

1. Informações Gerais

1.1. Título do Trabalho:

Rotas adaptativas para inspeção autônoma de subestação por meio do uso de VANT

1.2. Local:

Fundação CERTI

1.3. Orientador local:

Alexandre Marcondes

1.4. Orientador na UFSC:

Professor Ubirajara Franco Moreno

2. Desenvolvimento Previsto

1. Levantamento dos requisitos gerais, funcionais e não funcionais do projeto
2. Desenho da arquitetura do sistema
3. Mapeamento das alternativas de implementação dos módulos do sistema
4. Implementação de um ambiente virtual com VANT simulado na estação de comando
5. Implementação do ambiente de simulação no VANT

3. Desenvolvimento Efetivo

1. Levantamento dos requisitos gerais, funcionais e não funcionais do projeto
2. Desenho da arquitetura do sistema
3. Mapeamento das alternativas de implementação dos módulos do sistema
4. Implementação de um ambiente virtual com VANT simulado na estação de comando
5. Implementação do ambiente de simulação no VANT

4. Problemas Encontrados e Soluções Previstas

1. Levantamento dos requisitos gerais, funcionais e não funcionais do projeto

Elaboração de uma tabela de requisitos

2. Desenho da arquitetura do sistema:

Desenvolvimento de um diagrama conceitual envolvendo os principais módulos do sistema e como este estabelecem uma comunicação. Neste diagrama não estão presentes as tecnologias a serem utilizadas. Os principais módulos do sistema são:

- VANT
- RC (Controle Remoto)
- Gerenciador de voo
- Sistema gerenciador de módulos
- Estação de comando

3. Mapeamento das alternativas de implementação dos módulos do sistema

Levantamento das alternativas de tecnologias para os principais módulos de sistema

- VANT: Para o desenvolvimento e teste inicial optou-se pela utilização do VANT da DJI Inspire 1. A principal razão foi sua disponibilidade, já que a empresa onde está sendo realizado o projeto possui um a pronta disponibilidade. Ao término do desenvolvimento uma especificação para um VANT que atenda as demandas específicas do projeto deverá ser realizada visto o VANT utilizado não é aquisição para o projeto em específico.
- RC: O controle remoto, por consequência das justificativas do primeiro item, será o que vem de fábrica juntamente com o VANT DJI Inspire 1.
- Gerenciador de Voo: O gerenciador de voo neste projeto funcionará como uma ponte entre a estação de comando e o controlador de voo no VANT. Segundo a arquitetura de hardware da DJI o gerenciador de voo deve estar conectado no controle remoto durante a operação por meio de um cabo USB. Para esta opção estão disponíveis para uso, tablets ou celulares rodando sistema Android ou iOS. A SDK da DJI está disponível para Android ou iOS, neste projeto a opção utilizada foi a SDK para Android, devido a afinidade maior com programação na linguagem JAVA e também pelo fato de o sistema Android poder ser implementado em equipamentos mais baratos e genéricos. Neste caso o hardware utilizado foi um Moto C Plus com Android 7.0.0, porém qualquer outro que possuir um sistema Android ao menos na versão 5.0.0 é utilizável. Há também a possibilidade de se utilizar um sistema Android simulado, porém a SDK da SJI funciona bem apenas nas plataformas de arquitetura ARM, que quando simuladas em um ambiente x86 no Android Studio se provam extremamente custosas em

relação ao consumo de processador o que acaba dificultando a cadeia de desenvolvimento.

