

# INVESTIGAÇÃO DA LINGUAGEM PDDL NO PLANEJAMENTO DE MISSÕES PARA ROBÔS AÉREOS

LUIZ F. A. CANTONI\*, MARIO F. M. CAMPOS\*, LUIZ CHAIMOWICZ\*

*\*Universidade Federal de Minas Gerais  
Departamento de Ciência da Computação  
Laboratório de Visão Computacional e Robótica (VeRLab)  
Belo Horizonte, MG, Brasil*

Emails: `luiz.cantoni@gmail.com`, `{mario, chaimo}@dcc.ufmg.br`

**Abstract**— Unmanned Aerial Vehicle (UAV) mission planning is a complex task that is comprised of, among other things, determining which vehicles should be used and which tasks each vehicle has to perform in order to accomplish the desired outcome. Automated Planning is the area of AI that develops planning methodologies and techniques to automatically generate the sequences of actions necessary to solve problems in different domains. One important tool in this area is the Planning Domain Definition Language (PDDL). The present work studies the use of PDDL for UAV mission planning. We develop an experimental framework based on a flight simulator in order to run the PDDL-generated missions and compare the plans to the simulated reality. This allows us to refine and improve the models and to further explore PDDL and its limitations.

**Keywords**— Artificial Intelligence, Automated Planning, PDDL, UAV

**Resumo**— O planejamento de uma missão para robôs aéreos é um processo complexo e que exige, dentre outras coisas, determinar quais os veículos devem ser utilizados e quais as tarefas deverão ser executadas (por veículo) para o cumprimento da missão. No campo da inteligência artificial (IA), existe uma área denominada *Planejamento Automático* que desenvolve linguagens e algoritmos de planejamento, cujo objetivo é a geração automática de sequências de ações que resolvam um determinado problema, sendo uma das principais linguagens a *Planning Domain Definition Language* (PDDL). Neste trabalho, investiga-se a aplicação da PDDL no planejamento de missões para robôs aéreos. Por meio de um arcabouço experimental, os domínios desenvolvidos em PDDL são executados em um simulador de voo e os dados coletados são utilizados para comparar o planejado com o realizado, permitindo refinar o modelo e explorar a PDDL e suas limitações.

**Palavras-chave**— Inteligência Artificial, Planejamento automático, PDDL, VAANTs

## 1 Introdução

Um robô aéreo, também conhecido por veículo aéreo autônomo não tripulado (VAANT), pode ser definido como uma aeronave não tripulada que voa de forma parcialmente ou totalmente autônoma e que é utilizada para executar as mais diversas tarefas, seja nos contextos militar ou civil<sup>1</sup>.

Os desafios ao lidar com esse tipo de veículo, não residem apenas na fase de construção física da aeronave e no desenvolvimento do *hardware* e *software* que serão embarcados ou utilizados para controlar e monitorar o veículo à distância. A operação em si é algo que exige pessoas habilitadas e preparadas para trabalhar com um veículo de dinâmica complexa imerso em um ambiente com várias incertezas. Tarefas como guiar o veículo utilizando um controle remoto, monitorar os parâmetros do voo, obter fotografias de pontos de interesse ou planejar uma missão, demandam atenção e treinamento. Portanto, a autonomia desses veículos é um fator importante a ser considerado, pois tende a minimizar a necessidade de intervenção humana em sua operação.

O planejamento de uma missão para robôs aéreos é um processo complexo e que exige, dentre outras coisas, determinar quais serão os veículos utilizados e qual a sequência de tarefas de alto nível<sup>2</sup> que deverá ser executada para o cumpri-

mento da missão. Este trabalho fica ainda mais complexo para um ser humano caso múltiplos VAANTs sejam utilizados já que, dependendo da instância da missão, existem várias alternativas para completá-la. Assim, uma das demandas, é o desenvolvimento de *softwares* que facilitem esse processo (Nelson, 1995). Esses sistemas vão desde interfaces gráficas amigáveis que ajudam na guiagem e monitoramento dos VAANTs até algoritmos que determinam, de forma automática, a sequência de tarefas a ser executada pelos veículos.

