Mauro Sérgio Mafra Moreira

Controle de uma Formação VTNT-VANT Baseado em Espaço Nulo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na linha de pesquisa Robótica, Controle e Automação.

Universidade Federal do Espírito Santo – UFES
Centro Tecnológico
Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Mário Sarcinelli Filho
Coorientador: Prof. Dr. Alexandre Santos Brandão

Vitória, Espírito Santo
2020

Ficha catalográfica disponibilizada pelo Sistema Integrado
de Bibliotecas - SIBI/UFES e elaborada pelo autor

MOREIRA, Mauro Sérgio Mafra Moreira, 1980-
M835c Controle de uma formação VTNT-VANT baseado em espaço nulo / Mauro Sérgio Mafra Moreira. - 2020
91 f. : il.

Orientador: Mário Sarcinelli Filho.
Coorientador: Alexandre Santos Brandão.
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Robótica. 2. Controle de formação de robôs. 3. Espaço Nulo.
4. Modelagem dinâmica de VANT.
5. Controle não linear de VANT. 6. Formação de Robôs
I. Sarcinelli Filho, Mário. II. Brandão, Alexandre Santos
III. Universidade Federal do Espírito Santo. Centro Tecnológico.
IV. Controle de uma Formação VTNT-VANT Baseado em Espaço Nulo

CDU: 621.3

MAURO SÉRGIO MAFRA MOREIRA

CONTROLE DE UMA FORMAÇÃO VTNT-VANT BASEADO EM ESPAÇO NULO

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Engenharia Elétrica.

Aprovada em 28 de fevereiro de 2020.

COMISSÃO EXAMINADORA

Prof. Dr. Mário Sarcinelli Filho
Universidade Federal do Espírito Santo
Orientador

Prof. Dr. Alexandre Santos Brandão
Universidade Federal de Viçosa
Coorientador

Profa. Dra. Eliete Maria de Oliveira Caldeira
Universidade Federal do Espírito Santo

Prof. Dr. Igor Henrique Beloti Pizetta
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo - Campus Aracruz

Prof. Dr. Lucas Vago Santana
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo - Campus Linhares

Aos meus pais, filhos e à minha esposa.

Agradecimentos

Agradeço a todos aqueles que contribuíram direta e indiretamente para a realização deste trabalho, em especial à minha esposa Luciana por ter se dedicado, gerenciado e suportado a árdua sobrecarga de tarefa do núcleo familiar em especial no cuidado dos nossos filhos.

Aos meus tesouros mais valiosos, meus filhos Matheus e Lucas, hoje com 6 e 3 anos respectivamente, agradeço a cada um onde em suas respectivas maneiras souberam respeitar e ajudar. Por várias noites o trabalho de escrita e raciocínio se tornou mais suave ao vê-los. Matheus em seus questionamentos e explicações detalhadas sobre o que papai faz com Drones e Robôs e Lucas que sempre se sentava em meu colo e me dizia que iria ajudar a escrever.

Aos meus pais, irmã e sogros que me apoiaram na decisão de fazer o mestrado, agradeço ao grande suporte proporcionado, principalmente nos momentos de elevada carga de matérias e provas onde a presença e cuidados das crianças foi essencial para o sucesso acadêmico.

Ao professor orientador e amigo Mário Sarcinelli Filho, que esteve presente em todo o caminho percorrido, como uma bússola para o marinheiro, ele sempre me direcionou ao melhor caminho a ser seguido no mar do conhecimento.

Ao professor coorientador e amigo Alexandre Santos Brandão, por me receber em seu laboratório e auxiliar com valiosos conselhos sobre como desenvolver a fase experimental desta dissertação.

A toda a equipe de alunos e ex-alunos dos laboratórios LAI-VIX (Anthony Pinto, Vinícius Pacheco, Harisson Marciano, Diego Bertolani, Marcos Rabelo e Igor Pizetta) e NERO (Valentim Ernandes e Daniel Dourado) onde o trabalho em equipe foi essencial para o desenvolvimento de uma plataforma de comunicação estruturada para operar o Drone Bebop 2 e execução dos testes.

Agradeço também à UFES, e em particular ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela formação de que pude usufruir, e pelo suporte à realização da minha pesquisa para dissertação de mestrado.

Agradeço aos órgãos de fomento à pesquisa, no caso o CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - e a FAPES - Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, pelo suporte financeiro que permitiu a realização deste trabalho. Em particular, agradeço à FAPES pela Bolsa de Mestrado a mim concedida.

A todos que não citei, afinal são muitos nomes, também serei sempre grato.

Mauro Sérgio Mafra Moreira

"Quanto mais aumenta nosso conhecimento, mais evidente fica nossa ignorância."
John F. Kennedy

Resumo

Este trabalho propõe um controlador para guiar dois robôs, um de tração diferencial e o outro aéreo, em uma tarefa de posicionamento colaborativo. O projeto adota o paradigma de estrutura virtual, consistindo na estrutura virtual uma reta imaginária que interliga os dois robôs. Usando a técnica de espaço nulo a tarefa de controle é dividida em duas subtarefas, manter a forma e movê-la. Neste contexto, a formação é representada por um conjunto de variáveis de formação, que são derivadas das posições dos robôs. As variáveis de formação são agrupadas em dois conjuntos, um associado à forma e outro à posição da estrutura virtual. Essa técnica permite estabelecer maior prioridade a uma das subtarefas do conjunto, como, por exemplo, manter a forma da formação, deixando a subtarefa de mover a estrutura virtual com menor prioridade. Embora também se possa atribuir maior prioridade ao movimento da formação em detrimento de manter sua forma, para o propósito deste trabalho, que consiste em fazer o robô aéreo pousar sobre o terrestre, o que corresponderia à etapa final de uma entrega de pacote, por exemplo, os resultados experimentais aqui apresentados mostram que atribuir maior prioridade à manutenção da forma da formação é mais vantajoso, já que se está visando aplicações que exigem conhecer bem a posição do robô aéreo em relação ao terrestre ao longo do tempo, permitindo ao veículo aéreo pousar no terrestre.

Abstract

This work proposes a controller to autonomously guide a formation of two mobile robots, a differential drive wheeled one and an aerial one, in tasks of positioning. The controller is designed considering the paradigm of virtual structure, in which the virtual structure is the imaginary tridimensional straight line linking the two robots, and adopting the technique of null space, to split the whole control task in two sub-tasks, which are keeping and moving the formation. To do that the formation is described by a set of formation variables, which are derived from the individual robot positions. To apply the null-space technique, the formation variables are grouped into two sets, one associated to the shape of the virtual structure and the other associated to the position of the virtual structure. The use of the null space technique allows giving higher priority to the sub-task of keeping the formation shape, leaving the sub-task of moving the virtual structure to a new position with a lower priority, thus generating a rigid formation. Although the reverse situation, to give higher priority to the movement of the formation to a new position, generating a flexible formation, is also possible. However, simulated and experimental results have shown that to give higher priority to keep the shape of the formation is the best solution. The reason is this project is aiming at applications demanding that the position of the aerial robot with respect to the ground one be well known along time, allowing the aerial vehicle to land over the terrestrial one, which would be, for instance, the last phase of a package delivery mission.

Sumário

1 INTRODUÇÃO . . . 25
1.1 Definição do Problema . . . 27
1.2 Motivação . . . 28
1.3 Objetivos . . . 28
1.4 Estado da arte . . . 29
1.5 Hipótese Principal . . . 30
1.6 Estrutura do Trabalho . . . 30

2 ARQUITETURA . . . 33
2.1 Robôs Utilizados . . . 33
2.1.1 O Robô *Pioneer 3-DX* e seu Modelo Cinemático . . . 33
2.1.2 Modelo Dinâmico e Controlador do *Pioneer 3-DX* . . . 35
2.1.3 O Drone *Bebop 2* e seu Modelo Cinemático . . . 36
2.1.4 Modelagem Dinâmica e Controlador do *Bebop 2* . . . 38
2.1.5 Identificação dos parâmetros do modelo dinâmico do *Bebop 2* . . . 39
2.1.6 Validação do Modelo Dinâmico do *Bebop 2* . . . 41
2.2 Arquitetura Física . . . 43
2.2.1 Dimensões do Espaço de Teste . . . 43
2.2.2 Controle de Luminosidade . . . 43
2.3 Arquitetura de *Hardware* . . . 44
2.3.1 Visão Geral . . . 44
2.3.2 O sistema de Captura de Movimento *OptiTrack* . . . 44
2.3.3 Estrutura de Rede . . . 45
2.4 Arquitetura de *Software* . . . 45
2.4.1 O *Software Motive* . . . 45
2.4.2 ROS - *Robot Operating System* . . . 47
2.4.3 *Driver Bebop_Autonomy* . . . 48
2.4.4 A Plataforma *Aurora* . . . 49
2.4.5 *Driver* de Comunicação Aurora - ROS . . . 49

3 FORMAÇÕES DE MÚLTIPLOS ROBÔS . . . 51
3.1 Introdução . . . 51
3.1.1 Motivações . . . 51
3.2 Formação Planar (2D) . . . 52
3.3 Formação Espacial (3D) . . . 54

4 CONTROLE DE FORMAÇÃO . . . 57
4.1 Filosofia de Controle . . . 57
4.2 Sistema de Controle Multicamadas . . . 58
4.3 O Controle da Formação . . . 61

4.3.1 Controlador Clássico 61
4.3.2 Controle Baseado em Espaço Nulo 62
4.4 O Controlador Proposto 63

5 RESULTADOS 65
5.1 Comparação de Desempenho de Controladores 65
5.2 Pouso do VANT sobre o VTNT 69

6 CONCLUSÕES 71

REFERÊNCIAS 73

APÊNDICE A – *CONCEITOS BÁSICOS DE SISTEMAS LINEARES* 77
A.1 Escalonamento de Matrizes 77
A.1.1 Conceito Geral 77
A.1.2 Matriz Escalonada na Forma Reduzida 77
A.2 Independência e Dependência Linear 78
A.3 Posto de uma Matriz 78
A.4 Nulidade de uma Matriz 79
A.5 Matriz Identidade 79
A.6 Matriz Transposta 79
A.7 Matriz Inversa 80
A.8 Matriz Pseudo Inversa 80

Lista de ilustrações

Figura 1 – Robôs Terrestres . . . 25
Figura 2 – Tipos de VTNT . . . 26
Figura 3 – Tipos de VANT . . . 27
Figura 4 – VANTs - Veículos Aéreos Não Tripulados . . . 28
Figura 5 – Robô *Pioneer 3-DX* e caracterização da sua cinemática . . . 34
Figura 6 – O drone *Bebop 2* . . . 37
Figura 7 – Sinal aplicado ao *Bebop 2* para obtenção do modelo dinâmico simplificado . . . 39
Figura 8 – Parâmetros calculados para o modelo dinâmico do *Bebop 2* . . . 40
Figura 9 – Sinal de validação do modelo . . . 41
Figura 10 – Validação do modelo obtido . . . 42
Figura 11 – Tarefa de Posicionamento em Formação de Quadrimotores . . . 42
Figura 12 – Erros da tarefa de posicionamento da formação de dois drones *Bebop 2* . . . 43
Figura 13 – Câmeras usadas no sistema *OptiTrack* . . . 45
Figura 14 – Vista do local dos experimentos, veículos utilizados e algumas câmeras do *OptiTrack* . . . 46
Figura 15 – Topologia de hardware . . . 46
Figura 16 – Motive . . . 47
Figura 17 – Arquitetura de comunicação básica do ROS . . . 48
Figura 18 – Arquitetura de comunicação do *driver bebop_autonomy* . . . 49
Figura 19 – *Driver* de comunicação Aurora - ROS . . . 50
Figura 20 – Formação planar (2D). . . . 52
Figura 21 – Formação espacial (3D) . . . 55
Figura 22 – Diagrama de controle detalhado. . . . 58
Figura 23 – Fluxograma correspondente ao sistema de controle multicamadas . . . 59
Figura 24 – Controle baseado em espaço nulo . . . 63
Figura 25 – Controle baseado em espaço nulo com prioridade de forma . . . 64
Figura 26 – Diagrama de blocos para o controlador proposto, que atribui maior prioridade ao controle da forma da formação. . . . 64
Figura 27 – Percursos dos robôs no Controle Convencional. . . . 66
Figura 28 – Percursos dos robôs no Controle NSB com Prioridade de Forma. . . . 66
Figura 29 – Percursos dos robôs no Controle NSB com Prioridade de Posição. . . . 67
Figura 30 – Erros de formação para o Controle Convencional. . . . 67
Figura 31 – Erros de formação para o Controle NSB com Prioridade de Forma. . . . 68
Figura 32 – Erros de formação para o Controle NSB com Prioridade de Posição. . . . 68
Figura 33 – Pouso do VANT sobre o VTNT usando o Controle Convencional . . . 70
Figura 34 – Pouso do VANT sobre o VTNT usando o Controle NSB com Prioridade de Forma . . . 70