- Sistema gerenciador de módulos: Na estação de comando a opção de sistemas gerenciador de pacotes foi Robot Operating System (ROS). O ROS dentre os frameworks para robótica foi escolhido por possuir ampla comunidade de desenvolvimento, ser código aberto e possuir diversos módulos de path planning disponíveis, prontos para serem modificados de acordo com as necessidades de projeto. O ROS também não se limita apenas a soluções fechadas, a integração do equipamento em uma camada ROS possibilita que diversos pacotes para diferentes sensores, atuadores, robôs ou algoritmos possam ser integrados a solução, expandindo assim soluções de prateleira como o VANT da DJI. A versão do ROS utilizada será a Kinetic por possuir validação com numerosos pacotes de path planning.
- Estação de comando: Para a estação de comando foi decidido utilizar o notebook GL62M 7REX da MSI por possuir processador gráfico dedicado e poder de processamento suficiente para rodar as rotinas de path planning e visão 3D. Tendo em vista que melhores opções estão disponíveis no mercado, a opção por este notebook é também pela imediata disponibilidade. Ao final do desenvolvimento uma análise de melhorias será realizada de forma a identificar uma configuração que se adapte as especificidades do projeto. O sistema operacional escolhido foi Linux Ubuntu 16.04 por possuir validação com a versão ROS Kinetic.

4. Implementação de um ambiente virtual com VANT simulado na estação de comando

Para o ambiente de simulação foi escolhido o software Gazebo. A escolha por este software foi realizada com base na integração que possui com o framework ROS e por possuir soluções simulação de VANT genéricas implementadas. A instalação do Gazebo é inclusive realizada durante a instalação completa do ROS Kinetic

O Gazebo possui uma API que permite a troca de mensagens entre a camada ROS e o simulador. Os pacotes desenvolvidos para ROS podem utilizar as informações de simulação geradas pelo Gazebo para executar os algoritmos de path planning e experimentar a execução de rotas neste ambiente simulado.

Para a simulação do VANT foi escolhida uma solução de simulação de um VANT com quatro motores que aceita comandos de controle remoto conectados na estação de comando e apresenta uma câmera Kinect para a visualização do mapa, solução esta pode ser consultada em <https://www.wilselby.com/research/ros-integration/3d-mapping-navigation/>. Esta solução vem acompanhada também de uma interface de comandos de posição, que podem ser geradas por scripts internos, não limitando-se exclusivamente ao comando do controle remoto. Esta solução foi escolhida devido a possibilidade de se trabalhar com as opções de controle remoto e comando de posição simultaneamente, possibilitando a coleta de pontos e a posterior execução automática de trajetórias.

A implementação de controle consta de em duas malhas, posição e velocidade. Na malha externa a variável de entrada é posição, a realimentação vem através de um sensor GPS simulado, a saída são as velocidades angulares roll, pitch e yaw e impulso. No controlador interno a variável de entrada são velocidades angulares e impulso, a saída são velocidades dos 4 motores em RPM, a realimentação vem através de uma IMU simulada.

O VANT pode ser controlado com um controle genérico. Neste caso optou-se pela utilização de um controle remoto para vídeo-game que utiliza comunicação USB com a estação de comando. As funções de navegação podem ser ativadas através da tecla “L1” enquanto que a função de comando

manual pode ser ativada através da tecla “O”. Uma vez no modo manual, as funções “takeoff” e “land” podem ser ativadas através das teclas “Δ” e “X” respectivamente.

5. Implementação do ambiente de simulação no VANT

Suítes de simulação para VANT DJI Inspire 1 podem ser executadas em Windows, MacOS, iOS e Android. A solução para a estação de comando está no ambiente Linux, impossibilitando assim simulação direta do VANT na estação de comando. Neste caso a simulação é realizada utilizando o próprio gerenciador de voo, hardware com sistema Android. A SDK DJI para Android possui uma classe que permite a implementação de simulação. A simulação permite a execução das rotas ou comando direto através de controles virtuais ou o próprio controle remoto do VANT. Para realizar a simulação o gerenciador de voo deve estar conectado ao controle remoto da DJI, o VANT e o controle devem ser ligados e o VANT configurado para a posição de pouso.

O programa utilizado para realizar a simulação foi baseado e um exemplo disponível na secção de desenvolvedor no site do fabricante. Modificações foram realizadas de forma a corrigir um problema de compatibilidade de hardware e também para que a apresentação dos valores de orientação e posição seja na tela principal

5. Alterações do Anteprojeto com Justificativa (quando for o caso)