

UNIVERSIDADE DE SÃO PAULO

ÉRICO VERÍSSIMO HORTOLAN
JOÃO AUGUSTO DOS SANTOS ANDRADE
RENATO AGNELO GARIANI

Image SyncCopter

Trabalho de Formatura apresentado
à Escola Politécnica da Universidade
de São Paulo

Engenharia Elétrica - Departamento
de Sistemas Eletrônicos

Orientador: Prof. Dr. Celso
Kurashima

São Paulo
2013

UNIVERSIDADE DE SÃO PAULO

ÉRICO VERÍSSIMO HORTOLAN

JOÃO AUGUSTO DOS SANTOS ANDRADE

RENATO AGNELO GARIANI

Image SyncCopter

Trabalho de Formatura apresentado
à Escola Politécnica da Universidade
de São Paulo

São Paulo

2013

AGRADECIMENTOS

Ao Prof. Dr. Celso Kurashima, nosso orientador, pela grande contribuição e constante apoio e estímulo dados ao longo do desenvolvimento desse projeto.

Ao Prof. Dr. Armando Laganá, que nos acompanha desde o segundo ano da graduação.

Às nossas famílias por todo suporte no período da faculdade e, principalmente, durante o período empenhado neste trabalho de conclusão de curso.

Aos nossos amigos de turma, companheiros nessa jornada iniciada há cinco anos, por todos os risos, tristezas e alegrias compartilhados entre nós e a todos os demais que colaboraram durante o desenvolvimento deste trabalho.

RESUMO

A reconstrução de imagens e cenários tridimensionais (3D), a partir de imagens em duas dimensões (2D), se mostra atraente mediante as suas inúmeras possíveis aplicações. A nível de aplicação comercial, a geração de imagens em três dimensões pode ser utilizada, por exemplo, no monitoramento de uma construção em andamento ou de fronteiras territoriais, sendo que, no Brasil, este segmento de mercado é pouco explorado - isto pode ser inferido perante à existência de ações governamentais que visam a compra de tecnologias que permitam o reconhecimento de áreas previamente determinadas. Em adição a isto, nota-se que o uso de agentes voadores para esses fins também se apresenta como um mercado com baixo nível de exploração, pois Veículos Aéreos Não-Tripulados (VANTs) possuem alto custo, visto que este tipo de tecnologia eletrônica ainda é pouco desenvolvido em nosso país. Diante disso, o desenvolvimento de um sistema que permita a obtenção dessas imagens, por meio de agentes aéreos em voo sincronizado e de baixo custo, e o processamento destas é o objetivo deste trabalho de conclusão de curso.

Palavras Chave: Reconstrução Tridimensional. Veículo Aéreo Não-Tripulado (VANT). Quadricóptero. Processamento de Imagens. Controle Eletrônico.

ABSTRACT

The Three-dimensional reconstruction of images and scenarios, from two-dimensional images, revealed itself attractive ahead of the countless possible applications. As commercial application, three-dimensional generation could be used, for example, in construction in progress or territorial boundaries monitoring, inasmuch as in Brazil, this market segment is yet under explored – it can be inferred from the large number of governmental incentives for purchase of area recognition technology. In addition, it should be noted that the use of aerial elements for these purposes also is an unexplored market, since Unmanned Aerial Vehicles (UAVs) have high costs, as this kind of electronic technology is still little developed in our country. Therefore, developing a system to obtain two-dimensional images, from an aerial element, and do the image processing are the objective of this Course Conclusion Work.

Keywords: Three-dimensional Reconstruction. Unmanned Aerial Vehicle (UAV). Quadcopter. Image Processing. Electronic Control.

LISTA DE TABELAS

Tabela 1 – Requisitos de marketing e engenharia	14
Tabela 2 – Matriz de gestão de riscos	20
Tabela 3 – Características das soluções de câmeras	23
Tabela 4 – Consolidação AHP das soluções de câmeras	23
Tabela 5 – Características das soluções de formato	26
Tabela 6 – Consolidação AHP da solução de formato	26
Tabela 7 – Características das soluções de SIC	27
Tabela 8 – Consolidação AHP para solução de SIC	27
Tabela 9 – Características das soluções de motores	30
Tabela 10 – Consolidação AHP para solução de motores	30
Tabela 11 – Lista de materiais da Prova de Conceito	49
Tabela 12 – Lista de materiais – Protótipo Final	50
Tabela 13 – Comparativo de Custos	51
Tabela 14 – Cálculos do método AHP para câmeras	69
Tabela 15 – Cálculos do método AHP para solução de formato	71
Tabela 16 – Cálculo do método SHP para solução de SIC	72
Tabela 17 – Cálculo do método AHP para solução de motor	73

LISTA DE FIGURAS

Figura 1 – Logotipo do projeto.....	10
Figura 2 – Helicóptero de filmagem na indústria cinematográfica.....	11
Figura 3 – Objetivo do projeto	12
Figura 4 – Reconstrução de um modelo 3D	14
Figura 5 – Planejamento estratégico do projeto	16
Figura 6 – Diagrama do sistema de funcionamento proposto	17
Figura 7 – Estrutura analítica do projeto – Primeira Fase	18
Figura 8 – Estrutura analítica do projeto – Segunda Fase	18
Figura 9 – Utilização do <i>FSWEBCAM</i>	22
Figura 10 – Tela inicial do programa de reconstrução 3D.....	24
Figura 11 – Placa de desenvolvimento Arduino	27
Figura 12 – Raspberry Pi modelo B	27
Figura 13 – Estrutura interna de um motor brushless	29
Figura 14 – Exemplo de ESC comercial.....	29
Figura 15 – Circuito integrado MPU6050	31
Figura 16 – Interface do MPU6050	31
Figura 17 – Alternativas de circuito de driver	33
Figura 18 – Sinal esperado na saída.....	33
Figura 19 – Estrutura eletrônica do VANT	34
Figura 20 – Montagem para levantamento do modelo	35
Figura 21 – Lugar geométrico das raízes de malha aberta	37
Figura 22 – Diagrama de controle do sistema.....	38
Figura 23 – Diagrama do lugar das raízes do sistema com controle.....	39
Figura 24 – Diagrama de controle da rotação no eixo Z	40
Figura 25 – Comunicação TCP/IP	43
Figura 26 – Interface homem-máquina (IHM).....	44
Figura 27 – Estrutura proposta do VANT	46
Figura 28 – VANT construído para prova de conceito.....	47
Figura 29 – VANT em voo teste	48
Figura 30 – Algoritmo de obtenção das imagens via câmera USB	53
Figura 31 – Placa de circuito impresso do driver.....	53

Figura 32 – Exemplo de um controle PDD	54
Figura 33 – Sensor de nível de tensão da bateria	55
Figura 34 – Protótipo final	56
Figura 35 – Conjunto de imagens de verificação	58
Figura 36 – Nuvem de pontos e posições das câmeras	59
Figura 37 – Reconstrução 3D	59
Figura 38 – Imagem do site	60

SUMÁRIO

RESUMO.....	3
ABSTRACT	4
LISTA DE TABELAS.....	5
LISTA DE FIGURAS.....	6
SUMÁRIO.....	8
1. Introdução.....	10
1.1. Título, Sigla e Logotipo do Projeto.....	10
1.2. Motivação	10
1.3. Objetivos.....	11
2. Metodologia	13
2.1. Referencial Técnico.....	13
2.2. Análise de Requisitos	14
2.3. Descrição da Proposta	15
2.4. Gerenciamento do projeto	17
2.4.1. Plano de Gerenciamento.....	17
2.4.2. Cronograma.....	19
2.4.3. Risco.....	19
3. Desenvolvimento	21
3.1. Captura e processamento de imagens	21
3.1.1. Captura das Imagens	21
3.1.2. Reconstrução 3D	23
3.2. Agentes voadores.....	25
3.2.1. Definição dos componentes estruturais e eletrônicos	26
3.2.1.1. Sistema Integrado de Controle	26
3.2.1.2. Bateria	28
3.2.1.3. Motores.....	28
3.2.1.4. Sensor	30
3.2.1.5. BEC e ESC's	32
3.2.2. Modelagem física.....	34
3.2.3. Controle	37
3.3. Estação Base	41

3.3.1. Comunicação.....	41
3.3.2. Interface Homem-Máquina (IHM)	43
3.4. Prova de conceito	44
3.4.1. Primeira prova de conceito: Reconstrução 3D	44
3.4.2. Segunda prova de conceito: Voo controlado do agente voador	46
4. Resultados.....	49
4.1. Orçamento.....	49
4.2. Testes.....	51
4.2.1. Plano de Integração e testes	52
4.3. Protótipo final.....	55
4.3.1. Estrutura.....	56
4.3.2. Controle de Voo.....	57
4.4. Reconstrução 3D.....	58
5. Discussão	60
5.1. Site do projeto	60
5.2. Próximas Etapas.....	60
5.2.1. Captação de investimentos	61
5.2.2. Pesquisa e Desenvolvimento	61
5.3. Conclusão.....	62
Referências	64
APÊNDICE A – Cronograma	68
APÊNDICE B – Detalhamento do cálculo para solução de câmeras.....	69
APÊNDICE C – Detalhamento do cálculo para solução de formato	71
APÊNDICE D – Detalhamento do cálculo para solução de sistema de controle SIC	72
APÊNDICE E – Detalhamento do cálculo para solução de motor	73
APÊNDICE F – Rotinas MATLAB para identificação de parâmetros da saída do sistema	74
APÊNDICE G – Rotinas MATLAB para identificação de parâmetros da função de transferência do sistema	75
ANEXO A – Resumo das especificações do MPU6050	77
ANEXO B – Diagrama de blocos internos do MPU6050	78
ANEXO C – Raspberry PI - Especificações Técnicas.....	79

1. Introdução

1.1. Título, Sigla e Logotipo do Projeto

Título: Image SyncCopter

Sigla: ISC

O logotipo do projeto encontra-se na figura 1.

Figura 1 – Logotipo do projeto



Fonte: Própria

1.2. Motivação

A utilização de drones (ou VANTs – Veículos Aéreos Não-Tripulados) tem sido amplamente difundida atualmente, principalmente devido à sua versatilidade para aplicação em diversas situações. Sua utilização pode ser dividida em três naturezas distintas: entretenimento, aplicações comerciais e aplicações militares. No que diz respeito à aplicação voltada para entretenimento, os VANTs são utilizados principalmente como hobby, como modelos de aeromodelismo. Quanto ao uso comercial, um drone pode ser utilizado, por exemplo, para cobertura de eventos, como no caso recente das manifestações na cidade de São Paulo (1). Já no que se refere à aplicações militares, drones são utilizados tanto para monitoramento de fronteira (2), quanto para operações estratégicas do exército (3).

A utilização de modelos e cenários 3D possibilita a reprodução de uma riqueza de detalhes que não pode ser observada diretamente com fotos tradicionais. No âmbito do entretenimento, a utilização de técnicas para produzir imagens em 3 dimensões também tem grande impacto para os espectadores.

Para a produção de imagens em ambientes externos a partir de variados ângulos torna-se interessante a utilização de veículos aéreos não tripulados (VANTs). Tais veículos podem ser direcionados e permitem a realização de movimentos e planos de voo não obtidos facilmente com outras formas de filmagens,

como a utilização de helicópteros, amplamente difundida na indústria cinematográfica. Um destes helicópteros pode ser observado na figura 2.

Figura 2 – Helicóptero de filmagem na indústria cinematográfica



Fonte: <http://www.helicopterfilm.tv/helicopters.php>

Diante disso, as motivações deste projeto residem no alto volume de aplicações de um drone, na possibilidade de combinação de imagens para reconstrução de um modelo tridimensional, na complexidade da solução em engenharia e na oportunidade de se utilizar várias áreas do conhecimento do curso de Engenharia Elétrica.

1.3. Objetivos

O objetivo deste projeto é o desenvolvimento um veículo aéreo não tripulado capaz de obter imagens de um edifício alvo e enviá-las a uma estação base. As imagens serão transmitidas para um computador onde o conjunto de imagens será processado e um modelo 3D será reconstruído.



Fonte: Própria

2. Metodologia

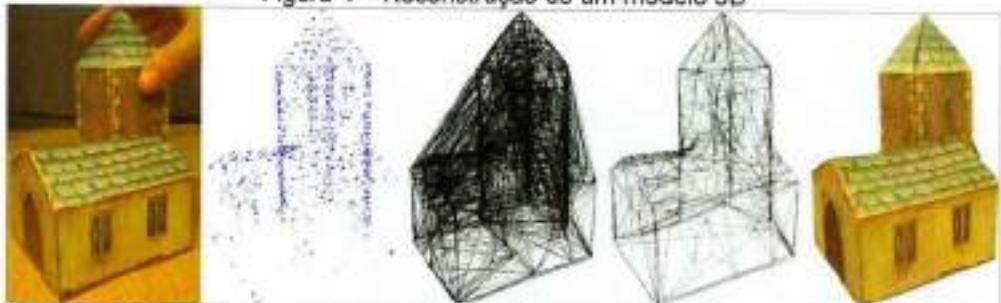
2.1. Referencial Técnico

A utilização de agentes voadores para fins diversos têm crescido nos últimos anos. Analogamente, o estudo de diferentes modos de processamento de imagens (em especial, o modelamento de estruturas em 3D a partir de figuras 2D) está em amplo desenvolvimento, seja em nível acadêmico, seja em nível comercial. Neste projeto, as duas frentes - a concepção dos agentes voadores e o modelamento de imagens - possuem alto grau de aplicabilidade e, portanto, estão em um processo constante de aprimoramento.

Como exemplo da utilização de agentes voadores podemos demonstrar o trabalho desenvolvido por DAS et al (4), que apresenta a construção de um sistema de monitoramento de temperatura de superfície através do uso de um quadricóptero equipado com uma câmera e o trabalho desenvolvido por Longhitano onde é estudada a utilização de VANTs para sensoriamento de impactos ambientais causados por acidentes rodoviários (5). Pounds, Mahony e Corke, ainda, demonstram em seu trabalho o desenvolvimento de um agente voador capaz de carregar uma carga útil de 1 kg com propósitos comerciais, diferenciando-se da maioria dos trabalhos cujo peso total não passa de 3 kg (6), enquanto Bouabdallah, Murrieri e Siegwart desenvolveram um quadricóptero pequeno para ambientes internos (7).

Em relação ao modelamento de imagens, são constantes os trabalhos que apresentam como foco a geração de cenários 3D por meio de um conjunto de imagens 2D. Artigos e papers cujo tema de pesquisa é o processamento de imagens são recorrentes na comunidade científica, especialmente algoritmos de conversão 2D - 3D (8). Muitos ramos dependentes de tecnologia computacional voltada ao tratamento de imagens, como a medicina e a construção civil, são propulsores para as pesquisas cujo objetivo é a criação de métodos matemáticos e algoritmos que consigam não somente identificar padrões visuais de um objeto através de fontes diferentes, mas também compilar esses padrões de modo a gerar uma reprodução tridimensional.

Figura 4 – Reconstrução de um modelo 3D



Fonte: http://mi.eng.cam.ac.uk/~qp202/my_papers/BMVC09/

2.2. Análise de Requisitos

Dados os objetivos definidos anteriormente, podemos esperar que o projeto atingisse os seguintes requisitos de marketing:

- 1 – Observar um objeto a distância com imagem 3D.
- 2 – Capturar cenas via agente voador.
- 3 – Controlar o agente à distância.
- 4 – Possuir custo abaixo das opções encontradas no mercado

Com estes requisitos de marketing espera-se que o projeto tenha sucesso e apresente apelo comercial quando concluído.

O agente voador terá um controle de voo embarcado com a função de estabilizar o quadricóptero. Possuirá também a função de transmitir a imagem capturada através de uma câmera de baixa resolução para uma estação em terra. Nesta, será executado o algoritmo que irá tratar a imagens de baixa resolução e gerar o modelo 3D.

Tendo sido definidos os requisitos de Marketing, pode-se mapear os requisitos de engenharia do projeto conforme tabela 1, de acordo com as definições anteriores e justificativas demonstradas.

Tabela 1 – Requisitos de marketing e engenharia

Requisitos de Marketing	Requisitos de Engenharia	Justificativas
1,2,4	Gerar um cenário 3D através de câmera de baixa resolução	Via encontrada criar cenário 3D e para minimizar custos

Requisitos de Marketing	Requisitos de Engenharia	Justificativas
2,3	Agente que captura imagem e as transmite para uma estação base	Sistema de comunicação comum de fácil aplicação
1,2,3	Estação base gerenciadora do agente e modelagem do cenário 3D	Um sistema de controle central que trabalhe as informações e gerencie as atividades
3,4	O agente voador deve pesar menos que 1,0 kg	Diminuir custos energéticos e financeiros
4	Voo deve durar no mínimo 5 minutos	Diminuir Custos
4	Custo menor que R\$ 2000,00	Orçamento limitado

As definições dos requisitos de engenharia mostram que o projeto deverá ter um custo menor que R\$ 2.000,00 e, o agente voador, massa menor que um quilograma com autonomia de voo de 5 minutos.

2.3. Descrição da Proposta

Perante as condições iniciais do projeto - um produto pré-determinado, com funções bem definidas - optou-se pela utilização da metodologia *top-down*, dada a clara definição do ponto de partida do projeto (a geração de uma imagem 3D a partir de imagens 2D, fornecidas pelo agente voador). Sendo assim, as estratégias foram divididas conforme a natureza conceitual de cada segmento.

Por meio das ferramentas de Planejamento Estratégico, como a análise MOST e a Pirâmide Estratégica (do Inglês *Mission, Objectives, Strategies e Tactics*) foi possível a segmentação do projeto final em dois subprojetos, permitindo uma delimitação mais precisa de cada uma das etapas a ser realizada, além de definir de modo objetivo e conciso quais os meios para se atingir o resultado esperado.