Uma das maneiras de se determinar uma sequência de ações de alto nível é utilizar as linguagens e algoritmos de planejamento provenientes da inteligência artificial (IA), especificamente de uma área denominada de *Planejamento Automático*. A *Planning Domain Definition Language* (PDDL) (Ghallab et al., 1998; Fox and Long, 2003) é um exemplo de linguagem que é utilizada para descrever um *Problema de Planejamento* e gerar uma sequência de ações (plano) que o resolva.

O objetivo principal deste trabalho é investigar o uso da PDDL no planejamento de missões para robôs aéreos. Basicamente, pretende-se estudar a viabilidade de se utilizar um arcabouço de planejamento clássico da IA na robótica área. Para realizar esse estudo, será desenvolvida uma missão que trata especialmente do deslocamento do veí-

possuem detalhes suficientes para a sua efetiva execução. Assim, uma questão é determinar que o veículo deve se deslocar até um local e a outra é como ele efetivamente realizará esse trajeto.

<sup>1</sup>Esse tipo de veículo é conhecido mundialmente pela sigla *unmanned aerial vehicle* (UAV).

<sup>2</sup>As tarefas são consideradas de alto nível, pois não

culo. Características como a duração, a velocidade, a distância e o consumo de combustível são modeladas utilizando a PDDL, verificando, assim, se a linguagem possui expressividade suficiente para modelar uma tarefa essencial para a maioria dos robôs aéreos. A partir do modelo em PDDL, os planos são gerados pelos planejadores e executados no simulador de voo X-Plane<sup>®</sup> por meio de um arcabouço experimental desenvolvido para este trabalho. Essa execução permite comparar o planejado (em PDDL) com o executado (no simulador de voo).

## 2 Referencial Teórico

### 2.1 Planejamento Automático

Planejamento Automático é uma área de pesquisa da IA que lida com a geração automática da sequência de ações necessárias para resolver um problema. Sendo assim, dado uma situação inicial, um conjunto de ações e uma situação final desejada, a tarefa de planejamento consiste em determinar uma sequência de ações que soluciona o problema, ou seja, uma sequência que atinja a situação desejada a partir da situação inicial.

A sequência de ações que soluciona um problema de planejamento é chamada de *plano* e o algoritmo que a produz é denominado *planejador*. O planejador recebe a descrição formal de um *problema de planejamento* especificado em uma determinada linguagem (PDDL, por exemplo) e usa essa representação explícita do problema para derivar as heurísticas e realizar as inferências e buscas necessárias para solucioná-lo.

O conceito de separação entre a representação de um problema e o método para resolvê-lo surgiu no fim da década de cinquenta com um programa de computador denominado *General Problem Solver* (Newell et al., 1959). A ideia dos autores era que esse programa fosse um solucionador de problemas gerais, utilizado para provar teoremas, jogar xadrez, entre outros. A sua estrutura geral de controle foi a base para o *Stanford Research Institute Problem Solver* (STRIPS) que se refere tanto à linguagem utilizada para representar um problema de planejamento quanto ao planejador (Fikes and Nilsson, 1971). A forma de representação de um domínio em STRIPS teve grande influência na área de planejamento e, devido a isso, ele é a base para linguagens modernas como a *action description language* (ADL) e a PDDL (Russell and Norvig, 2003).

### 2.2 PDDL

Criada em 1998, a PDDL é a linguagem oficial das competições internacionais de planejamento<sup>3</sup> que acontecem de dois em dois anos dentro da principal conferência de planejamento da área, denominada *International Conference on Automated Planning and Scheduling* (ICAPS). O objetivo principal da

linguagem, no contexto das competições, é permitir comparações de desempenho entre os planejadores em diferentes domínios (Ghallab et al., 1998).

Devido ao sucesso obtido nas competições, a linguagem evoluiu de forma significativa e já é utilizada em aplicações reais. Atualmente, existem cinco versões da PDDL. A cada versão novas características e mais expressividade são adicionadas à linguagem, permitindo que uma diversidade maior de problemas sejam especificados.

