

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT  
BACHARELADO EM ENGENHARIA ELÉTRICA**

**GIAN LUCAS NUNES DE ALMEIDA**

**GERAÇÃO AUTOMÁTICA DE CÓDIGO PARA A TRAJETÓRIA  
DE VANTS EM VOOS ASSISTIDOS**

**JOINVILLE, SC**

**2015**

**GIAN LUCAS NUNES DE ALMEIDA**

**GERAÇÃO AUTOMÁTICA DE CÓDIGO PARA A TRAJETÓRIA  
DE VANTS EM VOOS ASSISTIDOS**

Trabalho de Conclusão apresentado ao Curso de Bacharelado em Engenharia Elétrica do Centro de Ciências Tecnológicas, da Universidade do Estado de Santa Catarina, como requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Dr. André Bittencourt Leal  
Coorientador: M<sub>Eng.</sub> Ricardo F. Martins

**JOINVILLE, SC  
2015**

**GIAN LUCAS NUNES DE ALMEIDA**

**GERAÇÃO AUTOMÁTICA DE CÓDIGO PARA A TRAJETÓRIA  
DE VANTS EM VOOS ASSISTIDOS**

Trabalho de conclusão de curso, apresentado ao curso de Bacharelado em Engenharia Elétrica do Centro de Ciências tecnológicas, da Universidade do Estado de Santa Catarina, como requisito parcial para obtenção do título do grau em Engenharia Elétrica.

**Banca Examinadora**

Orientador:

\_\_\_\_\_  
Dr. André Bittencourt Leal  
Universidade do Estado de Santa Catarina

Coorientador:

\_\_\_\_\_  
M<sub>eng.</sub> Ricardo Ferreira Martins  
Universidade do Estado de Santa Catarina

Membro:

\_\_\_\_\_  
Dr. Celso José Faria de Araújo  
Universidade do Estado de Santa Catarina

Membro:

\_\_\_\_\_  
Dr. Charles Christian Miers  
Universidade do Estado de Santa Catarina

**JOINVILLE, 09/12/2015**

## **AGRADECIMENTOS**

Agradeço a todos que contribuíram em minha formação acadêmica durante o curso de graduação.

Aos professores André Bittencourt Leal e Ricardo Ferreira Martins pela orientação, dedicação, amizade e toda infraestrutura proporcionada para que este trabalho fosse possível.

Aos meus pais, Ivanir e Maria, minha irmã Andressa e demais familiares pelo apoio, compreensão e dedicação durante esta caminhada.

À minha namorada Aline, a qual me acompanhou desde o início da graduação, pelo seu amor e dedicação.

Aos colegas e amigos: Dhyony Lukas Serighelli, Vinicius Pelizzer Casara, Rodolfo Vanassi, Jardel Régis Teixeira, Guilherme da Silva Fisher, Murilo Brunel Rosa, Alisson Mengatto, Gillian Dal Posso, Eduardo Joaquin Vieira, Henrique Bruno, Nicolas Zapora e Daniel Camargo pela parceria ao longo do curso e pelas contribuições em diversos assuntos relacionados à graduação.

*Um pouco de conhecimento que age vale  
infinitamente mais do que conhecimento  
que é ocioso.*

Khalil Gibran

## RESUMO

Veículos aéreos não tripulados (VANTs) têm sido utilizados com grande frequência em diversas tarefas como busca, resgate, vigilância, monitoramento e até mesmo hobby. Neste trabalho, é apresentada uma pesquisa sobre VANTs e seus componentes, assim como uma revisão sobre Sistemas a Eventos Discretos (SEDs) para a criação de um modelo abstrato do voo e a criação de uma ferramenta para a geração de código, de forma automática, combinando com o uso de sensores complementares ao uso de novas trajetórias. Assim, os conceitos deste trabalho e as ferramentas desenvolvidas foram testados utilizando a técnica de simulação *Hardware-In-the-Loop* (HIL) e com a realização de ensaios reais. A simulação permite visualizar o comportamento do VANT, sem correr nenhum risco devido a uma possível queda.

**Palavras-chave:** *VANTs, SEDs, Geração Automática de Código, HIL.*

## **ABSTRACT**

Unmanned aerial vehicles (UAV) have been often nowadays utilized on different tasks such as: search, rescue, surveillance, monitoring and even like hobby. In this study, it is presented a research on UAVs its components, and a revision about Discrete Events Systems (DES) to create an abstract flight model and the creation of a tool to generate these code automatically, combining the use of complementary sensors to the use of new trajectories. Thus, the concepts and the developed tools were validated utilizing the simulation technique called as Hardware-In-the-Loop and with the realization of real flights. The simulation allows viewing the behavior of UAV, without any risk due to a possible fall.

**Keywords:** *UAV, DES, automatically code generation, HIL.*

## LISTA DE ILUSTRAÇÕES

Figura 2.1 - Primeiro VANT construído - <i>aviation torpedo</i> .....	15
Figura 2.2 - <i>Kettering Bug</i> .....	16
Figura 2.3 - B-17 <i>Flying Fortresses</i> . ....	17
Figura 2.4 - MQM-33 utilizado como primeiro drone de reconhecimento. ....	17
Figura 2.5 – <i>Firebee</i> .....	18
Figura 2.6 - Classificação dos VANTs .....	19
Figura 2.7 - RQ-4 <i>Global Hawk</i> .....	20
Figura 2.8 - Hexacoptero utilizado em resgate .....	21
Figura 2.9 - Multirotor GD-X8.....	21
Figura 2.10 - Modelo simples quadrotor .....	22
Figura 2.11 - Movimentos Básicos.....	23
Figura 2.12 - Sensor Ultrassônico .....	26
Figura 2.13 - Câmera PX4FLOW.....	27
Figura 2.14 - Módulo para telemetria .....	28
Figura 2.15 - Rádio controle (Transmissor e Receptor).....	28
Figura 2.16 - Controle de estabilidade e atitude. ....	29
Figura 2.17 - <i>Hardware APM 2.6</i> .....	30
Figura 2.18 - <i>Mission Planner, Software</i> utilizado pela APM .....	31
Figura 2.19 - Controlador <i>Pixhawk</i> .....	32
Figura 2.20 - <i>Naza lite</i> e módulo GPS/Bússola.....	33
Figura 2.21 - Controles Wii e Arduino utilizados no Multiwii .....	34
Figura 2.22 - Controlador Lisa/M .....	34
Figura 2.23 - Planejamento de voo para geoprocessamento. ....	36
Figura 3.1 - Funcionamento HIL .....	39
Figura 4.1 - Exemplo de um diagrama gráfico de um autômato de estados finitos ..	43
Figura 4.2 - Autômatos $G_1$ e $G_2$ antes da sincronização.....	45
Figura 4.3 - Resultado da Sincronização de $G_1$ e $G_2$ .....	45
Figura 4.4 - SED operando em malha fechada.....	46
Figura 5.1 - Planejamento de missão .....	49
Figura 5.2 - Primeiro esquema de envio de sub-missões.....	52
Figura 5.3 - Etapas da aplicação .....	54
Figura 6.1 – Nova importação no <i>Nadzoru</i> .....	57
Figura 6.2 - Criação de sub-missões (Exportação).....	57
Figura 6.3 - Resultado da Importação.....	58
Figura 6.4 - Proteção contra importações incorretas .....	58
Figura 6.5 - Arquivos criados pela operação CSM .....	60
Figura 6.6 - Botão AutoWD.....	61
Figura 6.7 - Fluxograma da estrutura da função AutoWD.....	63
Figura 6.8 - Comparação entre registros do <i>Mission Planner (a)</i> e API do <i>Google Maps (b)</i> .....	65
Figura 6.9 - Ensaio utilizando função <i>AutoWD</i> .....	66
Figura 6.10 - VANT utilizado para testes .....	67
Figura 6.11 - Marcadores utilizados (em vermelho).....	67
Figura 6.12 - Missão real importada no <i>Nadzoru</i> .....	68
Figura 6.13 - Registro dos movimentos do VANT.....	69

## LISTA DE ABREVIATURAS E SIGLAS

ADEF	Autômato Determinístico de Estados Finitos
GPS	<i>Global Positioning System</i>
GS	<i>Ground Station</i>
HIL	<i>Hardware-in-the-loop</i>
HOTOL	<i>Horizontal Take-off and Landing</i>
MP	<i>Mission-Planner</i>
SEDs	Sistemas a Eventos Discretos
VANT	Veículo Aéreo Não Tripulado
VTOL	<i>Vertical Take-off and Landing</i>
WP	<i>WayPoint</i>

## LISTA DE SÍMBOLOS

$\Sigma$	Conjunto de Eventos
$\Sigma_u$	Conjunto de Eventos Não Controláveis
$\Sigma^*$	Fechamento de Kleene
$L$	Linguagem de um autômato
$G_1    G_2$	Composição paralela do autômato $G_1$ com $G_2$

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>13</b>
<b>2</b>	<b>CARATERIZAÇÃO DE UM VANT .....</b>	<b>15</b>
2.1	HISTÓRIA.....	15
2.2	CLASSIFICAÇÃO DOS VANTS .....	19
2.3	CARACTERIZAÇÃO DO MULTIROTOR.....	20
2.4	COMPORTAMENTO EM VOO.....	21
2.5	CONTROLADORES DE VOO .....	24
2.5.1	Sensores.....	24
2.5.2	Módulo para telemetria .....	27
2.5.3	Rádio controle.....	28
2.5.4	Unidade de processamento .....	29
2.5.5	Controladores comerciais .....	29
2.6	MODOS DE VOO .....	35
2.6.1	Voo Manual.....	35
2.6.2	Voo Semiautomático.....	35
2.6.3	Voo Automático .....	36
2.7	CONSIDERAÇÕES PARCIAIS.....	37
<b>3</b>	<b>HARDWARE-IN-THE-LOOP (HIL).....</b>	<b>38</b>
<b>4</b>	<b>SISTEMAS A EVENTOS DISCRETOS E CONTROLE SUPERVISÓRIO ...</b>	<b>41</b>
4.1	SISTEMAS A EVENTOS DISCRETOS .....	41
4.2	REPRESENTAÇÃO FORMAL DOS SEDS .....	41
4.2.1	Representação de SEDs por Linguagens.....	41
4.2.2	Representação de SEDs por Autômatos .....	42
4.3	CONTROLE SUPERVISÓRIO DE SEDS.....	45
4.3.1	Especificação de controle .....	46
4.3.2	Controlabilidade de linguagens.....	47
4.4	CONSIDERAÇÕES PARCIAIS.....	47
<b>5</b>	<b>DESENVOLVIMENTO .....</b>	<b>49</b>
5.1	ELEMENTOS NECESSÁRIOS.....	50
5.2	ELEMENTOS UTILIZADOS.....	51
5.3	APLICAÇÃO .....	52
<b>6</b>	<b>IMPLEMENTAÇÃO E RESULTADOS .....</b>	<b>56</b>
6.1	IMPLEMENTAÇÃO.....	56

6.2	RESULTADOS.....	64
7	CONCLUSÃO .....	71
	REFERENCIAS .....	73

## 1 INTRODUÇÃO

Segundo ICAO (*International Civil Aviation Organization* - 2011), veículos aéreos não tripulados (VANTs) são todas as aeronaves que não possuem um piloto a bordo para serem guiadas. Os primeiros relatos de aplicações práticas dos VANTs datam de 1849, para fins militares na guerra entre Veneza e Áustria. Tratavam-se de balões carregando bombas (MUNARETTO, 2015). Entretanto, somente no final da década de 1950, durante a guerra do Vietnã, as pesquisas na área obtiveram resultados mais significativos (NONAMI, 2007). Atualmente, com o desenvolvimento e maior acesso à tecnologia, os mesmos têm sido aplicados em diversas áreas e para diversas tarefas, algumas delas descritas por Amahah (2009). Estas tarefas podem ser executadas de maneira pilotada remotamente por meio de um controle, havendo forte dependência da experiência e técnica do piloto, ou podem ser executadas de maneira automática (KABBABE, 2011).

Uma maneira comum para que a aeronave opere automaticamente, é que a mesma utilize coordenadas disponibilizadas através de uma estação posicionada em solo (*Ground Station* – GS), que envia a trajetória de uma missão pré-definida, sendo que esta trajetória não prevê o comportamento da aeronave no espaço aéreo e os obstáculos físicos a serem enfrentados. Segundo Scaramuzza (2013), para que a tarefa seja concluída, a placa controladora, que atua sobre os motores do VANT, possui um *firmware* que na maioria dos casos está preparado somente para fazer uso dos sensores acelerômetro, giroscópio e do GPS (*Global Positioning System*), como orientação. Este método de orientação utilizando o GPS só é possível em missões em locais abertos (*outdoor*), pois quando tratado de missões em locais fechados (*indoor*) o sinal deste sensor é prejudicado (SCARAMUZZA, 2013). Além disso, em missões em locais abertos o GPS não é suficiente para prever todas as adversidades a serem enfrentadas, como por exemplo, barreiras físicas elevadas ou até mesmo o ofuscamento do sinal os satélites (ACHTELIK, 2009).

Por estes motivos, é de grande interesse a modificação do *firmware* bem como do *software* da estação terrestre (GS) do VANT para o uso de um ou mais sensores complementares e o desenvolvimento de um modelo que obtenha maiores informações sobre o comportamento da aeronave no espaço aéreo, levando em considerações as adversidades da planta (SCARAMUZZA, 2013).

Uma abordagem interessante para a solução deste problema pode ser através da utilização da teoria de Sistemas a Eventos Discretos (SEDs), no qual a evolução do sistema não se dá pelo decorrer do tempo, mas sim pela ocorrência de eventos discretos no tempo (CASSANDRAS & LAFORTUNE, 2002), como por exemplo, a chegada de um VANT a um ponto de interesse.

A proposta deste trabalho é implementar uma ferramenta computacional capaz de levar em consideração a informação de sensores adicionais, embarcados ou não, a tomada de decisões em pontos intermediários da missão principal. Esta ferramenta utiliza um modelo em SEDs da missão principal, onde é possível a adição de novas transições entre os pontos intermediários a fim de gerar os pontos de decisão, particionando a missão principal em sub-missões. A partir destas sub-missões a ferramenta gera o código compatível para o VANT, respectivo às mesmas.

Na Seção 2 são apresentados os conceitos sobre VANTs, caracterização dos multirotores, comportamentos em voo, controladores de voo e considerações parciais. Na Seção 3 são apresentados os conceitos sobre a técnica de simulação *Hardware-in-the-loop* utilizada para realização de testes neste trabalho. A seção 4 apresenta uma revisão sobre Sistemas a Eventos Discretos (SEDs) e a representação gráfica por autômatos destes sistemas.

Com os conceitos apresentados nas seções 2, 3 e 4 é proposta uma solução para a utilização de sensores adicionais para tomada de decisão em pontos intermedirnos, apresentado na Seção 5. A implementação e os resultados obtidos são apresentados na Seção 6 à conclusão é apresentada na Seção 7 e por fim são apresentadas as referências bibliográficas.

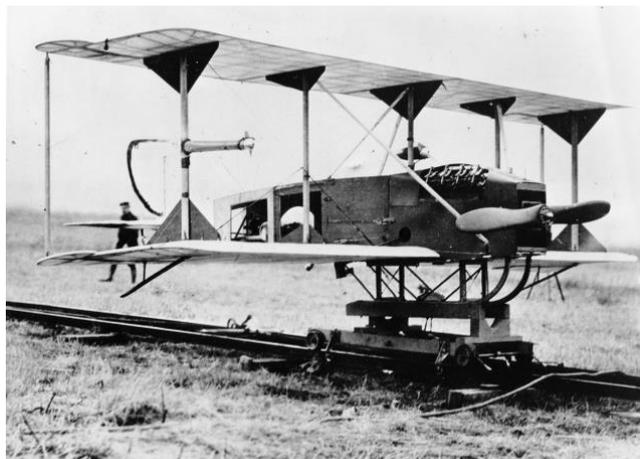
## 2 CARATERIZAÇÃO DE UM VANT

Para uma melhor compreensão do leitor com os termos utilizados neste trabalho e uma contextualização com o assunto de VANTs, é apresentado neste capítulo uma breve história sobre os mesmo, suas classificações, comportamentos de voo e principais componentes.

### 2.1 HISTÓRIA

Os primeiros relatos sobre VANTs datam da época de 1849, quando os austríacos lançaram 200 balões não pilotados armados com bombas para atacar a cidade de Veneza. No entanto, no ano de 1916, os americanos Lawrance e Sperry desenvolveram um VANT, ilustrado na Figura 2.1, no qual apresentava um giroscópio para estabilizar o corpo da aeronave, dando início ao controle de atitude. Eles chamaram esta aeronave de “*aviation torpedo*” que por sua vez levava consigo cargas explosivas, sendo esta uma versão mais antiga dos mísseis de cruzeiro (NONAMI *et al.*, 2010).

Figura 2.1 - Primeiro VANT construído - *Aviation Torpedo*.