A seguir, tem-se a ilustração dos subprojetos em uma Pirâmide Estratégica.

Figura 5 – Planejamento estratégico do projeto



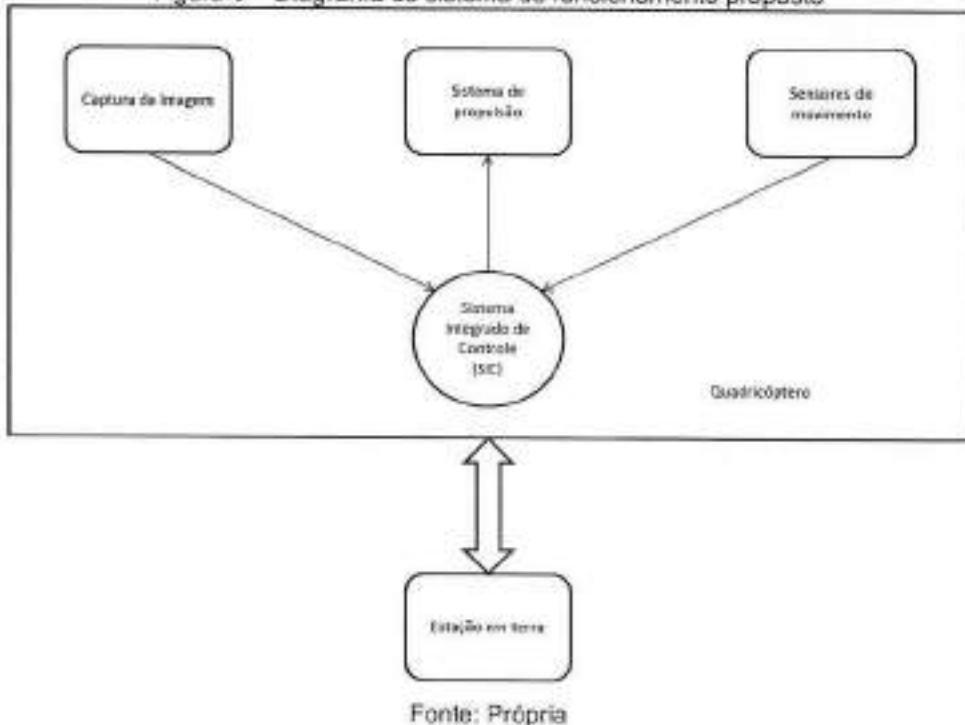
Fonte: Própria

Por conseguinte, o projeto pode ser dividido em duas partes suficientemente independentes que agregadas permitiriam o sucesso deste: a captura e processamento das imagens para a formação do modelo 3D e a construção do agente voador. Definidos os objetivos parciais, o passo seguinte foi a definição das estratégias para conseguir atingi-los. Logo, foi necessária a realização de uma análise de cenário para uma avaliação das possíveis soluções, resultando em uma abordagem prática e factível de ser executada para ambas as partes do projeto.

Partindo-se da divisão proposta, pode-se determinar que, para fins de processamento de dados, tem-se dois núcleos de principais, o primeiro alocado no quadricóptero, chamado de SIC (Sistema Integrado de Controle) e outro responsável pela reconstrução do modelo 3D a partir das imagens.

Assim, o diagrama abaixo representa os blocos funcionais necessários. No quadricóptero, o SIC será responsável por concentrar todas as informações do agente voador recebendo comandos, capturando a imagem e processando dados dos sensores. A estação em terra será responsável pela reconstrução do modelo 3D bem como fará a comunicação com o SIC.

Figura 6 – Diagrama do sistema de funcionamento proposto



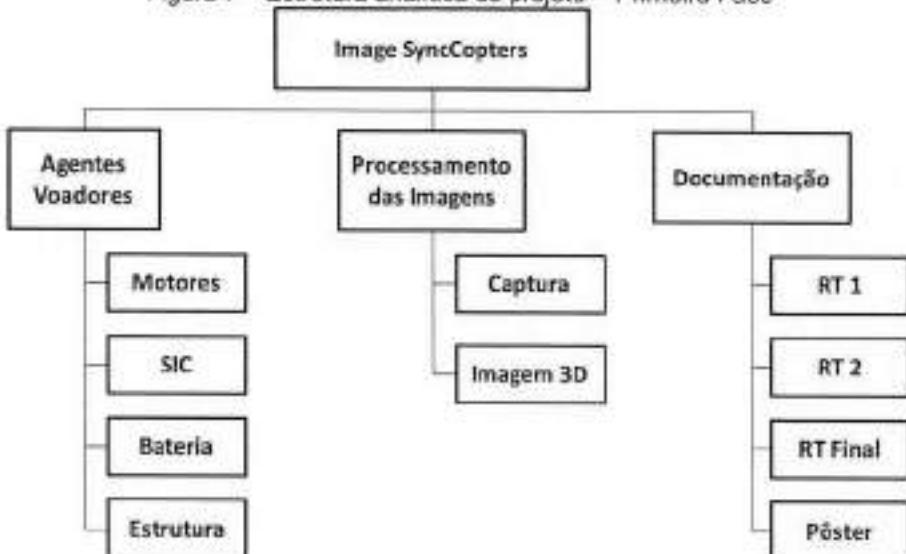
2.4. Gerenciamento do projeto

Para que os objetivos do projeto fossem alcançados, foi preciso a introdução de uma metodologia de gerenciamento de projetos para que fosse possível o acompanhamento de todas as atividades efetuadas neste trabalho. A metodologia acompanhada por suas ferramentas permitiu análises quantitativas e qualitativas das diversas etapas do projeto.

2.4.1. Plano de Gerenciamento

O projeto teve seu plano desenvolvido de acordo com a metodologia do PMI®, segundo as práticas definidas no PMBoK® (8). O projeto teve sua EAP (Estrutura analítica do projeto - WBS *Work Breakdown Structure*) elaborada de modo que foram definidos 3 subgrupos de trabalho compostos pelos entregáveis mostrados na figura abaixo, conforme análise estratégica realizada e considerando a produção de documentação necessária.

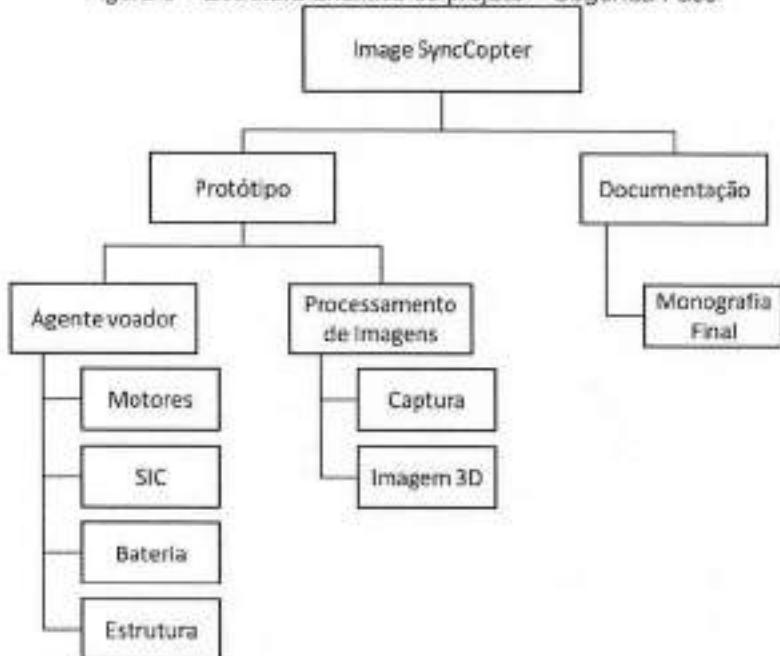
Figura 7 – Estrutura analítica do projeto – Primeira Fase



Fonte: Própria

Para a segunda etapa deste trabalho, foi preciso definir uma nova EAP, visto que as propostas iniciais foram cumpridas ao longo do primeiro período de desenvolvimento do projeto. A nova EAP contempla apenas a segunda fase do projeto, que engloba o protótipo e a entrega da monografia final.

Figura 8 – Estrutura analítica do projeto – Segunda Fase



Fonte: Própria

2.4.2. Cronograma

O cronograma do projeto, contemplando as atividades desenvolvidas ao longo do desenvolvimento do trabalho, pode ser observado no Apêndice A. Para desenvolvimento do cronograma foi utilizada a ferramenta Gantter® (9). Tal ferramenta baseada em navegador web além de não possuir custo para o projeto permitiu que o documento fosse mantido na nuvem, além do desenvolvimento colaborativo do cronograma, mostrando-se uma ferramenta eficiente frente às necessidades deste trabalho.

2.4.3. Risco

Outro ponto proposto pelo PMBoK® é o gerenciamento de riscos. Dentre as atividades previstas, a presença de riscos deve ser planejada com a finalidade de minimização dos impactos, caso estes venham a ocorrer.

A utilização das 9 áreas do conhecimento do PMBoK® para gerenciamento do projeto permite uma análise objetiva das áreas sujeitas a riscos durante a execução do projeto. Sendo as áreas identificadas: Recursos Humanos, Aquisições e Tempo.

Na área de Recursos Humanos a presença do risco está associada aos integrantes da equipe, uma vez que motivos pessoais ou de saúde podem vir a impedir o desenvolvimento do projeto. A perda de parte da equipe tem impacto moderado no projeto uma vez que não é possível a substituição do mesmo. Já a probabilidade de impacto deste risco pode ser considerada baixa.

Na área de Aquisições existe o risco associado ao grau de complexidade do projeto. Os componentes normalmente utilizados para este tipo de projeto não estão facilmente disponíveis no mercado nacional. Dessa forma o planejamento do projeto deve levar em conta a dificuldade de importação de equipamentos para o Brasil caso seja necessária tal abordagem. Tais fatores levam a classificar o impacto deste risco como moderado e sua probabilidade de ocorrência como baixa, uma vez que a definição da *Bill of Materials* permite uma visão clara sobre todas as aquisições necessárias para a execução do projeto.

Da mesma forma que a área de aquisições, a área de Tempo é sensível a importação de equipamentos dadas as dificuldades do processo. O recurso tempo

também é limitado dada a duração do semestre e horários disponíveis, dado que a dedicação ao projeto não será de tempo integral. Tal risco pode ser então classificado como tendo alto impacto para o projeto e com probabilidade de ocorrência moderada.

Distribuindo-se os riscos em uma matriz que relaciona a probabilidade de ocorrência com o impacto do risco no projeto temos a tabela abaixo. A área mais sensível do projeto pode então ser entendida como o recurso **Tempo**, seguida pelos **Recursos Humanos** e por **Aquisições**.

Tabela 2 – Matriz de gestão de riscos

Grau de Risco		Impacto		
		Alto	Moderado	Baixo
Probabilidade	Alto	Red	Yellow	Yellow
	Moderado	Tempo	Yellow	Green
	Baixo	Yellow	Recursos Humanos	Aquisições

Os riscos do projeto serão tratados com planos de contingência com a finalidade de minimizar a probabilidade de falha do projeto.

O risco de Recursos Humanos não possui plano adequado de contingência adequado uma vez que a substituição de integrantes da equipe não é possível. Quanto às aquisições, o plano é a aquisição de componentes em mercado nacional caso seja necessário, de modo que o projeto não sofra impacto com a importação de componentes. O risco de Tempo do projeto tem como plano de contingência a dedicação durante maior tempo dos integrantes da equipe para o projeto.

3. Desenvolvimento

O desenvolvimento do projeto versa sobre os estudos aprofundados de cada subprojeto – Reconstrução tridimensional e construção do agente voador – assim como a determinação das soluções que foram adotadas no protótipo final.

Ao fim do desenvolvimento se encontra o detalhamento das duas provas de conceito, efetuadas ao final do primeiro semestre de trabalho, com o intuito de ratificar as soluções escolhidas inicialmente para o atingimento das metas finais.

O detalhamento de cada componente, de suas características e comportamento, juntamente com o desenvolvimento das provas de conceito, podem ser vistos nas seções a seguir.

3.1. Captura e processamento de imagens

3.1.1. Captura das Imagens

A captura de imagens será realizada por meio de câmera de pequeno porte para obtenção de uma imagem com boa definição visual. Uma vez capturadas as imagens, elas serão transmitidas via Wi-Fi para um computador, no qual estará alocado o software que contém o algoritmo de combinação de imagens, responsável por gerar uma imagem em três dimensões.

Com o intuito de capturar várias imagens durante uma incursão aérea, foi desenvolvido um software na linguagem de programação Python responsável por configurar os parâmetros da câmera (frame, tempo de abertura da objetiva, quantas fotos serão tiradas) e, então, obter as fotos do objeto a ser reconstruído em três dimensões.

Cada imagem, por sua vez, deve possuir uma identificação, visto que a ordem das imagens interfere no processo de reconstrução, pois a é maior a correlação entre duas fotos que possuem uma defasagem menor que 15° entre os ângulos de captura. Para solucionar esse problema, as imagens tiradas possuem uma impressão digital de data conhecido como *timestamp*, permitindo a ordenação de modo simplificado.

Para executar o procedimento acima descrito, utilizou-se o aplicativo *fswebcam*, devido à simplicidade e à eficácia de seu funcionamento. Suas funções são realizadas no ambiente Linux, desde a calibração da câmera até salvar a imagem no cartão de memória, sendo que a própria inicialização do aplicativo pode ser feita via terminal ou através de um script Python. O fato das inicializações poderem ser feitas diretamente no terminal permite a integração desse programa de modo fácil aos demais softwares do projeto. Para ilustrar o uso desse aplicativo, tem-se um exemplo de utilização do aplicativo *fswebcam* a seguir.

Figura 9 – Utilização do *FSWEBCAM*



Fonte: <http://www.lynda.com/Linux-tutorials/Creating-interface-circuits/132717/143094-4.html>

As imagens, consequentemente, serão salvas diretamente do cartão de memória SD contido no agente voador. O processo de envio das mesmas para a estação base pode ser feito tanto via comunicação Wi-Fi quanto por meio físico, com a inserção do cartão diretamente no computador responsável pelo processamento de imagens.

Podemos então entender como componente principal no bloco de captura de imagem a câmera. Sendo assim, foi aplicada a metodologia de decisão AHP para definição da melhor solução dentre as alternativas levantadas.

Este componente teve como características críticas para o projeto definidas pelo grupo os indicadores Custo, Peso, Resolução e Conectividade USB por padrão.

A fim de aplicar-se a metodologia, foram consideradas as soluções e suas características apresentadas na tabela abaixo.

Tabela 3 – Características das soluções de câmeras

Opções	Custo (R\$)	Peso (mg)	Resolução (MP)	USB
Câmera Iphone 3G	6,69	20	2,0	Não
Mini Webcam convencional	11,76	50	1,3	Sim
Mini Webcam Y3000	80,64	25	8,0	Sim
Webcam de notebook	36,00	90	1,3	Sim

Os cálculos intermediários podem ser observados no Apêndice B. Para a câmera foi obtida a tabela final apresentando as pontuações individuais abaixo.

Tabela 4 – Consolidação AHP das soluções de câmeras

Opções	Custo	Peso	Resolução	USB	Pontuação
Câmera Iphone 3G	0,54	0,41	0,16	0,04	0,19
Mini Webcam convencional	0,31	0,17	0,10	0,32	0,29
Mini Webcam Y3000	0,05	0,33	0,63	0,32	0,28
Webcam de notebook	0,10	0,09	0,10	0,32	0,24

Assim, podemos observar que o método AHP indicou como solução mais adequada para o projeto a Mini Webcam convencional.

3.1.2. Reconstrução 3D

Sabendo a posição e a direção de cada câmera é possível a renderização da imagem 3D. Através da identificação de pontos de referência, como no reconhecimento de impressões digitais, é realizado o cálculo da distância de cada pixel em relação aos mesmos pixels das demais imagens adquiridas e assim é possível o posicionamento 3D do pixel para a formação do cenário. Através desta técnica é possível determinar o posicionamento relativo das câmeras para cada imagem, de modo que simplifica-se o processos de aquisição das mesmas, uma vez

que não é necessário o conhecimento prévio do posicionamento das mesmas e as fotos podem ser tiradas de posições que sejam levemente divergentes das demais sem prejuízo para a reconstrução 3D. No entanto, o ponto negativo dessa abordagem é a necessidade de processamento intensivo para a formação 3D.

O processamento de imagem é realizado por um software baseado no programa *Insight 3D* (11), de código aberto licenciado sobre os termos do GNU AGPL 3 (12). Optou-se pela plataforma do *Insight 3D* uma vez que, por ser um código aberto, não acrescenta nenhum custo ao orçamento do projeto. Tecnicamente, o *Insight 3D*, por definição, recebe uma série de fotos de um cenário real e, automaticamente, calcula as posições espaciais de cada pixel. Como resultado deste cálculo, tem-se uma nuvem de pontos que, após o processamento, permite que seja construído um polígono.

Figura 10 – Tela inicial do programa de reconstrução 3D



Fonte: Própria

Para se obter um software funcional, contudo, foram necessárias adaptações no código fonte, principalmente em relação às funções de tratamento geométrico – tais como as funções relacionam os pontos espaciais de cada imagem por meio de uma matriz de correlação usados na renderização do polígono texturizado. Tais alterações foram necessárias uma vez que o software foi inicialmente desenvolvido

para utilizar a biblioteca OpenCV v1.0, no entanto a versão utilizada neste projeto foi a v2.4.5 (13).