Um problema de planejamento em PDDL é descrito por dois arquivos denominados de *arquivo de domínio* e *arquivo de problema*. Ambos são passados como parâmetros para um planejador e este, por sua vez, gera (se possível) um plano. A Figura 1 mostra uma visão geral da resolução de um problema de planejamento em PDDL.

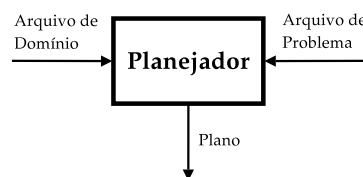


Figura 1: Visão geral da resolução de um problema de planejamento em PDDL.

Em um arquivo de domínio deve ser descrito o conhecimento a respeito do problema a ser resolvido. Esse conhecimento deve ser traduzido em tipos de objeto, predicados, funções e ações. Já no arquivo de problema, é descrita uma instância de um problema a ser resolvido. A instância é composta pelos estados inicial e final e por uma métrica (opcional). Sendo assim, um mesmo arquivo de domínio pode ser utilizado por diferentes instâncias (arquivos de problema) e essa é a principal razão da separação de um problema de planejamento em dois arquivos.

### 2.3 Planejamento de Missões para robôs aéreos

Não existe uma definição formal do que é um sistema de planejamento de missão para VAANTs. Alguns dos conceitos aqui descritos foram retirados de (Nelson, 1995). Segundo este autor, o processo de planejamento de missão é uma atividade baseada em um mapa e que tem como ponto de partida a solicitação do cumprimento de uma determinada tarefa. Por sua vez, um especialista deve desenvolver um plano que satisfaça os requisitos e as restrições da tarefa a ser cumprida. Geralmente, a tarefa inclui a observação de uma determinada área por meio dos sensores de um ou mais VAANTs e a interpretação dos dados obtidos dentro de um intervalo de tempo determinado. Além dos requisitos específicos da tarefa, existem outros critérios da missão que devem ser observados, como evitar a colisão dos veículos com o chão, evitar que os veículos fiquem sem combustível e obedecer as regulamentações do espaço aéreo.

Dentro do processo de planejar uma missão, os especialistas devem se preocupar com diversas

<sup>3</sup>International Planning Competition (IPC).

questões, muitas delas são atividades que exigem cálculos complexos e que, por isso, devem ser delegadas para computadores. Alguns exemplos são: planejar a trajetória a ser percorrida considerando as restrições não-holonômicas do veículo e evitando possíveis obstáculos, estimar a cobertura dos sensores para o correto mapeamento do solo, calcular as altitudes ao longo do voo evitando colisão com montanhas, calcular o tempo e a velocidade necessária para cumprir os prazos da missão, calcular a distância entre os veículos e as estações de solo evitando interrupções nas comunicações (Nelson, 1995).

Além dessas questões, existem outras de mais alto nível como definir quais os tipos e a quantidade de veículos utilizar, verificar se VAANTs em voo podem ser reaproveitados para cumprir a missão sem a necessidade de pouso e reconfiguração e, por último, deve-se determinar a sequência de ações que os veículos devem executar para cumprir a missão. Estas últimas indagações, apesar de não envolverem cálculos matemáticos complexos, são também desafiadoras para um ser humano, principalmente quando a missão é complexa, ou seja, quando existem diferentes possibilidades de completá-la. Uma das maneiras de responder a essas últimas questões é utilizar o planejamento automático proveniente da IA.

O único trabalho encontrado que utiliza o planejamento automático dentro do planejamento de missões para robôs aéreos foi o Gancet et al. (2005), entretanto, este trabalho não faz uso da PDDL, mas do SHOP2, uma planejador que implementa a metodologia *hierarchical task network* (HTN) (Russell and Norvig, 2003).

### 3 Metodologia

Nesta seção, será apresentada a metodologia concebida para investigar o uso da PDDL no planejamento de missões para robôs aéreos. Em primeiro lugar, mostra-se a métrica utilizada para comparar o planejado com o realizado e, logo em seguida, são detalhados os domínios criados em PDDL para a missão de navegação entre *waypoints*.