Fonte: <http://www.museumofflight.org/files/test.png>

A ideia do *aviation torpedo* foi levada até o exército americano, resultando em uma aeronave autônoma denominada *Kettering Bug* (Figura 2.2), que teve seu primeiro voo em 1918. Levava consigo uma carga de 135 kg de explosivos e

calculava o lançamento da mesma pelo número de revoluções que o seu motor realizava. Embora sua tecnologia fosse bem sucedida, não foi possível aplicá-la na primeira guerra mundial, acabando com o desenvolvimento total do projeto (DARLING, 2011).

Figura 2.2 - Kettering *Bug*.



Fonte: [http://www.daviddarling.info/encyclopedia/K/Kettering\\_Bug.html](http://www.daviddarling.info/encyclopedia/K/Kettering_Bug.html)

Após a primeira guerra mundial, foram criadas as primeiras aeronaves pilotadas por rádio frequência, que primeiramente eram denominadas de “*queen bee*”. Mas, com o passar do tempo, este nome se transformou e introduziu o termo “drone” (HOWETH, 1963).

Somente na segunda guerra mundial, o exército americano começou a utilizar aeronaves guiadas por rádio frequência que eram utilizadas como alvos para treinamento da artilharia antiaérea (DEGARMO, 2004). Foi também durante este período, que os nazistas desenvolveram inúmeros VANTs que podiam carregar cargas explosivas e viajavam em velocidades superiores a mil quilômetros por hora (WOLF, 1997).

Segundo Nonami *et al.* (2010), o desenvolvimento dos VANTs tomou vantagem sobre a guerra do Vietnã e da guerra fria, período este em que as pesquisas e o desenvolvimento nesta área estavam em alta escala. No ano de 1946, oito aeronaves B-17 (Figura 2.3) foram transformadas em drones para coletar informações de níveis de radioatividade em testes com bombas nucleares,

controlados em solo por um transmissor acoplado em um Jeep e em ar por outra aeronave (HOVERSTEN, 2013).

Figura 2.3 - B-17 *Flying Fortresses*.



Fonte: <http://www.eaa240.org/wp-content/uploads/2009/12/b-17memphisbelle-1-9-9-09.jpg>

Em 1950 surgiram os primeiros VANTs, que possibilitavam realizar missões de reconhecimento, ou seja, eram capazes de tirar fotografias. Estes consistiam em uma adaptação do modelo MQM-33 (

Figura 2.4), que por sua vez servia de alvo (JONES, 1956).

Figura 2.4 - MQM-33 utilizado como primeiro drone de reconhecimento.



Fonte: [http://en.wikipedia.org/wiki/File:Radioplane\\_Shelduck.JPG](http://en.wikipedia.org/wiki/File:Radioplane_Shelduck.JPG)

No ano de 1951 foi desenvolvido o primeiro VANT com propulsão a jato. Denominado "Firebee" (

Figura 2.5), ele era utilizado para o treinamento da artilharia antiaérea. Foi um dos “drones alvo” mais produzido de todos os tempos. Houve derivações deste modelo, que por sua vez foram utilizados como drones de reconhecimento em missões de espionagem pelos americanos durante a guerra fria e a guerra do Vietnã.

Figura 2.5 – *Firebee*



Fonte: [http:// wikimedia.org/wikipedia/commons/b/b8/BQM-34A\\_Firebee\\_I\\_1.JPG](http://wikimedia.org/wikipedia/commons/b/b8/BQM-34A_Firebee_I_1.JPG)

Segundo Horgan (2013), em 2013 mais de 50 países já faziam uso desta tecnologia. No entanto, os maiores desenvolvimentos e pesquisas em VANTs são realizados pelas forças militares americanas e pelas forças armadas de Israel.

No passado, os VANTs eram utilizados quase que exclusivamente para fins militares, mas na atualidade eles também têm sido muito utilizados pra fins civis, para diversas aplicações, conforme apresentado na Tabela 2.1.

Tabela 2.1 - Aplicações civis de VANTs

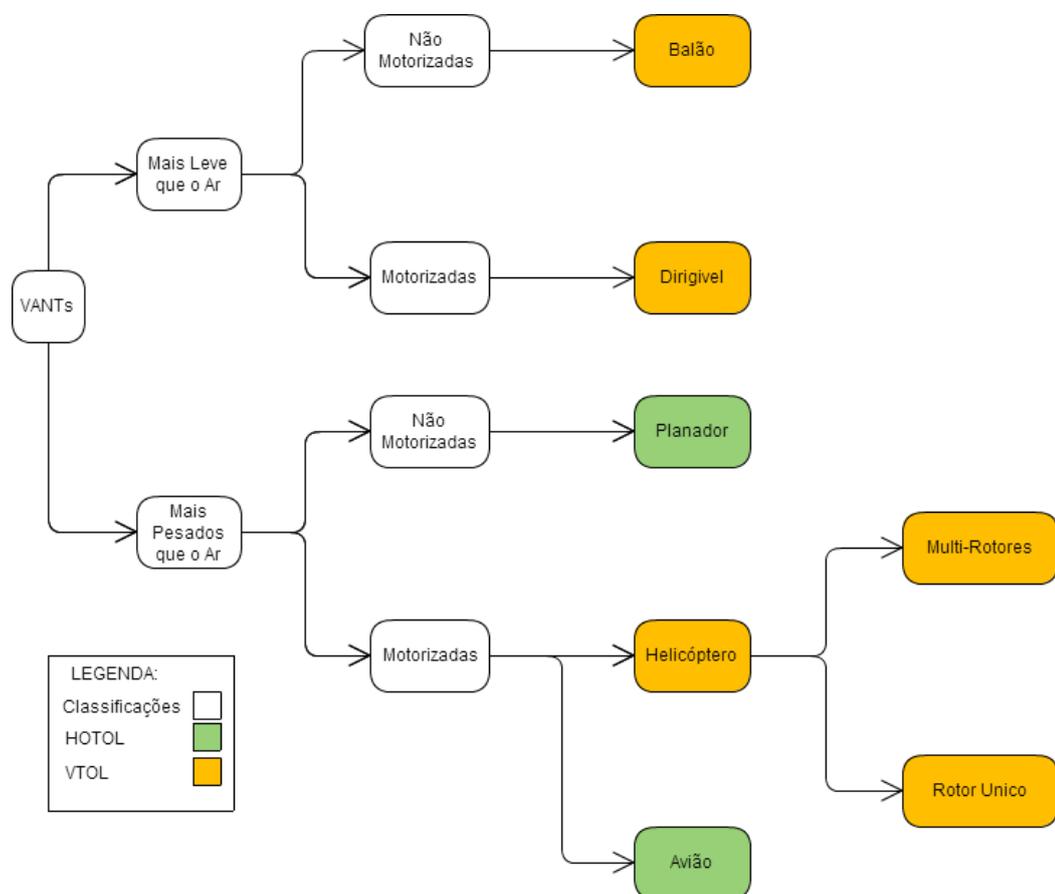
<b>Indústria</b>	<b>Uso</b>
Indústria agrícola	Fotografia aérea para monitoramento de crescimento e pragas nas plantações.
Telecomunicações	Plataforma retransmissora de sinais.
Energia	Monitoramento de linhas de transmissão. Medição de possíveis parques eólicos.
Televisão e cinema	Tomadas aéreas ou acrobáticas.
Salva vidas	Lançamento de boias para banhistas em situação de perigo.

Fonte: NONAMI *at et.* 2010

## 2.2 CLASSIFICAÇÃO DOS VANTS

Segundo Bouabdallah *et al.* (2004), os VANTS podem ser classificados de acordo com o principal modo de propulsão ou por serem mais ou menos densos que o ar, como mostra o diagrama da Figura 2.6. Dentre essas classificações, é possível citar também as características de decolagem: *Vertical Take-off and Landing* (VTOL) e *Horizontal Take-off and Landing* (HOTOL).

Figura 2.6 - Classificação dos VANTS



Fonte: Adaptação de PAULA (2012).

As aeronaves denominadas VTOL são aeronaves capazes de decolar e pousar verticalmente. Já as aeronaves HOTOL necessitam atingir uma determinada velocidade horizontal para poder decolar ou pousar. Em geral, as aeronaves VTOL possuem velocidades de cruzeiro menores que as aeronaves HOTOL.

Na maior parte dos casos, os VANTS que são HOTOL são aeronaves que possuem asa fixa (aviões) podendo assim carregar mais peso e possuir maior

autonomia de voo. Os primeiros VANTs criados eram todos modelos de asa fixa, por possuírem um controle de voo menos complexo. Um exemplo de um VANT de asa fixa é o *RQ-4 Global Hawk* (Figura 2.7) utilizado como aeronave militar de inspeção que pode inspecionar até 100.000 m<sup>2</sup> em um dia (PUTRICH, 2010).

Figura 2.7 - RQ-4 *Global Hawk*



Fonte: [http://www.northropgrumman.com/Photos/pgL\\_GH-10020\\_008.jpg](http://www.northropgrumman.com/Photos/pgL_GH-10020_008.jpg)

### 2.3 CARACTERIZAÇÃO DO Multirotor

Os multirotores são aeronaves derivadas dos helicópteros, que possuem ao menos dois rotores na posição horizontal, característica esta que os tornam aeronaves VTOL. Diferente das aeronaves de asas fixas (aviões), os multirotores são denominados asas rotativas, em razão de que sua sustentação no ar se dá por conta das hélices que estão acopladas aos seus rotores. Esta característica lhes dá uma vantagem em relação às aeronaves de asas fixas possibilitando pairar no ar (GUIMARÃES, 2012).

O número de rotores pode variar em função da aplicação de interesse. O aumento do número de rotores torna o VANT mais estável e permite que a este carregar mais peso, porém isto lhes causa um elevado consumo de combustível ou de bateria. As quantidades de motores comumente utilizadas são: três (tricóptero), quatro (quadricóptero), seis (hexacóptero) e oito (octacóptero), sendo mais estável o octacóptero e mais ágil o tricóptero. A Figura 2.8 ilustra um exemplo de um modelo similar de hexacóptero utilizado em um resgate de um civil que estava desaparecido há três dias e que foi encontrado pelo VANT em vinte minutos (BOLAND, 2014).

Figura 2.8 - Hexacopectero utilizado em resgate



Fonte: <http://www.business-opportunities.biz/wp-content/uploads/2014/11/drone.jpg>

Com o crescente uso de VANTs multirotores em aplicações civis, no qual existe afluência de pessoas ou os equipamentos transportados pelo VANT são de alto valor, é importante a utilização de aeronaves com número de motores superior a 4 ou com sobreposição de motores, como ilustrado na Figura 2.9.

Figura 2.9 - Multirotor GD-X8



Fonte: <http://www.quadrocopter.com/>

Nestes casos, em uma eventual falha de um motor, o VANT ainda pode manter sua estabilidade até que se consiga realizar seu pouso seguro, porém esta estratégia exige um maior consumo de bateria.

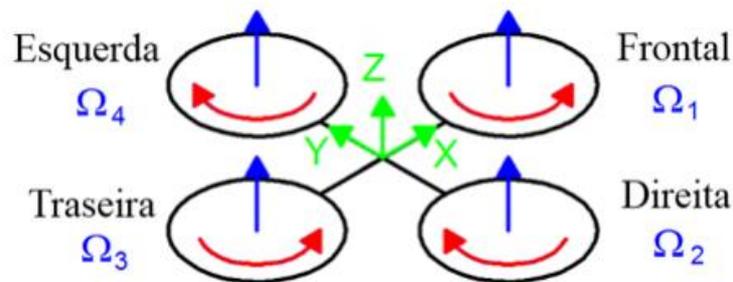
## 2.4 COMPORTAMENTO EM VOO

O comportamento dos VANTs multirotores pode ser definido por intermédio de quatro movimentos básicos: Altitude, Rolagem (*Roll*), Arfagem (*Pitch*) e Guinada

(Yaw). Para que isto seja possível é necessário que exista um algoritmo de estabilização do drone, o que é explicado com mais detalhes na Seção 2.5.4. Na seção 2.4 são explicados estes movimentos referentes a um modelo de quadricóptero.

Para facilitar o entendimento desses conceitos, considera-se um modelo simplificado de quadricóptero (Figura 2.10) levando em consideração que a estrutura seja completamente simétrica, esta escolha se justifica devido o fato de que se fossem levados em consideração os aspectos assimétricos, os modelos possuiriam maior complexidade inviabilizando o entendimento. O motor frontal e de traseira possuem sentido de giro anti-horário, já os motores da direita e esquerda possuem o sentido de giro horário.

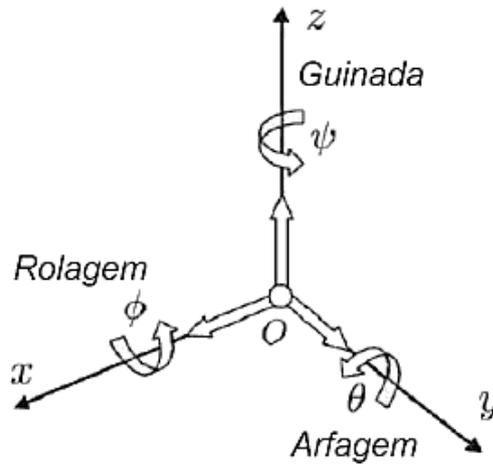
Figura 2.10 - Modelo simples de um quadricóptero



Fonte: (PAULA, 2012)

Segundo Paula (2012), a configuração do giro dos motores se dá pelo fato de eliminar os efeitos giroscópios ocasionados pelos torques dos motores. Para que ocorra a execução dos movimentos básicos (Figura 2.11) os motores devem modificar suas velocidades para atingirem o movimento desejado. A Tabela 2.2 indica a relação da velocidade angular dos motores para cada movimento.

Figura 2.11 - Movimentos Básicos



Fonte : (PAULA, 2012)

Os valores de  $\Omega_c$  são referentes a uma velocidade angular constante e suficiente para vencer a força gravitacional, já os valores de  $\Omega_v$  são referentes a uma velocidade angular variável, podendo ser positiva ou negativa, não excedendo os valores limítrofes do modelo. Utilizando o exemplo da arfagem frontal, ambos os motores da direita e esquerda mantêm suas velocidades constantes, o dianteiro reduz sua velocidade e o motor traseiro aumenta para que haja a inclinação frontal do VANT, realizando o movimento.

Tabela 2.2 - Relação da velocidade das hélices e o movimento desejado

Movimento	Velocidade angular para cada motor.			
	Frontal ( $\Omega_1$ )	Direita ( $\Omega_2$ )	Traseira ( $\Omega_3$ )	Esquerda ( $\Omega_4$ )
Pairado	$\Omega_c$	$\Omega_c$	$\Omega_c$	$\Omega_c$
Altitude (Z)	$\Omega_c \pm \Omega_v$	$\Omega_c \pm \Omega_v$	$\Omega_c \pm \Omega_v$	$\Omega_c \pm \Omega_v$
Arfagem ( $\Theta$ )	$\Omega_c \mp \Omega_v$	$\Omega_c$	$\Omega_c \pm \Omega_v$	$\Omega_c$
Rolagem ( $\Phi$ )	$\Omega_c$	$\Omega_c \pm \Omega_v$	$\Omega_c$	$\Omega_c \mp \Omega_v$
Guinada ( $\Psi$ )	$\Omega_c \pm \Omega_v$	$\Omega_c \mp \Omega_v$	$\Omega_c \pm \Omega_v$	$\Omega_c \mp \Omega_v$

A partir destes movimentos básicos o controlador de voo pode realizar a movimentação do VANT. Essa movimentação pode ser de maneira automática ou

pilotada, assim existem alguns modos de voo destinados para estas funções. A seção 2.5 trata dos controladores de voo e suas características.

## 2.5 CONTROLADORES DE VOO

O controlador de voo é uma das partes mais importantes de um VANT. Nela estão acoplados todos os sensores, módulo para telemetria, rádio controle, unidade de processamento e os *drivers* para acionamento dos motores. As subseções 2.5.1, 2.5.2, 2.5.3, 2.5.4 e 2.5.5 são apresentados os equipamentos acoplados ao controlador e seus modos de voo.

### 2.5.1 Sensores

Os sensores são os elementos responsáveis por prover informações sobre os estados da aeronave, tais como altitude, posição e atitude. Como a autonomia do voo é algo a ser maximizado, em alguns casos são utilizados sistemas micro eletromecânicos (*Microelectromechanical systems*), os quais são sensores muito pequenos e leves que utilizam tecnologia micrométrica para estimarem os valores das grandezas desejadas (JANG *et al.*, 2006). Os sensores listados nas subseções 2.5.1.1, 2.5.1.2, 2.5.1.3, 2.5.1.4, 2.5.1.5 e 2.5.1.6 podem ser utilizados tanto em ambientes abertos quanto fechados, com exceção do GPS, que só pode ser utilizado no primeiro caso.

#### 2.5.1.1 Acelerômetro

O acelerômetro é um sensor utilizado para medir a aceleração, sendo esta dividida em: estática, como a força da gravidade ou dinâmica como, as causadas por movimentos de aceleração ou frenagem de corpos. Com a utilização deste sensor é possível estimar a aceleração linear nos eixos cartesianos e, a partir disso, determinar em que direção o VANT está se deslocando (NONAMI *et al.*, 2010).