Lista de Abreviaturas

VANT	Veículo Aéreo Não Tripulado
VTNT	Veículo Terrestre Não Tripulado
UAV	do Inglês, *Unmanned Aerial Vehicle*
UGV	do Inglês, *Unmanned Ground Vehicle*
NSB	do Inglês, *Null Space Behavior*
GPS	do Inglês, *Global Position System*
PoE	do Inglês, *Power Over Ethernet*
CPU	do Inglês, *Central Processing Unit*
GPU	do Inglês, *Graphics Processing Unit*
SDK	do Inglês, *Software Development Kit*
ROS	do Inglês, *Robot Operating System*
LED	do Inglês, *Light Emitting Diode*

Lista de Símbolos

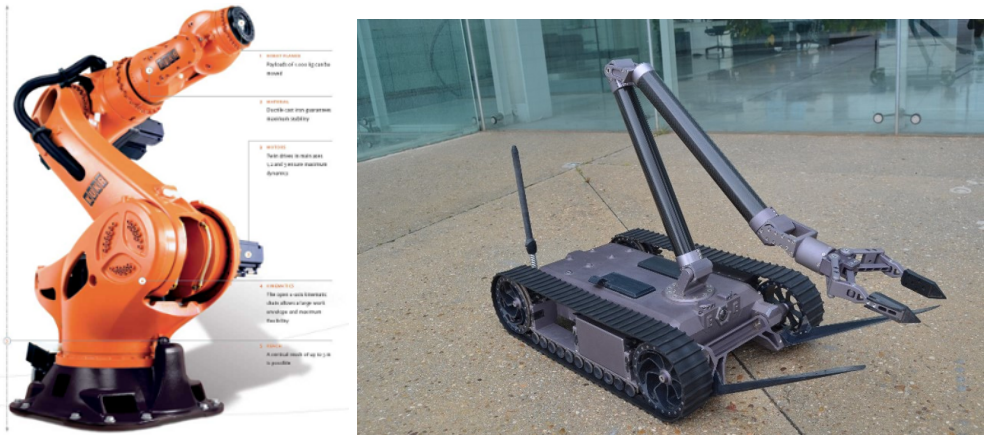
x	Posição de um veículo em relação ao eixo-x
y	Posição de um veículo em relação ao eixo-y
z	Posição de um veículo em relação ao eixo-z (Altitude)
ϕ	Arfagem - Orientação de um veículo ao Redor do eixo x
θ	Rolagem - Orientação de um veículo ao Redor do eixo y
ψ	Guinada - Orientação de um veículo ao Redor do eixo z
$\dot{x}$	Velocidade linear de um veículo em relação ao eixo-x
$\dot{y}$	Velocidade linear de um veículo em relação ao eixo-y
$\dot{z}$	Velocidade linear de um veículo em relação ao eixo-z (Altitude)
$\dot{\phi}$	Velocidade angular de um veículo ao redor do eixo x
$\dot{\theta}$	Velocidade angular de um veículo ao redor do eixo y
$\dot{\psi}$	Velocidade angular de um veículo ao redor do eixo z
$\ddot{x}$	Aceleração linear de um veículo em relação ao eixo-x
$\ddot{y}$	Aceleração linear de um veículo em relação ao eixo-y
$\ddot{z}$	Aceleração linear de um veículo em relação ao eixo-z (Altitude)
$\ddot{\phi}$	Aceleração angular de um veículo ao redor do eixo x
$\ddot{\theta}$	Aceleração angular de um veículo ao redor do eixo y
$\ddot{\psi}$	Aceleração angular de um veículo ao redor do eixo z
$\mathbf{X}$	Vetor dos estados de posição do veículo
$\dot{\mathbf{X}}$	Vetor dos estados de velocidade do veículo
$\ddot{\mathbf{X}}$	Vetor dos estados de aceleração do veículo
$\mathbf{q}$	Vetor dos estados de posição da formação
$\dot{\mathbf{q}}$	Vetor dos estados de velocidade da formação
$\ddot{\mathbf{q}}$	Vetor dos estados de aceleração da formação
ρ	Erro de distância entre o VANT e o VTNT
α	Erro de orientação entre o VANT e o VTNT no plano XY
β	Erro de orientação entre o VANT e o VTNT na projeção de ρ plano XY
$\mathbf{u}$	Vetor de sinais de controle
f_1, f_2	Matrizes de parâmetros dinâmicos do modelo do quadrimotor

1 Introdução

Uma definição oficial para o termo robô é dada pelo Robot Institute of America (RIA) (SPONG; HUTCHINSON; VIDYASAGAR, 2006): “Um robô é um manipulador reprogramável, multifuncional, projetado para movimentar material, ferramentas ou dispositivos especializados através de movimentos programáveis variados para desenvolver uma variedade de tarefas”.

Esta definição formal traz o primeiro contato com o conceito de robótica e apresenta algumas informações chave, como a ideia de programar atividades e movimentos, conceitos os quais serão trabalhados posteriormente nesta dissertação. Aproximando o conceito de robótica ao contexto de aplicação alvo neste projeto, pode-se desenvolver a definição de robôs segundo aspectos construtivos e de aplicação.

Do ponto de vista construtivo, pode-se classificar os robôs como terrestres ou aéreos. Os robôs terrestres podem ser do tipo fixos, como os braços robóticos (mais frequentemente chamados robôs manipuladores) amplamente utilizados na indústria automobilística ou veículos móveis que podem ter ou não um manipulador robótico embarcado. Braços Robóticos são muito utilizados em tarefas de precisão, que exigem alto grau de repetibilidade e área restrita de atuação. Normalmente são robustos e são alimentados continuamente por uma fonte de energia tipicamente de natureza elétrica ou hidráulica. Vide Figura 1



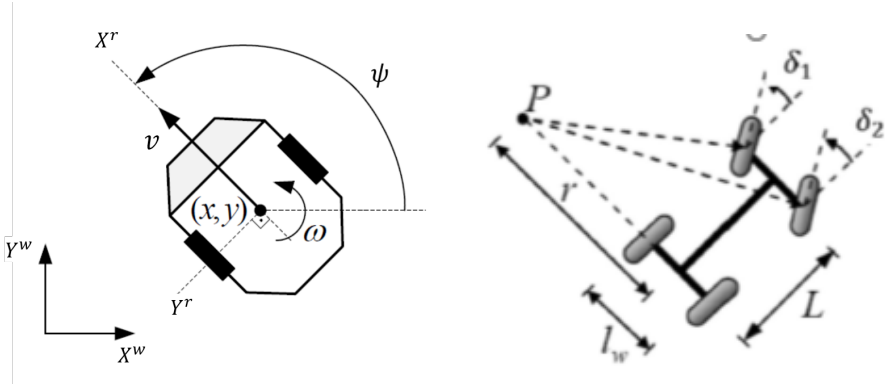
(a) Manipulador Robótico Fixo (b) VTNT Com Manipulador Robótico

Figura 1 – Robôs Terrestres

Veículos Terrestres Não Tripulados(VTNTs) possuem maior versatilidade e atuam em espaços mais amplos. Contudo, normalmente são dependentes de fontes de energia embarcadas, usualmente baterias elétricas recarregáveis. Apesar do consumo energético não ser exacerbado como em veículos aéreos do tipo helicóptero, a fonte de energia limitada impõe restrições construtivas as quais devem ponderar pela autonomia do veículo.

Os VTNTs podem ser do tipo de tração diferencial ou *car-like*. No primeiro caso, de construção mais simples, são usadas duas rodas fixas para tração e uma roda castor, sendo esta apenas para estabilização. Este tipo de veículo possui a capacidade de girar em torno do seu próprio eixo, conforme a Figura 2 (a) esta característica lhe permite alto grau de manobrabilidade e aliada a um artifício matemático, no qual o ponto de controle é deslocado do eixo das rodas, possibilita seu deslocamento em qualquer direção desejada no plano. Veículos do tipo *car-like* possuem maior estabilidade e usualmente maior capacidade de carga comparados com veículos do tipo tração diferencial. Contudo, não são capazes de girar em torno do seu próprio eixo, restrição

esta que impõe a necessidade de maiores áreas de manobras e um raio mínimo de curvatura (vide distância r na Figura 2 (b)), e, caso ele seja atingido, será necessário maior tempo e complexidade para a manobra do veículo.



(a) Robô de Tração Diferencial (b) Robô do Tipo Carro (*car-like*)

Figura 2 – Tipos de VTNT

Veículos Aéreos Não Tripulados (VANTs) podem ser classificados basicamente em dois tipos, veículos de asas fixas (aviões e planadores) e de asas rotativas (helicópteros e multimotores, estes mais conhecidos como drones). Cada um destes tipos de veículo possui pontos positivos e negativos, os quais devem ser analisados para melhor atender as necessidades de utilização.

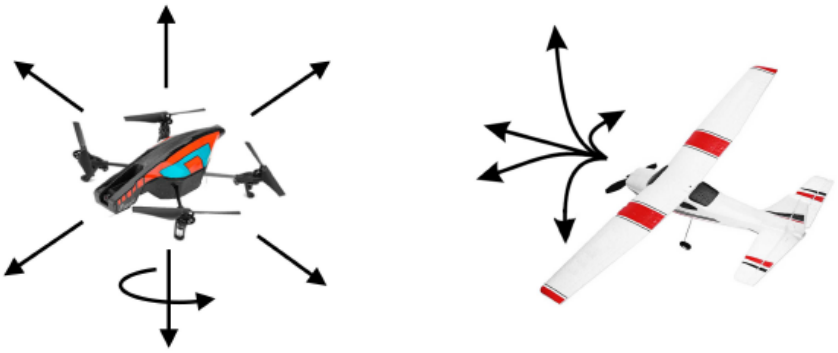
VANTs de asas fixas, geralmente possuem maior autonomia de voo e maior capacidade de carga, visto que sua sustentação em voo é realizada pelas asas fixas. O empuxo proporcionado pela diferença de pressão do ar acima e abaixo das asas é a principal fonte de sustentação da aeronave em voo, os propulsores são mais exigidos na decolagem, retomadas de velocidade e mudança de altitude. Ao se atingir a velocidade e altitude desejadas, utiliza-se uma parte reduzida da potência total dos motores para manutenção da aeronave no ar.

O ganho em autonomia e carga é contrabalanceado pela restrição de movimento que esta aeronave pode executar. Para que ela seja sustentada no ar ou tenha movimentos laterais, obrigatoriamente ela deve estar se deslocando para a frente, característica esta que restringe sua utilização em locais confinados ou com pouco espaço de manobra, sendo mais indicadas para utilização em longas distâncias. Vide Figura 3 item (b).

VANTs de asas rotativas, em geral possuem uma autonomia de voo inferior em relação aos VANTs de asas fixas, e sua capacidade de carga também é menor, pois a sustentação da aeronave é realizada pelas pás rotativas dos motores, os quais devem estar em funcionamento durante todo o tempo de voo, provendo o empuxo necessário para sustentação e movimentação. Em contrapartida, este modelo de aeronave apresenta alta versatilidade e capacidade de manobra, sendo mais indicados para utilização em ambientes com restrição de espaço ou que necessitem de precisão de posicionamento em baixas velocidades. São aeronaves versáteis para uso e o desenvolvimento das fontes de energia, principalmente a capacidade de carga das baterias, tem contribuído consideravelmente para o aumento de sua utilização. Vide Figura 3 item (a).