A PDDL2.2 foi a versão escolhida para este trabalho. Tentou-se utilizar versões mais recentes, como a 3.0, entretanto, os planejadores para essa versão não conseguiram gerar planos para os domínios tratados em tempo satisfatório.

#### 3.1 Qualidade do Plano

A palavra planejamento é utilizada em diversos contextos. Entretanto, deve-se notar que, independentemente da situação onde é aplicada, geralmente ela se refere ao ato de tentar estimar, a partir de dados históricos, utilizando um modelo ou ambos, o que será realizado no futuro.

Por isso, é fundamental comparar o planejado com o efetivamente realizado. Esse tipo de comparação permite refinar o modelo, calibrando-o com as características da realidade que precisam ser

melhor detalhadas. A metodologia descrita abaixo foi utilizada para comparar o planejado em PDDL com o realizado no simulador de voo.

Seja  $\psi = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$  o conjunto de todas as ações disponíveis no domínio. Cada ação desse conjunto, por sua vez, pode ser descrita pela seguinte tupla:  $\alpha_i = \{\phi_1, \phi_2, \dots, \phi_n\}$ , onde cada elemento desta tupla representa um parâmetro que precisa ser comparado<sup>4</sup>.

Considere ainda  $\pi_p = \langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle$  como sendo o plano gerado em PDDL e  $\pi_r = \langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle$  o mesmo plano, porém, executado no simulador de voo.

Uma das maneiras de realizar essa comparação é utilizando a seguinte fórmula de diferença:

$$diferenca_{ij} = \left( \frac{\pi_r^i(j)}{\pi_p^i(j)} - 1 \right) \times 100.$$

O índice  $i$  representa uma mesma ação nos conjuntos  $\pi_p$  e  $\pi_r$ , porém, instanciada de maneira distinta. O índice  $j$  representa um determinado parâmetro da respectiva ação  $i$ . A ação  $i$  no conjunto  $\pi_p$  possui os parâmetros instanciados com os valores planejados em PDDL. Já ação  $i$  no conjunto  $\pi_r$  possui os parâmetros instanciados com os valores realizados no simulador de voo. Portanto, a  $diferenca_{ij}$ , representa a diferença percentual entre o realizado e o planejado para um determinado parâmetro em uma determinada ação. Um percentual negativo significa que o valor realizado foi menor que o valor planejado e um percentual positivo o contrário.

Um exemplo do uso da fórmula pode ser observado na Tabela 1. A primeira ação teve o consumo de combustível realizado menor que o planejado. Já na segunda, gastou-se mais combustível do que o esperado.

Ação	Planejado	Realizado	Diferença (%)
Subir	4,31	3,45	-19,95
Descer	1,26	1,59	26,19

Tabela 1: Exemplo da aplicação da fórmula de diferença entre o planejado e o realizado.

#### 3.2 Missão de Navegação entre waypoints

Uma das tarefas fundamentais dos robôs aéreos é o deslocamento. Sendo assim, torna-se necessário modelá-la da maneira mais fiel possível, pois, a partir dela, é possível construir missões mais complexas que possuam ações específicas do problema a ser tratado.

Para isso, desenvolveu-se uma missão de navegação entre *waypoints*. Nesta missão, o veículo deve visitar um conjunto de *waypoints*, sendo que, cada um, descrito por uma latitude, longitude e altitude, deve ser visitado uma única vez. O objetivo da missão é modelar ação de deslocamento em PDDL o mais próximo possível da realidade, ou

<sup>4</sup>Neste trabalho, um parâmetro é uma das características modeladas em PDDL, como, por exemplo, a duração e a velocidade do deslocamento.

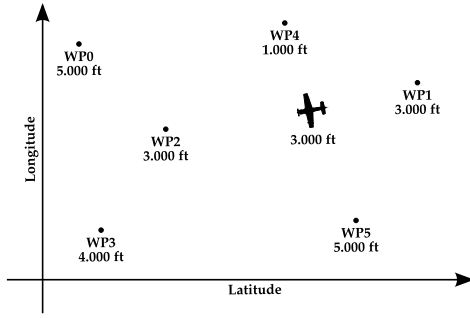


Figura 2: Visão geral da missão de navegação entre waypoints.

seja, uma ação onde a diferença entre o planejado e o realizado seja a menor possível. A Figura 2 mostra uma visão geral da missão. Nesta instância, o veículo iniciou a missão no WP0 e deve terminar no WP5, passando em sequência pelos waypoints 1,2,3 e 4.