#### 2.5.1.2 GPS

O GPS é responsável por fornecer a localização global, é um sistema de navegação baseado em satélites que orbitam a terra. Este sensor é o mais utilizado

para práticas de pilotagem *outdoor*, uma vez que em ambientes fechados (*indoor*) seu sinal fica prejudicado por obstáculos como telhados ou construções adjacentes (NONAMI *et al.*, 2010).

### **2.5.1.3 Giroscópio**

Sensor capaz de medir as variações angulares de algum objeto. Os VANTs são equipados com giroscópios, capazes de medir as variações em cada eixo, dessa maneira é possível realizar a medição da inclinação do VANT durante o voo (NONAMI *et al.*, 2010).

### **2.5.1.4 Magnetômetro**

Utilizado para fazer a medição da intensidade, direção e sentido do campo magnético. O campo magnético medido pelo magnetômetro é o próprio campo magnético da terra, que conseqüentemente faz com que o mesmo obtenha a orientação do norte geográfico da Terra. O magnetômetro é um substituto da bússola comum e, assim como nos sensores já citados, ele também é capaz de fazer a análise nos três eixos. (3DROBOTICS, 2015)

### **2.5.1.5 Barômetro**

O barômetro é um sensor capaz de indicar o valor da pressão atmosférica exercida sobre um determinado ponto. Assim, ele pode ser utilizado para estimar um valor mais preciso da altitude do VANT em relação ao GPS (JIONG *et al.*, 2010).

### **2.5.1.6 Sensores Adicionais de Navegação**

A informação de sensores adicionais para navegação automática ou autônoma é essencial uma vez que a informação destes sensores é processada por algoritmos que fazem o papel de evitar a colisão. Esta técnica é denominada *see-and-avoid* (ver e evitar). Nesta subseção são abordados os tipos de sensores adicionais que podem ser empregados nesta técnica.

### 2.5.1.6.1 Ultrassônicos

Os sensores ultrassônicos são sensores que têm a capacidade de enviar um sinal sonoro e por meio da reflexão deste sinal, no encontro de um objeto, realizar uma estimativa da distância através do tempo levado para esta onda retornar à origem.

São sensores baratos e leves, possuindo em torno de 10 gramas, tomando modelos com distância máxima aproximadamente de 3 metros. A Figura 2.12 ilustra um modelo que tem a capacidade de realizar medições entre 10 centímetros e 3 metros.

Figura 2.12 - Sensor Ultrassônico



Fonte: [http://ecx.images-amazon.com/images/I/51tbSvuOcAL.\\_SY300\\_.jpg](http://ecx.images-amazon.com/images/I/51tbSvuOcAL._SY300_.jpg)

Apesar de serem sensores de baixo custo e baixo peso, são sensores que possuem curtas distâncias de medição. Assim, uma vez que o controle utiliza a abordagem *see-and-avoid*, em casos em que o VANT se desloca em velocidade elevada, ele não possui capacidade para realizar uma manobra de evasão devido ao curto espaço de detecção proporcionado pelo sensor.

### 2.5.1.6.2 Câmeras

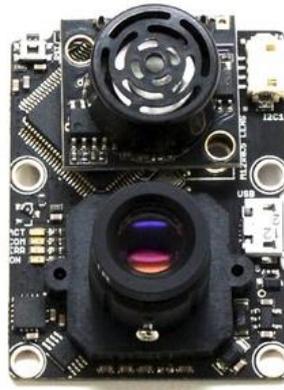
As câmeras já são utilizadas em diversas aplicações e em diversas áreas para reconhecimento de padrões e também podem ser empregadas em VANTs. Podem identificar objetos muito mais distantes que os sensores ultrassônicos e infravermelhos.

Entretanto, necessitam de algum *software* com algoritmos de reconhecimento de padrões, os quais possuem um custo computacional elevado. Esta característica limita o uso de câmeras para certas aplicações em VANTs, uma vez que seus

controladores de voo não possuem recursos necessários para o processamento dessas imagens.

Porém, algumas soluções comerciais já possuem um processador com poder de processamento elevado embarcado junto à câmera. Dentre elas podemos citar a câmera PX4FLOW, desenvolvida pela 3DR e ilustrada na Figura 2.13.

Figura 2.13 - Câmera PX4FLOW



Fonte: [https://pixhawk.org/\\_media/modules/px4flow-bottom.jpg?w=345&tok=f67039](https://pixhawk.org/_media/modules/px4flow-bottom.jpg?w=345&tok=f67039)

### 2.5.2 Módulo para telemetria

Este módulo (Figura 2.14) estabelece um *link* de comunicação em tempo real com o controlador de voo, repassando os dados de voo para o software do controlador (GS).

Esta é uma comunicação de duas vias, ou seja, o controlador pode conversar com o GS e vice-versa.

Figura 2.14 - Módulo para telemetria



Fonte: <https://store.3drobotics.com>

### 2.5.3 Rádio controle

O rádio controle é responsável por realizar a comunicação entre o usuário e o VANT de modo unilateral, ou seja, as informações são repassadas apenas do Rádio Controle (transmissor) para o seu receptor que está acoplado ao controlador VANT. Através do Rádio Controle é possível executar os comandos manuais de voo e também escolher os modos de voo. Existem modelos de rádio controle que podem alcançar dezenas de quilômetros de distância operacional. A Figura 2.15 ilustra o Rádio Controle junto com seu receptor.

Figura 2.15 - Rádio controle (Transmissor e Receptor)



Fonte : <http://www.leetshop.com/Photos/RC-RCS-p149801-1.jpg>

### 2.5.4 Unidade de processamento

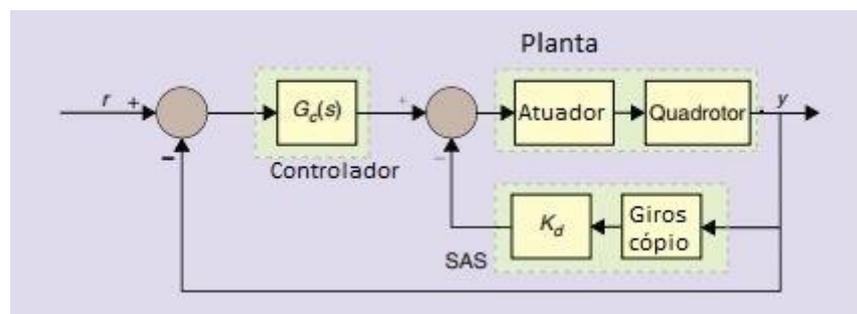
Segundo Paula (2012), a unidade de processamento é responsável pelo controle geral das aeronaves. Esta unidade faz a aquisição dos sinais obtidos pelos sensores e mediante a execução de algoritmos de controle faz o acionamento devido de cada motor para que este se mantenha estável.

Como está se trabalhando com muitos sensores e os mesmos estão sobre constante vibração mecânica, é comum a implementação de um filtro de Kalman. Como este item não é o foco do trabalho, maiores informações podem ser encontradas na referência (MARINA *et al.*, 2012).

Segundo Miller (2011), o controle sem a realimentação de erros de um VANT é instável, pois o modelo revelou que os polos estão alocados no lado direito do plano complexo (real-imaginário). Por esse motivo é necessária à utilização de algum método de controle com realimentação de erro dos sensores para que a estabilização do mesmo seja possível (LIM *et al.*, 2012).

Um método de controle para estabilização muito utilizado é o controle do tipo SAS (*Stability Augmentation System*). Já o controle de atitude é realizado por um controlador em um segundo *loop* com um integrador, conforme ilustrado na Figura 2.16. Para mais detalhes sobre o assunto sugere-se a leitura do trabalho de Stevens e Lewis (2003).

Figura 2.16 - Controle de estabilidade e atitude.



Fonte: Adaptação de LIM, PARK, *et al.*, (2012)

### 2.5.5 Controladores comerciais

Como a atividade com VANTs vem se tornando cada dia maior (MARINA *et al.*, 2012), as empresas mostraram interesse neste seguimento e passaram a produzir controladores. Em LIM *et al.*, (2012) é realizado um estudo de nove

controladores com suas principais características. Alguns desses controladores são apresentados nesta seção.

### 2.5.5.1 Arducopter Mega (APM)

O controlador APM é um projeto que se baseia na placa Arduino e também é *Open-Source*, ou seja, o código deste controlador (*firmware*), bem como o *hardware* e *software* são completamente abertos e disponíveis para os usuários.

Este controlador é comercializado pelo grupo *3DRobotics*, que fornece os três componentes: *software*, *hardware* e *firmware*. É importante citar também que este controlador pode ser utilizado em diversas plataformas, tais como: modelos de multirrotores, asas fixas, carros e helicópteros.

O *hardware* (Figura 2.17) consiste numa placa física que contém um microprocessador (unidade de processamento) ATMEGA2560 de 8 bits com uma frequência de operação e 16MHz e os sensores de giroscópio, acelerômetro, magnetômetro e barômetro, todos estes sendo sensores de três eixos. Em versões mais recentes do controlador o magnetômetro foi retirado da placa e adicionado ao módulo de GPS para evitar interferências magnéticas causadas pelo mesmo (3DROBOTICS, 2015).

Figura 2.17 - Hardware APM 2.6



Fonte: <https://store.3drobotics.com/products/apm-2-6-kit-1>

O *firmware* é o código que está contido dentro do *hardware* e configura a APM para o determinado veículo escolhido. Constantemente a empresa *3DRobotics* atualiza estes *firmwares* e disponibiliza o código dos mesmos para os usuários

realizarem suas modificações. Este *firmware* é transferido para o *hardware* através do *software* disponibilizado pela mesma.

O *software* (Figura 2.18), também chamado de *Ground Station* (GS), não é responsável apenas em fazer a transferência do *firmware*, mas também é utilizado para configuração dos parâmetros do usuário e visualizar os dados emitidos pelos sensores em tempo real por telemetria. Existe também a possibilidade de ser alterado o modo de voo pelo *software*.

O protocolo de comunicação utilizado pelo APM é o *MAVlink* (*Micro Air Vehicle Communication Protocol*), este protocolo de comunicação é o utilizado para comunicar o *software* (*Mission Planner*) com o controlador, que por sua vez, é responsável por repassar a missão configurada pelo *software* e os valores dos sensores.

É importante destacar que tanto o protocolo *MAVlink* quanto o *software* (GS), *hardware* e *firmware* do controlador são *Open-Source* sobre licença GPL, ou seja, seus códigos são abertos e disponíveis para todos realizarem suas modificações.

Figura 2.18 - *Mission Planner*, *Software* utilizado pela APM



Fonte: O autor

Ainda, tendo em vista que este controlador permite o uso da técnica *Hardware-in-the-loop* (HIL), a aplicação do trabalho final foi realizada em uma placa controladora deste modelo. A técnica *Hardware-in-the-loop* será brevemente apresentada na seção 3.

### 2.5.5.2 Pixhawk

O controlador de voo *Pixhawk* (Figura 2.19) foi desenvolvido pela *3DRobotics* em parceria com o grupo *Computer Vision and Geometry Lab (CVG)* do Instituto Federal de Tecnologia de Zurique. Assim como o controlador APM da *3DRobotics*, o *Pixhawk* é uma plataforma *Open-Source* que também possui suporte de uma comunidade voluntária de desenvolvedores (PX4, 2014).

Está equipada com os mesmos sensores citados na seção 2.5.1, mas possui uma unidade de processamento 168 MHz *Cortex M4F* de 32 bits, superior ao modelo APM. O controlador *Pixhawk* suporta as seguintes plataformas: Multirrotores, Aviões, Carros, Robôs Bípedes e Barcos (PX4, 2014).

Figura 2.19 - Controlador *Pixhawk*



Fonte: [http://regmedia.co.uk/2013/12/04/pixhawk\\_montage.jpg](http://regmedia.co.uk/2013/12/04/pixhawk_montage.jpg)

### 2.5.5.3 Naza

O controlador Naza é produzido pelo grupo DJI e, diferentemente dos controladores vistos até agora, este possui código fechado e eletrônica fechada, ou seja, o usuário não tem acesso para fazer modificações nos mesmos.

Possui algoritmos avançados de estabilização de atitude e um aprimorado modo de falha, entretanto este controlador não suporta muitas plataformas como os controladores citados anteriormente, suportando apenas multirrotores.

A Figura 2.20 mostra um controlador *Naza-M lite* com seu módulo de GPS/Bússola.

Figura 2.20 - *Naza lite* e módulo GPS/Bússola



Fonte: <http://www.dji.com/product/naza-m-lite>

#### 2.5.5.4 Multiwii

É um controlador de voo para quadrotores, desenvolvida por aeromodelistas. Este projeto utiliza uma placa Arduino como controlador central com o processador Atmel328p, sendo que a parte sensorial pode variar para cada usuário. De maneira similar aos controladores da 3DR, o controlador Multiwii também é uma plataforma *Open-Source*.

O nome Multiwii advém do fato de que no começo do projeto os sensores acelerômetro e giroscópio utilizados eram os mesmos dos controles do Nintendo Wii (LIM *et al.*, 2012). A Figura 2.21 ilustra os controles do Nintendo Wii ao lado do Arduino.

Figura 2.21 - Controles Wii e Arduino utilizados no Multiwii



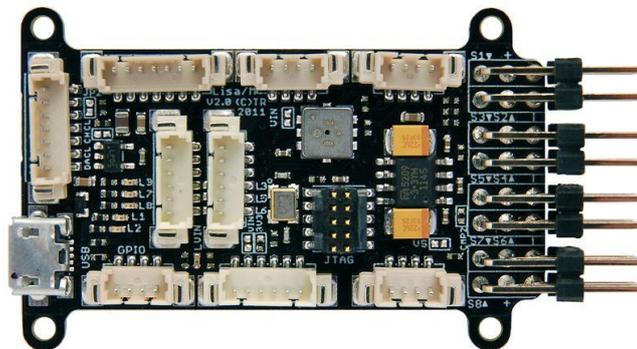
Fonte: <http://3.bp.blogspot.com/-SElpTEikj6Q/UWjMD35TE7I/AAAAAAAAADDs/DKZu3u3Nqss/s1600/controller.png>

### 2.5.5.5 Paparazzi Lisa/M

O controlador Lisa/M é um controlador de voo flexível, que pode ser utilizado tanto em multirotores quanto em asas fixas. Seu controlador possui um processador STM32 e também é um projeto *Open-Source*.

A comunidade que desenvolve o controlador Lisa possui uma documentação com relação ao uso deste controlador, tornando a sua utilização interessante, uma vez que várias informações estão disponíveis no site<sup>1</sup> desta comunidade, auxiliando novos usuários. A Figura 2.22 ilustra um controlador Lisa/M.

Figura 2.22 - Controlador Lisa/M



Fonte: [http://wiki.paparazziuav.org/wiki/File:LisaM\\_V2\\_0\\_TopView.JPG](http://wiki.paparazziuav.org/wiki/File:LisaM_V2_0_TopView.JPG)

<sup>1</sup> [http://wiki.paparazziuav.org/wiki/Lisa/M\\_v2.0](http://wiki.paparazziuav.org/wiki/Lisa/M_v2.0)

## 2.6 MODOS DE VOO

Recentemente os VANTs chegaram a um ponto de desenvolvimento em que o operador pode navegar em três tipos de voo: manual, semiautomático e automático (EISENBEISS, 2009).

### 2.6.1 Voo Manual

Neste modo de voo todos os graus de liberdade são controlados diretamente e livremente pelo piloto. Os comandos são recebidos pelo controle remoto enviado pelo piloto e nenhum controle automático é envolvido. Os dados do sistema, como bateria e sinal de rádio são observados via GS do VANT (EISENBEISS e SAUERBIE, 2011).

### 2.6.2 Voo Semiautomático

No modo semiautomático o piloto pode comandar diretamente os movimentos de rolagem, arfagem, guinada e altitude e o controlador de voo desempenha o papel de tornar a pilotagem mais simples, estabilizando o VANT na posição horizontal com base nos dados recebidos pelos sensores (EISENBEISS e SAUERBIE, 2011).

Os fabricantes de controladores 3DR e DJI pré-definem alguns modos secundários de voos semiautomáticos sendo os mais importantes:

- *Alt-Hold*: Consiste em manter a altitude atual do VANT;
- *Loiter*: Mantém o VANT parado no espaço baseado em sua posição adquirida pelo GPS. Se o VANT sofrer alguma modificação de localização em função de interferência externa, o controlador automaticamente atua a fim de trazê-lo para a posição desejada;
- *Circle*: Baseia-se em fixar o VANT na altitude atual e realizar um movimento circular com raio fixo pré-determinado via GS;
- *RTL (Return to Launch)*: faz com que o VANT retorne para a sua posição de lançamento, utilizando o GPS para realizar esta função.

### 2.6.3 Voo Automático

O piloto deixa o controle de altitude e posição para o controlador de voo, que por sua vez faz a movimentação com base na missão pré-definida por meio do GS. Como nos outros modos de voo, as informações dos sensores podem ser conferidos em tempo real via GS.