Mais detalhes sobre as características dos VANTs podem ser encontrados em (BRANDÃO; SARCINELLI-FILHO; CARELLI, 2013a)

Analisando os robôs do ponto de vista de aplicação, eles são comumente utilizados em condições onde há risco à vida humana, em tarefas repetitivas ou tarefas que requerem uma grande precisão em sua execução. Neste contexto, tem-se uma ampla gama de situações onde a utilização de robôs pode ser encontrada. Neste trabalho será dado enfoque no controle coordenado de robôs distintos que devem colaborar um com outro de



(a) VANTs de Asas Rotativas (b) VANTs de Asas Fixas

Figura 3 – Tipos de VANT

forma a obter melhores condições para movimentação de carga.

Este cenário é principalmente vislumbrado em empresas de logística, onde objetos devem ser manipulados constantemente ou até mesmo entregues de forma autônoma ao consumidor final. Outro exemplo onde a movimentação de carga pode ser visualizada é a inspeção de tubulações através de sensores que devem ser posicionados em vários pontos da estrutura de forma que haja contato físico com a mesma.

Tais aplicações normalmente envolvem posicionamento de objetos em locais de altitude elevada, condições insalubres e de alta repetibilidade de movimentos, próximo a gases tóxicos ou até mesmo em ambientes com risco de explosão. Em tais situações a utilização de robôs em substituição à mão de obra humana representa uma solução atrativa às empresas, pois os gastos com possíveis acidentes de trabalho superam em muito o investimento em automação.

Outro fator que corrobora com a robotização nas áreas citadas é a redução dos custos de operação, aliada aos ganhos de escala de produção, pois os robôs especializados podem executar tarefas repetitivas com alto grau de eficiência.

A utilização de VANTs e VTNTs em movimentação de carga e entrega de objetos tem ganhado expressivo progresso na robótica, a melhoria nas fontes energéticas que os alimentam propiciam maior autonomia e capacidade de carga. Este cenário aumenta a aplicabilidade de robôs nas áreas citadas.

As tarefas de movimentação de carga requerem um controle apurado, o qual deve garantir as melhores condições para que o objeto seja retirado de um ponto e posicionado em outro. Neste contexto, esta dissertação analisa alternativas de controle que possibilitem a melhor condição para que um VANT movimente um objeto de forma cooperativa com um VTNT.

1.1 Definição do Problema

A utilização de drones para inspeção e movimentação de carga está se tornando uma realidade. Melhorias construtivas e o desenvolvimento da capacidade computacional dos equipamentos têm viabilizado a tendência crescente de sua utilização a Figura 4 itens (a) e (b) ilustram exemplos de utilização de drones .

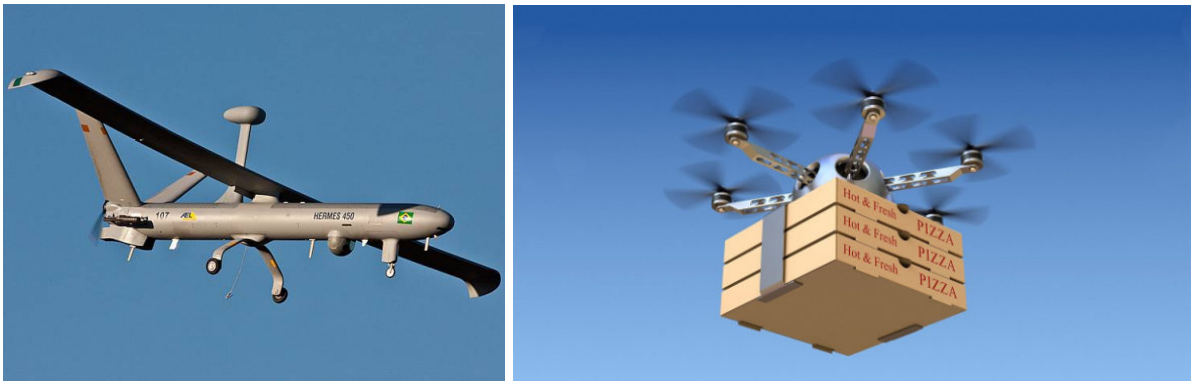
Empresas como VALE, Petrobrás, Shell, e de logística, como Amazon e DHL, têm intensificado suas pesquisas nesta área. Entretanto, a utilização de robôs para áreas industriais e logística ainda representa um

grande desafio, especialmente quando envolve o transporte de objetos.

Devido à autonomia limitada dos drones, intensificada pelo peso extra correspondente à carga, técnicas de otimização de controle, colaboração e movimentação são buscadas, para melhoria de desempenho e utilização dos robôs. A utilização de uma formação heterogênea, composta de robôs aéreos e terrestres, aliada à otimização do controle colaborativo formam uma solução que viabiliza o aumento da capacidade de movimentação de carga.

No cenário de entrega de objetos, uma possibilidade a ser considerada seria descrita como um robô terrestre transportar o drone e sua carga até um ponto próximo de onde será realizada a entrega do objeto. Uma vez que o robô terrestre esteja posicionado, o drone pode decolar e coletar a carga localizada no robô terrestre a fim de entregá-la a seu destino final, necessitando dessa forma uma menor quantidade de energia para realização da entrega.

Para que esta solução seja viável, é desejável que o drone se mantenha posicionado sobre o veículo terrestre da forma mais rápida e estável possível, provendo a melhor condição para que o objeto seja coletado ou entregue ao robô terrestre. É justamente este comportamento que esta pesquisa se propõe a otimizar.



(a) VANT de Asas Fixas (b) VANT de Asas Rotativas

Figura 4 – VANTs - Veículos Aéreos Não Tripulados

1.2 Motivação

Há várias razões que motivam pesquisas nesta área. Algumas delas, estritamente relacionadas ao escopo desta pesquisa são listadas a seguir:

- A utilização de robôs para movimentação de objetos ainda representa um grande desafio.
- Tem-se um mercado emergente em expansão e promissor para utilização de robôs no contexto de movimentação de carga.
- O setor logístico é estratégico, de forma que propostas de redução de tempo e aumento da confiabilidade de transporte representam ganhos potenciais para empresas do setor.

1.3 Objetivos

O objetivo desta dissertação é desenvolver um controlador baseado em comportamento, utilizando a técnica de espaço nulo que provenha as melhores condições para entrega/coleta de objetos em uma formação

heterogênea composta por um robô terrestre e um aéreo. Um exemplo seria o drone entregar/coletar um pacote em uma região localizada acima do robô terrestre, a proposta também contempla a otimização das condições de pouso de um drone sobre um robô terrestre.

1.4 Estado da arte

Nesta seção se discutem alguns trabalhos analisados durante o desenvolvimento desta pesquisa. Dessa forma se pode-se ter uma visão geral sobre o que está sendo estudado na área de controle de formação de robôs.

Estes estudos se iniciam na década de 80, conforme mencionado em (PARKER, 2016), onde o controle do movimento coordenado de robôs é citado. A cooperação entre robôs aéreos e terrestres se inicia com o mapeamento de áreas e geoposicionamento através da fusão de imagens (ver (SURMANN; BERNINGER; WORST, 2017)).

Em aplicações mais recentes, os VANTs vêm sendo utilizados para movimentação de objetos (MASONE; BÜLTHOFF; STEGAGNO, 2016; BAI; WEN, 2010; PIZETTA; BRANDÃO; SARCINELLI-FILHO, 2019b). Empresas como Amazon e DHL têm investido consideravelmente nesta área, a qual ainda representa um grande desafio, pois é necessário aliar autonomia dos equipamentos com o acréscimo de massa propiciado pelo objeto a ser transportado (VILLA; BRANDÃO; SARCINELLI-FILHO, 2019) (PIZETTA; BRANDÃO; SARCINELLI-FILHO, 2019a).

Neste contexto, soluções que reduzam o tempo destinado ao posicionamento aéreo auxiliam positivamente a quantidade de produtos manipulados, logo o estudo de controladores que otimizem a coleta e entrega de objetos é fundamental para que seu transporte seja realizado de uma forma eficiente.

Uma das formas de melhorar esta eficiência é utilizar modelos matemáticos que descrevem a dinâmica dos robôs de forma mais realista. Modelos dinâmicos aproximados podem ser obtidos ao se comparar um sinal aplicado ao robô com sua respectiva resposta. Esta relação entre os sinais de entrada e saída do robô permitem estimar parâmetros que representem a dinâmica do robô (SANTOS et al., 2017).

Vários níveis de detalhamento podem ser aplicados à modelagem dinâmica de VANTs, como discutido em (SANTOS, 2017), (SANTOS et al., 2017), (SANTANA et al., 2014) e (SANTANA, 2016). Nesta dissertação de mestrado estima-se o modelo dinâmico segundo as diretrizes descritas em tais referências, onde um modelo dinâmico de alto nível é obtido para o VANT AR.Drone 2.0, da Parrot, Inc. O detalhamento matemático desta modelagem é explicado no Capítulo 2, e será aplicado para a identificação dos parâmetros referentes ao drone Bebop 2, também da Parrot, Inc., que também será utilizado nesta dissertação.

Da mesma forma como realizado para a modelagem do VANT pode-se obter o modelo dinâmico do VTNT a partir de um sinal de controle conhecido e compará-lo com sua respectiva resposta, conforme é feito em (MARTINS, 2009). Neste caso particular, o VTNT utilizado, assim como seu modelo dinâmico, é o mesmo utilizado em **??** e em (MARTINS, 2009). Em particular, a identificação dos parâmetros do VTNT, um Pioneer 3-DX, da Mobile Robots Inc., foi realizada em (MARTINS; SARCINELLI-FILHO; CARELLI, 2017), e os parâmetros ali obtidos foram utilizados nesta dissertação.

Estimar de forma correta o modelo dinâmico dos robôs utilizados é o primeiro passo para melhoria no controle. Contudo, ainda se pode adotar estratégias de controle mais complexas, as quais afetam o comportamento dos robôs atuando em formação. Técnicas baseadas em comportamento são utilizadas em uma grande variedade de situações como estratégias de controle. Especificamente nesta dissertação, se utiliza a abordagem baseada em espaço nulo para modificar a forma como os robôs interagem uns com os outros, para manter a forma da estrutura virtual que caracteriza a formação e para mover a formação como um todo.

Aplicações desta abordagem de controle são encontradas em (ANTONELLI; ARRICHIELLO; CHIAVERINI, 2010), (CHIAVERINI; MEDDAHI, 2015), (ROSALES et al., 2016) e (MANSFELD et al., 2017), apenas para citar algumas delas. Portanto, o controle da formação se baseará na abordagem de estrutura virtual (LEWIS; TAN, 1997; ANTONELLI; CHIAVERINI, 2006; KITTS; MAS, 2009; ADAMEK; KITTS; MAS, 2015; BRANDÃO et al., 2015), com utilização da técnica de espaço nulo para hierarquização dos comportamentos envolvidos (ANTONELLI; ARRICHIELLO; CHIAVERINI, 2010; ROSALES et al., 2016).

É nesse contexto que esta dissertação discute o controle baseado em espaço nulo de uma formação de dois veículos (formação em linha) sendo um deles uma plataforma móvel terrestre do tipo tração diferencial (modelo Pioneer 3-DX) e o outro um quadrimotor (drone Bebop 2), navegando para posicionar-se em uma posição pre-definida, de forma que o drone possa pousar, no momento desejado, sobre a plataforma terrestre, emulando o encerramento de uma missão de entrega de pacotes.