Realizadas as modificações que permitiram utilizar o software de reconstrução tridimensional com total funcionalidade, verificou-se que o tempo de processamento das imagens. Para realizar uma reconstrução completa, conforme especificação pelo criador do Insight3D, o tempo total seria torno de dez minutos. Com o software otimizado para este projeto, desde a inserção da série de fotos até a geração do polígono texturizado, a duração desse processo é aproximadamente cinco minutos para se ter uma reconstrução tridimensional satisfatória. Contudo, este tempo de execução pode variar conforme a complexidade da estrutura que se deseja obter.

3.2. Agentes voadores

O agente voador é uma parte fundamental para o sucesso do projeto, uma vez que as imagens serão obtidas pela câmera acoplada a ele. Inicialmente, a construção do agente voador implicará no desenvolvimento de uma plataforma composta pelo conjunto circuito eletrônico, sensores e o conjunto de motores e hélices. Esta abordagem de subdivisão se apresenta bastante viável, visto que uma montagem simplificada de cada subsistema do agente voador implica em tempo de processo de produção e testes menor. No entanto, deve-se planejar maior tempo na integração e testes de funcionamento conjunto dos subsistemas.

Como condição inicial para a conclusão da concepção de cada agente, foi necessária a definição de que tipo de plataforma voadora iria ser utilizada: conforme o número de motores de sustentação, toda configuração de equilíbrio se modifica, uma vez que o empuxo resultante é diretamente proporcional ao número e disposição de motores. O método AHP decidiu pela utilização de quadricópteros como agente voador, pois a estabilidade é um ponto forte deste tipo de veículo aéreo, conforme pesos definidos no método. Além disto, existe ainda um vasto número de pesquisas e protótipos neste formato, conforme levantamento do referencial técnico.

Tabela 5 – Características das soluções de formato

Opções	Custo (R\$)	Peso (mg)	Estabilidade	Controlabilidade
1 hélice	168,00	160,00	Ruim	Ruim
2 hélices	218,40	250,00	Razoável	Razoável
3 hélices	268,80	340,00	Boa	Boa
4 hélices	319,20	430,00	Muito Boa	Muito Boa

Os cálculos intermediários podem ser observados no Apêndice C.

Tabela 6 – Consolidação AHP da solução de formato

Opções	Custo	Peso	Estabilidade	Controlabilidade	Pontuação
1 hélice	0,34	0,40	0,06	0,06	0,15
2 hélices	0,26	0,26	0,18	0,18	0,20
3 hélices	0,21	0,19	0,29	0,29	0,27
4 hélices	0,18	0,15	0,47	0,47	0,38

3.2.1. Definição dos componentes estruturais e eletrônicos

3.2.1.1. Sistema Integrado de Controle

Em relação ao agente voador, foi implementado um sistema de controle central, no qual são gerenciadas todas as funções de movimento do agente voador. Logo, a partir de um comando externo, o controle central processa o que foi solicitado e ativa os motores conforme a necessidade. Em conjunto com esse sistema, há uma realimentação referente ao movimento do quadricóptero, informando ao controle parâmetros como posição e velocidade (nos três eixos de deslocamento, X, Y e Z), para que a estabilidade de voo e de movimentação seja mantida ao longo de uma excursão – as correções são realizadas através de comandos dados a cada motor, especificamente. Esta realimentação é realizada por meio de um conjunto composto por acelerômetros e giroscópios.

Para o núcleo de computacional – SIC –, a análise se delimitou a escolha entre duas opções a serem utilizadas: Raspberry Pi® ou Arduino®, ambos

apresentando suas particularidades em relação à memória, capacidade de processamento e custo (14 e 15).

Figura 11 – Placa de desenvolvimento Arduino



Fonte: <http://loja.roledotecnologia.com.br/arduino-uno.html>

Figura 12 – Raspberry Pi modelo B



Fonte: <http://www.bit-tech.net/modding/2013/03/22/raspberry-pi-case-competition-update/1>

Diante da avaliação destes itens específicos (conforme pode ser visto na tabela 6), a solução encontrada foi a utilização do *Raspberry Pi*®.

Tabela 7 – Características das soluções de SIC

Opções	Custo	Capacidade de processamento	Integração do controle
Arduino	50,40	Normal	Boa
Raspberry Pi	117,60	Muito Boa	Boa

Tabela 8 – Consolidação AHP para solução de SIC

Opções	Custo	Capacidade de processamento	Integração do controle	Pontuação
Arduino	0,70	0,35	0,41	0,54
Raspberry Pi	0,30	2,83	2,45	1,47

Os cálculos intermediários podem ser observados no Apêndice D.

3.2.1.2. Bateria

A definição da bateria adequada para a alimentação do sistema de cada plataforma voadora deve levar em consideração tanto o tempo de duração quanto custo. Destes, o primeiro se mostra o mais relevante, visto que a autonomia de voo é um requisito de engenharia do projeto.

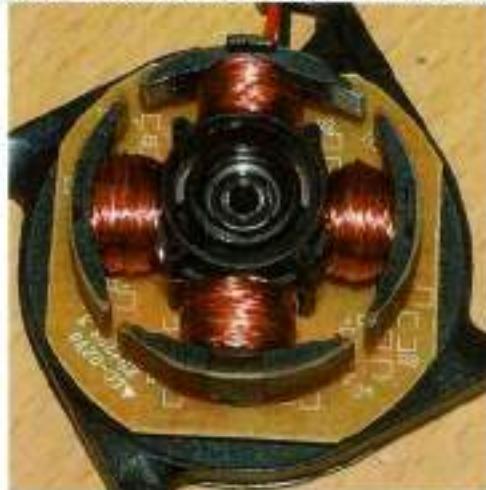
Em adição à verificação da duração de carga, analisou-se a relação peso e potência consumida; contudo, a definição do modelo específico da bateria somente foi realizado após a construção do protótipo da prova de conceito, devido à necessidade de se avaliar a relação potência consumida pelo peso. Entretanto, as baterias que apresentam melhor relação de capacidade de carga por peso são constituídas por íons ou polímeros de Lítio - denominadas baterias de Li-Ion e Li-Pol, respectivamente. Por conseguinte, optou-se por baterias de Li-Pol para a alimentação do circuito eletrônico dos agentes voadores uma vez que as baterias de Li-Ion exigem drivers de segurança para carga e descarga além de não serem capazes de fornecer alta corrente.

3.2.1.3. Motores

Foi necessário também definir o tipo de motor adequado ao quadricóptero. Em quadricópteros já projetados e implementados, notou-se que apenas dois tipos de motores são possíveis de serem usados: motores elétricos com escova e motores elétricos sem escova (motores à combustão não foram considerados, haja visto a ausência de trabalhos anteriores nos quais este tipo de motor estava presente para VANTs do tipo quadricóptero).

Diferentemente de Motores Brushed, os Motores Brushless não possuem escovas e seu princípio de funcionamento tem como base o acionamento sucessivo de bobinas, tal como reproduzidas na figura abaixo, de modo a criar um campo rotativo. Neste tipo de motor a carcaça externa possui imãs permanentes que, quando submetidos ao campo magnético criado pelo acionamento das espiras, produzem torque e movimento.

Figura 13 – Estrutura interna de um motor brushless.



Fonte: <http://www.howstuffworks.com/brushless-motor.htm>

Tais motores tem a vantagem de serem energeticamente mais eficientes que os com escova, mas necessitam de um sistema de controle que acione as bobinas na ordem correta. Este sistema de acionamento é conhecido como ESC (*Electronic Speed Controller*).

Figura 14 – Exemplo de ESC comercial



Fonte: http://www.hobbyking.com/hobbyking/store/uh_viewitem.asp?idproduct=6457

Para se definir corretamente o tipo de motor a ser implementado, foi realizada a avaliação de quais parâmetros são relevantes. Assim, foram avaliados o volume de projetos nos quais cada tipo de motor é usado, a controlabilidade, o custo, o peso

e um fator empuxo/R\$. A análise desses parâmetros está contida na tabela 8, na qual são ilustrados cada um dos parâmetros quantitativamente.

Tabela 9 – Características das soluções de motores

Opções	Custo (R\$)	Peso (mg)	Controlabilidade	Utilização	Empuxo/R\$
Brushless	30,00	80,00	Mais complicado	Mais utilizado	0,030
Brushed	26,00	88,00	Mais simples	Menos utilizado	0,015

Os cálculos intermediários podem ser observados no Apêndice E.

Tabela 10 – Consolidação AHP para solução de motores

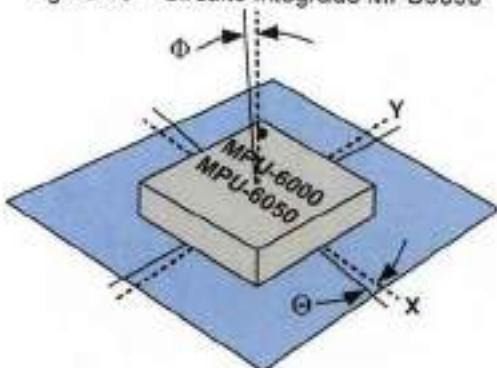
Opções	Custo	Peso	Controlabilidade	Utilização	Empuxo	Pontuação
Brushless	0,55	0,54	0,52	0,56	0,67	0,59
Brushed	0,45	0,46	0,48	0,44	0,33	0,53

3.2.1.4. Sensor

São ainda necessários sensores que permitam ao VANT a estimativa de sua posição, de modo que seja implementado um algoritmo de controle no SIC e a malha de controle seja fechada. Os sensores escolhidos foram acelerômetros e giroscópios. Acelerômetros são dispositivos eletromecânicos que permitem medir forças de aceleração em uma certa direção (16). Giroscópios são dispositivos baseados no princípio do momento angular, como uma roda de bicicleta que tende a se manter na vertical quando girando, e permitem obter alterações angulares nos eixos (17).

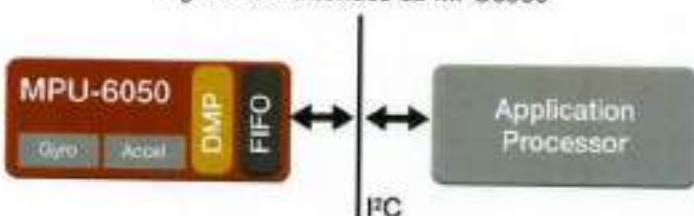
Para o sistema de controle serão usados 3 acelerômetros e 3 giroscópios formando um sistema de navegação inercial completo para os 3 eixos. Tal sistema de aquisição de movimento utilizará o circuito integrado MPU-6050 que possui os requisitos especificados e permite comunicação I2C para aquisição dos dados (18). Alguns requisitos adicionais sobre o MPU6050 podem ser observados no ANEXO A.

Figura 15 – Circuito integrado MPU6050



Fonte: http://produto.mercadolivre.com.br/MLB-452725642-modulo-giroscopio-acelerometro-mpu-6050-3-eixos-p-arduino-_JM

Figura 16 – Interface do MPU6050



Fonte: <http://invensense.com/mems/gyro/mpu6050.html>

O diagrama de blocos internos do MPU6050 pode ser observado no ANEXO B. O MPU6050 possui ainda um processador de ponto flutuante dedicado ao cálculo de dados de posicionamento, conhecido como DMP® - Digital Motion Processor®. Este processador permite a fusão de dados dos giroscópios e acelerômetros para a composição de uma representação em quatérnios da posição do quadricóptero. Para fins de utilização no controle, os quartérnios são convertidos em ângulos aeronáuticos conhecidos como *yaw*, *pitch* e *roll*. A utilização desta abordagem permite que seja transferido para o MPU6050 o processamento de toda a informação sobre posicionamento, somente sendo feita a leitura dos dados via interface I2C. O DMP® permite também a configuração de um filtro passa baixas no algoritmo de cálculo, tal filtro tem a vantagem de que seja obtida uma leitura mais robusta e imune a ruídos introduzidos pela vibração dos motores. Na figura abaixo pode-se observar os valores processados pelo DMP® juntamente com o cálculo do ângulo caso sejam utilizados diretamente os valores lidos do giroscópio, uma vez que o giroscópio representa a velocidade angular, a simples integração do valor lido representa a posição angular.



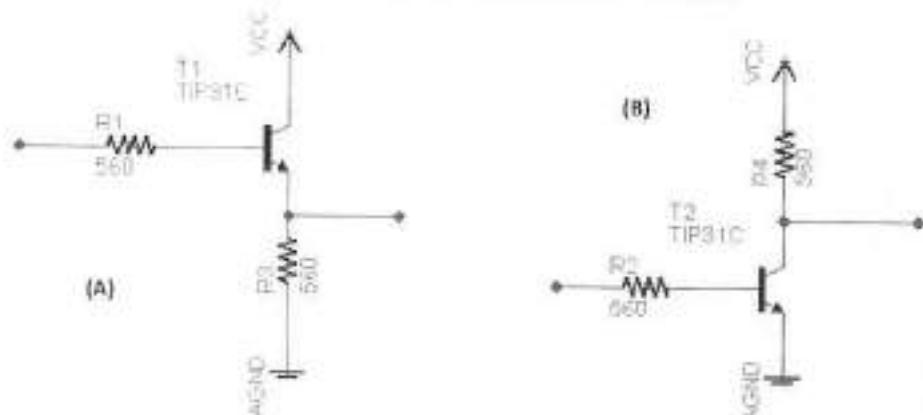
Do gráfico nota-se que devido aos ruídos de leitura do giroscópio, o valor da posição angular obtida pelo segundo método apresenta o efeito de "escorregamento". Tal efeito é compensado no DMP pela utilização dos acelerômetros para estimar a posição do vetor gravidade e utiliza este para correção do efeito de "escorregamento".

3.2.1.5. BEC e ESC's

São necessários ainda outros 2 componentes eletrônicos, sendo eles: circuito BEC (Battery Elimination Circuit) que permite a utilização de apenas uma bateria interligando todos os ESC's e o sistema de controle; e um circuito de Driver que permite o acionamento da entrada dos ESC's devido à baixa corrente fornecida pela saída do Raspberry Pi.

Foram estudadas 2 alternativas de circuito de Driver, como mostrado na figura abaixo. O primeiro circuito, identificado como (A), apresenta como característica de funcionamento que o sinal de saída apresenta o mesmo nível do sinal de entrada, ou seja, um sinal lógico Verdadeiro na entrada produz uma saída lógica Verdadeira. Já o circuito (B) possui comportamento de inversor lógico da entrada.

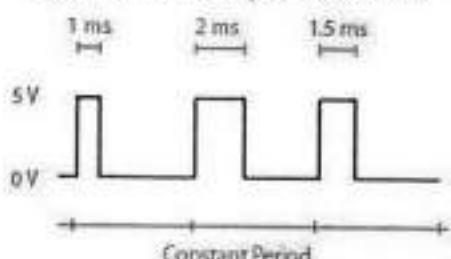
Figura 17 – Alternativas de circuito de driver



Fonte: Própria

Para decisão do circuito de Driver foi analisado o sinal produzido necessário para o acionamento do ESC. O sinal que deve ser produzido tem a seguinte característica: um time slot de 20ms deve receber um pulso de 1ms a 2ms de modo que 1ms representa potência 0% e 2ms, potência 100%.

Figura 18 – Sinal esperado na saída

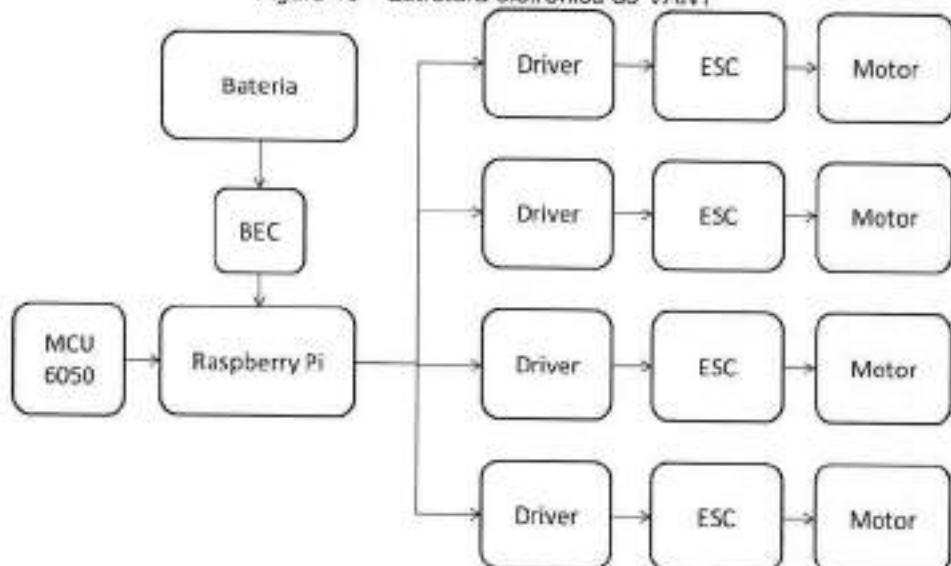


Fonte: Própria

De tal modo, a base do transistor permanecerá saturada entre 5% e 10% do tempo em (A) e entre 90% e 95% do tempo em (B). Assim, se torna evidente que o circuito proposto em (A) é superior, pois a potência dissipada no resistor de pull-down (ou pull-up no caso (B)) será inferior ao caso (B).

Portanto, determinados todos os componentes, principais e secundários, a estrutura eletrônica do VANT fica como mostrado na figura abaixo.