Para que este modo de voo funcione, é necessária a utilização do GS, uma vez que junto dele, o usuário planejará sua missão, definindo os pontos em que o VANT irá se deslocar até chegar ao seu destino final, denominados *WayPoints*.

Durante o período de planejamento da missão, o usuário tem o poder de inserir os valores de altitude destes pontos, bem como as rampas de subida e descida entre eles. Também é possível definir pontos onde os equipamentos carregados pelo VANT farão alguma atividade, como por exemplo, quando as câmeras farão a aquisição de imagens. A Figura 2.23 ilustra um planejamento de voo para fins de geoprocessamento, situação em que é necessário varrer uma determinada região capturando imagens.

Figura 2.23 - Planejamento de voo para geoprocessamento.



Fonte: o autor

É importante salientar que modos de voo automáticos não podem ser confundidos com modos de voo autônomos. O voo automático (também chamado de assistido) deve ser assistido por um piloto capacitado, para que em caso de falha, este assumo o comando. Já no modo autônomo ninguém pode enviar decisões para o VANT durante a execução da missão, somente é enviado o comando inicial de voo.

Segundo a ANAC (Agência Nacional de Aviação Civil), operações com VANTs ou aeromodelos operando em modo autônomo são expressamente proibidos, pois, o modo autônomo impossibilita a ação externa de um piloto em caso de falha.

Além disso, por ser um modo de voo estável, é um dos mais úteis e convencionais para se realizar voos de reconhecimento e aquisição de imagens (EISENBEISS e SAUERBIE, 2011). Em virtude destes fatos, o modo automático é o alvo deste trabalho.

## 2.7 CONSIDERAÇÕES PARCIAIS

Neste capítulo são apresentados os conceitos básicos e a história dos VANTs, bem como suas classificações. Também neste capítulo foram destacados os modelos de VANT multirotores, pois estes são alvo deste trabalho, por possuírem características como permanecerem parados no ar ou decolarem verticalmente.

O controlador é o principal elemento dos VANTs, pois é nele que são tratados todos os dados adquiridos dos sensores. Neste capítulo são apresentados alguns modelos de controladores destacando suas principais características.

O propósito deste capítulo é apresentar o comportamento de VANT quadrotor, através de seus movimentos e estilos de voos, mostrar a função do controlador de voo e abordar sobre alguns sensores usados em VANTs. Todas estas questões são importantes para o leitor compreenda mais facilmente a técnica utilizada neste trabalho para a simulação do VANT, a qual é abordada no capítulo 3.

### 3 HARDWARE-IN-THE-LOOP (HIL)

Neste capítulo aborda-se sobre a técnica denominada *Hardware-in-the-loop* (*HIL*) bem como suas características e a utilização da mesma neste trabalho.

#### 3.1 CARACTERIZAÇÃO DA TÉCNICA *HIL*

A técnica *HIL* é caracterizada pela operação de componentes reais em conexão com componentes simulados em tempo real. O processo consiste em simular parcialmente ou totalmente atuadores, sensores e processos físicos (ISERMANN *et al.* 1999).

Os benefícios da utilização do *HIL* são:

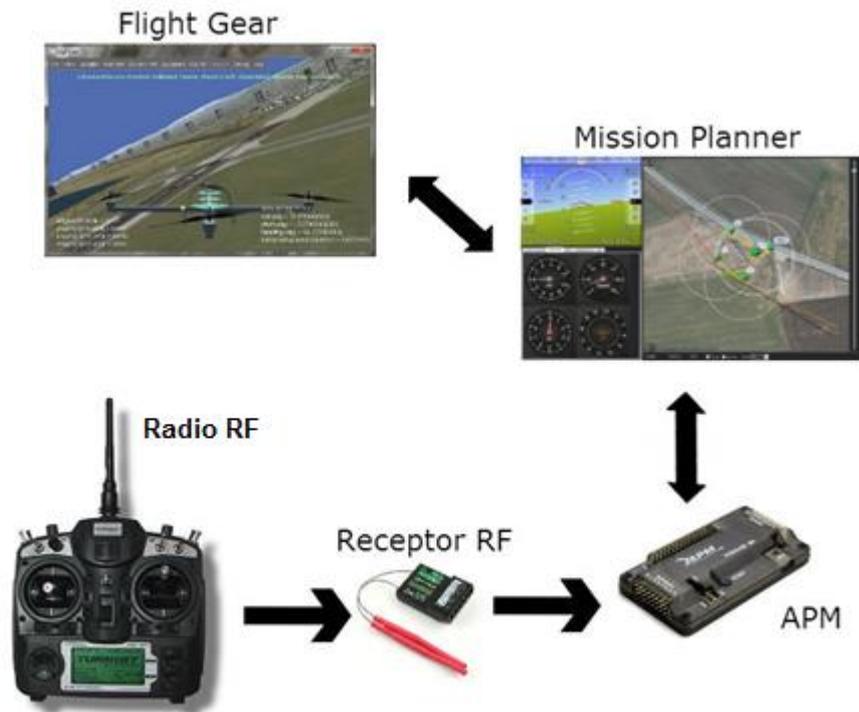
- Realizar os testes do controle do *hardware* e software sem a necessidade da real operação dos processos;
- Testar o controle do *hardware* e software sobre condições adversas extremas;
- Testar os efeitos causados pelas falhas dos atuadores ou sensores;
- Reproduzir experimentos com frequência de repetição;
- Economizar tempo e dinheiro no desenvolvimento.
- Evitar possíveis avarias em ensaios físicos.

Neste trabalho é utilizada a técnica *HIL* na forma ilustrada na Figura 3.1 Assim, para a simulação do VANT será utilizado como *hardware* o controlador APM em conjunto com o seu software *Mission Planner*, que por sua vez estará conectado com o simulador de voo *Flight Gear* (APM, 2009). É interessante salientar que utilizando esta técnica não é necessária a estrutura física do VANT, somente seu controlador para a realização desta técnica.

A partir do momento em que o controlador estiver conectado com o *software*, é necessária a utilização do *firmware* específico para a utilização do *HIL*. Este *firmware* busca os valores de todos os sensores físicos do VANT no simulador, visto que estes estão sendo emulados no mesmo. Adquiridos estes valores, o controlador atua nas saídas a fim de controlar os motores, cujos movimentos são reproduzidos no simulador. Neste caso, o controlador somente é responsável por fazer os cálculos

para estabilizar o VANT e receber possíveis instruções do piloto, via rádio controle ou GS.

Figura 3.1 - Funcionamento HIL



Fonte: O autor.

A realização desta técnica, utilizando o controlador APM, se mostrou eficaz neste trabalho, pois o comportamento simulado . Porém, para que fosse possível a utilização do mesmo, foi necessária a realização de testes para levantar as versões dos três elementos utilizados em questão (*Mission Planner*, *Flight Gear* e *Firmware*).

Este fato ocorreu porque a versão do *firmware* do controlador que suporta a técnica HIL é uma versão que foi criada em 2013, e a comunidade que desenvolve os *firmwares* passou a não se interessar mais por esta técnica, não dando continuidade aos desenvolvimentos. Entretanto, como o *software Mission Planner* continuou sendo desenvolvido, alguns problemas começaram a aparecer, com relação a esta técnica (APM, 2009). Um dos problemas percebidos consiste no fato de que os valores de velocidade e altitude de voo não são exibidos corretamente no GS. A Tabela 3.1.1 ilustra as versões do *Software* e *Firmware* utilizados neste trabalho para que a técnica HIL se mostrasse eficaz.

Tabela 3.1 - Versão dos *Softwares*

<b>Software</b>	<b>Versão</b>
<i>Flight Gear</i>	2.0
<i>Mission Planner</i>	1.3.10
HIL	2.9.1

A escolha desta técnica de simulação pode ser justificada pelo fato de estarmos utilizando o controlador de voo de forma similar à utilização real, ou seja, os cálculos e as ordens impostas ao VANT são realizados da mesma forma que a real, eliminando uma possível queda, divergindo do modo real, somente pelo fato dos sensores e o comportamento dos motores estarem sendo emulados no simulador.

A transição entre a técnica de simulação HIL e a utilização real do VANT, também é um grande potencial desta técnica, uma vez que só é necessária a mudança dos parâmetros do *firmware* para que o mesmo deixe de operar em modo HIL e passe a operar controlando o sistema real.

A técnica de simulação HIL abordada neste capítulo tem grande importância na sequência do trabalho, uma vez que esta foi utilizada como meio principal de validação da proposta deste trabalho.

### 3.2 CONSIDERAÇÕES PARCIAIS

Neste capítulo foi apresentado a técnica de simulação *Hardware-in-the-loop*, que é uma técnica que consiste em simular alguns componentes juntamente com componentes reais. Também foram mostradas as versões dos softwares utilizados para que a técnica fosse possível de ser utilizada para os testes deste trabalho.

No capítulo 4, é realizado uma revisão de Sistemas a Eventos Discretos e representação gráfica utilizando autômatos, que foram utilizados para representar a missão completa.

## 4 SISTEMAS A EVENTOS DISCRETOS E CONTROLE SUPERVISÓRIO

Neste capítulo são abordados os conceitos sobre os Sistemas a Eventos Discretos (SEDs) e como os mesmos são representados por modelos formais. Com o conhecimento dos SEDs é apresentada uma técnica de controle utilizando como base estes modelos.

### 4.1 SISTEMAS A EVENTOS DISCRETOS

Segundo Cassandras e Lafortune (2008), SED é um sistema de estados discretos cuja dinâmica é dirigida a eventos, ou seja, a mudança de estados depende da ocorrência de eventos discretos assíncronos ao longo do tempo.

Estes eventos discretos por sua vez, são gerados através de ocorrências relacionadas com o sistema em questão, podendo ser associados a ações específicas como o acionamento de um sensor ou um conjunto de ações satisfeitas, como um sensor permanecer ativado por um determinado período de tempo.

Estas ocorrências de eventos discretos causam mudanças internas no sistema, as quais podem ou não se manifestar como ações externas a um observador. Segundo Cury (2001), estas mudanças do sistema são caracterizadas por mudanças abruptas e instantâneas em intervalos de tempo irregulares e desconhecidos, ou seja, após a percepção da ocorrência do evento, o sistema reage imediatamente e acomoda-se em tempo nulo em um novo estado.

### 4.2 REPRESENTAÇÃO FORMAL DOS SEDS

Segundo Cassandras e Lafortune (2008), os SEDs não podem ser modelados adequadamente por equações diferenciais, de modo que se deve adotar outro formalismo que seja capaz de representar o comportamento deste tipo de sistema. Nesta seção é apresentadas as noções básicas sobre linguagem e autômatos, os quais consistem em metodologias formais para tal representação.

#### 4.2.1 Representação de SEDs por Linguagens

Os SEDs possuem um conjunto de eventos associados a eles denominado  $\Sigma$ . Este conjunto  $\Sigma$  é considerado o alfabeto de uma linguagem, sendo as sequências

destes eventos as palavras, que também podem ser chamadas de cadeias. Assim, um SED é modelado por uma linguagem cujas palavras representam as sequências de eventos possíveis de ocorrer no sistema (CASSANDRAS e LAFORTUNE, 2008).

Por exemplo, seja  $\Sigma = \{a, c, s\}$  um conjunto de eventos, pode-se definir as linguagens  $L_1 = \{aa, ca, s\}$  e  $L_2 = \{\text{todas as cadeias formadas com duas letras começando por } a\}$ .

É possível citar também, que o conjunto formado por todas as possíveis cadeias finitas formadas pelos elementos contidos em  $\Sigma$  é denotado por  $\Sigma^*$ , incluindo também a cadeia vazia denominada de  $\mathcal{E}$  (CURY, 2001).

#### 4.2.2 Representação de SEDs por Autômatos

Os autômatos determinísticos de estados finitos (ADEF) são capazes de representar a linguagem de um SED e são definidos por uma quintupla:

$$G = \{X, \Sigma, f, x_0, X_m\}$$

Na qual:

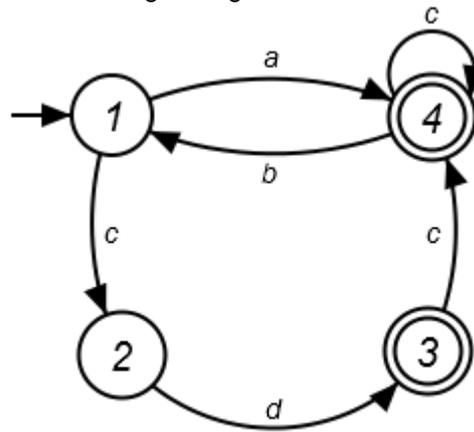
- $X$  é o conjunto finito de estados do autômato;
- $\Sigma$  é o conjunto de eventos que definem o alfabeto;
- $f : X \times \Sigma \rightarrow X$  é a função de transição de estados :  $f(x, \sigma) = y$  diz que existe uma transição rotulada pelo evento  $\sigma$  do estado  $x$  para o estado  $y$ ;
- $x_0$  é o estado inicial;
- $X_m \subseteq X$  é o conjunto de estados marcados.

Este autômato é dito determinístico, pois  $f$  é uma função de  $X \times \Sigma$  para  $X$ , ou seja, não pode haver duas transições com o mesmo rótulo saindo do mesmo estado.

O autômato por sua vez pode ser representado por um grafo orientado, onde os nós representam os estados e os arcos orientados representam as transições entre os estados. O estado inicial é identificado por uma seta apontando para ele e os estados marcados são representados por círculos duplos (CURY, 2001).

A Figura 4.1 é um exemplo de um autômato determinístico da forma  $G = \{X, \Sigma, f, x_0, X_m\}$  tal que:

Figura 4.1 - Exemplo de um diagrama gráfico de um autômato de estados finitos



Fonte: O autor

- $X = \{1, 2, 3, 4\}$ ;
- $\Sigma = \{a, b, c, d\}$ ;
- $f(1, a) = 4, f(1, c) = 2, f(2, d) = 3, f(3, c) = 4, f(4, b) = 1$  e  $f(4, c) = 4$ ;
- $x_0 = 1$ ;
- $X_m = \{3,4\}$ ;

Como é possível visualizar a Figura 4.1, neste autômato existe um caso em que  $f(x, \sigma) = x$ , ou seja, o evento  $\sigma$  leva para o mesmo estado  $x$ . Neste caso, diz-se que este estado possui um auto laço.

#### 4.2.2.1 Composição de Autômatos

A modelagem de um SED utilizando ADEF pode ser realizada fazendo o uso de dois métodos: abordando globalmente ou localmente (CURY, 2001). Quando tratada da maneira global, é necessário ser feita uma análise completa do sistema para que se representem todas as transições possíveis de eventos.

Entretanto, quando é tratado de sistemas de grande porte este método global possui uma grande complexidade, aliado a isso, existe a possibilidade de inclusão ou retirada de algum equipamento do processo, requerendo assim a reconstrução do modelo.

Por sua vez, a modelagem local do sistema parte do princípio que possam ser realizados modelos locais para cada subsistema, podendo assim compor os mesmos e obter um modelo global do sistema (CURY, 2001). Esta abordagem local

concede ao projetista maior facilidade de modelagem além de flexibilidade na alteração em algum subsistema já modelado.

Para que esta técnica local se torne equivalente à global é necessário executar os eventos compartilhados em ambos ADEFs, mantendo suas características próprias. Dessa forma a composição paralela garante estas condições compondo os ADEFs em um só.