Como forma de guiar o trabalho e os experimentos, foram definidas quatro características a serem modeladas: *velocidade*, *distância*, *altitude* e *consumo de combustível*. Entende-se que modelar essas características de uma maneira fiel diminui a diferença entre o planejado e o realizado. Além disso, elas podem ser consideradas características essenciais no deslocamento de um veículo aéreo.

A abordagem utilizada foi iterativa e incremental. Um determinado domínio em PDDL era desenvolvido e experimentado no simulador. Os dados coletados na simulação eram utilizados para comparar o planejado com o realizado. Caso houvesse muita divergência, um novo domínio era construído tentando modelar com mais detalhes os parâmetros que obtiveram maior diferença.

A Figura 3 mostra uma ação de deslocamento em PDDL. Uma ação é composta por um nome, parâmetros, duração (se for uma ação durativa), condições e efeitos. No exemplo da figura, os parâmetros são o veículo e os waypoints de origem e destino. As condições descrevem o que deve ser verdadeiro para que essa ação possa ser considerada aplicável, já os efeitos descrevem como o estado do mundo mudará caso a ação seja aplicada. Essa ação, além de ser temporal, lida com variáveis de estado numéricas, como, por exemplo, a distância entre os dois waypoints, representada pela função (*distance ?l1 ?l2*).

O termo *at start* aparece apenas em ações temporais e é uma palavra reservada da linguagem que funciona como um marcador temporal. Embora exista como especificar ações com duração contínua em PDDL, poucos planejadores conseguem lidar com elas, devido a complexidade computacional envolvida. Geralmente, as ações temporais tratadas pelos planejadores são discretas e para descrever o que deve ser verdadeiro no início, durante e no fim da ação, utiliza-se os termos *at start*, *over all* e *at end*, respectivamente. A seguir será apresentado como as características foram modeladas em PDDL.

```
(:durative-action go
:parameters (?uav - UAV ?l1 - Wpt ?l2 - Wpt)
:duration (= ?duration
(/ (distance ?l1 ?l2) (speed ?uav)))
:condition
(and
(at start (not (visited ?uav ?l2)))
(at start (atLocation ?uav ?l1))
(at start (reachable ?l1 ?l2)))
; o veículo deve possuir combustível
; suficiente para se deslocar
; (fuelLevel >= (distance/speed)*fuelFlow)
(at start (>= (fuelLevel ?uav)
(* (/ (distance ?l1 ?l2) (speed ?uav))
(fuelFlow ?uav))))
)
:effect
(and
(at start (not (atLocation ?uav ?l1)))
(at start (decrease (fuelLevel ?uav)
(* (/ (distance ?l1 ?l2) (speed ?uav))
(fuelFlow ?uav))))
(at end (atLocation ?uav ?l2))
(at end (visited ?uav ?l2)))
)
```

Figura 3: Ação de deslocamento modelada em PDDL.

A velocidade horizontal de uma aeronave é comumente expressa utilizando a unidade knot (1 knot equivale a 1 nm/h). Se o fator vento não for considerado, pode-se dizer que, basicamente, existem dois tipos de velocidade horizontal em uma aeronave, são elas: *indicated airspeed* (IAS) e *true airspeed* (TAS)<sup>5</sup>. A IAS decresce com o aumento da altitude enquanto a TAS aumenta (Cantoni, 2010).