Ao longo do desenvolvimento da pesquisa, os resultados parciais obtidos foram relatados em artigos publicados em conferências, uma internacional e outra nacional, a saber:

- Mauro Sérgio Mafra Moreira, Alexandre Santos Brandão e Mário Sarcinelli Filho, Null Space Based Formation Control for a UAV Landing on a UGV. Em: 2019 International Conference on Unmanned Aircraft Systems (ICUAS'19), Atlanta, GA, USA, Junho de 2019, pp. 1389-1397. (DOI: 10.1109/ICUAS.2019.8797820)

- Mauro Sérgio Mafra Moreira, Alexandre Santos Brandão, Mário Sarcinelli Filho e Sara Jorge e Silva, Controle de uma Formação VTNT-VANT Baseado em Espaço Nulo. Em: 14o. Simpósio Brasileiro de Automação Inteligente (SBAI 2019), Ouro Preto, MG, Brasil, Outubro de 2019. (DOI: 10.17648/sbai-2019-111245)

1.5 Hipótese Principal

De forma condensada, pode-se descrever a hipótese investigada neste trabalho como sendo: é viável utilizar o controle de formação em linha de um VTNT e um VANT, adicionando-se a técnica de espaço nulo, para assegurar o pouso do VANT sobre o VTNT no contexto da parte final de uma tarefa de entrega de pacotes, quando o VANT deve retornar à sua base.

Para concluir pela veracidade dessa hipótese, todo o desenvolvimento teórico é validado por resultados simulados e experimentais, apresentados no decorrer do texto, conforme brevemente descrito na próxima seção.

1.6 Estrutura do Trabalho

A organização desta dissertação segue a estrutura abaixo:

Capítulo 1: Introdução

Neste capítulo são descritos os tipos mais comuns de robôs existentes, é realizado um breve comentário sobre as vantagens e desvantagens dos modelos citados bem como é feita a apresentação do problema relativo à movimentação de carga, contexto em que o problema tratado se insere.

Capítulo 2 Arquitetura

Neste capítulo são descritas as arquiteturas lógica e física utilizadas nas simulações e controle dos robôs.

Também são detalhados os equipamentos utilizados e as características que auxiliam a replicação do caso de estudo desta dissertação.

Capítulo 3: Formação de Robôs

Neste capítulo é descrito o conceito de formação, ou sistemas de múltiplos robôs. O capítulo se inicia por uma abordagem mais simples correspondente à formação em duas dimensões (2D), apta apenas a navegar no plano do chão, e, após esta descrição, o conceito é extrapolado para a abordagem de formação em três dimensões (3D), apta a navegar no espaço.

Capítulo 4: Controle de Formação

Neste capítulo é apresentada a estratégia de controle proposta para o problema descrito no Capítulo 1

Capítulo 5: Resultados

Este capítulo apresenta os resultados de simulação e experimentais utilizando a estrutura de controle proposta.

Capítulo 6: Conclusões e Trabalhos Futuros

Este capítulo elenca as principais conclusões do trabalho realizado, assim como propostas para continuação da pesquisa.

2 Arquitetura

Este capítulo descreve o que foi utilizado, dos pontos de vista físico e lógico, para desenvolver esta pesquisa. Após sua leitura o leitor deve ser capaz de compreender o ambiente de teste e replicá-lo, caso necessário. Serão explicados dados técnicos dos robôs, arquitetura de *hardware*, *software*, interfaces desenvolvidas e ambiente físico utilizado.

2.1 Robôs Utilizados

Foram utilizados dois robôs atuando de forma cooperativa em uma formação heterogênea. Mais precisamente, um robô terrestre do tipo uniciclo Pioneer 3-DX, da Mobile Robots, Inc., e um drone modelo Bebop 2, da Parrot Drones SAS, cujas características principais serão detalhadas nas próximas seções.

2.1.1 O Robô *Pioneer 3-DX* e seu Modelo Cinemático

O Pioneer 3-DX é um robô terrestre do tipo uniciclo, o qual possui duas rodas de tração independentemente acionadas, em suas laterais, e uma roda castor para apoio da estrutura, em sua parte traseira. Esta configuração permite que ele se desloque para a frente ou para trás (na direção x^r do seu sistema de coordenadas), ou gire em torno do seu próprio eixo vertical (z^r), ou ainda se move combinando essas duas possibilidades. Ou seja, tal robô possui dois graus de liberdade. Ele é um robô versátil, e pode ser configurado com sensores conforme necessidade do pesquisador. Em geral, ele já vem equipado com sensores nas rodas do tipo *encoder*, que geram informações de deslocamento nas direções x e y do sistema de coordenadas global, considerando-se como origem de tal sistema de coordenadas o ponto onde o robô está ao ser ligado. Através da eletrônica interna embarcada, tal robô é acionado quando se lhe fornece comandos de velocidade linear em seu eixo x^r e angular em torno do seu eixo z^r (SANTANA; BRANDÃO; SARCINELLI-FILHO, 2016; MARTINS; SARCINELLI-FILHO; CARELLI, 2017).

Há modelos do Pioneer 3-DX os quais possuem um computador embarcado, nesse caso o usuário executa a lógica de comando diretamente no robô utilizando a biblioteca de interface do fabricante. Contudo, o modelo utilizado nesta pesquisa não possui tal computador a bordo, dessa forma um *hardware* externo foi adicionado o qual recebe e envia comandos através de uma porta de comunicação serial RS-232. Duas opções de *hardware* embarcados foram utilizados em momentos distintos da pesquisa: um *notebook* rodando o sistema operacional *Windows 7* e uma placa *Raspberry PI 3 B+* rodando o sistema operacional *Raspbian*, baseado em *Linux Ubuntu*. Para o escopo desta pesquisa a as duas soluções apresentaram resultados idênticos.

Os dados lidos do robô são expressos pelo vetor

$$\mathbf{u} = \begin{bmatrix} v & \omega \end{bmatrix}^T, \tag{2.1}$$

onde v e ω são, respectivamente, a velocidade linear e angular instantâneas, tomadas a cada intervalo de 100 ms, que é o período de amostragem padrão do robô.

Da mesma forma, os sinais de controle enviados ao robô são expressos pelo vetor

$$\mathbf{u}_d = \begin{bmatrix} v_d & \omega_d \end{bmatrix}^T, \tag{2.2}$$

onde v_d e ω_d são, respectivamente, a velocidade linear e angular desejadas.

O modelo cinemático de um robô do tipo uniciclo, como o Pioneer 3-DX, é descrito por

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{A}_P \begin{bmatrix} \upsilon \\ \omega \end{bmatrix}, \tag{2.3}$$

onde $\begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix}^T$ é o vetor de velocidades nos eixos x e y globais, ψ é o ângulo de orientação do robô, definido como o ângulo entre os eixos x global e x^r do robô, $\begin{bmatrix} \upsilon & \omega \end{bmatrix}^T$ é o vetor de velocidades linear e angular instantâneas e $a > 0$) é a distância do ponto de controle $\mathbf{h}$ ao centro do eixo virtual que une as rodas de tração do veículo (ver Figura 5(b)). Por sua vez, a matriz

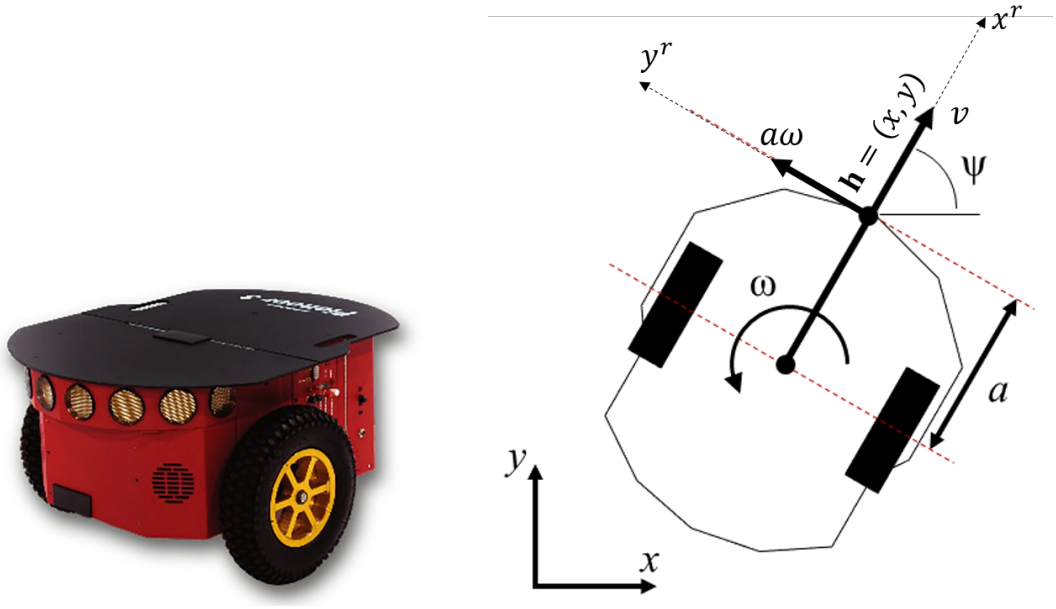
$$\mathbf{A}_P = \begin{bmatrix} \cos(\psi) & -a\,\mathrm{sen}(\psi) \\ \mathrm{sen}(\psi) & a\cos(\psi) \end{bmatrix} \tag{2.4}$$

é a matriz de cinemática direta do robô.

O modelo descrito na Equação (2.3) representa uma simplificação do modelo real, pois desconsidera os efeitos de inércia do robô. Na verdade, tal equação descreve apenas a cinemática do veículo, ou seja as relações entre posição e velocidade, considerando que o veículo não tem massa, logo não tem inércia. Tal modelo foi escolhido para o desenvolvimento do controle do robô terrestre devido ao cenário de testes envolver velocidades reduzidas.

Ao inverter a Equação (2.3) obtém-se a Equação (2.5) que representa o modelo de cinemática inversa, a partir do qual se pode calcular a velocidade linear e angular do robô em função das velocidades $\dot{x}$ e $\dot{y}$ referenciadas ao mundo.

$$\begin{bmatrix} \upsilon \\ \omega \end{bmatrix} = \mathbf{A}_P^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{2.5}$$



(a) Robô real (b) Descrição da cinemática

Figura 5 – Robô *Pioneer 3-DX* e caracterização da sua cinemática

onde a matriz

$$\mathbf{A}_P^{-1} = \begin{bmatrix} \cos(\psi) & \operatorname{sen}(\psi) \\ \frac{-\operatorname{sen}(\psi)}{a} & \frac{\cos(\psi)}{a} \end{bmatrix} \tag{2.6}$$

é a matriz de cinemática inversa do robô.

Utilizando a Equação (2.5) obtém-se as velocidades instantâneas do robô, as quais são comparadas com as velocidades desejadas $\mathbf{u}_r = \begin{bmatrix} \upsilon_r & \omega_r \end{bmatrix}^T$ para o robô, geradas pelo controlador da formação. As diferenças entre estas velocidades, adicionadas a uma componente proporcional ao erro de posição ponderado, compõem o sinal de controle que efetivamente será enviado ao robô, conforme caracterizado na próxima subseção.

2.1.2 Modelo Dinâmico e Controlador do *Pioneer 3-DX*

Uma abordagem mais próxima da realidade, e de maior complexidade, pode ser adotada ao se contemplar a dinâmica do robô nas equações de controle. Neste caso o comando de velocidade enviado ao robô é composto pela referência de velocidade proveniente do controlador cinemático, corrigida por um bloco adicional de controle, o qual se define como um compensador dinâmico, correção esta que visa compensar os efeitos da dinâmica do robô.