#### 4.2.2.2 Composição paralela

Sejam dois autômatos  $G_1 = \{X_1, \Sigma_1, f_1, x_{0_1}, X_{m_1}\}$  e  $G_2 = \{X_2, \Sigma_2, f_2, x_{0_2}, X_{m_2}\}$ , a composição paralela de  $G_1$  com  $G_2$ , representada por  $G_1 || G_2$  é definida da seguinte maneira (CASSANDRAS e LAFORTUNE, 2008):

$$G_1 || G_2 = Ac \{X_1 \times X_2, \Sigma_1 \times \Sigma_2, f_{1||2}, (x_{0_1}, x_{0_2}), X_{m_1} \times X_{m_2}\}$$

Tal que as operações  $X_1 \times X_2, \Sigma_1 \times \Sigma_2$  e  $X_{m_1} \times X_{m_2}$  são a função produto, que pode ser vista em mais detalhes em (CASSANDRAS e LAFORTUNE, 2008, p. 99) e:

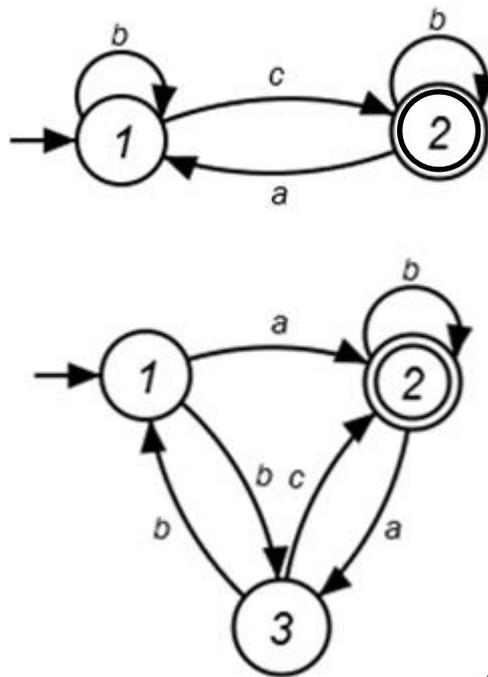
$$f_{1||2} : (X_1 \times X_2) \times (\Sigma_1 \cup \Sigma_2) \rightarrow (X_1 \times X_2)$$

Ou seja:

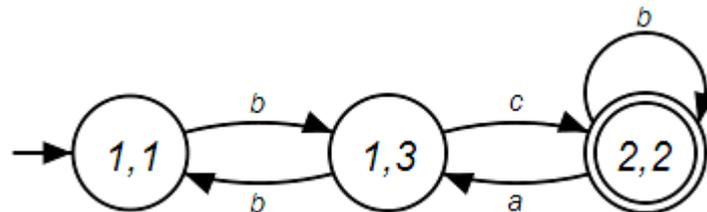
$$f_{1||2}((x_1, x_2), \sigma) = \begin{cases} (f_1(x_1, \sigma), f_2(x_2, \sigma)) & \text{se } \sigma \in \Sigma_1 \cap \Sigma_2 \text{ e } \sigma \in \Sigma_1(x_1) \cup \Sigma_2(x_2) \\ (f_1(x_1, \sigma), x_2) & \text{se } \sigma \in \Sigma_1 \text{ e } \sigma \notin \Sigma_2 \text{ e } \sigma \in \Sigma_1(x_1) \\ (x_1, f_2(x_2, \sigma)) & \text{se } \sigma \in \Sigma_2 \text{ e } \sigma \notin \Sigma_1 \text{ e } \sigma \in \Sigma_2(x_2) \\ \text{Indefinida caso contrário} & \end{cases}$$

Assim, um evento comum em  $\Sigma_1$  e  $\Sigma_2$  só pode ser executado sincronamente entre os dois autômatos, sendo que os demais eventos podem ocorrer de forma assíncrona, ou seja, de modo independente em cada autômato.

A Figura 4.2 exemplifica dois autômatos antes da sincronização e a Figura 4.3 apresenta o resultado da sincronização.

Figura 4.2 - Autômatos  $G_1$  e  $G_2$  antes da sincronização

Fonte: O Autor

Figura 4.3 - Resultado da Sincronização de  $G_1$  e  $G_2$ 

Fonte: O Autor

### 4.3 CONTROLE SUPERVISÓRIO DE SEDS

Na teoria do controle supervisorio de SEDs os eventos devem ser classificados como controláveis ou não controláveis. Por exemplo, o sinal de um microcontrolador que realiza o acionamento de um motor é um evento controlável, já uma leitura de um sensor é considerada um evento não controlável. Segundo Queiroz (2004), o objetivo do controle supervisorio é gerar um indivíduo denominado supervisor, que desabilitando os eventos controláveis evite que o sistema

(comumente chamado de planta) viole um conjunto de regras, intituladas especificações de controle. Desta forma, o supervisor deve garantir que a planta apresente apenas comportamentos que não violem as especificações de controle. Segundo Cury (2001), considera-se que o supervisor S atua em uma estrutura de malha fechada com a planta G (ver Figura 4.4) onde S observa os eventos ocorridos em G e toma as decisões nos eventos que são controláveis.



Fonte: Adaptação de (CURY, 2001)

### 4.3.1 Especificação de controle

Segundo Queiroz (2004), dentre as principais especificações de controle estão as que podem ser classificadas como: segurança e vivacidade. As especificações de segurança procuram garantir a integridade física dos equipamentos bem como dos usuários do sistema. Já as especificações de vivacidade podem ser vistas como as responsáveis para que o sistema sempre possa completar as tarefas propostas.

De forma semelhante à modelagem da planta, as especificações de controle também podem ser representadas por ADEFs. A modelagem das mesmas pode ser vista com mais detalhes em Cassandras e Lafortune (2008, p. 140). São estas especificações de controle que compostas via composição paralela em conjunto com o modelo da planta G geram a linguagem alvo. Esta linguagem alvo contém todas as transições possíveis de serem realizadas pela planta G e que atendem as especificações de controle. Entretanto, podem existir cadeias de eventos não controláveis, ou seja, estas cadeias não podem ser realizáveis a partir de uma ação de controle (CURY, 2001).

Para que haja a solução do problema de controle supervisorório é essencial o conceito de controlabilidade de linguagens.

### 4.3.2 Controlabilidade de linguagens

Segundo Cury (2001), dada uma planta  $G$ , com comportamento  $(L(G), L_m(G))$  e estrutura de controle  $\Gamma$ , definido sobre o conjunto de eventos  $\Sigma$  e a linguagem  $K \subseteq L(G)$ ,  $K$  é dita ser controlável com respeito a  $G$  se  $\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K}$ .

Na qual:

- $\overline{K}$  é o prefixo-fechamento de  $K$ ;
- $\Sigma_u$  é o conjunto de eventos não controláveis;
- $L(G)$  a linguagem gerada do autômato  $G$ .

Em palavras, uma linguagem  $K$  é controlável se para todos os seus prefixos  $s \in \overline{K}$ , a ocorrência de qualquer evento não controlável  $u \in \Sigma_u$  não leva ao descumprimento de nenhuma especificação de controle, ou seja,  $su \in \overline{K}$ . Segundo Cassandras e Lafortune (2008), a expressão da linguagem pode ser reescrita da seguinte maneira: para todo  $s \in \overline{K}$ , para todo  $e \in \Sigma_u$ ,  $se \in L(G) \rightarrow se \in \overline{K}$ .

Como dito anteriormente, este resultado é fundamental para a solução do problema de controle supervisor. Para que seja possível o cálculo do supervisor, são necessários algoritmos que a partir da linguagem alvo eliminem as transições que levam a maus estados. Estes maus estados são os estados da linguagem alvo ( $K$ ) que tentam proibir eventos não controláveis que são possíveis de ocorrer no estado correspondente da planta. Por este motivo existe a necessidade da realização do cálculo da máxima sublinguagem de  $K$  que é controlável em relação a  $L(G)$ , ou seja, deve-se calcular o que é conhecido como  $SupC(K,G)$ , como citado na Seção 4.3.

## 4.4 CONSIDERAÇÕES PARCIAIS

Neste capítulo, são abordados conceitos de sistemas a eventos discretos bem como a representação dos mesmos utilizando autômatos. Além disso, fez-se uma breve explanação sobre a teoria de controle supervisor de SEDs.

A utilização destes conceitos neste trabalho se restringe à modelagem de missões usando ADEFs, possibilitando ainda ao usuário adicionar novas transições, gerando assim as sub-missões que serão tratadas nos capítulos seguintes.

A utilização do controle supervísório e as operações entre autômatos não foi empregada neste trabalho, porém pode ser utilizada em trabalhos futuros na criação de modelos em SEDs do comportamento de mais aeronaves operando no mesmo espaço aéreo, sendo gerenciadas por um centralizador (*Ground Station*).

## 5 DESENVOLVIMENTO

A proposta inicial deste trabalho era a criação de módulos de geração de código automático para a implantação de novos sensores embarcados no VANT, havendo assim a necessidade de serem realizadas modificações no *firmware* do controlador APM, pois o mesmo não estava programado para tratar estas informações.

A adição destes novos sensores visa que o modo de voo automático pudesse ter maior flexibilidade, ou seja, possuir o poder de tomar decisões locais para mudança de sua rota de uma missão pré-definida, mediante a leitura dos sensores.

Para que fosse possível a adição destes novos sensores, optou-se pela criação de modelos matemáticos que possibilitassem representar o comportamento dos sensores e também a missão como um todo, utilizando a teoria de SEDs.

Até o momento da definição do trabalho, o GS estava preparado somente para que uma única missão já pré-estabelecida fosse enviada para o VANT, ou seja, o usuário criava a missão, enviava para o VANT em solo e o mesmo executava esta missão em modo automático até o último *WayPoint* determinado pelo usuário. A Figura 5.1 ilustra o planejamento dos *WayPoints* no *Mission Planner* de uma missão com quatorze pontos sobre o mapa.

Figura 5.1 - Planejamento de missão



Fonte: O autor

Porém, durante o decorrer do trabalho optou-se por encaminhar a leitura destes sensores para o GS, assim centralizando as decisões tomadas pelo VANT com relação a sua nova rota.

Esta decisão foi tomada visando à realização de ensaios simulados e ensaios reais, experimentando de maneira total o trabalho, tendo em vista que, estes ensaios reais demandam maior tempo e assim, não havendo tempo hábil para a total implementação proposta na definição deste trabalho.

Todavia, para a que o VANT pudesse realizar uma nova rota mediante as informações provenientes do sensor, o GS necessitaria que estas novas rotas estivessem previstas, ou seja, existia a necessidade de criar estas novas rotas. Com esta necessidade, a escolha de particionar a missão principal mostrou-se viável, assim, transformando a missão principal em pequenos trechos denominados neste trabalho de sub-missões.

Para que os conceitos deste trabalho pudessem ser testados, optou-se pela utilização da técnica de simulação *HIL* descrita seção 3, que traz uma flexibilidade para o trabalho, uma vez que os testes realizados nesta técnica, utilizam o mesmo controlador utilizado para os testes reais, assim, facilitando a validação real.

Diante destes fatos, foram necessários alguns elementos com o fim de implementar e validar os conceitos abordados no escopo deste trabalho. A seção 5.1 trata destes elementos.

## 5.1 ELEMENTOS NECESSÁRIOS

Para que o andamento do trabalho fosse possível, optou-se pela utilização da técnica HIL, citada na Seção 3. Esta técnica foi escolhida por vários motivos, um deles é o risco elevado de falhas em testes reais, mediante as modificações realizadas, na qual uma queda pode acarretar em danos irreversíveis ao controlador e outros equipamentos do VANT, impossibilitando a continuidade do trabalho.

Outro motivo bastante relevante é a realização de testes rápidos sem que houvesse necessidade de criar um ambiente controlado para realização dos testes físicos, assim, não ficando a mercê de condições climáticas adversas.

Para utilizar a técnica foram necessários os seguintes elementos:

- Controlador de voo com *firmware* na opção *HIL*;

- *Ground Station* (GS), que cria o canal de comunicação entre o simulador e o controlador;
- Simulador, para simulação gráfica do VANT e dos valores dos sensores;
- Rádio Controle;
- Computador.

Esta estratégia não se baseia somente em *software*, mas também utiliza o *hardware* do controlador de voo, ou seja, o *hardware* utilizado em testes reais é o mesmo utilizado na simulação, distintos somente no *firmware* utilizado nas duas aplicações, já como citado anteriormente, tornando a validação real mais simples, uma vez que só é necessário modificar a versão do *firmware* para a aplicação física do VANT.

## 5.2 ELEMENTOS UTILIZADOS

Neste trabalho, os elementos utilizados da técnica são citados na Tabela 5.1. Todos estes softwares escolhidos são *Open-Source*, ou seja, não é necessária a compra de uma licença de utilização e possuem código aberto, no qual novos usuários podem desenvolver novas funcionalidades, assim incentivando a utilização de soluções livres em possíveis trabalhos futuros.

Tabela 5.1 - Elementos Utilizados no trabalho

	<b>Software</b>	<b>Versão</b>	<b>Licença</b>
<b>Simulador</b>	<i>Flight Gear</i>	2.0	GPLv2
<b>Ground Station</b>	<i>Mission Planner</i>	1.3.10	GPLv3
<b>Controlador</b>	APM	2.6	GPLv3
<b>Firmware</b>	HIL	2.9.1	GPLv3

Fonte: O Autor.

Foram também realizados testes com outras versões destes softwares e outros simuladores (*X-Plane 10*), porém somente as versões citadas na Tabela 5.1

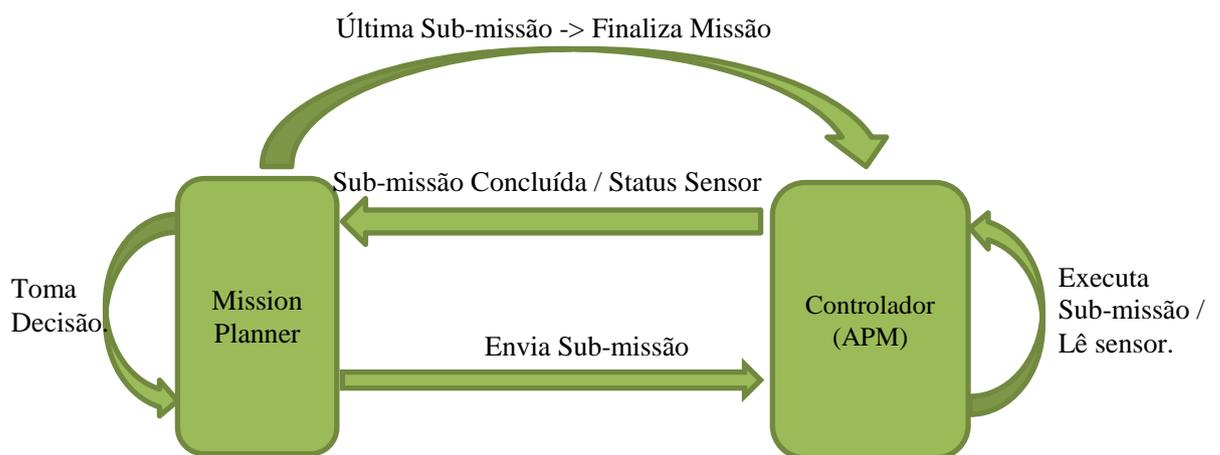
se mostraram satisfatórios, pois os testes utilizando a técnica *HIL* funcionaram, o que é observado no simulador *X-Plane 10*.

### 5.3 APLICAÇÃO

Como já citado no início desta seção, a proposta é proporcionar uma maior flexibilidade para as missões automáticas, sendo como proposta inicial a implementação de módulos de geração automática de código para novos sensores que tomariam decisões no decorrer de uma missão, criando novas rotas a serem tomadas.

Sendo assim, foi realizado um estudo da estrutura do código do *firmware* utilizado para a implementação de um novo modo de voo onde seriam levados em consideração os sensores. Entretanto, no decorrer do estudo da estrutura, foi delimitada a problemática do trabalho para que as decisões só fossem tomadas em pontos intermediários pré-definidos pelo usuário no momento do planejamento da missão (*Way Points*), fracionando a missão principal em sub-missões. A Figura 5.2 ilustra o primeiro esquema utilizado.

Figura 5.2 - Primeiro esquema de envio de sub-missões.



Fonte: O Autor

Como descrito na Seção 5, esta decisão ocorreu devido ao interesse em validar por completo o trabalho com testes reais. Entretanto, em decorrência dos

mesmos serem realizados em ambientes abertos, existe uma dependência das condições meteorológicas.

Porém, o GS utilizado não estava preparado para receber a informação de novas sub-missões, ou seja, o usuário só poderia criar uma missão principal que seria enviada de maneira integral para o VANT e executada pelo mesmo de maneira automática, executando-a até o fim. Durante a execução desta, o GS não possuía nenhum controle sobre o VANT, com exceção o comando de abortar.

Desta maneira, se fez a necessidade do uso de uma ferramenta que pudesse entender os arquivos gerados da missão principal pelo GS e o usuário tivesse a possibilidade de criar sub-missões derivadas da missão principal mediante ao acionamento de um sensor genérico.

A partir desta condição foi utilizada a técnica de controle supervísório citada na Seção 4 utilizando o software *Nadzoru*<sup>2</sup>. Este *software* foi desenvolvido e implementado por Yuri Kaszubowski Lopes com contribuições de Eduardo Harbs sobre supervisão de André B. Leal e Roberto S. U. Rosso Jr., no grupo de Automação de Sistemas e Robótica (GASR) da UDESC.

A utilização do *Nadzoru* no contexto deste trabalho se dá da seguinte forma. Inicialmente é realizada a importação no *Nadzoru* dos arquivos gerados pelo GS e então estes são transformados em ADEFs, em arquivo passível de manipulação pelo *Nadzoru*. Desta forma, o autômato obtido pode ser modificado de modo a contemplar novas transições entre os pontos da missão.

Após este procedimento o *software Nadzoru* cria as novas sub-missões gerando os arquivos de texto das mesmas, que são utilizadas pelo GS em decorrência da ativação do sensor e enviadas para o VANT.