A ação descrita na Figura 3, considera a IAS como velocidade para o planejamento, representada pela função (*speed ?uav*). Já a Figura 4, mostra como é realizado o cálculo da duração utilizando a TAS. Note que a TAS é calculada dinamicamente utilizando a IAS, um fator de ajuste e a altitude atual da aeronave, maiores detalhes podem ser encontrados em Cantoni (2010).

```
(*
(/
(distance ?l1 ?l2)
(* (speed ?uav)
(+ 1 (* (altitude ?l1) (tasFactor)))));TAS
)(fuelFlow ?uav)
```

Figura 4: Código em PDDL para o cálculo do combustível gasto. Este código utiliza a TAS como velocidade planejada.

O consumo de combustível, representado pela função (*fuelFlow ?uav*), é uma variável muito sensível a diversos parâmetros. Além das condições atmosféricas, a altitude e o peso do avião também influem diretamente. Portanto, criar um modelo para o comportamento dessa característica não é algo trivial. A alternativa adotada foi utilizar um valor conservador.

No caso da altitude, dois domínios foram desenvolvidos. A diferença entre eles é que o segundo considera as fases de voo (cruzeiro, subida, descida), cada uma representada por uma ação diferente. Dessa maneira, enquanto no primeiro domínio apenas uma velocidade e taxa de consumo

<sup>5</sup>A IAS é a velocidade indicada e a TAS é a velocidade verdadeira ou aerodinâmica.

de combustível podem ser utilizadas no planejamento, no segundo domínio pode-se ter velocidades e taxas de consumo diferentes pra cada fase de voo, deixando o plano gerado mais fiel à realidade.

#### 4 Arcabouço experimental

O arcabouço desenvolvido para a realização dos experimentos é composto por diversos módulos. Alguns foram desenvolvidos para este trabalho, outros são *softwares* desenvolvidos por terceiros. A Figura 5 mostra uma visão geral do arcabouço.

Os módulos de planejamento, execução e o *X-Plane Interface* (X-Pi) foram desenvolvidos em Java, sendo que esse último, é uma biblioteca que faz a interface de comunicação com o simulador X-Plane®, permitindo que comandos sejam enviados e dados recebidos do simulador. Por ser independente do restante dos módulos, pode ser utilizada em outros projetos. O seu código é livre e de código aberto e está publicado em <http://code.google.com/p/x-pi/>.

A utilização de simuladores de voo de uso pessoal como o X-Plane®, *Microsoft Flight Simulator*® (MsFS) e *FlightGear Flight Simulator* (FgFS) é algo comum em pesquisas científicas. Esse tipo de abordagem permite redução significativa no tempo de desenvolvimento, pois os simuladores de voo preveem um ambiente que permite utilizá-los na validação de sistemas embarcados reais (Cantoni et al., 2009).

Neste trabalho, utilizou-se o piloto automático do X-Plane® para controlar a aeronave de maneira autônoma. Mais especificamente, o Módulo de Execução, que possui as ações PDDL instanciadas para execução, utiliza a biblioteca X-Pi para comunicação com o simulador, enviando os dados de altitude, velocidade e direção que devem ser mantidos pelo piloto automático.

#### 5 Resultados

Os experimentos apresentados nesta seção, foram utilizados para refinar os modelos desenvolvidos em PDDL. Dessa maneira, a construção e testes de um determinado domínio envolvia, basicamente, a codificação em PDDL, a execução no simulador e a análise dos dados. Caso algum parâmetro apresentasse uma diferença significativa entre o planejado e o realizado, o domínio era ajustado e novos experimentos realizados. Sendo assim, além de conduzir a uma modelagem mais fiel, a parte experimental permitiu compreender melhor a PDDL e suas limitações. Para este trabalho, selecionou-se o planejador *LPG-td* (Gerevini et al., 2004). O *LPG-td* consegue lidar com a versão 2.2 da PDDL e é uma extensão do planejador *LPG*.

Por ser uma missão de navegação entre *waypoints*, as ações de decolagem e pouso não foram consideradas. Todo experimento tinha início com o avião já em voo. Nos experimentos com a característica velocidade, o veículo voava por 10 *waypoints*, cada um separado por 4 nm do seu antecessor. Já

nos experimentos com a característica altitude, o veículo voava também por 10 *waypoints*, cada um separado por 1000 ft de altitude do seu antecessor.