Portanto, o sinal de velocidade proveniente do controlador cinemático é enviado a um compensador dinâmico, o qual manipula tais velocidades com o objetivo de anular (na prática atenuar) os efeitos da dinâmica do robô. Neste trabalho utiliza-se o compensador dinâmico proposto em (MARTINS; SARCINELLI-FILHO; CARELLI, 2017; MARTINS, 2009), cuja estabilidade foi provada pelos autores. A lei de controle correspondente a tal compensador é caracterizada como

$$\mathbf{u}_d = \mathbf{H}(\dot{\mathbf{u}}_r + \mathbf{K}_{d1} \tanh(\mathbf{K}_{d2}\tilde{\mathbf{u}})) + \mathbf{C}\mathbf{u}_r, \tag{2.7}$$

onde $\tilde{\mathbf{u}} = [\tilde{v} \quad \tilde{w}]^T$ é o erro das velocidades linear e angular, respectivamente, $\tilde{v} = v_r - v$ e $\tilde{\omega} = \omega_r - \omega$, $\mathbf{K}_{d1}$ e $\mathbf{K}_{d2}$ são matrizes de ganho diagonais definidas positivas e $\mathbf{H}$ e $\mathbf{C}$ são matrizes relacionadas ao modelo dinâmico do *Pioneer 3-DX*, conforme (MARTINS; SARCINELLI-FILHO; CARELLI, 2017). Considerando os parâmetros dinâmicos $\theta_i, i = 1, \cdots, 6$ (ver (MARTINS; SARCINELLI-FILHO; CARELLI, 2017)), as matrizes $\mathbf{H}$ e $\mathbf{C}$ podem ser representadas como

$$\mathbf{H} = \begin{bmatrix} \theta_1 & 0 \\ 0 & \theta_2 \end{bmatrix}$$

e

$$\mathbf{C} = \begin{bmatrix} \theta_4 & -\theta_3\omega \\ \theta_5\omega & \theta_6 \end{bmatrix}.$$

Os parâmetros $\theta_i, i = 1, \cdots, 6$, são referentes às constantes físicas do robô, tais como massa, tamanho das rodas, resistência elétrica dos motores, entre outras. Nos experimentos realizados neste trabalho foram utilizados os mesmos parâmetros θ identificados em (MARTINS, 2009) através do método dos mínimos quadrados para o *Pioneer 3-DX* sem acessórios extras (sensor *laser* e/ou câmera).

Conforme descrito anteriormente, o modelo foi levantado em estudos anteriores os quais podem ser encontrados em (RABELO; BRANDÃO; SARCINELLI-FILHO, 2018) e (MARTINS, 2009). Os parâmetros θ utilizados nesta dissertação são tais que

$$\mathbf{H} = \begin{bmatrix} 0,534 & 0 \\ 0 & 0,217 \end{bmatrix}$$

e

$$\mathbf{C} = \begin{bmatrix} 0,956 & 0,013 \\ -0,084 & 1,059 \end{bmatrix}$$

Estes parâmetros representam uma boa aproximação do modelo real do robô *Pioneer 3-DX*, o que pode ser comprovado através dos estudos citados e dos testes executados nesta dissertação.

2.1.3 O Drone *Bebop 2* e seu Modelo Cinemático

O *Bebop 2* é um quadrimotor de asas rotativas produzido pela empresa *Parrot Drones SAS*. Ele possui uma câmera frontal de 14 mega-pixel do tipo olho de peixe, com resolução *FullHD* 1080p e estabilização ótica. Ele também possui processadores CPU(do Inglês: *Central Processig Unit*) *dualcore* e GPU (do Inglês: *Graphics Processing Unit*) *quadcore* embarcados.

O quadrimotor é fornecido com instrumentos embarcados, como sensores de altitude, GPS, acelerômetros, sensor de nível de carga da bateria e duas câmeras, os quais podem ser acessados através de *software* fornecido pelo fabricante ou através de um *driver* 2.4.3 de comunicação, baseado no SDK (do Inglês: *Software Development Kit*) oficial da Parrot.

Sensores adicionais ainda podem ser acoplados ao corpo do *Bebop 2*, como o módulo de desvio de obstáculo desenvolvido pela própria Parrot. Contudo, não serão utilizados módulos adicionais neste trabalho.

Para realizar a comunicação com o *Bebop 2* se utilizou o *driver* denominado *Bebop_Autonomy* 2.4.4 juntamente com o ROS (do Inglês: *Robot Operating System*) 2.4.2. Dessa forma o acesso às variáveis internas do quadrimotor é realizado através de **nós** responsáveis por publicar e escutar mensagens (uma descrição mais detalhada é apresentada na Subsecção 2.4.2, que é destinada a este tópico).

Do ponto de vista operacional, o *Bebop 2* possui seis graus de liberdade, conforme mostrado na Figura 6. Contudo, apenas quatro sinais de controle são enviados a ele, configurando dessa forma um sistema sub-atuado, como é o caso de qualquer aeronave. Os sinais de controle enviados são

$$u_d = [u_\phi \quad u_\theta \quad u_{\dot{z}} \quad u_{\dot{\psi}}]^T \in [-1, 1], \tag{2.8}$$

onde:

- u_ϕ controla o ângulo de rolagem, responsável pela movimentação esquerda-direita, ou seja, equivale a um comando de velocidade $u_{\dot{y}}$, conforme detalhado em (SANTANA, 2016);
- u_θ controla o ângulo de arfagem, que resulta na movimentação para frente e para trás, ou seja, equivale a um comando de velocidade $u_{\dot{x}}$, conforme detalhado em (SANTANA, 2016);
- $u_{\dot{z}}$ controla a velocidade vertical;
- $u_{\dot{\psi}}$ é responsável pela taxa de guinada, que rotaciona o robô ao redor do seu eixo z.

Quanto à cinemática do *Bebop 2*, ela é descrita considerando-se que o veículo não possui massa, ou seja, as velocidades comandadas são atingidas instantaneamente. Isso quer dizer que as velocidades em coordenadas

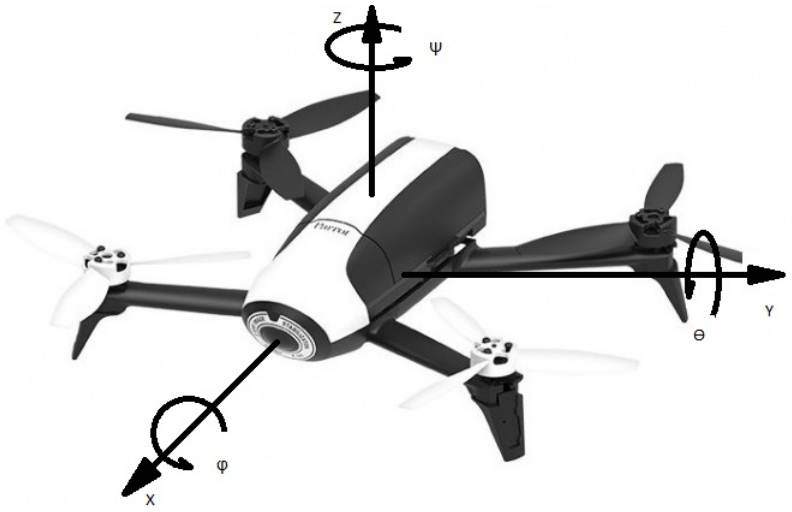


Figura 6 – O drone *Bebop 2*

do veículo são iguais aos valores comandados. Daí, as velocidades em coordenadas globais são simplesmente as projeções das velocidades em coordenadas do veículo nos eixos do sistema global de coordenadas, considerando o ângulo de guinada ψ do veículo. Portanto,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix} = \mathbf{A}_B \begin{bmatrix} u_{\dot{x}} \\ u_{\dot{y}} \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix}, \tag{2.9}$$

sendo a matriz

$$\mathbf{A}_B = \begin{bmatrix} \cos(\psi) & -\operatorname{sen}(\psi) & 0 & 0 \\ \operatorname{sen}(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

a matriz de cinemática direta do *Bebop 2*. No caso desta dissertação somente será utilizada a submatriz de cinemática direta

$$\begin{bmatrix} \cos(\psi) & -\operatorname{sen}(\psi) & 0 \\ \operatorname{sen}(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

já que apenas a posição $\begin{bmatrix} x & y & z \end{bmatrix}$ do veículo é de interesse.

A partir da Equação (2.9) se pode escrever a cinemática inversa do *Bebop 2* como

$$\begin{bmatrix} u_{\dot{x}} \\ u_{\dot{y}} \\ u_{\dot{z}} \\ u_{\dot{\psi}} \end{bmatrix} = \mathbf{A}_B^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\psi} \end{bmatrix}, \tag{2.11}$$

sendo

$$\mathbf{A}_B^{-1} = \begin{bmatrix} \cos(\psi) & \operatorname{sen}(\psi) & 0 & 0 \\ -\operatorname{sen}(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

2.1.4 Modelagem Dinâmica e Controlador do *Bebop 2*

Para o projeto do controlador a ser utilizado para o quadrimotor será considerado o modelo dinâmico proposto em (SANTANA et al., 2014; SANTANA; BRANDÃO; SARCINELLI-FILHO, 2016; SANTANA, 2016) e já utilizado em (SANTOS, 2017), (SANTOS et al., 2017), (SANTOS et al., 2017) e (SANTOS et al., 2019), dado por

$$\ddot{\mathbf{X}} = \mathbf{f}_1\mathbf{U} - \mathbf{f}_2\dot{\mathbf{X}}, \tag{2.13}$$

a partir do qual se projeta o controlador baseado na dinâmica inversa do veículo, dado por

$$\mathbf{U} = \mathbf{f}_1^{-1}(\nu + \mathbf{f}_2\dot{\mathbf{X}}) \tag{2.14}$$

onde

- $\mathbf{U}$ é um vetor que contém o sinal de controle enviado ao drone;
- $\mathbf{f}_1^{-1}$ representa uma matriz de constantes positivas características de cada drone;
- ν é dado por $\nu = \ddot{\mathbf{X}}_d + \mathbf{K}_p\tilde{\mathbf{X}} + \mathbf{K}_d\dot{\tilde{\mathbf{X}}}$;
- $\mathbf{K}_p$ e $\mathbf{K}_d$ são matrizes diagonais positivas definidas (ganhos);
- $\tilde{\mathbf{X}} = \mathbf{X}_d - \mathbf{X} = \begin{bmatrix} x_d - x & y_d - y & z_d - y & \psi_d - \psi \end{bmatrix}^T$; e
- $\mathbf{f}_2$ também representa uma matriz de constantes positivas características de cada drone.

Para analisar a estabilidade do sistema proposto toma-se o modelo dinâmico do sistema descrito na Equação (2.13), substituindo-se $\mathbf{U}$ dado por (2.14), obtendo-se

$$\ddot{\tilde{\mathbf{X}}} + \mathbf{K}_d\dot{\tilde{\mathbf{X}}} + \mathbf{K}_p\tilde{\mathbf{X}} = \mathbf{0}, \tag{2.15}$$

que representa a equação em malha fechada que rege a dinâmica do erro do sistema. Utilizando a teoria de Lyapunov e a partir da Equação (2.15) vê-se que $\tilde{\mathbf{X}} \to \mathbf{0}$ quando $t \to \infty$. Para chegar a tal conclusão, seja a função candidata de Lyapunov dada por

$$V(\tilde{\mathbf{X}}, \dot{\tilde{\mathbf{X}}}) = \frac{1}{2}(\tilde{\mathbf{X}}^T\mathbf{K}_p\tilde{\mathbf{X}} + \dot{\tilde{\mathbf{X}}}^T\dot{\tilde{\mathbf{X}}}) \geq \mathbf{0} \tag{2.16}$$

e sua derivada primeira, que é dada por

$$\dot{V}(\tilde{\mathbf{X}}, \dot{\tilde{\mathbf{X}}}) = (\tilde{\mathbf{X}}^T\mathbf{K}_p\dot{\tilde{\mathbf{X}}} + \dot{\tilde{\mathbf{X}}}^T\ddot{\tilde{\mathbf{X}}}). \tag{2.17}$$

A partir de (2.15) se obtém que

$$\ddot{\tilde{\mathbf{X}}} = -\mathbf{K}_d\dot{\tilde{\mathbf{X}}} - \mathbf{K}_p\tilde{\mathbf{X}},$$

e substituindo tal valor em (2.17) se obtém que

$$\dot{V}(\tilde{\mathbf{X}}, \dot{\tilde{\mathbf{X}}}) = -\dot{\tilde{\mathbf{X}}}^T \mathbf{K}_d \dot{\tilde{\mathbf{X}}} \leq \mathbf{0}. \tag{2.18}$$

A partir de (2.16) e (2.18), e tendo em conta a teoria de estabilidade de Lyapunov e o Teorema de La Salle para sistemas autônomos (VIDYASAGAR, 1993), conclui-se que o erro $\tilde{\mathbf{X}}$ e sua derivada $\dot{\tilde{\mathbf{X}}}$ tendem a zero assintoticamente, ou seja, vê-se que o controlador proposto é capaz de guiar o quadrimotor para a posição desejada, considerando um erro ζ aceitável e um tempo finito suficientemente grande para convergência.