Figura 19 – Estrutura eletrônica do VANT



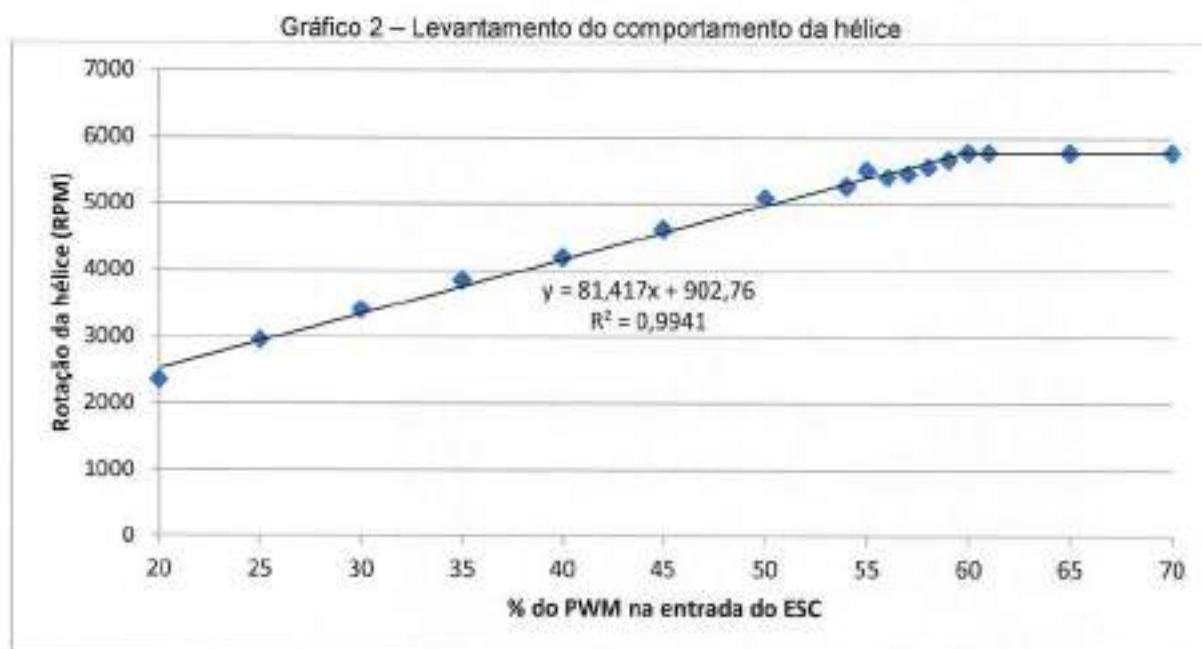
Fonte: Própria

Realizadas todas as avaliações e, portanto, as decisões referentes aos componentes e tecnologias que serão adotadas, pode-se gerar a lista de materiais (no Inglês, Bill of Materials - BOM) necessárias para a construção do agente voador. Para validação da concepção desta parcela do projeto, uma prova de conceito será efetuada, cuja função é verificar o voo controlado da plataforma voadora.

3.2.2. Modelagem física

Com o objetivo de controlar o quadricóptero, antes se fez necessário obter o modelo físico do VANT.

Como primeira etapa para a obtenção deste modelo, foi validada a hipótese de a resposta do quadricóptero ser linear. Para tal, o conjunto ESC-Motor-Hélice foi estimulado com um sinal PWM variável possibilitando a medida da resposta em termos de RPM da hélice.

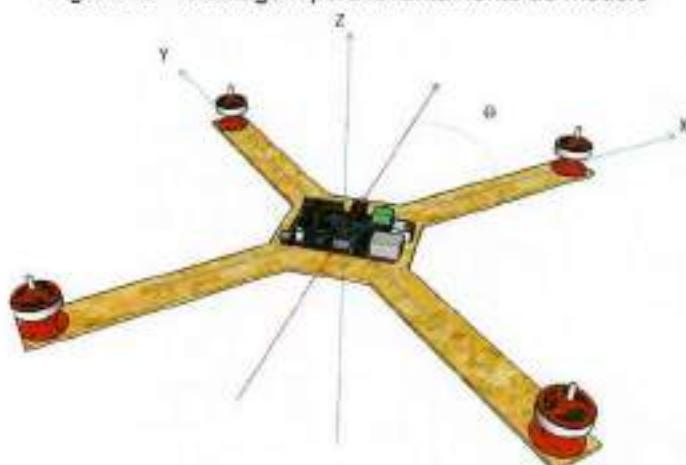


Fonte: Própria

A partir do gráfico verifica-se que o comportamento da rotação do motor-hélice é linear com saturação.

A próxima etapa para levantamento do modelo foi a abordagem do comportamento de cada eixo individualmente. Para tanto, estando um dos eixos fixado, o outro eixo permanece livre para movimentar-se e pode-se definir o ângulo θ , conforme disposto na figura abaixo.

Figura 20 – Montagem para levantamento do modelo



Fonte: Própria

O método utilizado para determinação do modelo para o quadricóptero foi o levantamento da resposta em frequência. Esta abordagem consiste da introdução de entradas senoidais no sistema e da medida da saída do mesmo. A saída apresenta a mesma frequência de oscilação da entrada, no entanto cada frequência possui características próprias de ganho e defasagem.

Os valores extraídos do quadricóptero foram submetidos à rotina do MATLAB apresentada no Apêndice F para extração dos parâmetros. Assim, obtiveram-se os pontos experimentais de ganho e fase para diversos valores de frequência utilizados para o levantamento do modelo.

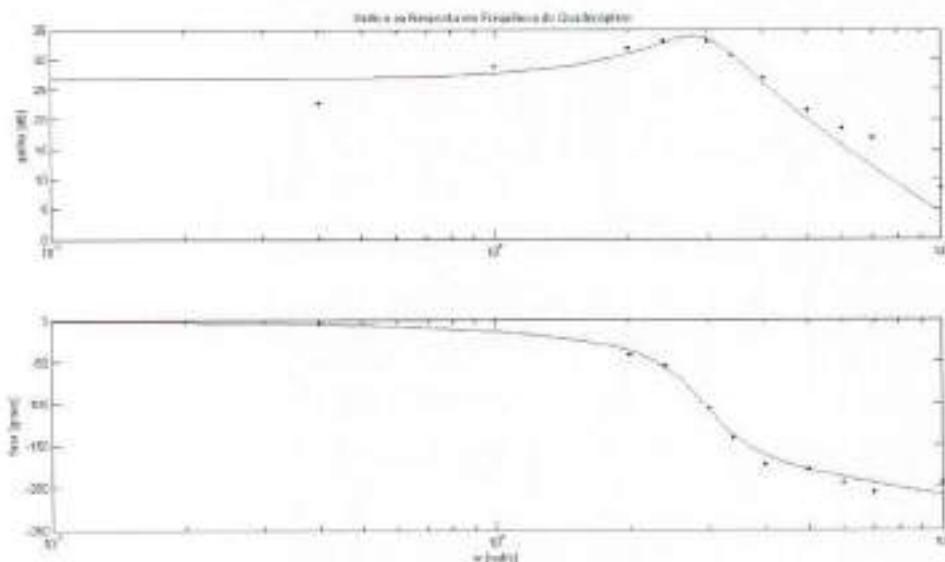
A partir dos pontos experimentais é possível determinar que o sistema é de 3ª ordem, devido à taxa de decaimento do ganho de 60dB/década a partir de 4 rad/s e à defasagem do sistema se aproximar de -270° para valores elevados de frequência. Tais pontos foram submetidos às rotinas do MATLAB do Apêndice G, que calculam os coeficientes para o sistema minimizando o erro quadrático em relação aos pontos experimentais.

Os coeficientes obtidos podem ser observados na equação abaixo, sendo possível também a observação da curva do modelo e dos pontos experimentais no gráfico 2.

Equação 1 – Modelo do quadricóptero

$$H(s) = \frac{2533}{s^3 + 14.63s^2 + 25.87s + 115.6}$$

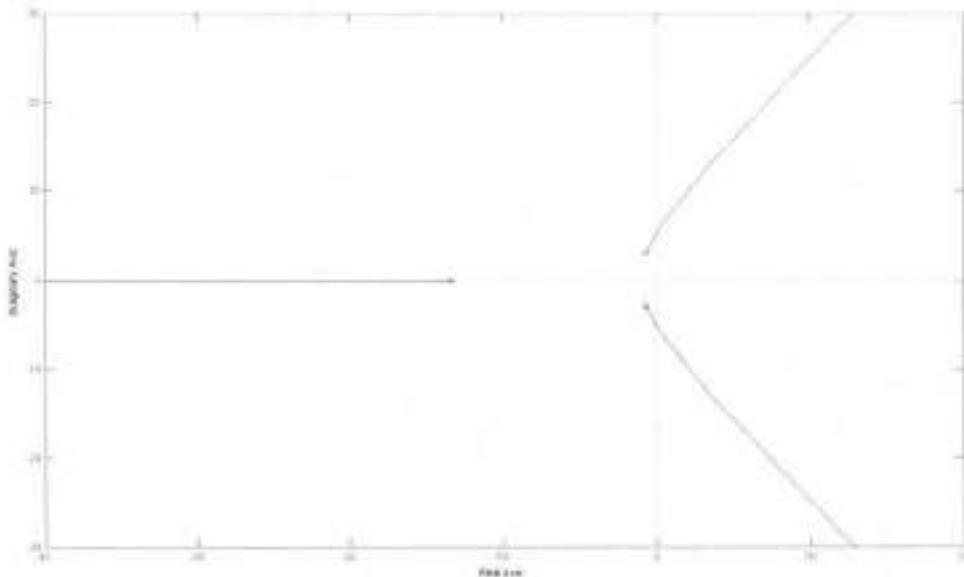
Gráfico 3 – Resposta em frequência experimental e do modelo



3.2.3. Controle

Partindo da resposta em frequência supracitada, de inicio, uma ferramenta útil para uma análise é o lugar geométrico das raízes (LGR) que fecha a malha do circuito com um controle proporcional cuja figura está logo abaixo. Conforme a figura abaixo, se pode observar que, com um controle proporcional com o aumento do ganho, o sistema sai do polo de malha aberta e vai para os polos de malha fechada ou para o infinito, caracterizando uma instabilidade. Assim, ao se obter o gráfico do LGR, observa-se que o sistema é instável: a instabilidade pode ser vista pela tendência dos polos se deslocarem para o semi-plano direito, resultando em uma exponencial crescente aplicada a anti-transformada de Laplace.

Figura 21 – Lugar geométrico das raízes de malha aberta



Fonte: Própria

Sendo assim, o foco passa a ser implementar uma função que, utilizando o Método do Lugar das Raízes (38), consiga deslocar os polos e zeros do sistema, tornando-o estável. A partir disso, sabendo que a função de transferência de malha aberta do sistema possui três polos, é imediato, a partir da teoria de Controle Moderno, adicionar uma segunda função de transferência com dois zeros. Esta

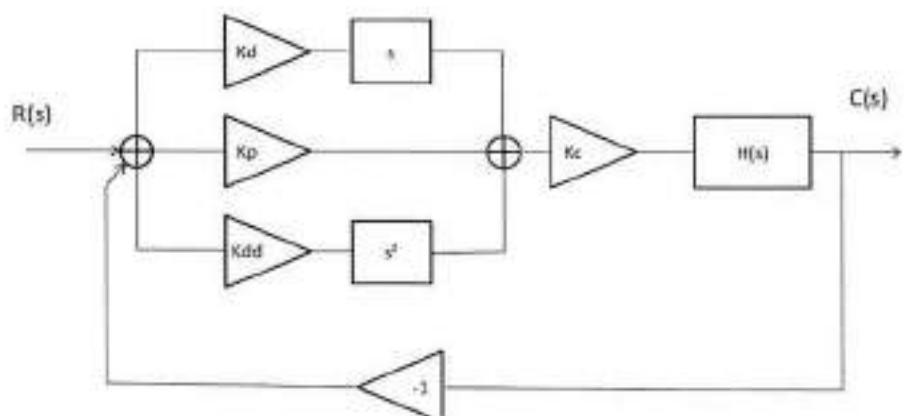
função, ilustrada na equação abaixo, permite que o sistema final atinja a estabilidade.

Equação 2 – Equação do sistema de controle

$$G(s) = K_{dd} \times s^2 + K_d \times s + K_p$$

O modelo proposto para malha fechada, com realimentação, está apresentado no diagrama abaixo.

Figura 22 – Diagrama de controle do sistema



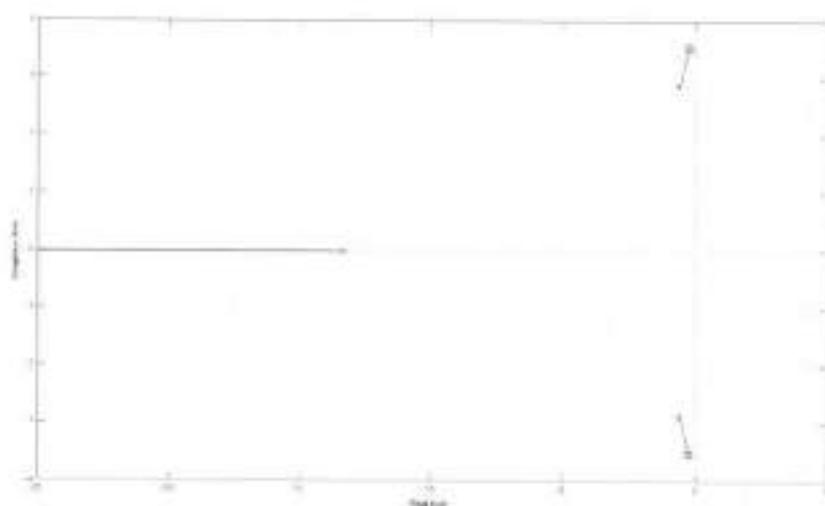
Fonte: Própria

No modelo acima:

- $R(s)$: Entrada do sistema – ângulo desejado;
- $C(s)$: Saída do sistema – ângulo informado pelo acelerômetro;
- K_p , K_d e K_{dd} : Ganho dos controladores proporcional, derivativo e derivativo de segunda ordem, respectivamente;
- K_c : Ganho do sistema de controle;
- $H(s)$: Função de transferência do quadricóptero.

A partir do modelo de controle definido, se obtém o novo LGR com as raízes de $G(s)$, conforme pode ser visto a seguir.

Figura 23 – Diagrama do lugar das raízes do sistema com controle



Fonte: Própria

Na figura acima, nota-se que os polos e zeros do sistema não estão mais no semi-plano direito, representando a estabilidade atingida. Por meio da ferramenta MATLAB, foi possível determinar os coeficientes numéricos da equação 2. Logo, a função $G(s)$ é dada por:

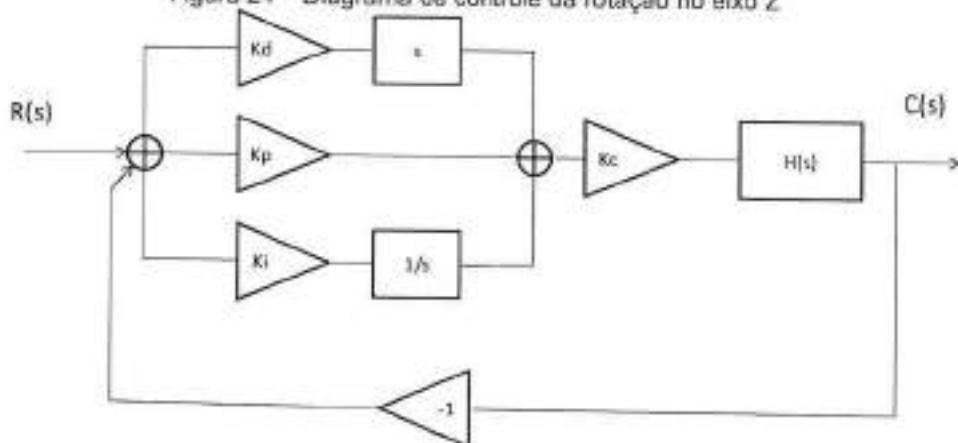
Equação 3 – Equação do sistema de controle

$$G(s) = 0.2 \times s^2 + 0.1 \times s + 2.5, \text{ com } K_c = 0.2$$

Concluído o modelo de controle para equilibrar um eixo do VANT, modelou-se, analogamente, o sistema de controle para o outro eixo. Logo, se tem o controle no plano XY.

Por fim, existe a necessidade de se controlar a posição angular no eixo Z, para se finalizar o sistema de controle de movimento do quadricóptero nos três eixos. Para isto, utiliza-se um controle PID (Proporcional Integrador Derivativo), cujo diagrama está representado abaixo.

Figura 24 – Diagrama de controle da rotação no eixo Z



Fonte: Própria.

Para o modelo ilustrado acima:

- $R(s)$: Entrada do sistema – ângulo desejado;
- $C(s)$: Saída do sistema – ângulo informado pelo acelerômetro;
- K_p , K_d e K_i : Ganho dos controladores proporcional, derivativo e integral, respectivamente;
- K_c : Ganho do sistema de controle;
- $H(s)$: Função de transferência do quadricóptero.

No VANT, o controle do eixo Z atua no incremento da potência dos motores, aumentando *duty cycle* (do Inglês, ciclo de trabalho) do PWM de cada um deles. Para se manter a estabilidade na direção normal ao plano dos motores, o controle PID acelera dois motores opostos, fazendo com que o momento linear seja nulo no eixo Z.

Portanto, por meio dos três modos de controle acima citados, um para cada eixo, é possível se executar o controle eletrônico do voo do agente voador. Da mesma forma que os demais softwares escritos neste projeto – exceto o programa de reconstrução 3D, escrito em C – os programas de controle foram escritos na linguagem Python.

3.3. Estação Base

O processo de reconstrução tridimensional de imagens, devido ao alto número de cálculos para definição das posições espaciais de cada pixel, deve ser feito em um computador que tenha alta capacidade de processamento.