Sendo assim, a aplicação total possui as seguintes etapas:

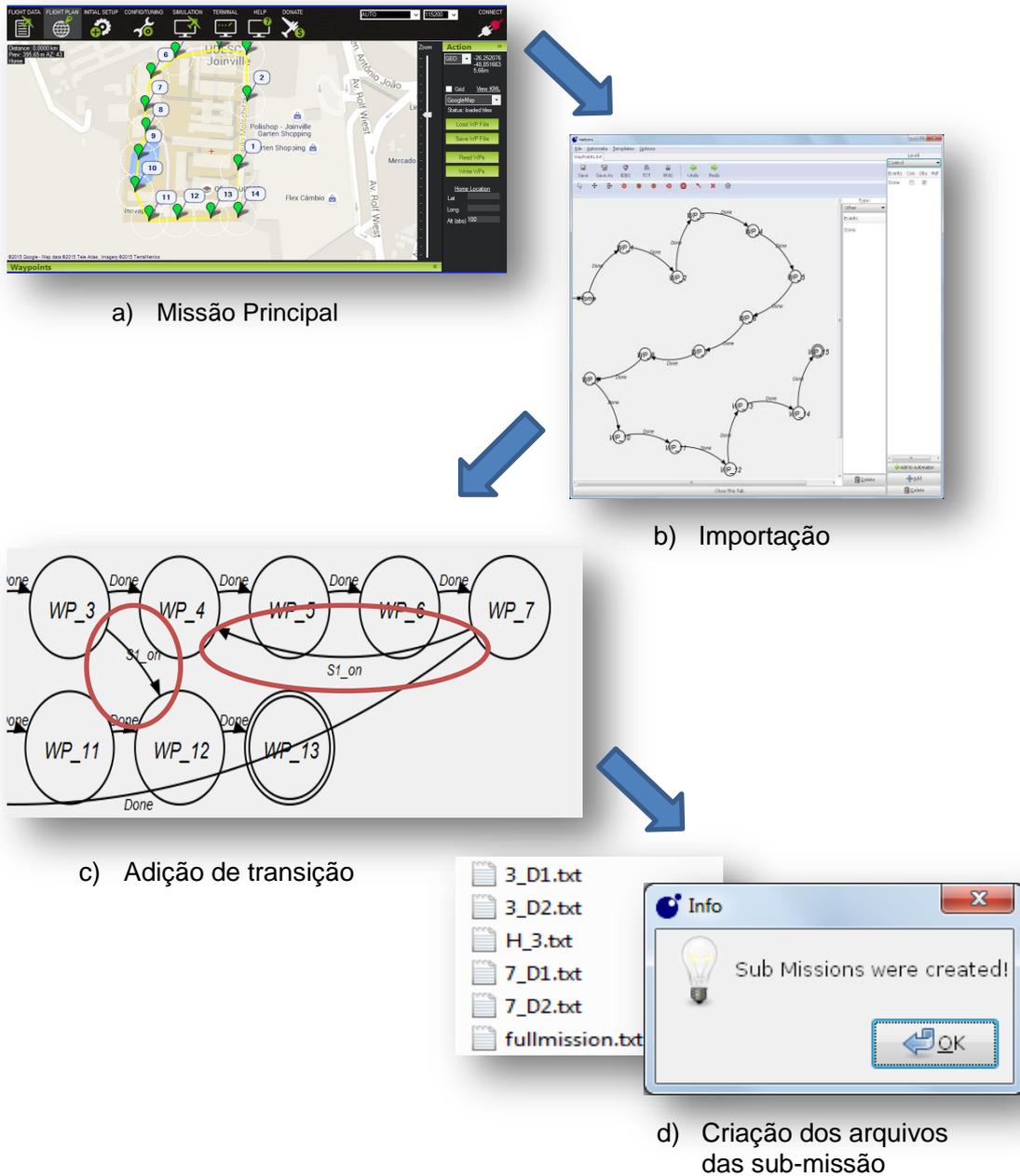
- Criação da missão principal (Figura 5.3 (a));
- Importação da missão principal para o software *Nadzoru* transformando-o em ADEF (Figura 5.3 (b));
- Adição de novas transições entre os *WayPoints* (pontos intermediários da missão) (Figura 5.3 (c));

---

<sup>2</sup> O *Software Nadzoru* possui várias operações voltadas para a teoria de controle supervísório de SEDs e diversas funções têm sido desenvolvidos pela equipe do GASR e também pelo seu idealizador, que atualmente realiza doutorado pleno na Universidade de Sheffield, no Reino Unido. Para mais detalhes sobre a ferramenta sugere-se consultar LOPES, et al., (2015).

- Criação das sub-missões e geração automática dos arquivos de texto das mesmas utilizados pelo GS.(Figura 5.3 (d));
- Utilização dos arquivos de texto das sub-missões pelo GS, mediante o status do sensor.

Figura 5.3 - Etapas da aplicação



Fonte: o Autor

O objetivo principal desta aplicação é criar um modo de voo automático com o controle da missão no GS, ou seja, dependendo do status do sensor o GS enviará uma sub-missão, se esta sub-missão chegou ao fim, realiza um teste para ver se existe uma decisão neste ponto.

#### 5.4 CONSIDERAÇÕES PARCIAIS

Neste capítulo foram detalhados os elementos necessários para que a aplicação fosse possível, bem como os elementos utilizados. As versões destes elementos são mencionadas, uma vez que é necessária a utilização destas versões para que a aplicação opere de maneira correta.

Também foi apresentada a aplicação deste trabalho que utiliza a representação gráfica de autômatos da missão principal no *software Nadzoru* realizando a criação de novas transições nos pontos de decisão mediante a leitura do sensor virtual na estação terrestre. O capítulo 6 trata da implementação realizada e os resultados simulados e reais obtidos através da mesma.

## 6 IMPLEMENTAÇÃO E RESULTADOS

Neste capítulo são apresentados a implementação da proposta bem como os testes realizados utilizando a técnica de simulação HIL e ensaios realizados fisicamente.

### 6.1 IMPLEMENTAÇÃO

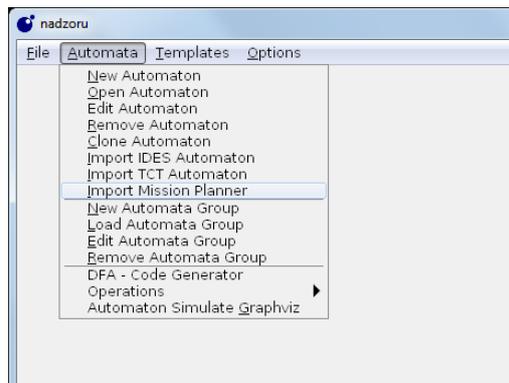
No início deste trabalho, as implementações seriam realizadas no *firmware* do controlador, adicionando novos sensores que pudessem flexibilizar o modo de voo automático, tomando decisões localmente mediante a leitura desses sensores. Porém, como citado na Seção 5, as informações destes sensores são enviadas para a GS, assim centralizando as tomadas de decisão.

O propósito desta implementação é criar um modo de voo automático, no qual a missão principal seria particionada em sub-missões, em que sobre alguns *WayPoints*, pré-definidos pelo usuário, havendo a tomada de decisão de acordo com a leitura dos sensores.

Todo este processo de particionamento da missão principal em sub-missões foi realizado no *software Nadzoru*. Este software foi desenvolvido na linguagem LUA de programação, que por sua vez, é uma linguagem de *script* imperativa e procedural (LUA, 2005). O código fonte do *Nadzoru* foi disponibilizado pelo próprio criador e está sobre as licenças GNU LGPLv3.

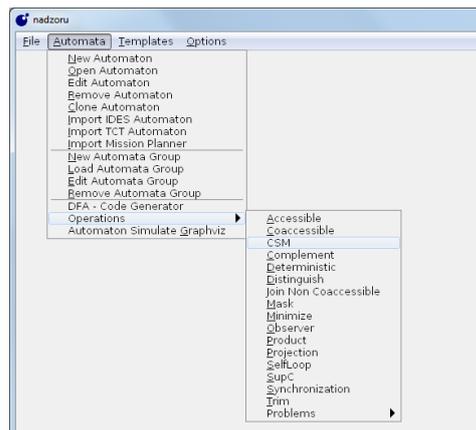
No *Nadzoru* foram criados os módulos de importação de missões do *Mission Planner* e os módulos de exportação de sub-missões. A Figura 6.1 e Figura 6.2 ilustram o novo modo de importação e exportação das sub-missões na tela do *Nadzoru*, respectivamente.

O módulo de importação tem o objetivo utilizar o arquivo de texto, gerado pelo *Mission Planner* (GS) da missão principal, com os *WayPoints* e suas respectivas coordenadas geográficas, e transformar o mesmo em uma representação gráfica de um Autômato Finito Determinístico (AFD). Esta importação transforma os *WayPoints* da missão em estados no autômato, mantendo o rótulo deste estado compatível com o número do *WayPoint*, utilizando um prefixo “WP\_”. Feito isto, as informações de coordenadas geográficas dos *WayPoints* são mantidas em uma variável interna do programa.

Figura 6.1 – Nova importação no *Nadzoru*

Fonte: o Autor

Figura 6.2 - Criação de sub-missões (Exportação)

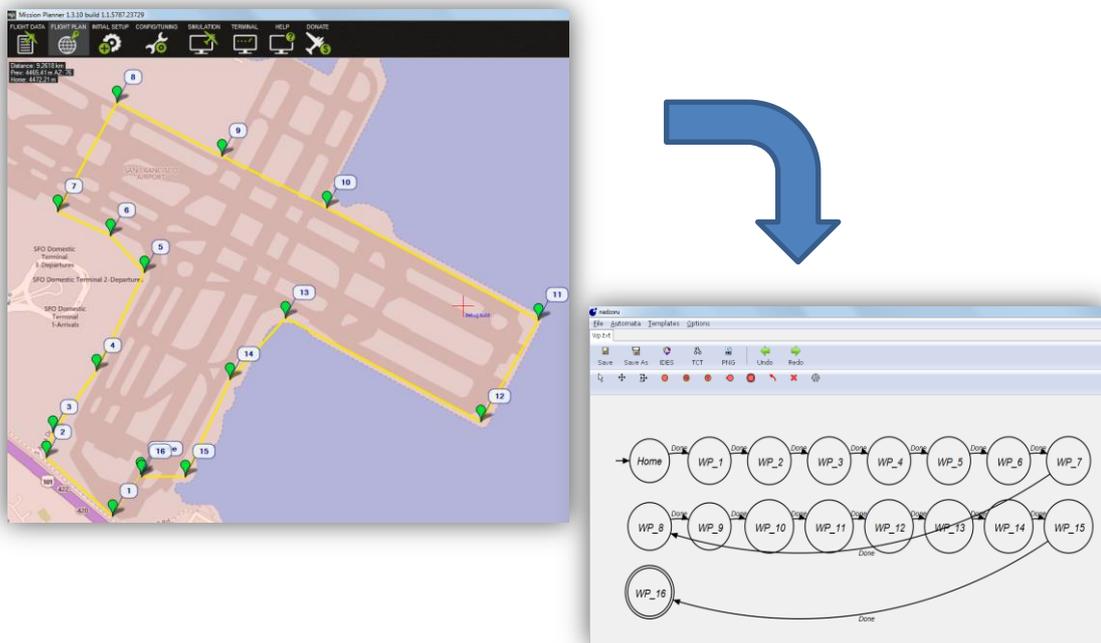


Fonte: o Autor

Após estes procedimentos, ocorre a interligação dos estados automaticamente, com o evento *Done*, simbolizando que a transição entre um *WayPoint* e o outro ocorreu.

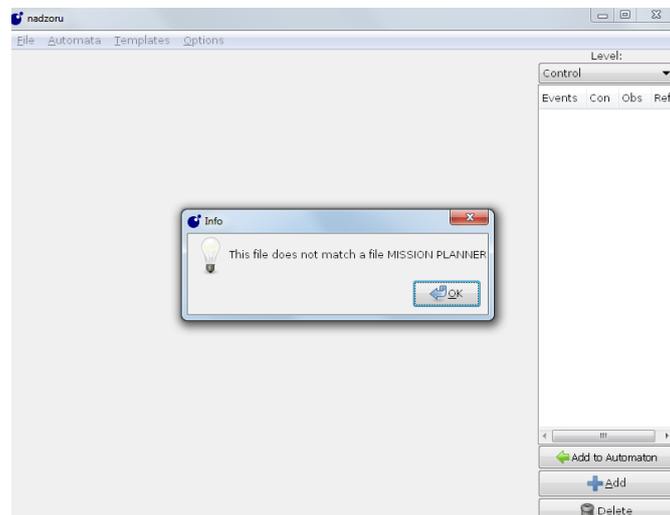
Esta importação toma como estado inicial do autômato o *WayPoint Home* da missão principal, já o evento final (marcado) é obtido como o último *WayPoint* da missão. A Figura 6.3 ilustra um exemplo de importação no *Nadzoru* de uma missão gerada no *Mission Planner*. Neste módulo também está implementado um dispositivo que impede que o usuário importe um arquivo que não tenha origem do *Mission Planner* (Figura 6.4).

Figura 6.3 - Resultado da Importação



Fonte: o Autor

Figura 6.4 - Proteção contra importações incorretas



Fonte: o Autor

Uma vez criado o modelo, o usuário pode criar novas transições entre os estados, assim impondo que nos estados de origem das novas transições, sejam adicionados pontos de decisão, em consequência de que por padrão os arquivos

vindos do *Software Mission Planner*, existem somente uma transição de saída em cada ponto da missão. Estas informações de transições de saídas são muito importantes, uma vez que as mesmas são utilizadas no módulo de criação de sub-missões.

O módulo de criação de sub-missões, denominado neste trabalho de “CSM” (*Create Sub-Missions*), faz a utilização do ADEF gerado anteriormente, concebido da importação da missão principal. Através deste ADEF é feito o uso das informações dos rótulos dos estados que representam os *WayPoints* e também é utilizada a informação das transições de saída existentes no estado, como citado anteriormente.

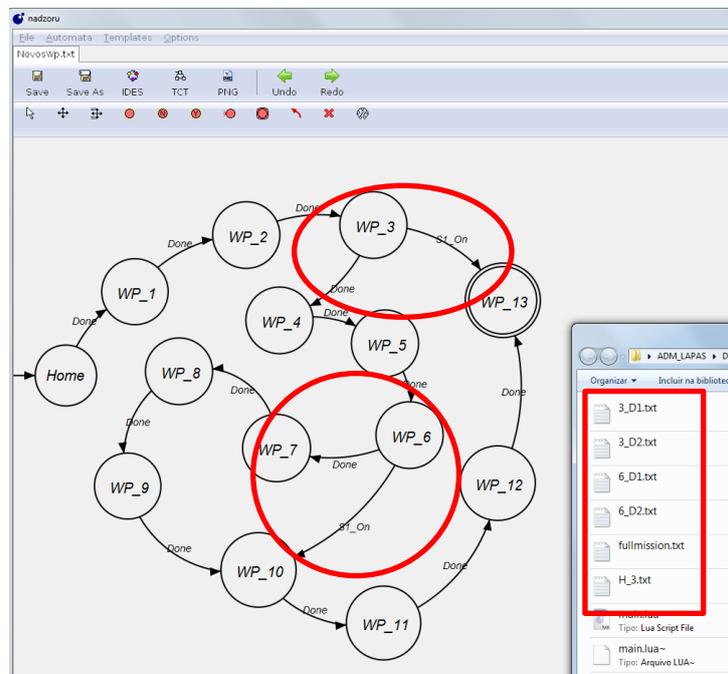
Estas informações são cruciais para a criação de novas sub-missões, uma vez que por padrão só existe uma transição de saída entre os estados, ou seja, se existir mais de uma transição de saída, aquele *WayPoint* será tratado com ponto de decisão, assim fracionando a missão principal até aquele ponto. A partir destes pontos de decisão são criadas duas novas sub-missões, a sub-missão D1 e D2, que levam o prefixo do nome do *WayPoint* com decisão, por exemplo, se no *WayPoint* 3 existir um ponto de decisão, ou seja, duas transições de saída, serão criadas as sub-missões “3\_D1” e “3\_D2”.

Neste trabalho optou-se por utilizar a sub-missão D1 como a sequência da missão principal e que deve ser tomada mediante a não ativação do sensor, e a sub-missão D2 como a sub-missão que deve ser tomada mediante a ativação do sensor. A sub-missão D1 é constituída com a sequência do *WayPoint* seguinte à decisão, até o próximo ponto de decisão ou fim da missão. Já a sub-missão D2 é constituída pela sequência do *WayPoint* destino da transição criada no ponto de decisão, até o fim da missão ou até o próximo ponto de decisão.

Para que toda a missão principal seja contemplada pelas sub-missões, quando existe a primeira ocorrência de ponto de decisão, o sistema cria uma sub-missão com o prefixo “H\_” seguido do ponto de decisão, indicando que o voo ocorrerá até aquele ponto, partindo da origem. Feitas as transições no modelo e utilizada a operação “CSM”, citada anteriormente, o *software Nadzoru* transforma estas sub-missões em arquivos de texto, no mesmo formato utilizado pelo *Mission Planner*, podendo assim ser enviadas para o VANT. A implementação também realiza a criação de um arquivo denominado “*fullmission*” apenas para ser utilizada como controle na utilização das sub-missões no *Mission Planner*.

A Figura 6.5 mostra os arquivos gerados mediante o uso desta ferramenta, utilizando um exemplo de missão, com pontos de decisão representados pelo autômato da mesma figura, onde os mesmos estão sendo destacados. É possível tomar o exemplo do estado referente ao *WayPoint* 3 onde existem duas transições: uma transição padrão “Done” que leva ao *WayPoint* 4, e outra transição “S1\_on” que leva ao *WayPoint* 13. Neste caso são geradas duas sub-missões referentes a este ponto: 3\_D1 que é sub-missão referente a transição “Done” e 3\_D2 referente a transição “S1\_on”.

