10);
67
68 [] (z & o=110) // contagem de 110 %

```

```

69 -> (z'=false);
70
71 [] (z & g=1) // Posicao pouso
72 -> (g'=0) & (b'=1)&(d'=0); // ida para a base
73
74 [] (z & f=1) // Posicao fotografia
75 -> (w'=1) & (f'=0); // ida para Waypoint
76
77 [] (z & b=1 & e=0) // Posicao Base
78 -> (1-a) : (f'=1) & (b'=0) & (d'=d+bw) // ida para fotografia passando por Waypoint
79 + (a):(e'=1);
80
81 [] (z & w=1 & e=0) // Posicao Waypoint
82 -> (1-a) : (g'=1) & (w'=0) & (d'=d+wb) & (po'=po+1) // ida para o pouso passando por Base
83
84 + (a): (e'=1);
85
86 [] (z & r=1 & e=0) // Posicao Rota Alternativa
87 -> (1-a) : (w'=1) & (r'=0) & (d'=d+rw) // ida para Waypoint
88 + (a) : (e'=1);
89
90 endmodule
91
92 // FIM BLOCO B
93
94 // INICIO BLOCO C
95
96 module BDI // Ponderacao BDI
97
98 [] (z & b=1 & e=1) // Decisao BDI na Base
99 -> (1 - o/100) : (e'=0)
100 + o/100:(e'=2);
101
102 [] (z & w=1 & e=1) // Decisao BDI no Waypoint
103 -> (1 - o/100) : (e'=0)
104 + o/100:(e'=2);
105
106 [] (z & r=1 & e=1) // Decisao BDI na Rota Alternativa
107 -> (1 - o/100) : (e'=0)
108 + o/100 : (e'=2);
109
110 endmodule
111
112 // FIM BLOCO C
113
114 // INICIO BLOCO D
115
116 module HITL // Interacao Humana
117
118 [] (z & b=1 & e=2) // Interacao Humana na Base
119 -> (p/(a)) : (r'=1) & (b'=0) & (d'=d+br) & (e'=0) // decisao desviar
120 + (v/(a)) : (w'=1) & (b'=0) & (d'=d+bw) & (e'=0) // decisao voltar
121 + (t/(a)) : (b'=b) & (d'=d+bb) & (e'=0); // decisao atrasar
122
123 [] (z & w=1 & e=2) // Interacao Humana no Waypoint
124 -> (p/(a)) : (r'=1) & (w'=0) & (d'=d+wr) & (e'=0) // decisao desviar
125 + (v/(a)) : (b'=1) & (w'=0) & (d'=d+wb) & (e'=0) // decisao voltar
126 + (t/(a)) : (w'=w) & (d'=d+ww) & (e'=0); // decisao atrasar
127
128 [] (z & r=1 & e=2) // Interacao Humana na Rota Alternativa
129 -> (p/(a)) : (b'=1) & (r'=0) & (d'=d+rb) & (e'=0) // decisao desviar
130 + (v/(a)) : (r'=0) & (w'=1) & (d'=d+rw) & (e'=0) // decisao voltar
131 + (t/(a)) : (r'=r) & (d'=d+rr) & (e'=0); // decisao atrasar
132
133 endmodule
134
135 // FIM BLOCO D
136
137 rewards "dist" // Levantamento das distancias

```

```
138
139 g=1: d; // Distancia percorrida ate o pouso
140
141 endrewards
```