O Raspberry Pi, contudo, não possui a capacidade de processamento necessária para se realizar uma reconstrução satisfatória, visto que sua CPU (do Inglês, *Central Processing Unit*) é limitada em termos computacionais: trata-se de um processador ARM de 700 MHz, do tipo ARM1176JZF-S core (família ARM11).

Para solucionar o problema da capacidade computacional, foi definido que o processo de reconstrução 3D, assim como o controle do quadricóptero, seria realizado via um computador em terra. Os comandos são enviados via comunicação sem fio para tornar possível o controle remoto do agente voador, visualizados por meio de uma interface gráfica, realizando a função de Interface Homem-Máquina (IHM). De modo geral, a estação base é responsável pela união dos dois subprojetos em um único sistema, caracterizando o projeto Image SyncCopter.

Nas seções seguintes, há o detalhamento da comunicação entre estação base e o quadricóptero e, também, da Interface Homem-Máquina.

3.3.1. Comunicação

Paralelo ao sistema de controle de voo e ao tratamento de imagens foi desenvolvido o sistema de comunicação entre o SIC e a estação em terra. Em relação à comunicação, são necessários três requisitos:

- Garantia de comando enviado entre a estação terra e o quadrimotor: Para que se possa controlar o VANT, é necessário poder enviar os comandos à distância. Quando for solicitado que o quadricóptero se movimente, é essencial que a ordem chegue ao agente voador.
- Garantia da entrega das imagens do quadricóptero para a estação terra: Ao capturar as imagens no VANT é necessário enviar para a estação base; assim é preciso certificar que a imagem foi entregue.

- Criar um canal de comunicação de implementação simples: como o escopo do projeto é o tratamento de imagens e o voo controlado do VANT, não é necessário implementar um sistema comunicação muito complexo, bastando apenas se ter um sistema que satisfaça os dois itens anteriores.

Após definidas as necessidades de comunicação, foram analisados três modelos de comunicação sem fio:

1. Comunicação Wi-Fi utilizando o protocolo UDP/IP;
2. Comunicação Wi-Fi utilizando o protocolo TCP/IP;
3. Comunicação em rádio frequência (RF).

Dentre as três opções acima, a terceira tem um nível de complexidade maior por ser preciso fazer adaptações para um perfeito acoplamento dos módulos de comunicação entre a unidade de controle e a estação base.

A primeira opção não atende ao critério de garantia de entrega dos pacotes de informação: o protocolo UDP/IP não garante a entrega da informação, além de não haver garantia de que a informação seja reconstruída de maneira correta no receptor caso ocorra a troca de ordem desses pacotes durante a transmissão de dados.

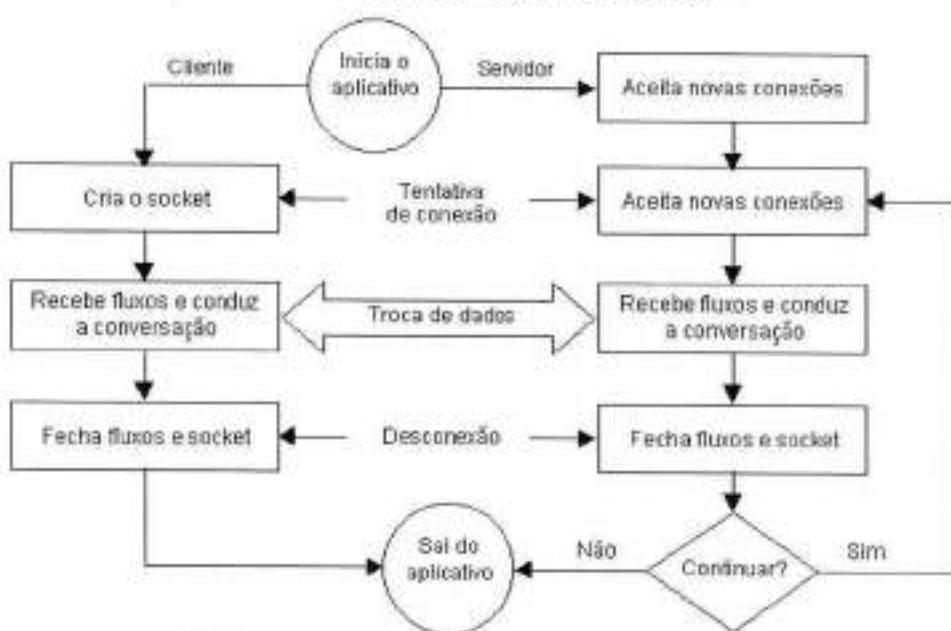
A segunda opção, portanto, é a mais indicada para atender os requisitos, pois cria um canal no meio. Por outro lado, caso haja troca dos pacotes, eles são reordenados através do próprio protocolo. Na situação de perda de pacotes, além do caso de perda de pacotes há o reenvio da informação.

Logo, a partir da escolha do uso do protocolo TCP/IP, a estrutura da comunicação foi definida através da relação "servidor-cliente": o servidor é a unidade de controle que está no VANT, para que este forneça as imagens quando capturadas. A estação terra, por sua vez, ficou definida como cliente, para solicitar as imagens e também enviar comandos básicos de movimento para o quadricóptero (subir, transladar e rotacionar).

Para estabelecer o canal de comunicação foram seguidos os passos básicos de comunicação TCP/IP e a porta 5000 foi determinada como porta-padrão, dentro

de uma rede Wi-Fi local. Os passos do protocolo de comunicação podem ser vistos na próxima figura.

Figura 25 – Comunicação TCP/IP



Fonte: <http://commons.wikimedia.org/wiki/File:300px-Tcp-handshake.png>

Com as características do sistema e a função de cada elemento na comunicação, o software de comunicação foi escrito. Visto que todos programas deste trabalho foram escritos em Python, por conveni\u00eancia seguiu-se a esta tend\u00eancia para os programa Servidor e Cliente escrevendo-os nessa linguagem.

3.3.2. Interface Homem-M\u00e1quina (IHM)

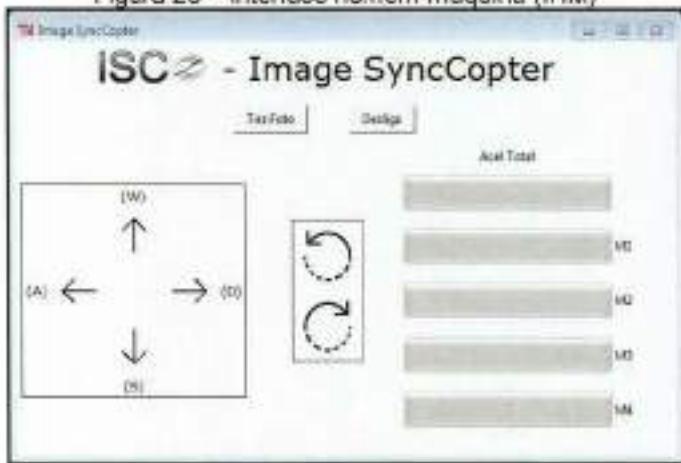
A Interface Homem-M\u00e1quina (IHM) possui a fun\u00e7\u00e3o de criar um canal de intera\u00e7\u00e3o entre o usu\u00e1rio e o quadric\u00f3ptero. Na IHM do Image SyncCopter, o usu\u00e1rio pode efetuar o controle remoto das fun\u00e7\u00e3es de todo sistema e visualizar par\u00e1metros de voo do quadric\u00f3ptero em tempo real. Assim, foi preciso desenvolver uma interface gr\u00e1fica – comumente conhecida como GUI (do Ingl\u00e9s, *Graphical User Interface*) – para disponibilizar visualmente as informa\u00e7\u00e3es advindas do sistema.

Como ambiente de desenvolvimento utilizou-se a linguagem Python atrav\u00e9s do pacote de *GuiProgramming* (Programa\u00e7\u00e3o de GUI) chamado Tkinter. A escolha

dessa ferramenta foi feita mediante seu uso amplamente difundido na criação de interfaces gráficas.

Definida a ferramenta de desenvolvimento, definiu-se que os parâmetros a ser expostos aos usuários seriam referentes ao controle do quadricóptero e da obtenção de imagens. A principal aplicação da IHM, portanto, é permitir a imersão do usuário durante o funcionamento do sistema, tornando maior o grau de realidade aumentada sentido por ele.

Figura 26 – Interface homem-máquina (IHM)



Fonte: Própria

3.4. Prova de conceito

Para validação dos conceitos do projeto proposto, foram determinados dois objetivos para serem atingidos na etapa inicial (primeiros seis meses) de desenvolvimento deste trabalho: a primeira prova de conceito consiste em reconstruir a imagem 3D de um objeto por meio de imagens previamente tiradas ao passo que a segunda tem por objetivo a construção e no voo controlado de um quadricóptero. Tanto a primeira quanto a segunda prova de conceito serão realizadas em ambientes controlados, isto é, em condições particulares que implicam em soluções simplificadas do escopo do projeto.

3.4.1. Primeira prova de conceito: Reconstrução 3D

A proposta final deste projeto, referente ao processamento de imagens, é reconstruir um cenário 3D a partir de imagens bidimensionais obtidas durante o voo. Desse modo, com intuito de se verificar conceitualmente essa proposta, realizou uma simulação do processamento de imagens em um ambiente controlado. Esta simulação consiste na realização da reconstrução 3D de um objeto, a partir de imagens previamente obtidas. Para realização satisfatória da prova de conceito, a simulação foi dividida em três partes:

(I) Obtenção de uma biblioteca de código aberto: nesta etapa, o grupo buscou soluções de código aberto para, além de evitar o custo vinculado a um software provado, ser possível efetuar eventuais mudanças no comportamento do programa responsável pela computação gráfica das imagens. Foi determinado, então, a utilização da biblioteca Open CV, pois esta é vastamente difundida em aplicações relacionadas à modelagem e ao processamento de imagens.

(II) Correção, verificação e compilação do código do software gráfico: a utilização de um programa computacional que permita a modelagem 3D através de fontes bidimensionais é vinculada, intrinsecamente, a um código de execução que acessa os algoritmos de modelagem e processamento de imagens. Logo, foi determinado o uso, como software base, do Insight3D, também de código aberto, para que fossem possíveis eventuais mudanças na programação, caso as funções executadas necessitem ser modificadas. Contudo, cabe salientar que foi preciso efetuar a correção de muitas instruções do código do Insight3D, pois muitas de suas funções não atendiam plenamente as necessidades do escopo da verificação de conceito.

(III) Execução do software e obtenção da imagem 3D: na etapa final da prova de conceito relacionada ao tratamento de imagens, executou-se o processamento, via software base, para obtenção da reprodução 3D. As imagens que serviram de modelo foram obtidas de modo manual, ou seja, as fotos foram tiradas manualmente por meio de uma câmera USB (diferentemente da proposição do projeto final, no qual este processo será feito via software).

Finalizadas essas três etapas, iniciou-se a parte de verificação com exemplos de sequências de fotos reais. No entanto, cabe salientar que inicialmente o software advindo do programa Insight3D não apresentou desempenho satisfatório, sendo necessário um alto volume de modificações no código fonte. Realizadas essas

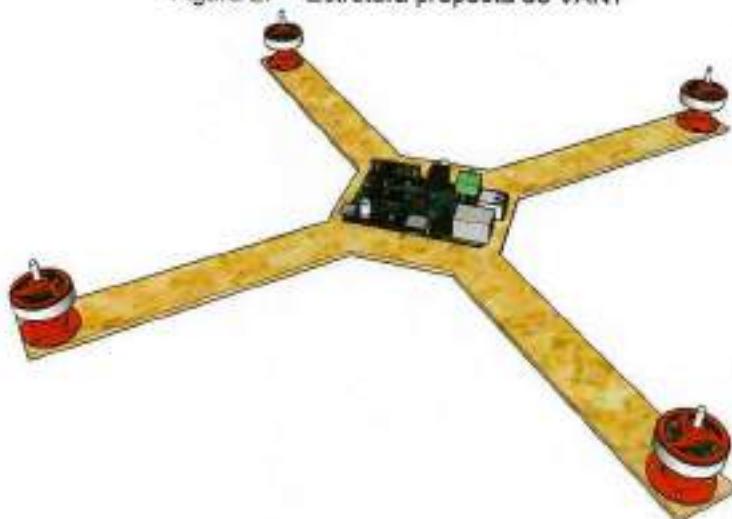
modificações, obteve-se um software de reconstrução funcional, tornando possível a reconstrução 3D.

3.4.2. Segunda prova de conceito: Voo controlado do agente voador

Na segunda prova de conceito, os objetivos versaram sobre a construção do quadricóptero e seu controle de voo.

No aspecto que tange a construção, foi gerado um protótipo com uma estrutura bem definida, conforme figura abaixo, em conjunto com a colocação dos componentes eletrônicos necessários. Porém, para se obter sucesso em relação à estrutura, analisou-se duas opções de materiais estruturais: plástico e madeira MDF. De forma empírica, optou-se pela madeira MDF para ser utilizada no modelo definitivo, visto que o primeiro protótipo, feito de plástico PVC apresentou falhas estruturais durante os testes de voo - o principal problema detectado foi a flexão das hastes de sustentação dos motores. Além disso, o quadricóptero da prova de conceito deve atender ao requisito de peso máximo – a plataforma voadora completa deve pesar menos de 2 kg – o que influenciou na escolha do material do protótipo final.

Figura 27 – Estrutura proposta do VANT



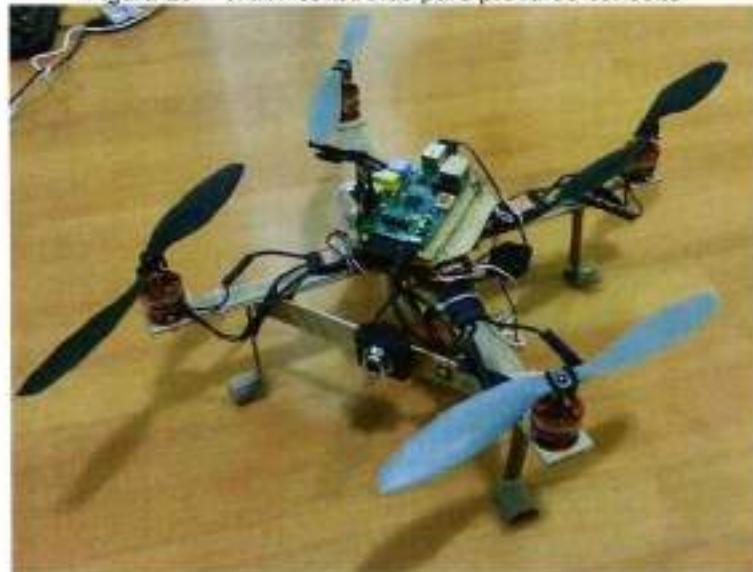
Fonte: Própria

Finalizada a construção, o principal objetivo é desenvolver e implementar o algoritmo que irá simular o comportamento durante o voo do quadricóptero. Para

isto, foram necessários alguns pré-requisitos, como programação em linguagem Python (principal linguagem de programação do Raspberry Pi), em ambiente Linux, e determinação prática da diferença de desempenho entre os motores. Estando atendidos os pré-requisitos, iniciou-se o desenvolvimento do algoritmo.

Inicialmente, verificou-se a diferença de performance de cada um dos motores, pois isto implica diretamente no equilíbrio do agente voador. Para isto, foram ligados os quatro motores em rotação baixa e, então, foi iniciada uma aceleração progressiva, verificando qual motor era capaz de levantar sua base do solo. Assim, repetindo o processo para cada um dos quatro motores, foi possível estabelecer um nível inicial de rotação, para que, ao se movimentar o quadricóptero, fossem evitados desequilíbrios oriundos de diferenças de aceleração.

Figura 28 – VANT construído para prova de conceito



Fonte: Própria

Funcionalmente, o algoritmo deve executar as funções básicas de movimento de um quadricóptero: subida (ou decolagem), estabilização em voo e descida (ou pouso). O modelo utilizado para controlar os motores foi a alimentação por PWM (Pulse Width Modulation), visto que seu controle é de fácil execução e, por conseguinte, simples programação.

Por fim, implementou-se o programa de teste, contendo o algoritmo de controle, no SIC, por meio do Raspberry Pi, para a execução dos primeiros ensaios.

Após as verificações práticas, obteve-se êxito nesta prova de conceito, como pode ser visualizado na figura abaixo, na qual o quadricóptero está em pleno voo.

Figura 29 – VANT em voo teste



Fonte: Própria

Em relação ao protótipo final, são previstas mudanças significativas, tanto em relação à estrutura quanto em relação à funcionalidade. No que se refere à estrutura, a plataforma voadora apresentará uma estrutura feita totalmente de madeira MDF, tendo uma relação peso/resistência estrutural satisfatória. Já o algoritmo de controle não irá conter somente as funções básicas, mas sim um sistema de realimentação de dados responsável por realimentar o SIC com informações vindas do conjunto acelerômetro/giroscópio. Por meio dessas informações, o SIC automaticamente calculará a velocidade de cada motor, equilibrando naturalmente o agente voador.

4. Resultados

Após a conclusão satisfatória das provas de conceito, foi iniciado o processo de aprimoramento do quadricóptero e da reconstrução tridimensional, para que ambos fossem desenvolvidos para atender os requisitos de engenharia e de marketing deste projeto.

4.1. Orçamento

De acordo com a escolha dos componentes com o método AHP abordada anteriormente, foi definida a uma prévia da lista de materiais, com os custos estimados a partir da comparação das alternativas. Esta lista é a base de componentes para a prova de conceito referente ao agente voador, sendo uma prévia dos custos totais do projeto. Abaixo, o orçamento inicial referente à lista de materiais.