A Tabela 2 mostra a diferença percentual entre o planejado e o realizado para a característica velocidade. Os resultados mostram que utilizar a TAS como velocidade planejada faz uma diferença significativa na qualidade do plano gerado.

Já na Tabela 3, dois comportamentos podem ser observados. Ao considerar uma distância curta (4 nm), nota-se que utilizar as fases de voo no planejamento é algo interessante, entretanto, quando a distância aumenta, nesse caso para 20 nm, a diferença entre o planejado e realizado vai ficando cada vez menos significativa. A explicação para esse comportamento é que o veículo permaneceu na velocidade planejada de cruzeiro por mais tempo do que quando a distância são de 4 nm e essa é a única velocidade planejada para o domínio que utiliza apenas uma ação para o deslocamento, ou seja, o veículo atingia a altitude do próximo *waypoint* bem antes de alcançá-lo. Em relação ao consumo de combustível, a diferença aconteceu, pois foi realizada uma escolha conservadora para a taxa de consumo.

Outros experimentos realizados com o arcabouço, bem como uma discussão mais detalhada dos resultados podem ser encontrados em Cantoni (2010).

Altitude Domínio	5.000 ft		10.000 ft	
	IAS	TAS	IAS	TAS
Duração	-6,62	0,36	-13,84	-0,90
Combustível	-7,72	-0,31	-12,93	0,93
Distância	-0,14	-0,17	-0,20	-0,14
Velocidade	7,25	-0,22	16,26	1,10
Taxa de Consumo	2,32	2,87	1,40	1,87

Tabela 2: Diferença percentual entre o planejado e o realizado para a característica velocidade.

Distância Domínio	4 nm		20 nm	
	s/fase	c/fase	s/fase	c/fase
Duração	23,25	0,55	4,13	-0,90
Combustível	13,15	-4,27	-14,69	-8,90
Distância	0,25	0,24	0,16	0,16
Velocidade	-18,91	-3,05	-3,83	0,78
Taxa Consumo	-7,87	1,88	-18,01	-10,68

Tabela 3: Diferença percentual entre o planejado e o realizado para a característica altitude.

#### 6 Conclusões

Este trabalho apresentou uma aplicação da linguagem de planejamento automático PDDL no auxílio ao problema de planejamento de missão para robôs aéreos. A linguagem e os algoritmos de planejamento que operam sobre ela foram utilizados para gerar sequências de ações que os robôs aéreos devem seguir para cumprir determinada missão. Até onde se tem conhecimento, essa é a primeira

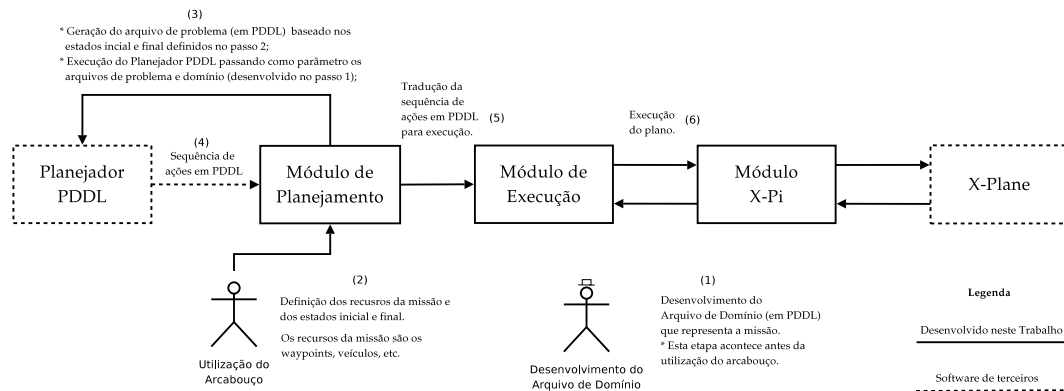


Figura 5: Visão geral do arcabouço desenvolvido para a execução dos planos gerados em PDDL.

vez que a PDDL é aplicada dentro desse contexto. Assim, este trabalho pode servir como um ponto de partida em pesquisas mais específicas sobre o tema.