2.1.5 Identificação dos parâmetros do modelo dinâmico do *Bebop 2*

Os parâmetros do modelo dinâmico do quadrimotor, correspondentes às matrizes $\mathbf{f}_1$ e $\mathbf{f}_2$ em (2.13), foram obtidos seguindo o procedimento descrito em (SANTOS, 2017), com o diferencial que neste caso foi utilizado um sistema de captura de movimento de alta precisão denominado *Optitrack* (ver a Subsecção 2.3.2), sistema este que permite identificar a posição do veículo no espaço ao longo do tempo.

Aplicando um sinal de controle conhecido ao quadrimotor (excitação), pode-se realizar o levantamento dos parâmetros dinâmicos do modelo simplificado a partir da relação entre o sinal aplicado (entrada do sistema) e sua respectiva resposta (saída do sistema). No caso, o sinal de excitação é conhecido, e a resposta foi medida usando-se o sistema *OptiTrack*.

Foi utilizado o método mínimos quadrados para levantar os parâmetros que melhor se ajustavam ao conjunto de dados obtidos nos testes. Note-se que este método oferece um resultado otimizado, uma vez que minimiza o erro e proporciona a curva que melhor se ajusta aos dados. Utilizando a Equação (2.13), foi possível estimar um modelo dinâmico simplificado representado pelas duas matrizes quadradas constantes denominadas $\mathbf{f}_1 = diag(K_1 \quad K_3 \quad K_5 \quad K_7)$ e $\mathbf{f}_2 = diag(K_2 \quad K_4 \quad K_6 \quad K_8)$.

Para cada um dos quatro graus de liberdade utilizados no controle do quadrimotor, ou seja, para cada uma das variáveis $\mathbf{u}_d = \begin{bmatrix} \phi & \theta & \dot{z} & \dot{\psi} \end{bmatrix}^T$ foi aplicado um sinal de excitação senoidal independente, dado pela Equação 2.19. As respectivas respostas foram gravadas e analisadas em uma etapa de pós processamento. O sinal de controle aplicado para a identificação dos parâmetros é descrito por

$$\mathbf{U} = \frac{0,4}{4,5}[3\,\mathrm{sen}(0,2\pi t) + \mathrm{sen}(0,6\pi t) + 0,5\,\mathrm{sen}(\pi t)], \tag{2.19}$$

o qual pode ser visto de forma gráfica na Figura 7.

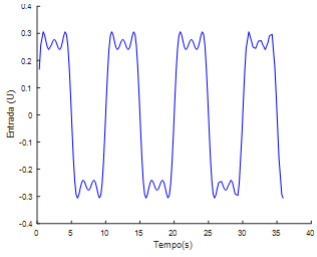


Figura 7 – Sinal aplicado ao *Bebop 2* para obtenção do modelo dinâmico simplificado

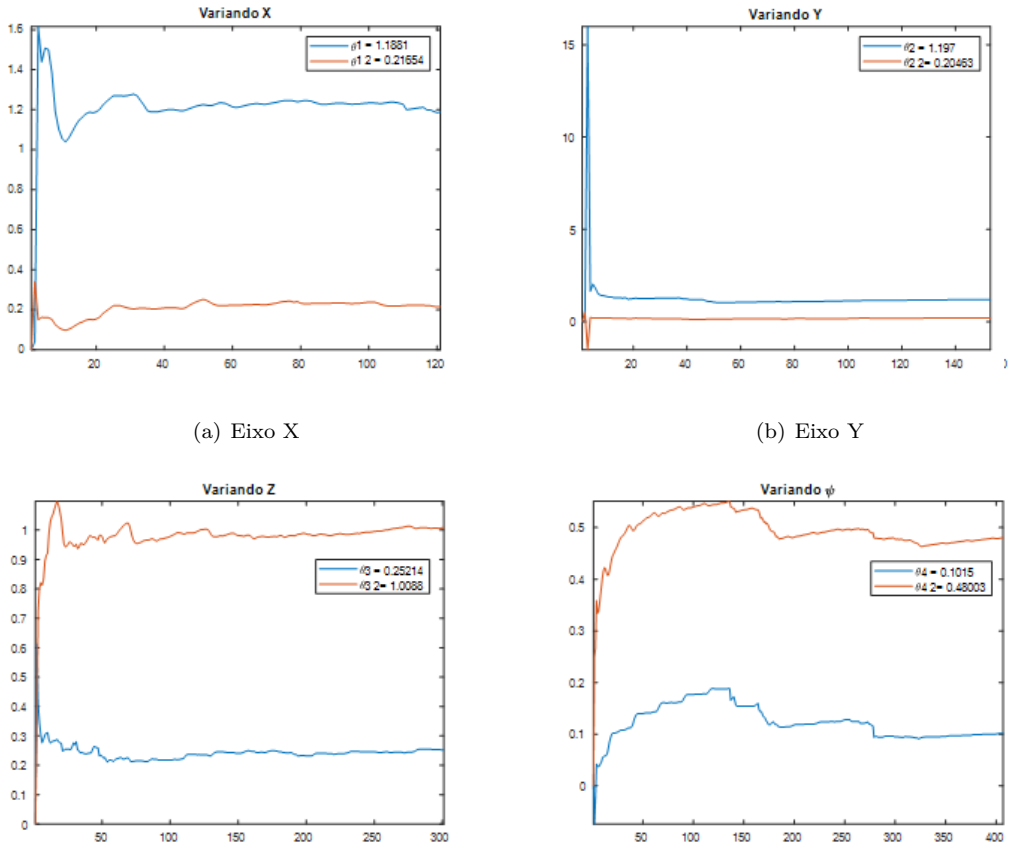
Já o erro de aproximação e os parâmetros podem ser visualizados na Figura 8

As matrizes diagonais $\mathbf{f}_1 = diag(K_1 \quad K_3 \quad K_5 \quad K_7)$ e $\mathbf{f}_2 = diag(K_2 \quad K_4 \quad K_6 \quad K_8)$ obtidas foram

$$\mathbf{f}_1 = \begin{bmatrix} 0,8417 & 0 & 0 & 0 \\ 0 & 0,8354 & 0 & 0 \\ 0 & 0 & 3,966 & 0 \\ 0 & 0 & 0 & 9,8524 \end{bmatrix} \tag{2.20}$$

e

$$\mathbf{f}_2 = \begin{bmatrix} 0,18227 & 0 & 0 & 0 \\ 0 & 0,17095 & 0 & 0 \\ 0 & 0 & 4,001 & 0 \\ 0 & 0 & 0 & 4,7295 \end{bmatrix}. \tag{2.21}$$



(a) Eixo X

(b) Eixo Y

(c) Eixo Z

(d) Orientação Psi

Figura 8 – Parâmetros calculados para o modelo dinâmico do *Bebop 2*

2.1.6 Validação do Modelo Dinâmico do *Bebop 2*

Para validação do modelo obtido foi aplicado um sinal senoidal distinto do sinal utilizado para a identificação dos parâmetros, caracterizado por

$$U = 0,2[\,\mathrm{sen}(0,2\pi t) + \,\mathrm{sen}(0,4\pi t)], \tag{2.22}$$

e a resposta correspondente foi medida. A Figura 9 mostra o sinal de validação aplicado, e as respectivas velocidades obtidas como resposta estão representadas na Figura 10, para cada eixo modelado.

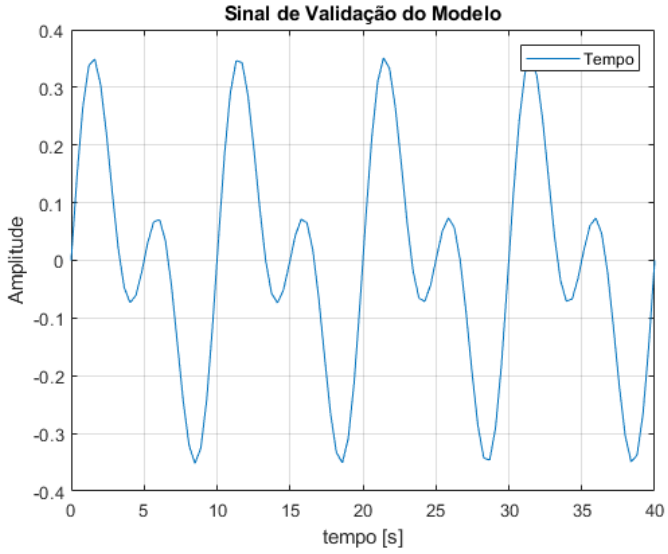
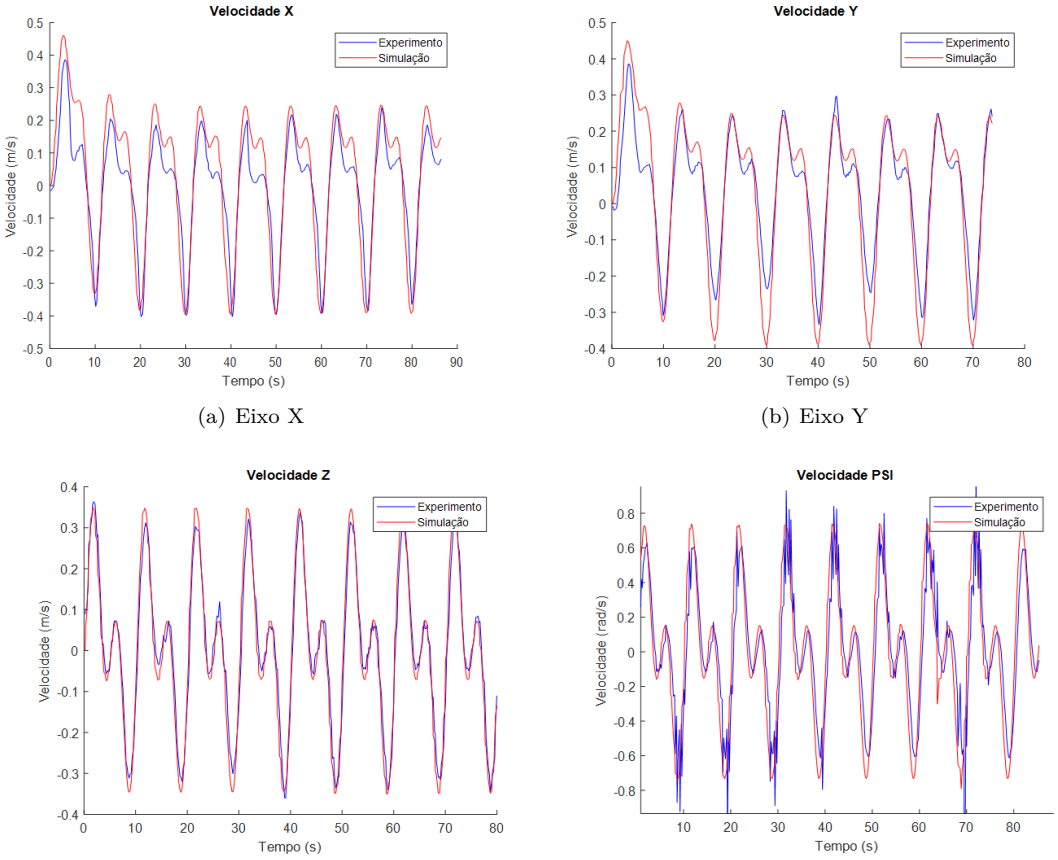


Figura 9 – Sinal de validação do modelo

Além do teste com o sinal em (2.22), também se avaliou a qualidade do modelo através de sua aplicação no controle de uma formação de dois VANTs. Os veículos foram programados para sair de uma posição inicial e chegar em uma posição final fixa, com o objetivo de avaliar o erro em regime estacionário da formação utilizando os modelos obtidos.