Tabela 11 – Lista de materiais da Prova de Conceito

Partes	Quantidade	Custo Unitário	Custo Total
Câmeras	1	11,76	11,76
Motores	4	35,00	140,00
Raspberry Pi	1	117,60	117,60
Hélices	4	4,00	16,00
Bateria	2	50,40	100,80
Miscelâneas		200,00	200,00
TOTAL			586,16

Pode-se notar que a lista inicial de material não contempla elementos relacionados ao controle de estabilidade ao agente voador ou ao armazenamento de imagens.

Em relação ao protótipo final, no qual o quadricóptero foi aprimorado tanto em termos estruturais quanto em termos eletrônicos, outros custos materiais foram adicionados ao orçamento, tais como acelerômetros, giroscópios e medidores de

níveis de bateria. Esses novos custos foram integrados ao orçamento prévio de modo gradativo, durante o desenvolvimento de cada parte do projeto.

Logo, fez-se necessária a elaboração de um novo orçamento, ajustado aos gastos reais ao fim da construção do agente voador. A parte desse trabalho relacionada à reconstrução tridimensional, assim como calculado previamente, não representou gastos, visto que todos os softwares utilizados são de código aberto. As modificações e novos custos orçamentários podem ser vistos na tabela a seguir.

Tabela 12 – Lista de materiais – Protótipo Final

Material	Utilização	Qtd	Custo Unit. (R\$)	Custo Total (R\$)
Madeira MDF (em cm ²)	Estrutura	50	0,25	12,25
Raspberry Pi	Processador	1	160,74	160,74
Cartão de Memória SD	Memória	1	11,99	11,99
Hélices	Estrutura	4	10,10	40,40
Acelerômetro / Giroscópio	Eletrônica Embarcada	1	20,97	20,97
Motor c/ escovas	Estrutura	1	3,00	3,00
Transistores, diodos e chaves	Eletrônica Analógica	15	5,00	5,00
Adaptador Wi-Fi	Eletrônica Embarcada	1	29,80	29,80
Motores s/ escovas	Estrutura	4	35,00	140,00
Hélices	Estrutura	6	15,99	95,94
Bateria	Alimentação	1	69,99	69,99
ESC's	Eletrônica Embarcada	4	16,61	66,45
Carregador da bateria	Eletrônica Embarcada	1	39,90	39,90
Papel Transfer	Eletrônica Analógica	1	4,00	4,00
Câmera	Reconstrução 3D	1	11,76	11,76
Software 3D	Reconstrução 3D	1	0,00	0,00
Miscelâneas	Diversos		14,55	14,55
Total				726,74

Portanto, para se obter um protótipo funcional, é preciso um investimento de aproximadamente R\$ 730,00. Ao se fazer um comparativo com um modelo

comercial já estabelecido no mercado (como o A.R. Parrot®), cujo preço médio é de R\$ 2.500,00, conclui-se que o Image SyncCopter possui um preço bastante inferior.

Para efeito de produção de um produto final e de comparação com modelos comerciais, é necessário considerar os montantes relacionados à mão-de-obra empregada no processo de Pesquisa e Desenvolvimento (P&D), deve-se aplicar a seguinte divisão de gastos do projeto: 60% para gastos com P&D, 40% para gastos com materiais.

Tabela 13 – Comparativo de Custos

Modelo	Descrição de Custos	Custo Parcial (R\$)	Custo Total (R\$)
Image SyncCopter	P&D	1.090,11	1.816,85
	Material	726,74	
A.R. Parrot®	Aquisição	2.500,00	2.625,00
	Frete de entrega (5%)	125,00	

Conclui-se, por meio da comparativo acima, que o protótipo Image SyncCopter apresenta um custo total 31% menor em relação ao modelo comercial, atendendo ao requisito de marketing referente ao baixo custo de produção deste projeto.

4.2. Testes

Foram elaborados testes cuja finalidade é verificar a viabilidade das soluções pontuais apresentadas para cada segmento. A principal função desses testes é tornar possível a conclusão do projeto em sua totalidade. Para isto, elaborou-se um plano de integração e testes dos subsistemas, de modo que a ordem de realização fosse diretamente proporcional ao impacto dele no contexto geral do trabalho. Cada teste tem por objetivo colocar em funcionamento, nas condições reais de utilização, o subsistema em questão, analisando sua robustez e sua confiabilidade.

Alguns desses testes, contudo, já estão englobados nas provas de conceito, sendo que, no atual momento do desenvolvimento do projeto, foi realizada apenas sua verificação funcional, retificando o conceito previamente analisado.

As análises dos resultados dos testes são fundamentais para que seja possível a correção de incoerências entre o que se espera na teoria e o que se verifica na prática. Desse modo, o plano de integração foi determinado de maneira objetiva e detalhada, visando ilustrar o comportamento de cada módulo do projeto.

4.2.1. Plano de Integração e testes

Quanto ao quadricóptero, foi definido que cada sistema seria integrado e testado segundo a ordem:

I. Montagem e testes do Raspberry Pi

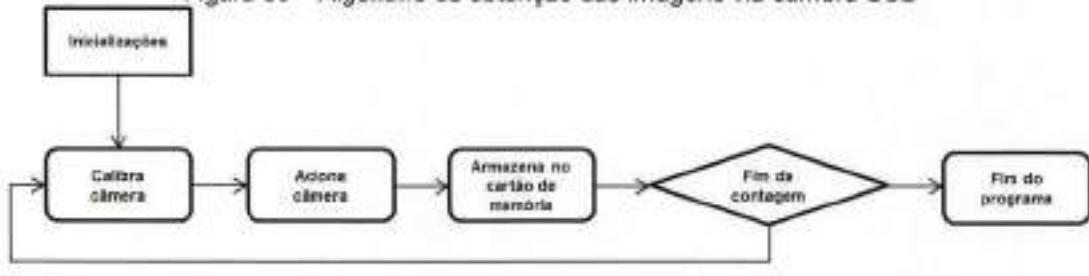
Realização feita através da alimentação da placa, instalação do sistema operacional, configuração dos drivers de vídeo, conexão da saída HDMI a um monitor compatível e testes gerais de desempenho e estabilidade.

Esta etapa é essencial, pois nela que se concluiu sobre a resposta satisfatória do sistema em relação às especificações do projeto, uma vez que o sistema operacional Linux permite a criação dos ambientes de programação necessários para desenvolvimento dos programas e rotinas dos softwares. Além disso, determinou-se que Python seria a linguagem de programação utilizada nos softwares de controle e de comunicação do agente voador, pois trata-se de uma linguagem interpretada, não havendo a necessidade de compilação.

II. Barramento USB e conexão da câmera

Realizado através da conexão da câmera à porta USB, testes em linha de comando do funcionamento da câmera e escrita de um programa capaz de capturar imagens em sequência e salvá-las no cartão de memória. De princípio simples, a conclusão efetiva do teste é vista no armazenamento das fotos tiradas com a câmera via software, mediante software cujo algoritmo está descrito abaixo.

Figura 30 – Algoritmo de obtenção das imagens via câmera USB

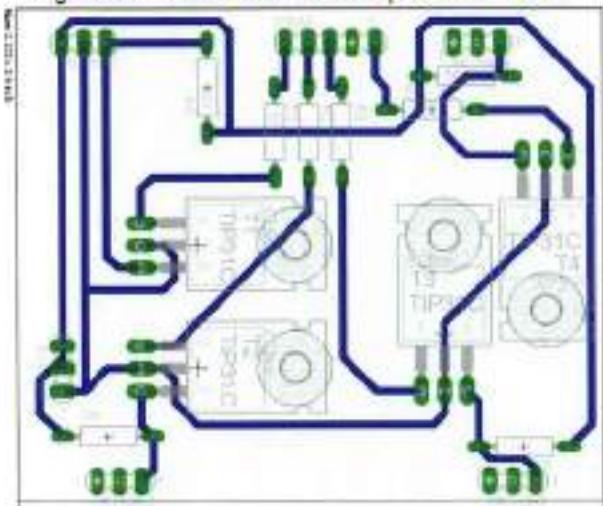


Fonte: Própria

III. Driver

Testes da placa de driver desenvolvida, sendo realizada a conexão da mesma ao barramento GPIO do Raspberry Pi e, com a utilização de um osciloscópio, constata-se a saída do pulso conforme esperado (de amplitude +3,3V). No entanto, para alimentação dos ESC's, é necessária a inserção dos drivers de tensão para se atingir um pulso de amplitude de +5V – conforme visto na seção 2.2 - Construção do agente voador. Assim, a integração da placa de driver com os ESC's conectados aos motores, realizou-se a conexão destes aos motores e à placa de driver de tensão, resultando em quatros motores funcionais.

Figura 31 – Placa de circuito impresso do driver



Fonte: Própria

IV. Acelerômetro e Giroscópio

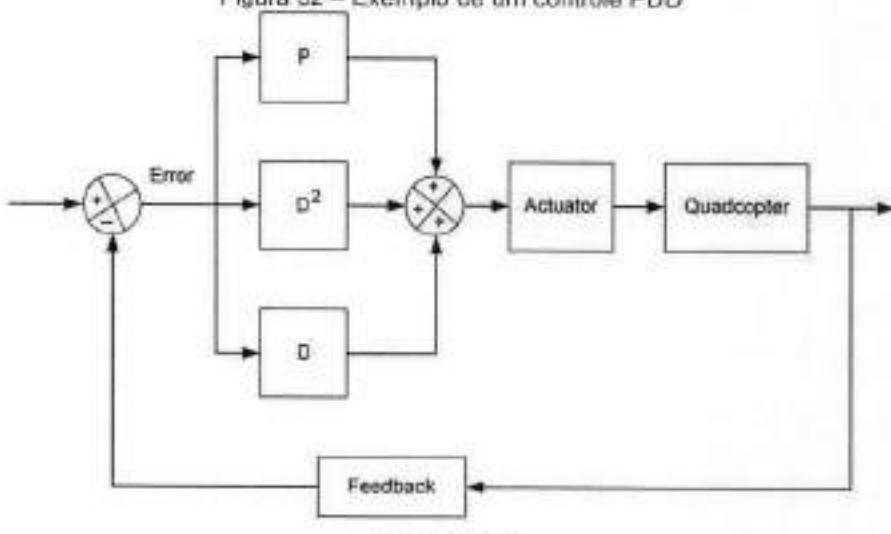
Conexão do MPU6050 e teste de funcionalidade, efetuada conectando-se o CI MPU6050 (acelerômetro e giroscópio) ao Raspberry Pi através da interface I2C. Para a verificação da funcionalidade, utilizou-se um programa de teste do protocolo

I2C, cuja função é de ler o registrador de endereço do MPU6050, após sua inicialização. Cabe salientar que a inicialização dos valores do MPU6050 é vital para o equilíbrio espacial da plataforma voadora, pois é por meio das informações de posição que o SIC irá coordenar a potência de cada motor. Por fim, analisou-se a aquisição de dados do MPU6050, através dos valores armazenados nos registradores dos giroscópios e acelerômetros.

V. Sistema Integrado de Controle

Integração de atuadores e sensores, realizado através da criação de um programa que execute o controle realimentado – adotou-se um controle PDD (Controle Proporcional Derivativo de segunda ordem).

Figura 32 – Exemplo de um controle PDD



Fonte: Própria

Após a formulação do controlador PDD via software, fez-se necessária a verificação da robustez desse controle durante um voo do agente voador. Assim, o teste de voo foi realizado com o objetivo de provar o controle realimentado desenvolvido em ambiente real com condições normais de uso, permitindo a análise das instabilidades vibracionais e fletoras da plataforma de voo.

VIII. Funcionamento

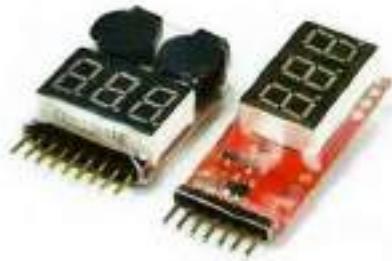
Os testes de funcionamento versam sobre a comunicação Wi-Fi entre o quadricóptero e estação base e, por fim, da autonomia da bateria durante um voo. O

teste de comunicação Wi-Fi do quadricóptero com um computador, realizado através de um programa no Raspberry Pi que se comunique com outro programa no computador sendo possível o controle dos movimentos através de comandos enviados via WiFi com a finalidade de verificar o requisito de engenharia correspondente.

Como resultado desse teste, a comunicação entre dois sistemas com o sistema operacional Linux, contidos na estação base e no agente voador, se mostrou satisfatória. No entanto, há a dependência de uma rede sem fio para que seja possível que os comandos inseridos no computador sejam transmitidos ao quadricóptero.

Por outro lado, o teste de autonomia de voo, foi realizado através do voo sustentado durante certo período com a finalidade de avaliar o requisito de engenharia definido no inicio do projeto. Foram realizados testes cujo objetivo foi verificar por quanto tempo o agente voador permanece ligado com os motores em funcionamento, sem que a variação da tensão da bateria atinja um nível mínimo. Empiricamente, a tensão mínima em cada célula da bateria não deve ser inferior à +3V. Para fazer o monitoramento da tensão da bateria é utilizado um sensor que, além de emitir um sinal sonoro ao se atingir o nível mínimo de tensão, mostra em um display de sete segmentos a diferença de potencial em cada célula. Na figura abaixo, um sensor análogo ao utilizado neste projeto.

Figura 33 – Sensor de nível de tensão da bateria



Fonte: <http://www.amazon.com/Bluecell-Digital-Battery-Voltage-Checker/dp/B0081Y6QC8>

4.3. Protótipo final

O protótipo definitivo da plataforma voadora, conforme mencionado na seção 2.3.2 – *Segunda prova de conceito: Voo controlado do agente voador* é um

melhoramento físico em relação ao quadricóptero construído inicialmente. As principais diferenças são referentes à mudança do material da estrutura (troca de plástico PVC por madeira MDF) e ao melhor posicionamento de componentes e drivers (compactação em uma única placa de circuito impresso). Em conjunto às mudanças estruturais, o controle eletrônico do agente voador foi aprimorado, permitindo uma melhora significativa na controlabilidade do protótipo.

4.3.1. Estrutura

A modificação estrutural elimina a existência de um momento fletor significativo durante o voo do quadricóptero, influenciando diretamente na estabilidade aérea e indiretamente na qualidade das imagens obtidas, uma vez que trepidações na câmera podem gerar ruídos e borões nas fotografias geradas.

A realocação dos componentes e dos drivers em uma única placa de circuito impresso implicou em uma melhor organização física da eletrônica embarcada do agente voador. Isto, além de dar mais proteção aos componentes durante um voo, permite uma localização mais rápida de possíveis problemas de hardware. Diante da movimentação durante as incursões aéreas do quadricóptero, faz-se necessário uma robustez em relação à resistência física do local onde os componentes estão alocados. O protótipo final montado pode ser visto na figura abaixo.

Figura 34 – Protótipo final



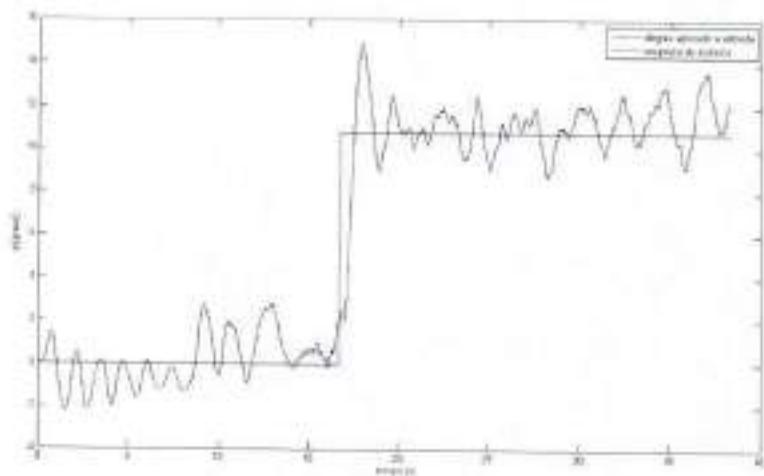
Fonte: Própria

4.3.2. Controle de Voo

O desenvolvimento dos controladores de voo, sendo eles controles do tipo PDD, para os eixos X e Y e do tipo PID, para o eixo Z, possibilitaram o atingimento do objetivo proposto inicialmente: executar voos estáveis.

Partindo do sistema de controle aplicado apresentado na seção 3.2.3 – *Controle*, é possível verificar a viabilidade desta solução por meio de simulações de movimentos do quadricóptero. Uma das simulações realizadas foi a aplicação de um degrau na entrada do sistema – podendo ser essa entrada no eixo X, Y ou Z.

Gráfico 4 – Resposta ao degrau do sistema



Conforme pode ser visto no gráfico acima, a aplicação de um degrau na entrada do sistema de controle gera uma resposta coerente na saída deste, visto que podem ser notados dois patamares bem distintos: um de nível lógico baixo, medições em torno do zero, e outro de nível lógico alto, em torno 11 graus. Logo, comparando-se a uma curva simulada na ferramenta MATLAB com a curva de saída do sistema real, a resposta mostra-se satisfatória, considerando um erro estacionário de 2° - sendo este erro desprezível mediante as dimensões do quadricóptero, no qual uma inclinação de 4° implica em uma altura de aproximadamente 3mm.

O procedimento de resposta do sistema de controle, portanto, mostra-se eficaz, uma vez que é possível interpretar os movimentos do quadricóptero como degraus de duração infinitesimal e, também, de diferentes amplitudes.

4.4. Reconstrução 3D

Quanto ao software de reconstrução 3D será utilizado como base o software Insight3D. No entanto, em primeiro momento este software não apresentou funcionamento como esperado sendo necessária a execução de diversas modificações no mesmo.