Figura 6.5 - Arquivos criados pela operação CSM



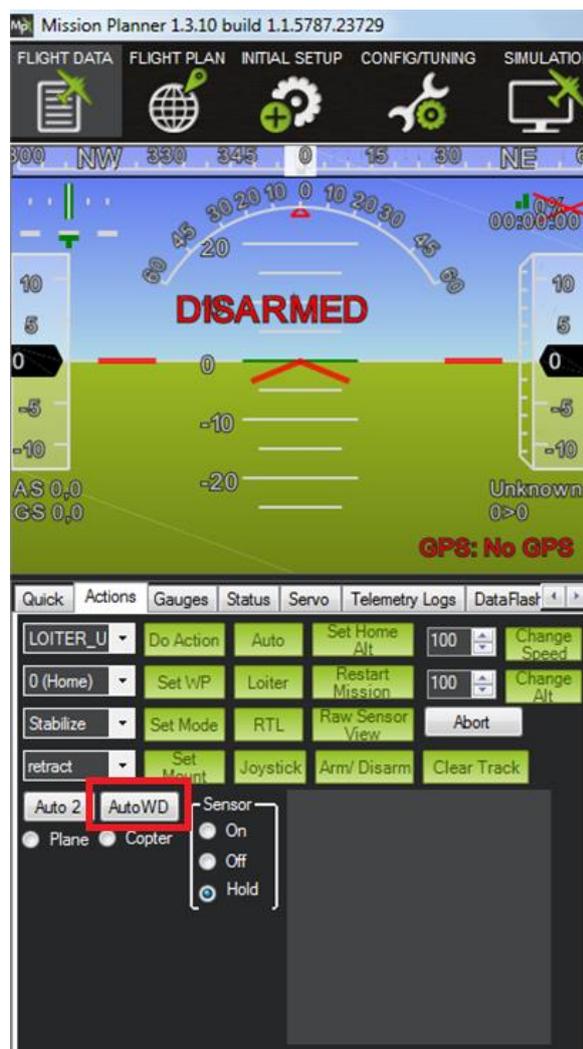
Fonte: o Autor

Após a criação destes módulos no *Nadzoru*, identificou-se a necessidade de realizar algumas mudanças no *software Mission Planner*, uma vez que este é responsável por enviar as missões para o VANT. O código fonte do *Mission Planner* foi obtido através de Gomes (2014), pois foram utilizadas algumas funcionalidades implementadas neste trabalho, entretanto o código fonte original pode ser obtido através do repositório GitHub (OBOURNE, 2014).

Diferentemente do *Nadzoru*, o *Mission Planner* foi desenvolvido na linguagem de programação C# e neste trabalho seu código foi compilado na ferramenta *Visual Studio 2013*.

Como o processo de missão automática no *Mission Planner* não previa mais de uma missão, foi necessário criar um novo modo automático no *Mission Planner* tal que fosse possível escolher entre as sub-missões, qual deveria ser enviada para o VANT mediante a leitura dos sensores. Assim, neste trabalho foi desenvolvido um modo denominado de *AutoWD (Automatic With Decisions)*, o qual foi integrado ao *Mission Planner*. A Figura 6.6 ilustra o botão (destacado em vermelho) responsável por ativar este modo no *Mission Planner*.

Figura 6.6 - Botão AutoWD



Fonte: o Autor

Logo abaixo do botão *AutoWD* podem ser visualizados os botões *Plane* e *Copter* que selecionam qual é o tipo de VANT que está sendo utilizado. Neste momento é importante informar que neste trabalho a função *AutoWD* está habilitada somente para o VANT do tipo *Copter*, ou seja, para os veículos aéreos de asas rotativas.

Além disso, é possível visualizar uma área no qual foi criado um sensor virtual. A escolha inicial deste trabalho era que o sensor estivesse embarcado o que foi substituído durante o desenvolvimento por um sensor virtual no *Mission Planner*. A ideia da utilização deste sensor é a possibilidade de substituir o mesmo por qualquer tipo de sensor: embarcado na aeronave ou estar em solo próximo de alguma região de interesse, o qual enviaria as informações do seu status. Neste trabalho optou-se por utilizar um sensor que possui três posições, sendo elas: *On* (Ativado), *Off* (Desligado) e *Hold* (espera).

Quando ativado o modo *AutoWD*, o *Mission Planner* inicia uma rotina de repetição buscando pelos arquivos das sub-missões. Primeiramente esta rotina procura pela sub-missão inicial, que possui um formato contendo no início de seu título “H\_”, como citado anteriormente, e envia a mesma para o VANT.

Após o envio desta sub-missão, o mesmo entra em modo de voo automático padrão para que a mesma seja executada. Junto com a informação da sub-missão inicial, existe também qual será o primeiro ponto de decisão, informação esta que está subsequente aos caracteres “H\_”.

Quando o VANT realiza a primeira sub-missão e chega ao ponto de decisão, o mesmo entra em modo de voo *Loiter*, o sensor é lido e é tomada a decisão mediante a sua posição, sendo elas:

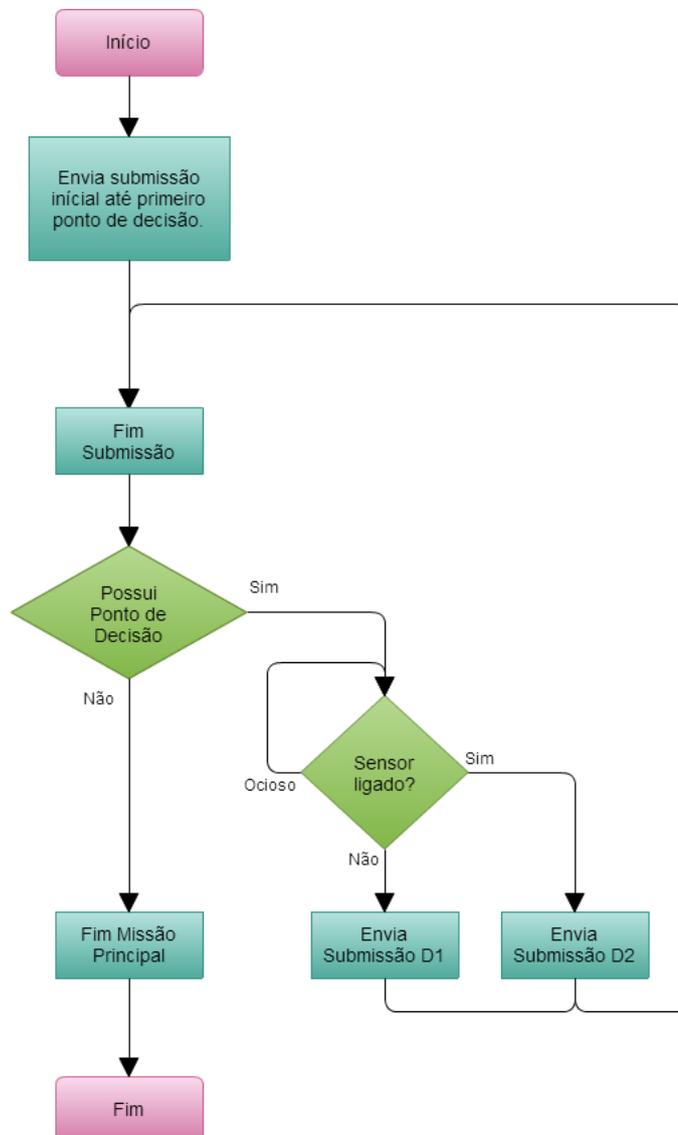
- Decisão 1 (envio da sub-missão D1) – Sensor desativado;
- Decisão 2 (envio da sub-missão D2) – Sensor ativado;
- Aguardando sensor (VANT parado e estabilizado) – Sensor posição Hold;

Estas decisões são similares aos demais pontos de decisão. Após a decisão ter sido tomada, é enviada para o VANT a sub-missão seguinte, sendo que em

seguida ao envio o VANT entra em modo de voo automático para que seja realizada a execução desta sub-missão.

Similar ao primeiro caso, quando o VANT chega ao fim da sub-missão o mesmo entra em modo de voo *Loiter* e analisa se existem arquivos das sub-missões naquele determinado *WayPoint*. Caso estes arquivos existam, o sensor é analisado e a decisão é tomada, entretanto se estes arquivos não existirem significa que a missão principal chegou ao seu fim. A Figura 6.7 ilustra um fluxograma desta nova funcionalidade.

Figura 6.7 - Fluxograma da estrutura da função AutoWD



Fonte: o Autor

Todas estas etapas são registradas em um visor desenvolvido em Gomes (2014), informando ao usuário:

- Nome da sub-missão enviada;
- Sub-missão Completa;
- Analisando o status do sensor;
- Status do Sensor;
- Missão Completa.

Também houve a utilização do botão “*Abort*” desenvolvido em Gomes (2014), o qual tem a funcionalidade a abortar o modo *AutoWD*, gerando assim maior segurança para a operação, uma vez que se ocorrer algum problema durante a execução da mesma o piloto pode retomar os controles manualmente.

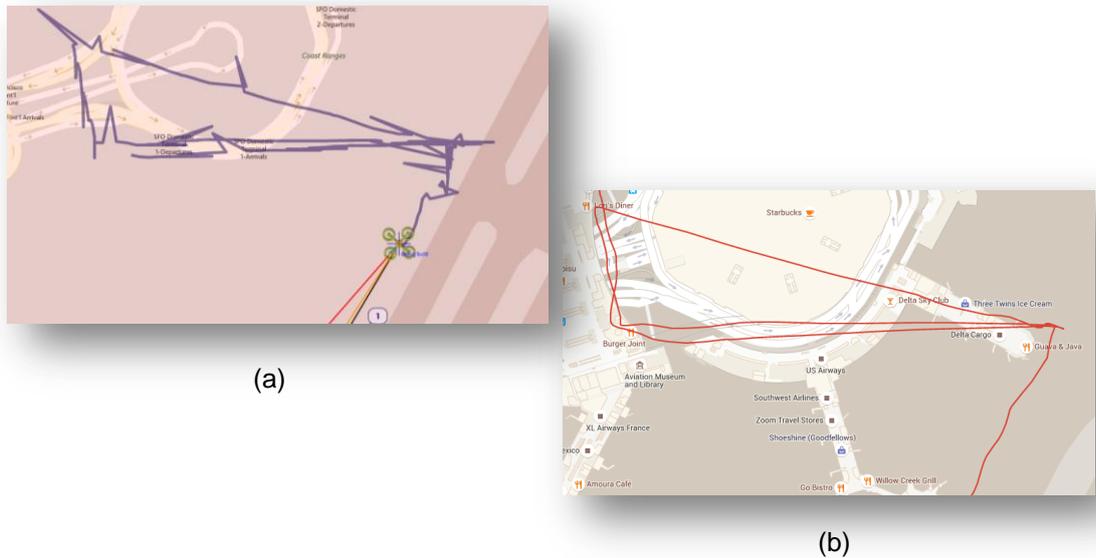
## 6.2 RESULTADOS

Nesta seção são apresentados os resultados obtidos executando o modo *AutoWD* apresentado na Seção 6.1 e implementado neste trabalho. Como descrito anteriormente, este modo só foi implementado para modelos de asa rotativa, sendo assim os resultados apresentados nesta seção são apenas resultados obtidos com esses modelos.

O *Mission Planner* possui um registro da posição que o VANT se locomove durante a missão, porém este registro é limitado, pois mantém um número pequeno de pontos, e apresentou problemas gráficos durante a representação dos dados de trajetória. Diante desses fatos, foi criada uma função na aba de simulação que habilita que os pontos de latitude e longitude possam ser salvados em um arquivo de texto, periodicamente a cada 500 milissegundos. Estes pontos podem ser utilizados para serem enviados a uma API do *Google Maps* possibilitando traçar a rota que o VANT executou durante o teste sem os problemas gráficos.

A Figura 6.8 mostra um comparativo dos registros do *Mission Planner* (esquerda) e dos registros adquiridos via arquivo de texto e enviados para a API do *Google Maps* (direita).

Figura 6.8 - Comparação entre registros do *Mission Planner* (a) e API do *Google Maps* (b)



Fonte: o Autor

Sendo assim, todos os resultados mostrados neste trabalho serão os registros obtidos através dos valores de longitude e latitude salvados durante a execução do voo e representado na API do *Google Maps*.

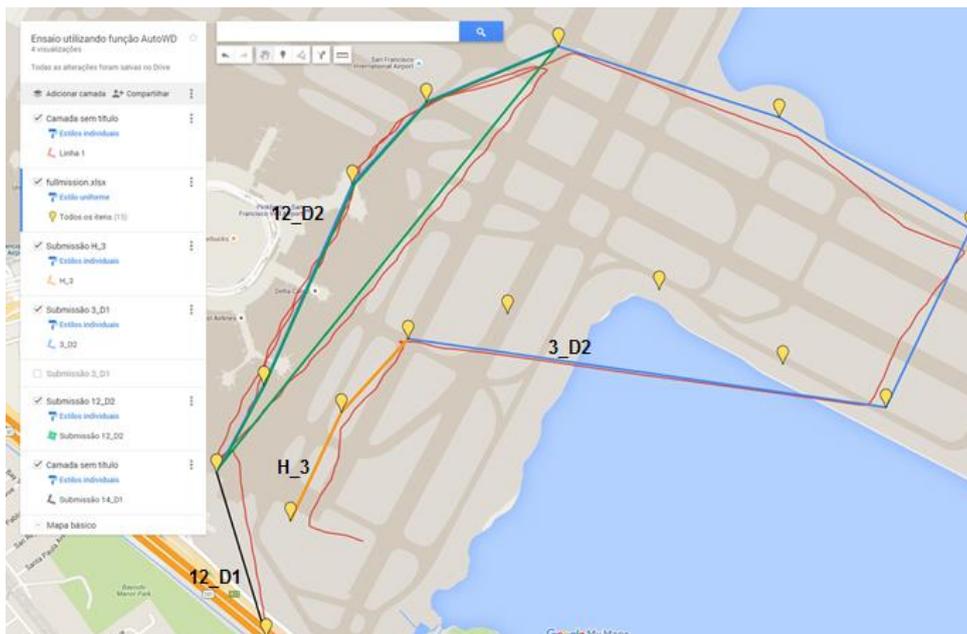
Para a realização dos testes foi planejada uma missão com 15 *WayPoints*. A missão principal foi importada para o *Nadzoru* e adicionada uma transição do *WayPoint* 3 para o 7 e uma transição do *WayPoint* 14 para o 10, totalizando 5 sub-missões.

Após o início da simulação o sensor foi ajustado para a posição “On” e como previsto, o *Mission Planner* tomou a decisão D2 enviando para o VANT a sub-missão 3\_D2. Ao chegar ao próximo ponto de decisão (14), o sensor foi posto novamente na posição “On” de modo que foi enviada para o VANT a sub-missão 14\_D2, fazendo com que o mesmo retornasse para o *WayPoint* 10. Como a sequência levava novamente para o *WayPoint* 14 e este é um ponto de decisão, o sensor foi colocado para posição “off”, levando o *Mission Planner* a enviar a sub-missão 14\_D1. Como

nenhum ponto de decisão foi encontrado e o VANT chegou ao *WayPoint* final, a missão principal foi finalizada.

A Figura 6.9 ilustra o deslocamento do VANT no decorrer desta missão, sendo a linha vermelha o trajeto realizado pelo VANT e as linhas laranja (H\_3), azul (3\_D2), verde (12\_D2) e preto (12\_D1) as sub-missões utilizadas neste ensaio.

Figura 6.9 - Ensaio utilizando função *AutoWD*



Fonte: o Autor

É perceptível que em alguns pontos a simulação se mostrou distante dos *WayPoints* definidos. Nestes ensaios, esta distância foi de aproximadamente 40 metros. Porém, estas diferenças se dão pelo fato de que os controladores internos do VANT que fazem o controle da trajetória não estavam com seus parâmetros completamente ajustados para que houvesse melhores resultados, sendo que este não era o foco principal deste trabalho.

Mais detalhes do voo simulado podem ser vistos no link <http://ricardofm.com/download/TccGianOk.mp4>, no qual foi disponibilizado o vídeo do voo simulado com algumas informações adicionais.

Durante o período final do trabalho foram realizados ensaios reais em um aeroclube da cidade de Joinville, respeitando a legislação vigente sobre VANTs. Nestes ensaios, foi utilizado um modelo de seis motores que pode ser visualizado com mais detalhes na Figura 6.10.

Figura 6.10 - VANT utilizado para testes



Fonte: o Autor

Para facilitar a criação da missão e verificar se os pontos que o VANT transitava eram os mesmos pontos definidos na estação terrestre (GS), foram utilizados marcadores no chão do local. A Figura 6.11 ilustra os marcadores dispostos.