Os drones foram posicionados em $\mathbf{X}_1 = [0\ m \quad 0\ m \quad 1\ m \quad 0\ rad]^T$ e $\mathbf{X}_2 = [-6\ m \quad 3\ m \quad 1\ m \quad 0\ rad]^T$, e a formação desejada é caracterizada pelas variáveis $\mathbf{q}_{des} = [6\ m \quad 3\ m \quad 5\ m \quad 0\ rad \quad 0\ rad]^T$, onde $[6\ m \quad 3\ m \quad 5\ m]^T$ representa a posição final desejada para o primeiro drone e $[5\ m \quad 0\ rad \quad 0\ rad]^T$ corresponde à distância final desejada entre os dois drones e os ângulos entre eles (tal descrição corresponde a ter os dois drones na mesma altura, em voo pairado, um a 5 m do outro na direção x). Além disso, $\dot{\mathbf{q}}_{des} = \mathbf{0}$, o que quer dizer que os dois drones devem permanecer em voo pairado após atingirem suas posições desejadas, até o final do experimento. Vê-se pelos gráficos das Figuras 11 e 12 que a formação foi de fato capaz de realizar a tarefa de posicionamento com sucesso, validando a utilização do modelo aqui identificado no projeto dos controladores utilizados em cada drone. Note-se que tais figuras apresentam resultados obtidos por simulação e em experimentos reais, quando foi utilizada informação de posição e velocidade fornecidas pela unidade de medição inercial e o módulo de GPS disponível a bordo dos drones.
Os erros de formação são ilustrados na figura 12.

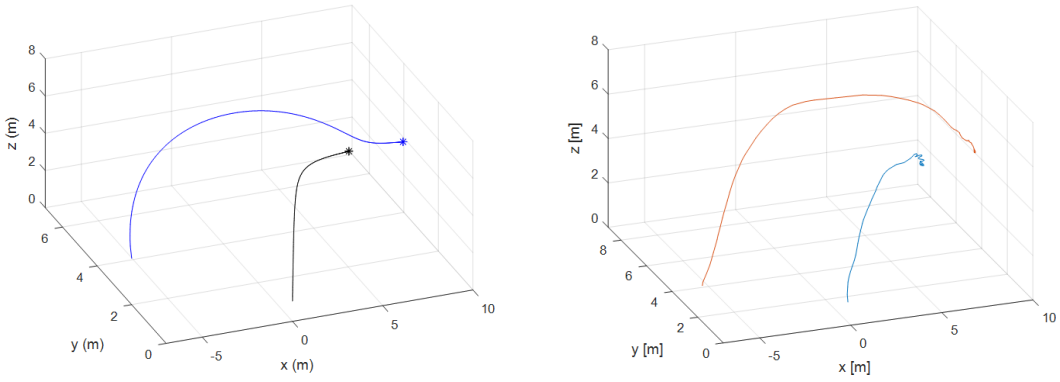
Conforme observado pela análise gráfica, o modelo apresentou bons resultados, sendo, portanto apto a ser utilizado no contexto desta pesquisa. Vale ressaltar que o modelo identificado consiste em uma aproximação simplificada do comportamento dinâmico do *Bebop 2*, apresentando boa aproximação porque os ângulos de



(a) Eixo X (b) Eixo Y

(c) Eixo Z (d) Orientação Psi

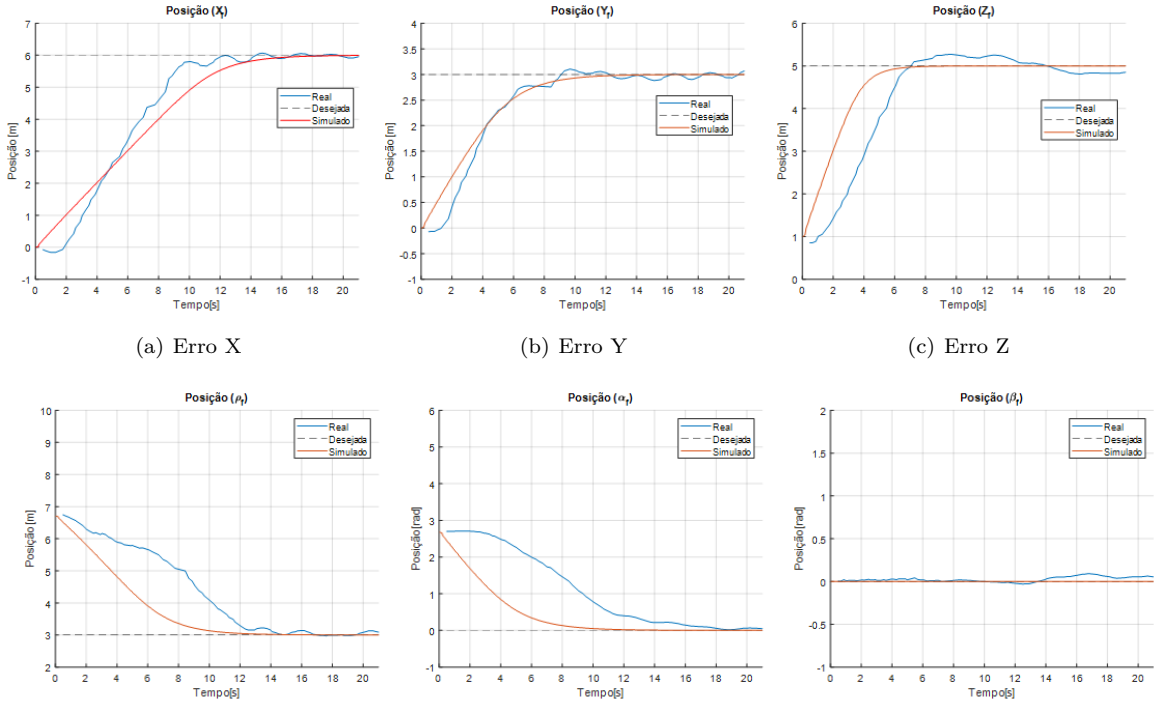
Figura 10 – Validação do modelo obtido



(a) Simulação (b) Experimento

Figura 11 – Tarefa de Posicionamento em Formação de Quadrimotores

arfagem e rolagem são limitados, pelo piloto automático a bordo do veículo, a valores bem pequenos (menores que 10°), conforme é discutido em detalhes em (SANTANA, 2016).



(a) Erro X (b) Erro Y (c) Erro Z

(d) Erro Rho (e) Erro Alpha (f) Erro Beta

Figura 12 – Erros da tarefa de posicionamento da formação de dois drones *Bebop 2*

2.2 Arquitetura Física

2.2.1 Dimensões do Espaço de Teste

O ambiente de teste consiste em uma sala de 6 metros de comprimento por 5 metros de largura e 3 metros de altura. Há mesas em apenas uma das arestas da sala, as quais são utilizadas por computadores e equipamentos de rede, não representando uma área útil de teste. Em resumo o experimento foi realizado em uma área útil de $72m^3$ (6 m X 4 m X 3 m) A sala possui um piso liso, não contendo obstáculos e imperfeições que afetem o deslocamento do VTNT.

2.2.2 Controle de Luminosidade

Para melhorar a iluminação para as câmeras do sistema de captura de movimento *Optitrack*, todas as janelas foram cobertas com material opaco, sendo a iluminação do ambiente realizada por lâmpadas fluorescentes.

Foram removidos ao máximo os materiais reflexivos que não fazem parte do corpo de prova, medida esta necessária para evitar falhas na detecção dos marcadores acoplados aos veículos, conforme necessidade do sistema de captura de movimento.

Como cuidados adicionais e para evitar fontes de luz indesejadas, a iluminação proveniente dos componentes eletrônicos, como LEDs (do Inglês: *Light Emitting Diode*), presentes na placa *Raspberry* e em *notebooks* são cobertos por uma estrutura composta de isopor coberto por um filme de material maleável e opaco.

2.3 Arquitetura de *Hardware*

Nesta seção serão abordados os aspectos de *hardware* utilizados tais como especificações, modelos dos equipamentos e características importantes que foram consideradas para a correta execução do experimento.

2.3.1 Visão Geral

Conforme mostrado na Figura 15, utilizou-se um sistema híbrido composto de equipamentos executando os sistemas operacionais Linux e Windows 10. Uma lista detalhada dos equipamentos utilizados é fornecida abaixo:

- *Sistema OptiTrack*: Computador DELL XPS com processador INTEL *Core* I7 de sétima geração, 24GB de Memoria RAM, disco SSD de 500GB e GPU compartilhada;
- *Linux ROS*: *Notebook* Samsung com processador INTEL *Core* I7 de sétima geração, 8GB de memória RAM DDR4 e disco SSD de 500GB;
- *Windows Matlab*: *Notebook* DELL com processador INTEL *Core* I7 de sétima geração, 8GB de memória RAM DDR4 e disco rígido de 1TB;
- *Roteador Wifi*: Modelo Tplink Gigabit com transmissão *wifi* de 2.4GHz e 5GHz;
- *Switch*: Switch Gigabit com capacidade de alimentação PoE (do Inglês: *Power Over Ethernet*);
- *Câmera OptiTrack*: 4 câmeras modelo Prime 13 e 4 câmeras modelo Prime 41;
- *Joystick*: modelo XBOX360 para computador.

2.3.2 O sistema de Captura de Movimento *OptiTrack*

O sistema de captura de movimento *OptiTrack* consiste em um sistema de localização tridimensional de código fechado, comercialmente vendido pela empresa NaturalPoint, Inc., cujo nome de fantasia é *OptiTrack*.

Este sistema é capaz de montar um corpo rígido virtual através de marcadores fixados no objeto de interesse, e informar sua localização no espaço tridimensional a partir da localização na imagem dos *blobs* de alta luminosidade correspondentes aos marcadores colocados nos objetos rastreados, imagens essas captadas por câmeras especializadas.

A informação de posição é distribuída via *broadcast* à rede, e pode ser captada por *softwares* como Matlab ou através do ROS, sistema que será discutido na Seção 2.4.2.

Nesta pesquisa utilizaram-se 8 câmeras de captura de vídeo, sendo 4 delas do modelo Prime 41, com maior abertura angular, e 4 câmeras do modelo Prime 13, com abertura angular mais restrita. A Figura 13, em suas partes (a) e (b) mostra tais câmeras. Elas são alimentadas via *PoE* e podem ser posicionadas a uma distância de até 20 metros do ponto de alimentação. No ambiente onde se realizaram os experimentos elas estão posicionadas em suportes fixos à parede da sala, a uma altura superior a 2 metros.

Por sua vez, a Figura 14 dá uma vista do local onde se realizaram os experimentos, mostrando os dois veículos com os marcadores acoplados à sua superfície e algumas das câmeras do *OptiTrack*. Observe-se que o *Pioneer 3-DX* possui uma plataforma sobre sua estrutura, que constitui uma superfície lisa para o pouso do VANT durante os experimentos.

Para cada corpo rígido cujo movimento se queria capturar pelo *OptiTrack* foram utilizados pelo menos 4 marcadores, sendo os mesmos distribuídos de forma assimétrica sobre o objeto a ser captado, no caso o

(a) Câmera Prime 13 (b) Câmera Prime 41

Figura 13 – Câmeras usadas no sistema *OptiTrack*

VTNT e o VANT. Tal assimetria dos marcadores é importante para garantir a diferenciação do corpo rígido, caso haja objetos com dimensões próximas. Este seria o caso, por exemplo, de algum estudo envolvendo a utilização de dois *Pioneers* ou dois *Bebops 2* simultaneamente.

2.3.3 Estrutura de Rede

A comunicação entre os computadores e câmeras é realizada através de cabos *Ethernet* conectados a um *switch* padrão Gigabit. Já o VANT se comunica com os computadores através de rede *wifi* operando em 2,4GHz, enquanto que para comunicação com o VTNT, que não tem computador a bordo, foi utilizada uma placa *Raspberry* Pi 3 modelo B plus ou um *notebook*. Em ambos os casos posicionados no topo de sua estrutura e também se conectando à mesma rede *wifi* de 2,4GHz utilizada pelo drone. Tal estrutura está ilustrada na Figura 15, na qual um detalhe que deve ser ressaltado é que estão ilustrados dois drones, não um drone e um robô a rodas. Entretanto, a estrutura para um drone e um robô terrestre, em sua essência, é idêntica àquela mostrada na figura.