Para verificação da eficiência do software modificado, utilizou-se um conjunto de imagens reais de um objeto – neste caso, dias paredes de uma casa – de modo a se verificar como seria o comportamento do algoritmo de reconstrução tridimensional. Este conjunto de imagens bidimensionais deve obedecer apenas uma condição de contorno: a posição relativa das câmeras não pode superar a defasagem de 15°, pois senão há o risco de existir uma grande distorção na nuvem de pontos – pontos estes responsáveis pela localização espacial de cada pixel da imagem. As fotos escolhidas para verificação da funcionalidade do software estão na figura a seguir.

Figura 35 – Conjunto de imagens de verificação



Fonte: Própria

Tal solução utiliza a biblioteca OpenCV para manipulação das imagens. A partir das imagens entradas é calculada a correlação entre duas imagens para determinar quais pontos representam a mesma posição espacial. A partir desta nuvem de pontos são calculadas as distâncias relativas com as quais é possível triangular a posição de cada câmera para cada foto.

Figura 36 – Nuvem de pontos e posições das câmeras



Fonte: Própria

Como resultado final, tem-se a reconstrução da aresta contida entre as duas paredes evidenciadas no conjunto de fotos tomadas como teste. Verifica-se, portanto, a eficiência das melhorias executadas no software base proveniente do Insight 3D.

Figura 37 – Reconstrução 3D



Fonte: Própria

5. Discussão

5.1. Site do projeto

Com o intuito de apresentar de uma forma sintetizada as informações deste trabalho foi criado o site <http://imagesynccopters.sourceforge.net>. Neste estão contidas informações como os objetivos do projeto, os códigos de reconstrução 3D e de controle, além de ilustrar as principais características dos elementos do Image SyncCopter. Para o desenvolvimento do site foi utilizada linguagem HTML.

Figura 38 – Imagem do site



Fonte: Própria

5.2. Próximas Etapas

Ao fim do projeto, foram obtidos protótipos funcionais que atendem satisfatoriamente os requisitos de Marketing e Engenharia. Contudo, são protótipos, necessitando de uma etapa complementar de pesquisa e desenvolvimento (P&D) para que se obtenha, de fato, um produto comercializável. Assim, como etapas complementares à finalização desse trabalho, têm-se a miniaturização do quadricóptero, desenvolvimento do algoritmo de controle de voo e aprimoramento do software de reconstrução tridimensional.

O desenvolvimento desses três itens permitirá ao Image SyncCopter competir no mercado com os demais produtos já estabelecidos, como o A.R. Parrot®, tanto

em termos de performance quanto em relação ao custo. Para isto, a captação de investimentos, assim como um tempo de maior aplicação nos processos de P&D são vitais para conclusão das etapas complementares.

Como etapa final, após o desenvolvimento de um produto comercializável, pode-se fundar uma empresa *start-up*, que além de vender o produto Image SyncCopter pode fornecer os serviços orientados às especificações do projeto, como reconhecimento de perfis topográficos.

5.2.1. Captação de investimentos

Por se tratar de um projeto de caráter acadêmico, há a possibilidade de se efetuar a captação de recursos financeiros por meio de instituições estaduais ou federais, como a Fapesp e o BNDES. Essas instituições proporcionam investimentos com capital investido a fundo perdido – ou seja, não há a obrigatoriedade de se devolver os montantes fornecidos.

Nesse aspecto, o PIPE (Pesquisa Inovativa em Pequenas Empresas) se mostra como principal alternativa. O PIPE, por definição, tem o objetivo de apoiar a execução de pesquisas em inovação, científica ou tecnológica, em pequenas empresas que residam no Estado de São Paulo. Os montantes de investimentos oriundos do PIPE podem somar R\$ 200 mil em um primeiro momento, havendo a possibilidade de atingir R\$ 1 milhão em um estágio avançado de desenvolvimento.

5.2.2. Pesquisa e Desenvolvimento

Obtidos os investimentos necessários para se financiar as pesquisas relacionadas ao melhoramento do protótipo, pode-se iniciar o processo de desenvolvimento do produto Image SyncCopter.

No que tange o desenvolvimento da miniaturização e do algoritmo de controle, é preciso a aquisição de novos componentes eletrônicos, tais como motores que possuam a mesma relação empuxo/potência que os motores atualmente usados, porém com menores dimensões. Já a estabilização de voo via controle eletrônico pode ser aprimorada com a aquisição de uma estação eletrônica de controle embarcado (fornecidos por empresas como a Parrot® e a Open Pilot®).

ou por meio de otimização do algoritmo de controle desenvolvido para o protótipo. Analogamente ao controle de voo, o melhoramento do software de reconstrução tridimensional é feito por meio da otimização do algoritmo previamente desenvolvido.

Todas essas ações, sejam elas relacionadas às aquisições ou ao desenvolvimento de softwares, possuem como base os investimentos para financiar os gastos com materiais, para compra de componentes eletrônicos e estruturais, e gastos com pessoal, nas atividades de desenvolvimento de software.

5.3. Conclusão

O projeto de formatura, por se tratar de um trabalho de conclusão de curso que visa integrar as áreas do conhecimento do curso de Engenharia Elétrica, permite aos alunos aplicar seus conhecimentos em um projeto. Diante disso, a escolha do tema deste trabalho buscou reunir áreas como Automação e Controle, Eletrônica Embarcada e Filtros Digitais, resultando em um projeto de alta complexidade.

Com o fechamento das atividades de desenvolvimento, pode-se afirmar que o resultado do projeto está alinhado com as expectativas do grupo no que se refere à funcionalidade. Funcionalmente, obteve-se um protótipo capaz de efetuar a captação de imagens de um objeto e fazer a reconstrução tridimensional, atendendo aos requisitos de Marketing e de Engenharia do projeto.

Para o atingimento das metas propostas a princípio, foi necessária a divisão do trabalho em dois subprojetos: o primeiro responsável pela reconstrução tridimensional e o segundo na construção de um quadricóptero capaz de realizar voos estáveis. A metodologia utilizada foi realizar de provas de conceito, cada uma sendo responsável por simular o comportamento de um subsistema em um ambiente controlado.

Em conjunto com a parte técnica, foi preciso desenvolver a capacidade de gestão de projetos, haja vista o volume de atividades necessárias para se atingir o objetivo proposto. Nesse aspecto, a utilização de conhecimentos de gerenciamento de custos, riscos e prazos se mostraram muito efetivas, principalmente na coordenação das decisões a serem tomadas durante a execução do projeto.

Por fim, pode-se afirmar que a gestão das atividades a serem realizadas e o aprofundamento técnico em cada segmento da Engenharia Elétrica abordado, foram

essenciais para o sucesso do projeto, tanto em termos de prazo quanto em termos de entrega de objetivos.

Referências

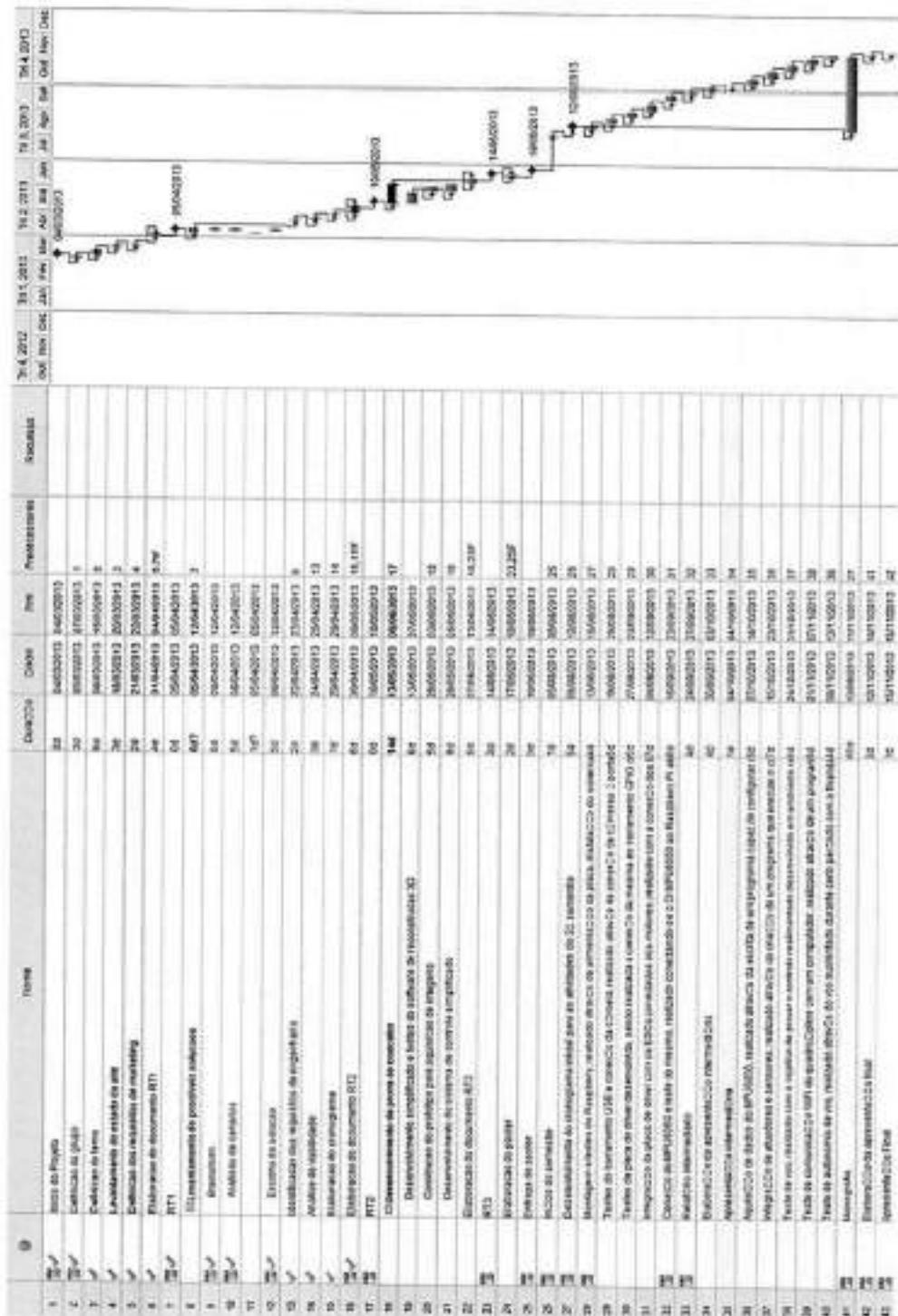
1. Drone registra imagens aéreas de protesto na Paulista; veja. **Folha de São Paulo**, 20 jun. 2013. Disponível em: <<http://www1.folha.uol.com.br/multimidia/videocasts/2013/06/1298766-drone-registra-imagens-aereas-de-protesto-na-paulista-veja.shtml>>. Acesso em: 21 jun. 2013.
2. Em 1ª ação conjunta de drones, FAB e PF apreendem drogas na fronteira. **G1 – RPCTV**, 23 mai. 2013. Disponível em: <<http://g1.globo.com/pr/parana/noticia/2013/05/em-1-acao-conjunta-de-drones-fab-e-pf-apreendem-drogas-na-fronteira.html>>. Acesso em: 21 jun. 2013.
3. Ataque de drone dos EUA mata militantes islâmicos no Paquistão. **G1**, 24 set. 2012. Disponível em: <<http://g1.globo.com/mundo/noticia/2012/09/ataque-de-drone-norte-americano-mata-militantes-islamicos-no-paquistao.html>>. Acesso em: 20 jun. 2013.
4. DAS, Jnaneshwar et al. **Environmental Sensing using Land-based Spectrally-selective Cameras and a Quadcopter**. In: The 13th International Symposium on Experimental Robotics, 2012 (ISER 2012). p. 259-272.
5. Longhitano, George Alfredo. **VANTS para sensoriamento remoto: aplicabilidade na avaliação e monitoramento de impactos ambientais causados por acidentes com cargas perigosas**. São Paulo, 2010. 147 p.
6. Pounds, P; Mahony, R; Corke, P. **Modelling and control of a large quadrotor robot**. CONTROL ENGINEERING PRACTICE, v. 18, n. 7, p. 691-699. 2010.
7. Bouabdallah, S; Murrieri, P; Siegwart, R. **Design and control of an indoor micro quadrotor**. IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2004, p. 4393-4398. 2004.
8. HARADA, K; SUCIATI, N. **Converting 2D Image into Sequence of Curves on 3D Flat Model**. IEEE, 2009.
9. PMI. **PMBoK - Project Management Body of Knowledge**. São Paulo: Saraiva, 2009, 4ª edição.
10. GANTTER. Disponível em: <<http://gantter.com/>>. Acesso em: 30 abr. 2013.
11. Insight 3D. Disponível em: <<http://insight3d.sourceforge.net>>. Acesso em: 29 mar. 2013.

12. GNU Operating System. **GNU AFFERO GENERAL PUBLIC LICENSE**. Disponível em: <<http://www.gnu.org/licenses/agpl-3.0.html>>. Acesso em: 10 set. 2013.
13. OPENCV. **Open Source Computer Vision**. Disponível em: <<http://opencv.org/>>. Acesso em: 08 de Agosto de 2013.
14. Raspberry Pi. Disponível em: <<http://www.raspberrypi.org>>. Acesso em: 01 mai. 2013.
15. Arduino. Disponível em: <<http://www.arduino.cc>>. Acesso em: 01 mai. 2013.
16. DimensionEngineering. **A beginner's guide to accelerometers**. Disponível em: <<http://www.dimensionengineering.com/info/accelerometers>>. Acesso em: 14 mai. 2013.
17. How Stuff Works. **How Gyroscope Works**. Disponível em: <<http://www.howstuffworks.com/gyroscope.htm>>. Acesso em: 14 mai. 2013.
18. InvenSense. **MPU-6000 and MPU-6050 Product Specification Revision 3.3**. InvenSense, 2012.
19. University of Pennsylvania. **Construction with Quadrotors**. Disponível em: <<http://www.upenn.edu/spotlights/quadrotors>>. Acesso em: 27 mar. 2013.
20. AeroQuad. **AeroQuad - The Open Source Quadcopter**. Disponível em: <<http://aeroquad.com/content.php>>. Acesso em: 26 mar. 2013.
21. RC Groups. Disponível em: <<http://www.rcgroups.com/>>. Acesso em: 28 mar. 2013.
22. Coolfactor. **Hacking the Wii Motion Plus**. Disponível em: <<http://www.coolfactor.org/blog/2010/01/31/hacking-the-wii-motion-plus/>>. Acesso em: 13 mar. 2013.
23. MULTIWII. **MultiWii**. Disponível em: <<http://www.multiwii.com/>>. Acesso em: 17 mar. 2013.
24. QUAN, L. **Image-based Modeling**. Springer, 1st Edition.

25. Portal Gestão. **Como tomar decisões eficazes.** Disponível em: <<http://www.portal-gestao.com/gestao/item/1454-como-tomar-decis%C3%B5es-eficazes.html>>. Acesso em: 30 abr. 2013.
26. Bouabdallah, Samir, Andre Noth, and Roland Siegwart. **PID vs LQ control techniques applied to an indoor micro quadrotor.** *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on.* Vol. 3. IEEE, 2004.
27. Protocolo de comunicação I²C. **I²C – bus specification and user manual.** Disponível em: <http://www.nxp.com/documents/user_manual/UM10204.pdf>. Acesso em: 07 jul. 2013.
28. RIGHETTI, X.; THALMANN, D. **Proposition of a modular I2C-based wearable architecture.** IEEE, 2010.
29. The Python Tutorial. **The Python Tutorial.** Disponível em: <docs.python.org/tutorial>. Acesso em: 27 jul. 2013.
30. Protocolo de comunicação SPI. **Serial Peripheral Interface.** Disponível em: <www1.microchip.com/downloads/en/devicedoc/spi.pdf>. Acesso em: 28 jul. 2013.
31. SUNDARAM, G.S.; PATIBANDALA, B. **Bluetooth communication using a touchscreen interface with the Raspberry Pi.** IEEE, 2013.
32. DOOBESOVA, Z. **Programming Language Python for Data Processing.** IEEE, 2011.
33. TkInter. **TkInter – Gui Programming.** Disponível em: <docs.python.org/2/library/tkinter.html>. Acesso em: 09 ago. 2013.
34. Controle Proporcional Integral Derivativo aplicado a um Quadricóptero. Disponível em: <<http://blog.oscarliang.net/quadcopter-pid-explained-tuning>>. Acesso em: 08 set. 2013.
35. PIPE – Pesquisa Inovativa em Pequenas Empresas. **PIPE.** Disponível em: <www.fapesp.br>. Acesso em: 15 nov. 2013.
36. OpenPilot. **OpenPilot – The Next Generation Open Source UAV Autopilot.** Disponível em: <www.openpilot.org>. Acesso em: 15 nov. 2013.