Figura 6.11 - Marcadores utilizados (em vermelho)

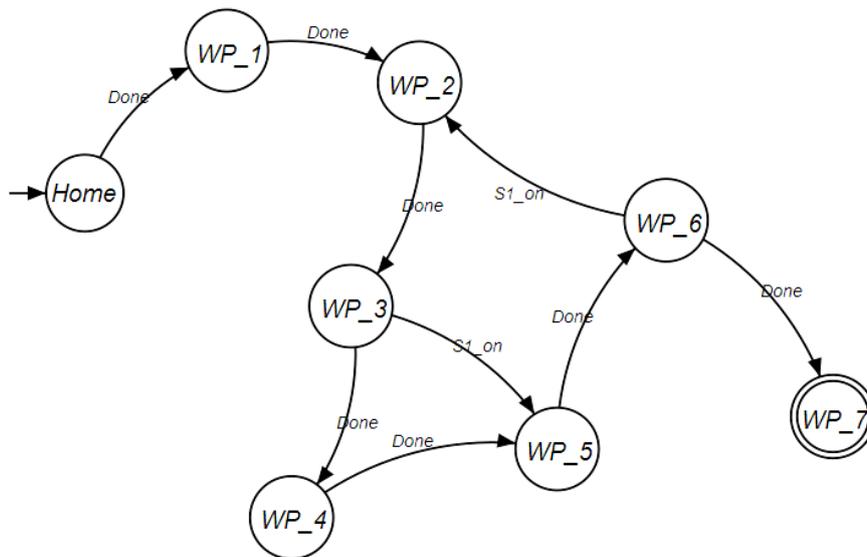


Fonte: o Autor

Foi proposto um cenário com 7 pontos intermediários (*WayPoints*), sendo que em dois destes pontos (3 e 6) ocorria a análise do sensor para determinar a decisão a ser tomada. No ponto 3 o VANT tinha a possibilidade de ser enviado diretamente para o ponto 5 se o sensor estivesse ligado, ou continuaria sua trajetória original

caso contrário. Já no ponto 6 o VANT poderia retornar ao ponto 2 caso o sensor estivesse ligado, caso contrário o mesmo seguiria até o último ponto (7). A Figura 6.12 ilustra o modelo da missão principal importada no *Nadzoru*, com as transições adicionais nos estados WP\_3 e WP\_6.

Figura 6.12 - Missão real importada no *Nadzoru*



Fonte: o Autor

Diante deste cenário foram impostas as seguintes condições. Na primeira passagem no ponto 3 o sensor seria ligado (o que é feito no MP) e seria enviado a sub-missão “3\_D2”, deslocando o VANT até o ponto 5, já na primeira vez que o mesmo atingisse o ponto 6 o sensor também seria ligado forçando o VANT a voltar até o ponto 2, enviando a missão “6\_D2”. Na segunda vez que o ponto 3 fosse atingido, seria desligado o sensor, forçando o envio da sub-missão “3\_D1” onde o VANT se locomoveria sequencialmente até o próximo ponto de decisão. De maneira similar, na segunda vez em que o VANT passasse pelo ponto 6, o sensor seria desligado novamente e o VANT seria encaminhado para o ponto 7, último ponto planejado da missão.

O trajeto desta missão foi adquirido pela própria ferramenta *Mission Planner*, sendo que nestas circunstâncias o mesmo não apresentou erros gráficos no registro da movimentação do VANT. A Figura 6.13 ilustra a trajetória percorrida pelo VANT

com os pontos definidos no início do ensaio, baseados nos marcadores do solo. Para uma melhor visualização da trajetória, a sequência de *WayPoints* percorridos pelo VANT durante o teste real foram os seguintes: {1,2,3,5,6,2,3,4,5,6,7}.

Figura 6.13 - Registro dos movimentos do VANT



Fonte: o Autor

Durante a utilização da abordagem real, o VANT não apresentou nenhum erro gráfico que era observado na técnica *HIL*. Além disso, é importante salientar que a precisão dos pontos foi de aproximadamente 1 metro tendo em vista que a distância entre os pontos eram de aproximadamente 15m, onde as condições de vento estavam em torno de 5km/h<sup>3</sup>.

Algo importante a ser destacado foram algumas mudanças realizadas no *Mission Planner* após ser realizado um primeiro teste real. Neste teste o VANT não executou a missão, em decorrência de que nos pontos de decisão o *Mission Planner* ativava modo de voo *Stabilize*, fazendo com que o controle do VANT volta-se a ser do piloto, sendo que este comportamento ocasionou uma queda do VANT.

<sup>3</sup> Mais detalhes do voo real podem ser vistos no link <http://ricardofm.com/download/TccGianPratica.mp4>, no qual foi disponibilizado o vídeo do voo com algumas informações adicionais.

Este comportamento não se manifestou durante os testes de simulação realizados com esta versão do *Mission Planner* que utilizava modo de voo *Stabilize* nos pontos de decisão. A maneira encontrada para correção deste problema foi trocar o modo de voo *Stabilize* para o modo de *Loiter* que mantém o VANT parado no ar, como citado na seção 2.6.2.

Entretanto ao utilizar a nova versão do *Mission Planner* durante simulação o VANT apresentou um comportamento instável devido ao uso *Loiter*. Deste modo, foram criadas duas versões para o *Mission Planner*, sendo uma delas utilizada nos ensaios de simulação, onde nos pontos de decisão o modo de voo *Stabilize* é acionado e outra versão utilizado no ensaio real, onde o modo *Loiter* é acionado nos pontos de decisão.

Ambas estas versões do *Mission Planner*, bem como a versão do *Nadzoru* utilizado neste trabalho será anexado a um DVD junto com este trabalho, bem como algumas instruções para a compilação do código fonte.

### 6.3 CONSIDERAÇÕES PARCIAIS

Neste capítulo foram apresentados às implementações realizadas na estação terrestre *Mission Planner*, bem como no *software Nadzoru* que representa graficamente os Sistemas a Eventos Discretos (SEDs) utilizando autômatos.

Na primeira ferramenta foi implementada uma função denominada *AutoWD* que toma decisões mediante ao estado do sensor virtual, também implementado na estação terrestre. No *software Nadzoru* foi implementado a ferramenta de importação de missões do *Mission Planner* e o módulo de geração de sub-missões a partir da adição de novas transições.

Através destas modificações foram realizados testes simulados e reais, no qual os testes simulados apresentaram problemas gráficos com relação ao registro da trajetória do VANT. Este problema gráfico foi solucionado através da reprodução dos pontos salvos da trajetória na API do *Google Maps*. O teste real não apresentou problemas com relação aos registro da trajetória, porém ambos os testes obtiveram sucesso ao completar a missão. No total foram realizados em torno de dez testes simulados e dois testes reais.

## 7 CONCLUSÃO

Neste trabalho foi realizado um estudo sobre Veículos Aéreos Não Tripulados (VANTs), a técnica de simulação *Hardware in the Loop* e em paralelo um estudo sobre a teoria de controle supervisorio de Sistemas a Eventos Discretos (SEDs). Assim, identificou-se a possível utilização desta teoria para a modelagem da missão principal, levando em consideração a utilização de novos sensores.

Os testes foram realizados através da técnica de simulação *HIL*, utilizando o controlador de voo APM 2.6, o simulador *Flight Gear* e como estação terrestre o software *Mission Planner*. Para a realização da implementação também foi necessária a utilização do software *Nadzoru*, no qual a missão principal foi representada por autômatos e onde foram realizadas as adições das novas transições mediante a leitura dos sensores. Além da utilização destes softwares, foram implementadas algumas funções destinadas para a aplicação, tanto no *Mission Planner*, como no *Nadzoru*.

Tratando do ambiente de simulação *HIL*, o mesmo apresentou problemas relacionados à versão dos três elementos (*GS*, *Firmware* e Simulador), relatando problemas durante o voo. Porém, foram encontradas três versões compatíveis que minimizavam estes problemas, viabilizando a utilização da técnica.

Com estes elementos funcionando, pôde-se realizar a validação deste trabalho no simulador e em um voo real. Durante este voo real foi verificado, um desempenho satisfatório da aplicação desenvolvida, uma vez que, o VANT executou a missão principal tomando decisões impostas pelo usuário somente nos pontos pré-selecionados.

O teste simulado apresentou alguns problemas com relação ao registro da missão, que registrou valores incompatíveis com a realidade, gerando erros, onde foi substituído por uma aplicação disponibilizada pelo *Google Maps*, mostrando o registro graficamente a partir dos pontos salvados durante a missão.

Já os testes reais se mostraram muito satisfatórios, uma vez que os problemas relatados na simulação não se manifestaram e a missão foi executada sem problemas, respeitando as tomadas de decisões decorrentes do estado do sensor.

Uma das aplicações que podem ser listadas para a utilização deste trabalho é o uso em plataformas de petróleo. Neste tipo de aplicação os VANTs realizam a

inspeção automática da plataforma mediante a uma missão, porém antes de passarem por um determinado trajeto que possa haver risco de explosão devido a gás, sensores posicionados neste trajeto fazem a leitura dos níveis de gás e decidem se realmente é possível à passagem pelo VANT naquele ponto.

Outra aplicação possível é a inspeção de aero geradores. Neste tipo de inspeção o VANT examina primeiro as partes críticas do mesmo, e através da análise das imagens geradas toma decisões para seguir a inspeção ou reportar a estação terrestre que existe um problema a ser solucionado. Caso isto ocorra o VANT retorna a missão é abortada para que os operadores realizem a manutenção necessária.

A realização deste trabalho proporcionou uma solução para os problemas de registro gráfico em testes simulados do *Mission Planner* utilizando uma API do *Google Maps*. Outra contribuição proporcionada também, foi a adição de novas transições entre os *WayPoints*, uma vez que o *Mission Planner* apenas proporcionava uma única transição de saída, gerando assim as sub-missões enviadas para o VANT em voo.

Levando em consideração os estudos realizados neste trabalho e os desenvolvimentos, sugere-se como continuidade:

- Implementação de sensores embarcados no VANT realizando a técnica *see-and-avoid*;
- Gerenciamento de múltiplas aeronaves no espaço aéreo, utilizando a teoria controle supervisorio de SEDs;
- Implementação de um sistema de *Geofence*, visando à segurança do espaço aéreo e de civis,;
- Utilização de um sistema de câmeras embarcadas para navegação assistida em locais sem sinal *GPS*.

## REFERENCIAS

- 3DROBOTICS. APM 2.6 Set. **3DRobotics**, 2015. Disponível em: <<https://store.3drobotics.com/products/apm-2-6-kit-1>>. Acesso em: 28 Abril 2015.
- APM. HIL Quad Simulator. **APM**, 2009. Disponível em: <<http://copter.ardupilot.com/wiki/common-simulation/hil-quad/>>. Acesso em: 10 Maio 2015.
- BOLAND, B. Drone Locates Missing 82 Year-Old Man in 20 Minutes After He'd Been Missing for Three Days. **CNS News**, 28 Julho 2014. Disponível em: <<http://www.cnsnews.com/mrctv-blog/barbara-boland/drone-locates-missing-82-year-old-man-20-minutes-after-hed-been-missing>>. Acesso em: 27 Abril 2015.
- BOUABDALLAH, S.; MURRIERI, P.; SIEGWART, R. In Proceedings of International Conference on Robotics and. **Design and Control of an Indoor Micro Quadrotor**, 2004.
- CASSANDRAS, C.; LAFORTUNE, S. **Introduction to Discrete Event Systems**. [S.l.]: Springer Science & Business Media, 2008.
- CURY, J. E. R. **Teoria de Controle Supervisório de Sistemas a Eventos Discretos**. Canela: [s.n.], 2001.
- DARLING, D. Kettering Bug. **The Worlds of David Darling**, 2011. Disponível em: <[http://www.daviddarling.info/encyclopedia/K/Kettering\\_Bug.html](http://www.daviddarling.info/encyclopedia/K/Kettering_Bug.html)>. Acesso em: 20 Abril 2015.
- DEGARMO, M. T. **Issues Concerning Integration of Unmanned**. McLean: MITRE, 2004.
- EISENBEISS, H. **UAV Photogrammetry**. Zurich: [s.n.], 2009.
- EISENBEISS, H.; SAUERBIE, M. Investigation of uav systems and flight modes for photogrammetric applications. **Remote Sensing and Photogrammetry Society**, 19 Dezembro 2011. 400-421.
- GOMES, V. H. **Uma implementação para planejamento e execução de voo autônomo de VANTs**. UDESC. Joinville. 2014.
- GUIMARÃES, J. P. F. **Controle de atitude e altitude para um veículo aéreo não tripulado**. [S.l.]. 2012.
- HORGAN, J. Unmanned Flight. **National Geographic**, 2013. Disponível em: <<http://ngm.nationalgeographic.com/2013/03/unmanned-flight/horgan-text>>. Acesso em: 25 Abril 2015.
- HOVERSTEN, P. Were drones used in the Bikini bomb tests? **Air&Space**, 2013. Disponível em: <<http://www.airspacemag.com/need-to-know/were-drones-used-in-the-bikini-bomb-tests-6578388/?no-ist>>. Acesso em: 25 Abril 2015.

HOWETH, C. L. S. Radio Controlled Aircraft. In: HOWETH, C. L. S. **History of Communications-Electronics in the United States Navy**. [S.l.]: [s.n.], 1963. Cap. XL, p. 479-493.

ISERMANN, R.; SCHAFFNIT, J.; SINSEL, S. Hardware-in-the-loop simulation for the design and testing of engine-control systems. **Control Engineering Practice**, p. 643–653, 1999.

JANG, J. S.; CROSSBOW TECHNOL., I. . S. J. C.; LICCARDO, D. Automation of Small UAVs using a Low Cost Mems Sensor and Embedded Computing Platform. **IEEE**, p. 1 - 9, 2006.

JIONG, Y.; LEI, Z.; JIANGPING, D. GPS/SINS/BARO Integrated Navigation System for UAV. **IEEE**, Kunming, p. 19 - 25, Julho 2010.

JONES, P. N. Periscopes For Airplanes. **Popular Mechanics Magazine**, p. 142-144, Junho 1956.

LIM, H.; PARK, J.; LEE, D.; KIM, H. J. Build Your Own Quadrotor. **IEEE**, p. 33 - 45, Setembro 2012.

LOPES, Y. K.; PINHEIRO, L. P.; LEAL, A. B.; ROSSO, R. U. J. Nadzoru: A Software Tool for Supervisory Control of Discrete Event Systems. **IFAC International Workshop on Dependable Control and Discrete Systems**, 2015.

LUA. About LUA. <http://www.lua.org/about.html>, 2005. Disponível em: <<http://www.lua.org/about.html>>. Acesso em: 15 Novembro 2015.

MARINA, H. G. D.; PEREDA, F. J.; GIRON-SIERRA, J. M.; ESPINOSA, F. UAV Attitude Estimation Using Unscented Kalman Filter and TRIAD. **IEEE**, p. 4465 - 4474, Novembro 2012.

MILLER, D. S. **Open Loop System Identificaiton of a Micro Quadrotor Helicopter from Closed Loop Data**. University of Maryland. Washington, p. 157. 2011.

MUNARETTO, L. **Vant e Drones**. 1ª. ed. São José dos Campos: [s.n.], 2015.

NONAMI, K.; KENDOUL, F.; SUZUKI, S.; WANG, W.; NAKAZAWA, D. **Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles**. [S.l.]: Springer, 2010.

OBOURNE, M. Mission Planner Ground Control Station, 2014. Disponível em: <<https://github.com/diydrones/MissionPlanner>>. Acesso em: 10 Junho 2015.

PAULA, J. C. D. **DESENVOLVIMENTO DE UM VANT DO TIPO QUADRIRROTOR PARA OBTENÇÃO DE IMAGENS AÉREAS EM ALTA DEFINIÇÃO**. Curitiba. 2012.

PUTRICH, G. Next generation of Global Hawks ready to roll. **Flight Global**, 2010. Disponível em: <<http://www.flightglobal.com/news/articles/next-generation-of-global-hawks-ready-to-roll-346116/>>. Acesso em: 25 Abril 2015.

PX4. **Pixhawk**, 2014. Disponível em: <<https://pixhawk.org/choice>>. Acesso em: 5 Maio 2015.

QUEIROZ, M. H. D. **Controle Supervisório Modular e Multitarefa de Sistemas Compostos**. Florianópolis, p. 150. 2004.

SCARAMUZZA, D. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. **IEEE Robotics & Automation Magazine**, 2013. 1-10.

STEVENS, B. L.; LEWIS, F. L. **Aircraft Control and Simulation**. 2<sup>a</sup>. ed. [S.l.]: John Wiley & Sons, 2003.

STRONG, A. B.; ISRAEL, G. K. Remote Piloted Aerial Vehicles : An Anthology, 2002. Disponível em: <[http://www.ctie.monash.edu/hargrave/rpav\\_home.html](http://www.ctie.monash.edu/hargrave/rpav_home.html)>. Acesso em: 20 Abril 2015.

WOLF, W. **German Guided Missiles: Henschel Hs 293 and Ruhrstahl SD 1400X**. [S.l.]: [s.n.], 1997.