Mesmo com as limitações impostas no modelo, como ausência de ventos e umidade, um dos principais desafios foi expressar, em um linguagem utilizada para o planejamento simbólico, algumas características de um veículo de dinâmica complexa. As diversas modelagens realizadas permitiram explorar várias características da linguagem e, embora a linguagem possua mecanismos interessantes, como a possibilidade de utilizar variáveis de estado numéricas e definir ações temporais, existem algumas limitações importantes. Entre estas, pode-se citar a impossibilidade de utilizar funções como seno, cosseno, raiz quadrada e entre outras dentro da linguagem. A PDDL permite apenas o uso das quatro operações aritméticas básicas. Isso impede que problemas mais complexos sejam resolvidos a partir da própria linguagem. Somada a essas, está a limitação de não permitir chamadas a funções externas desenvolvidas em outras linguagens. Isso impede, por exemplo, o cálculo mais realista da distância que o veículo percorrerá em uma navegação entre dois *waypoints*.

Um dos principais aspectos trabalhados, foi a execução dos planos gerados em PDDL no simulador de voo X-Plane® por meio do arcabouço experimental desenvolvido neste trabalho. Essa execução permitiu extrair os dados necessários para comparar o planejado em PDDL com o realizado no simulador. Ao final, os resultados mostraram que os modelos construídos foram conservadores em relação às características mencionadas acima. Este é um aspecto importante quando se trata de planejamento para veículos aéreos em geral.

Um possível trabalho futuro é aprofundar a análise da PDDL em missões mais complexas. Em Cantoni (2010), por exemplo, a linguagem é utilizada para gerar planos de combate a incêndios florestais, com ações específicas desse domínio, como lançar água no incêndio e reabastecer. Nesse caso, o planejador PDDL não retorna a mesma sequência de ações, como neste trabalho, mas sequências de ações que dependem diretamente do estado inicial

e final. Portanto, a PDDL é utilizada para tomada de decisão a respeito de quais veículos serão utilizados e qual a sequência de ações cada um deverá seguir para completar a missão. Pretende-se dar continuidade nesse trabalho e executar a missão de combate a incêndios florestais no arcabouço experimental com múltiplos veículos heterogêneos.

## Referências

- Cantoni, L. F. A. (2010). *Avaliação do uso da linguagem PDDL no planejamento de missões para robôs aéreos*, Dissertação de mestrado, UFMG.
- Cantoni, L. F. A., Alves Neto, A., Chaimowicz, L. and Campos, M. F. M. (2009). Análise Comparativa entre Microsoft Flight Simulator e Flightgear Flight Simulator em Testes Hardware-in-the-loop, *IX Simpósio Brasileiro de Automação Inteligente (SBAI'09)*, Brasília, Brazil.
- Fikes, R. and Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving, *IJCAI*, pp. 608–620.
- Fox, M. and Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning domains, *J. Artif. Intell. Res. (JAIR)* **20**: 61–124.
- Gancet, J., Hattenberger, G., Alami, R. and Lacroix, S. (2005). Task planning and control for a multi-uav system: architecture and algorithms, *IROS'05: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1017–1022.
- Gerevini, A., Saetti, A., Serina, I. and Toninelli, P. (2004). LPG-TD: a Fully Automated Planner for PDDL2.2 Domains (short paper), *14th Int. Conf. on Automated Planning and Scheduling (ICAPS-04)*, Whistler, Canada.
- Ghallab, M., Nationale, E., Aeronautiques, C., Isi, C. K., Golden, K., Penberthy, S., Smith, D. E., Sun, Y., Weld, D. and Mcdermott, C. D. (1998). PDDL - the Planning Domain Definition Language, version 1.2, *Technical report*.
- Nelson, M. (1995). UAV mission planning, *DSTO Formal Reports. Report.: DSTO-TR-0164*, p. 38.
- Newell, A., Shaw, J. C. and Simon, H. A. (1959). Report on a general problem-solving program, *IFIP Congress*, pp. 256–264.
- Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, 2a edn, Prentice-Hall, Englewood Cliffs, NJ, chapter Planning, pp. 375–416.