37. Ogatta, Katsuhiko. **Controle Moderno 2º Edição** -Capítulo 6- Análise pelo Método do Lugar das Raízes.

APÊNDICE A – Cronograma



APÊNDICE B – Detalhamento do cálculo para solução de câmeras

Tabela 14 – Cálculos do método AHP para câmeras

Tipo	Custo (R\$)	Peso	Resolução	USB	Pesos
Custo (R\$)	1,00	6,00	4,00	0,20	0,24
Peso	0,17	1,00	0,50	0,11	0,05
Resolução	0,25	2,00	1,00	0,20	0,09
USB	5,00	9,00	5,00	1,00	0,62

Custo (R\$)	Câmera Iphone 3G	Mini Webcam convencional	Webcam Y3000	Webcam de Notebook	Peso
Câmera Iphone 3G	1,00	1,76	12,06	5,38	0,54
Mini Webcam convencional	0,57	1,00	6,86	3,06	0,31
Webcam Y3000	0,08	0,15	1,00	0,45	0,05
Webcam de Notebook	0,19	0,33	2,24	1,00	0,10

Peso	Câmera Iphone 3G	Mini Webcam convencional	Webcam Y3000	Webcam de Notebook	Peso
Câmera Iphone 3G	1,00	2,50	1,25	4,50	0,41
Mini Webcam convencional	0,40	1,00	0,50	1,80	0,17
Webcam Y3000	0,80	2,00	1,00	3,60	0,33
Webcam de Notebook	0,22	0,56	0,28	1,00	0,09

Resolução	Câmera Iphone 3G	Mini Webcam convencional	Webcam Y3000	Webcam de Notebook	Peso
Câmera Iphone 3G	1,00	1,54	0,25	1,54	0,16

Resolução	Câmera Iphone 3G	Mini Webcam convencional	Webcam Y3000	Webcam de Notebook	Peso
Mini Webcam convencional	0,65	1,00	0,16	1,80	0,10
Webcam Y3000	4,00	6,15	1,00	6,15	0,63
Webcam de Notebook	0,65	1,00	0,16	1,00	0,10
<hr/>					
Resolução	Câmera Iphone 3G	Mini Webcam convencional	Webcam Y3000	Webcam de Notebook	Peso
Câmera Iphone 3G	1,00	0,11	0,11	0,11	0,04
Mini Webcam convencional	9,00	1,00	1,00	1,00	0,32
Webcam Y3000	9,00	1,00	1,00	1,00	0,32
Webcam de Notebook	9,00	1,00	1,00	1,00	0,32
<hr/>					

APÊNDICE C – Detalhamento do cálculo para solução de formato

Tabela 15 – Cálculos do método AHP para solução de formato

Tipos	Custo (R\$)	Peso	Estabilidade	Controlabilidade	Pesos
Custo (R\$)	1,00	6,00	0,33	2,00	0,26
Peso	0,17	1,00	0,14	0,33	0,05
Estabilidade	3,00	7,00	1,00	4,00	0,55
Controlabilidade	0,50	3,00	0,25	1,00	0,14

Custo (R\$)	1 hélice	2 hélices	3 hélices	4 hélices	Peso
1 hélice	1,00	1,30	1,60	1,90	0,34
2 hélices	0,77	1,00	1,23	1,46	0,26
3 hélices	0,63	0,81	1,00	1,19	0,21
4 hélices	0,53	0,68	0,84	1,00	0,18

Peso	1 hélice	2 hélices	3 hélices	4 hélices	Peso
1 hélice	1,00	1,56	2,13	2,69	0,40
2 hélices	0,64	1,00	1,36	1,72	0,26
3 hélices	0,47	0,74	1,00	1,26	0,19
4 hélices	0,37	0,58	0,79	1,00	0,15

Estabilidade	1 hélice	2 hélices	3 hélices	4 hélices	Peso
1 hélice	1,00	0,33	0,20	0,13	0,06
2 hélices	3,00	1,00	0,60	0,38	0,18
3 hélices	5,00	1,67	1,00	0,63	0,29
4 hélices	8,00	2,67	1,60	1,00	0,47

Controlabilidade	1 hélice	2 hélices	3 hélices	4 hélices	Peso
1 hélice	1,00	0,33	0,20	0,13	0,06
2 hélices	3,00	1,00	0,60	0,38	0,18
3 hélices	5,00	1,67	1,00	0,63	0,29
4 hélices	8,00	2,67	1,60	1,00	0,47

APÊNDICE D – Detalhamento do cálculo para solução de sistema de controle SIC

Tabela 16 – Cálculo do método SHP para solução de SIC

Tipos	Custo (R\$)	Capacidade de processamento	Integração do controle	Pesos
Custo (R\$)	1,00	1,00	4,00	0,48
Capacidade de processamento	1,00	1,00	0,17	0,17
Integração do controle	0,25	6,00	1,00	0,35
Custo (R\$)	Arduino	Raspberry Pi	Peso	
Arduino	1,00	2,33	0,70	
Raspberry Pi	0,43	1,00	0,30	
Capacidade de processamento	Arduino	Raspberry Pi	Peso	
Arduino	1,00	0,13	0,35	
Raspberry Pi	8,00	1,00	2,83	
Integração do controle	Arduino	Raspberry Pi	Peso	
Arduino	1,00	0,17	0,41	
Raspberry Pi	6,00	1,00	2,45	

APÊNDICE E – Detalhamento do cálculo para solução de motor

Tabela 17 – Cálculo do método AHP para solução de motor

Tipo	Custo (R\$)	Peso (mg)	Controlabilidade	Utilização	Pesos
Custo (R\$)	1,00	0,33	0,25	0,25	0,07
Peso (mg)	3,00	1,00	0,14	0,14	0,09
Controlabilidade	4,00	7,00	1,00	0,25	0,28
Utilização	4,00	7,00	4,00	1,00	0,57
<hr/>					
Custo (R\$)	Brushless		Brushed		Pesos
Brushless			1,00		0,46
Brushed			1,15		0,54
<hr/>					
Peso (mg)	Brushless		Brushed		Pesos
Brushless			1,00		0,52
Brushed			0,91		0,48
<hr/>					
Controlabilidade	Brushless		Brushed		Pesos
Brushless			1,00		0,33
Brushed			2,00		0,67
<hr/>					
Utilização	Brushless		Brushed		Pesos
Brushless			1,00		0,56
Brushed			0,80		0,44

APÊNDICE F – Rotinas MATLAB para identificação de parâmetros da saída do sistema

```

function [A,w,phi]=senofit(t,y,Ao,wo,phio)
    % Exibe o ajuste inicial
    plot(t,y,'g',t,Ao*sin(wo*t + phio),'r');
    tit = sprintf('Ao = %g ; wo = %g ; phio = %g',Ao,wo,phio);
    title(tit);
    legend('curva original','aproximação inicial');
    xlabel('PRESSIONE QUALQUER TECLA PARA INICIAR O AJUSTE');
    figure(gcf);
    pause;
    % Otimização
    Awphio = [ Ao ; wo ; phio ];      % empacota em um vetor
    options = optimset('Display','off','TolFun',1e-4,'TolX',1e-4);
    % Invoca o algoritmo
    Awphi = fminunc(@(Awphi) senofit_erquad(Awphi,t,y),Awphio,options);
    A = Awphi(1); w = Awphi(2); phi = Awphi(3);    % Retorna os valores

```

```

function f=senofit_erquad(Awphi,t,y)
    t=t(:); y=y(:);                  % convertido em vetores coluna
    yy = Awphi(1)*sin(Awphi(2)*t+Awphi(3)); % yy é uma aproximação para y
    f = norm( yy-y );                % erro quadratico entre yy e y
    plot(t,y,'g',t,yy,'r');          % traça o gráfico com o ajuste
    tit = sprintf('A = %g ; w = %g ; phi = %g',Awphi);
    title(tit);
    drawnow; figure(gcf);           % atualiza a tela imediatamente

```

APÊNDICE G – Rotinas MATLAB para identificação de parâmetros da função de transferência do sistema

```
% Condições Iniciais
numDeno = [1400 0.5 4 13 62];
% Otimização da função de transferência
[num, den] = bodefit(freq,G,P, numDeno);
% Função de transferência encontrada
quadmodel = tf(num, den)
% Exibindo resposta em frequência da solução encontrada com os pontos
% experimentais
[mag, phase, w] = bode(quadmodel);
magdb = 20*log10(reshape(mag, size(mag,3), 1));
phase = reshape(phase, size(mag, 3), 1);
ganho = 20*log10(G);
phi = 180/pi() * P;
subplot(2,1,1); semilogx(w, magdb, 'r', freq, ganho, '.');
xlim([0.1 10]);
title('Gráfico da Resposta em Frequência do Quadricóptero');
ylabel('ganho [dB]');
subplot(2,1,2); semilogx(freq, phi, '.', w, phase, 'r');
xlim([0.1 10]);
ylabel('fase [graus]'); xlabel('w [rad/s]');
```

```
function [num, den]=bodefit(w, G, F, numDeno)
    sis = tf(numDeno(1), numDeno(2:5));
    [GG,FF]=bode(sis,w); % GG,FF são equivalentes a G,F
    GG=squeeze(GG); FF=squeeze(FF)*pi/180;
    subplot(211); semilogx(w,20*log10(GG),'go',w,20*log10(G),'r');
    ylabel('ganho [dB]');
    subplot(212); semilogx(w,180/pi*FF,'go',w,180/pi*F,'r');
    ylabel('fase [graus]'); xlabel('w [rad/s]')
    drawnow; figure(gcf); % atualiza a tela imediatamente
```

```

xlabel('PRESSIONE QUALQUER TECLA PARA INICIAR O AJUSTE');
figure(gcf);
pause;
options = optimset('Display','off','TolFun',1e-4,'TolX',1e-4);
numDen = fminunc(@(numDen) bodefit_erquad(numDen,w,G,F),numDen,options);
num = numDen(1); den = numDen(2:5); % Retorna os valores

function f=bodefit_erquad(numDen, w, G, F)
w=w(:); G=G(:); F=F(:); % converte em vetores coluna
num = numDen(1);
den = numDen(2:5);
sis = tf(num, den);
S=G.*cos(F) + 1j*G.*sin(F); % converte G e F em fasores.
[GG,FF]=bode(sis,w); % GG,FF são equivalentes a G,F
GG=squeeze(GG); FF=squeeze(FF)*pi/180;
SS=GG.*cos(FF) + 1j*GG.*sin(FF); % SS é equivalente a S
f=norm(SS-S); % erro quadrático
% gráfico de acompanhamento
subplot(211); semilogx(w,20*log10(GG),'go',w,20*log10(G),'r');
ylabel('ganho [dB]');
subplot(212); semilogx(w,180/pi*FF,'go',w,180/pi*F,'r');
ylabel('fase [graus]'); xlabel('w [rad/s]');
drawnow; figure(gcf); % atualiza a tela imediatamente

```

ANEXO A – Resumo das especificações do MPU6050

7.7 Three-Axis MEMS Gyroscope with 16-bit ADCs and Signal Conditioning

The MPU-60X0 consists of three independent vibratory MEMS rate gyroscopes, which detect rotation about the X-, Y-, and Z- Axes. When the gyros are rotated about any of the sense axes, the Coriolis Effect causes a vibration that is detected by a capacitive pickoff. The resulting signal is amplified, demodulated, and filtered to produce a voltage that is proportional to the angular rate. This voltage is digitized using individual on-chip 16-bit Analog-to-Digital Converters (ADCs) to sample each axis. The full-scale range of the gyro sensors may be digitally programmed to ± 250 , ± 500 , ± 1000 , or ± 2000 degrees per second (dps). The ADC sample rate is programmable from 8,000 samples per second, down to 3.9 samples per second, and user-selectable low-pass filters enable a wide range of cut-off frequencies.

7.8 Three-Axis MEMS Accelerometer with 16-bit ADCs and Signal Conditioning

The MPU-60X0's 3-Axis accelerometer uses separate proof masses for each axis. Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially. The MPU-60X0's architecture reduces the accelerometers' susceptibility to fabrication variations as well as to thermal drift. When the device is placed on a flat surface, it will measure 0g on the X- and Y-axes and +1g on the Z-axis. The accelerometers' scale factor is calibrated at the factory and is nominally independent of supply voltage. Each sensor has a dedicated sigma-delta ADC for providing digital outputs. The full scale range of the digital output can be adjusted to $\pm 2g$, $\pm 4g$, $\pm 8g$, or $\pm 16g$.

7.9 Digital Motion Processor

The embedded Digital Motion Processor (DMP) is located within the MPU-60X0 and offloads computation of motion processing algorithms from the host processor. The DMP acquires data from accelerometers, gyroscopes, and additional 3rd party sensors such as magnetometers, and processes the data. The resulting data can be read from the DMP's registers, or can be buffered in a FIFO. The DMP has access to one of the MPU's external pins, which can be used for generating interrupts.

The purpose of the DMP is to offload both timing requirements and processing power from the host processor. Typically, motion processing algorithms should be run at a high rate, often around 200Hz, in order to provide accurate results with low latency. This is required even if the application updates at a much lower rate; for example, a low power user interface may update as slowly as 5Hz, but the motion processing should still run at 200Hz. The DMP can be used as a tool in order to minimize power, simplify timing, simplify the software architecture, and save valuable MIPS on the host processor for use in the application.

7.10 Primary I²C and SPI Serial Communications Interfaces

The MPU-60X0 communicates to a system processor using either a SPI (MPU-6000 only) or an I²C serial interface. The MPU-60X0 always acts as a slave when communicating to the system processor. The LSB of the I²C slave address is set by pin 9 (A0).

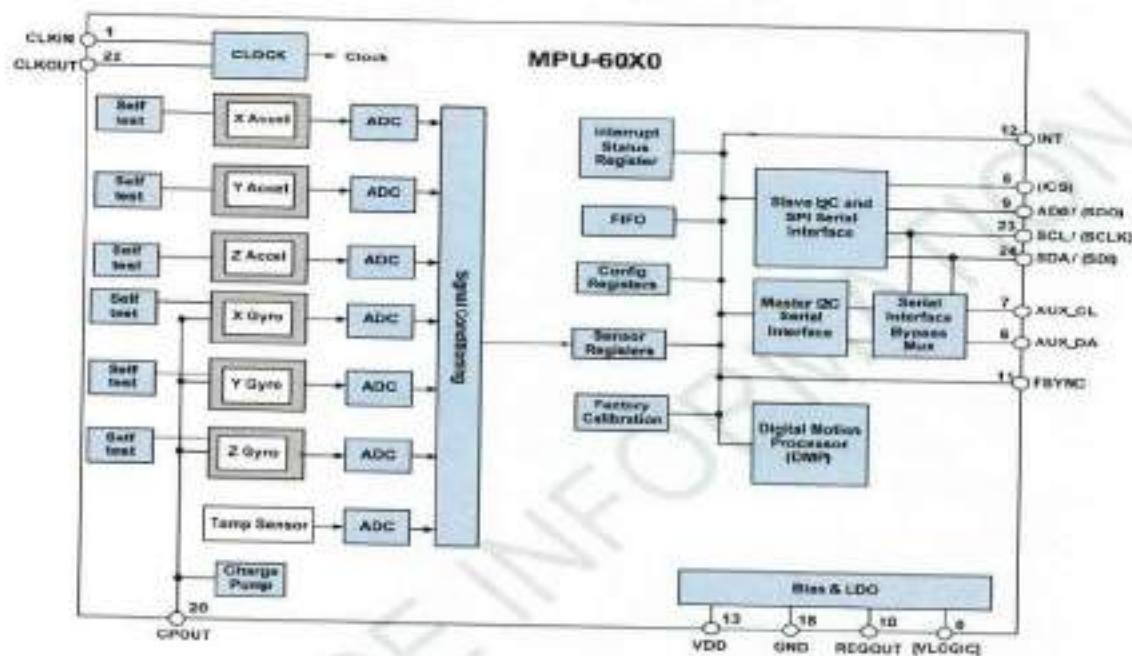
The logic levels for communications between the MPU-60X0 and its master are as follows:

- MPU-6000: The logic level for communications with the master is set by the voltage on VDD
- MPU-6050: The logic level for communications with the master is set by the voltage on VLOGIC

For further information regarding the logic levels of the MPU-6050, please refer to Section 10.

Fonte: MPU-6000/MPU-6050 Product Specification.

ANEXO B – Diagrama de blocos internos do MPU6050



Fonte: MPU-6000/MPU-6050 Product Specification.

ANEXO C – Raspberry Pi - Especificações Técnicas

Specifications [edit]

	Model A	Model B
Target price ^[1]	US\$25 Ext tax (GBP £16 Exc VAT)	US\$35 Ext tax (GBP £22 Exc VAT)
System-on-a-chip (SoC); ^[1]	Broadcom BCM2835 (CPU + GPU. SDRAM is a separate chip stacked on top)	
CPU:	700 MHz ARM11 ARM1176JZF-S core	
GPU:	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1.0, H.264 high-profile encode/decode	
Memory (SDRAM); ^[2]	256 MB (planned with 128 MB, upgraded to 256 MB on 29 Feb 2012); 512 MB (since 15 Oct 2012)	256 MB (until 15 Oct 2012); 512 MB (since 15 Oct 2012)
USB 2.0 ports:	1 (provided by the BCM2835)	2 (via integrated USB hub)
Video outputs; ^[3]	Composite video Composite RCA, HDMI (not at the same time)	
Audio outputs; ^[3]	TRS connector 3.5 mm jack, HDMI	
Audio inputs:	none, but a USB mic or sound-card could be added	
Onboard Storage:	Secure Digital/SD / MMC / SDIO card slot	
Onboard Network; ^[1]	None	10/100 wired Ethernet RJ45
Low-level peripherals:	General Purpose Input/Output (GPIO) pins, Serial Peripheral Interface Bus (SPI), PC, PS/2 ^[2] , Universal asynchronous receiver/transmitter (UART)	
Real-time clock; ^[3]	NONE	
Power ratings:	300 mA, (1.5 W); ^[1]	700 mA, (3.5 W)
Power source; ^[1]	5 V (DC) via Micro USB type B or GPIO header	
Size:	85.0 x 56.0 mm x 15mm	85.0 x 56.0 mm x 17mm
Weight:	31g	40g

Fonte: Embedded Linux Wiki. Disponível em: <http://elinux.org/RPi_Hardware>.