2.4 Arquitetura de *Software*

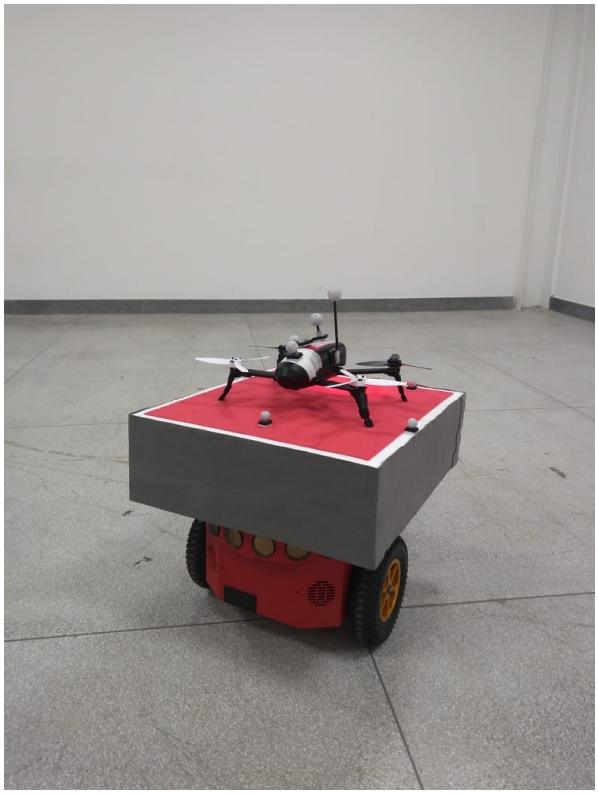
Nesta seção é realizada uma abordagem da arquitetura de *software* utilizada. São citados os *softwares* necessários para execução dos experimentos, assim como as interfaces desenvolvidas para troca de informação entre os equipamentos.

2.4.1 O *Software Motive*

Motive é um software proprietário da empresa NaturalPoint, desenvolvedora e distribuidora do sistema *OptiTrack* e pode ser baixado no site do mesmo, após o devido licenciamento. Ele é projetado para controlar a captura de imagens de objetos em movimento, possibilitando o rastreamento de objetos em várias aplicações. O *software* provê interfaces para captura e processamento de dados em três dimensões, os quais podem ser armazenados ou disponibilizados via *streaming* pela rede a outras aplicações clientes.

Ele obtém informações em 3D, reconstruindo a posição do objeto utilizando imagens de múltiplas câmeras em 2D e a reflexão de marcadores posicionados em pontos de interesse do objeto a ser rastreado. O *software* é

(a) Vista geral



(b) Veículos com marcadores

Figura 14 – Vista do local dos experimentos, veículos utilizados e algumas câmeras do *OptiTrack*

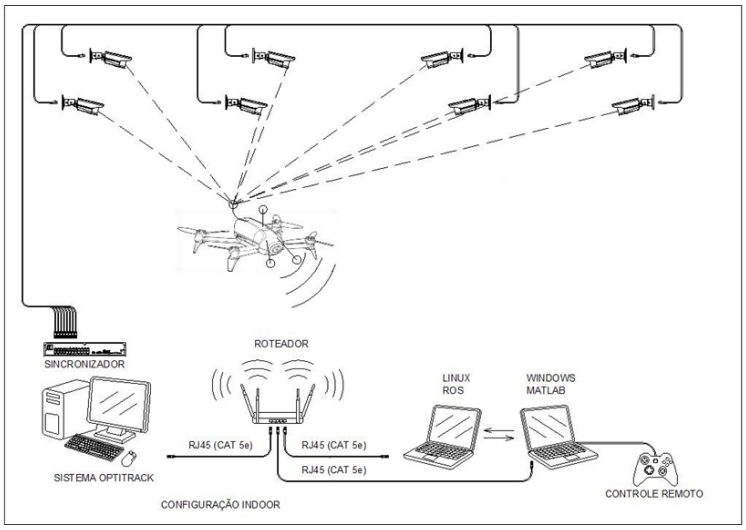


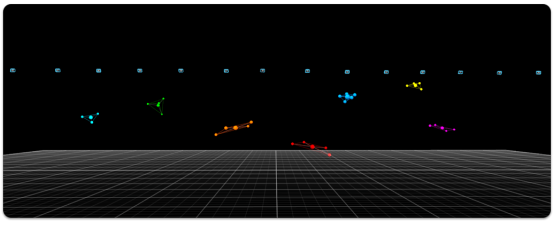
Figura 15 – Topologia de hardware

capaz de rastrear até 6 graus de liberdade, sendo 3 de posição e 3 de orientação.

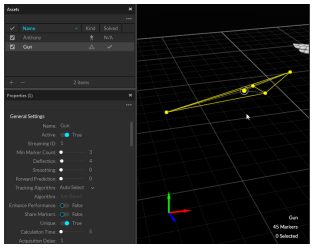
O rastreamento de múltiplos corpos rígidos em 3D simultaneamente também é possível, bastando para isso que marcadores em posições distintas sejam fixados nos corpos que serão rastreados pelo *software*. Para o estudo desenvolvido, será realizado o rastreamento simultâneo de dois robôs, sendo um terrestre, o *Pioneer*

3-DX, e um aéreo, o quadrimotor *Bebop 2*.

A Figura 16 mostra algumas telas do *software Motive*, ilustrando o rastreamento de objetos em movimento tridimensional. Cada conjunto de pontos, associados a uma cor, representam um objeto rastreado pelo *Software Motive*, conforme pode-se observar na figura da 16 item *(a)* tem-se sete objetos detectados. Na Subfigura *(b)* tem-se um detalhe do objeto o qual pode-se verificar seu centro de massa, ponto amarelo de maior destaque na figura e algumas configurações da interface gráfica do *Software Motive*.



(a) Exemplo: Sete objetos rastreados



(b) Tela de configuração do *Software Motive*

Figura 16 – Motive

2.4.2 ROS - *Robot Operating System*

Há uma ampla gama de material disponível na *internet* sobre o que é o ROS, um sistema operacional específico para robôs, e onde ele pode ser aplicado. Este tópico, porém, se aterá aos conceitos necessários para o leitor compreender sua utilização no contexto desta dissertação.

O ROS é um *framework* de código aberto e flexível que realiza a comunicação de baixo nível com uma grande variedade de robôs, inclusive os dois utilizados nesta dissertação. Ele é composto por um conjunto de bibliotecas e ferramentas que facilitam o desenvolvimento de comportamentos complexos de robôs atuando individualmente ou em conjunto.

De forma simplificada, o ROS opera com um serviço central denominado **nó Mestre**, o qual coordena a comunicação entre todos os **nós** da rede. Este nó possui as informações dos agentes que compõem a rede, juntamente com suas respectivas mensagens, que são publicadas e recebidas.

Ao ingressar na rede, o novo **nó** busca o **nó Mestre** e lhe informa quais tipos de mensagens são publicadas e quais são recebidas por ele. Caso o nó necessite receber alguma informação da rede, uma solicitação é enviada ao **nó Mestre**, denominada REQUISIÇÃO DE ASSINATURA ou *subscrições de tópico (subcription)*. Caso o tópico solicitado esteja disponível na rede, o **nó Mestre** informa ao nó requisitante o endereço do tópico divulgador (*Advertising*) e a partir deste ponto, toda a comunicação é realizada entre o nó ouvinte e o nó divulgador. A Figura 17 ilustra esta comunicação.

Nesta pesquisa utilizou-se um servidor Linux onde é executado o *Nodo Master* do ROS, juntamente com o *driver* de comunicação do *Bebop 2*. Toda a programação neste nó é desenvolvida em C++ e Python, e é responsável pela comunicação de baixo nível com os robôs.

Para realizar a interface com este conjunto, há um *Nodo Ouvinte/Divulgador*, o qual é responsável pelo controle dos robôs. Nele são executados o *loop* de controle o qual, contempla as proteções necessárias para assegurar a segurança das pessoas e a integridade dos equipamentos durante a realização dos experimentos.

Este nó é executado em um computador independente, rodando o sistema operacional Windows 10 com o aplicativo Matlab R2018a instalado. Através do Matlab é criado um nó de controle que se comunica com o

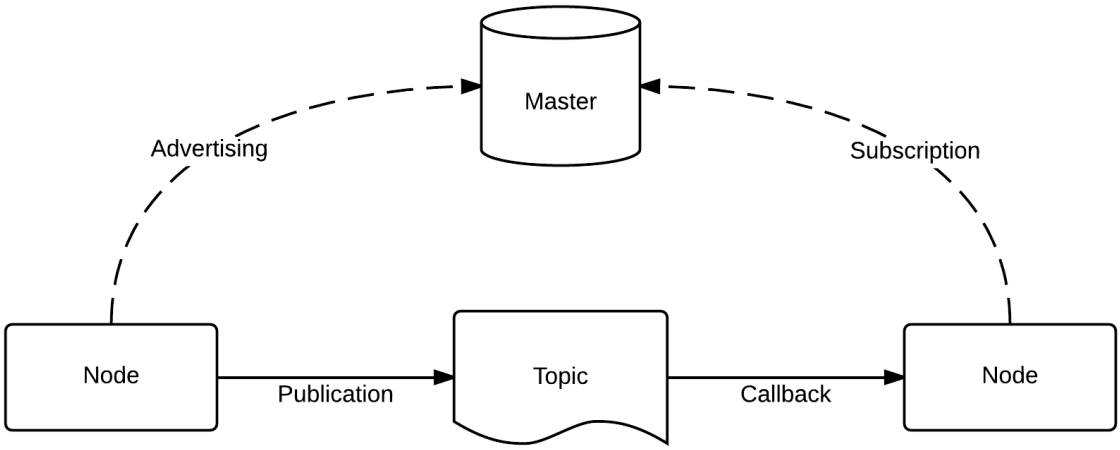


Figura 17 – Arquitetura de comunicação básica do ROS

Nodo Master localizado no computador Linux. Esta arquitetura será melhor detalhada na Subseção 2.4.5

Aprofundando mais o detalhamento da arquitetura, há várias distribuições do ROS que podem ser encontradas no site oficial do ROS. Neste trabalho utilizou-se a versão denominada *ROS-Kinect Kame* instalada em um computador com sistema operacional Ubuntu versão 16.04.

As configurações e versões de *softwares* descritas acima foram utilizadas afim de garantir a compatibilidade e estabilidade do sistema com o *driver* de comunicação *bebop_automomy*, tratado na Subseção 2.4.3. Versões mais recentes do sistema operacional, juntamente com o *ROS* instalado, foram testadas, mas até o momento em que esta dissertação foi escrita tais versões não apresentavam todas as funcionalidades necessárias para o correto funcionamento dos sistemas.

2.4.3 *Driver Bebop_Autonomy*

O *driver* de comunicação *bebop_autonomy* tem como base o próprio SDK oficial disponibilizado pela Parrot. Ele foi desenvolvido no *Autonomy Lab* da *Simon Fraser University* por Mani Monajjemi e alguns colaboradores, e mantido por Sepehr MohaimenianPour (também do *Autonomy Lab*, *Simon Fraser University*), Thomas Bamford (do *Dynamic Systems Lab, University of Toronto*) e Tobias Naegeli (*Advanced Interactive Technologies Lab, ETH Zürich*).

Desenvolvido na linguagem de programação C++, o *driver* oferece uma série de tópicos e recursos que possibilitam que o usuário do *Bebop 2* acesse os dados internos e envie comandos para o quadrimotor. Explicando de forma superficial, o *driver* de comunicação publica tópicos que enviam mensagens com informações de odometria e *status* do quadrimotor, e recebe mensagens do tipo comandos de velocidade.

O *driver* de comunicação baixado em fevereiro de 2019 e utilizado nesta pesquisa foi executado na versão 16.04 do sistema operacional Ubuntu, juntamente com o sistema ROS citado na Subseção 2.4.2.

A Figura 18 ilustra a arquitetura do *driver* de comunicação *bebop_autonomy*.

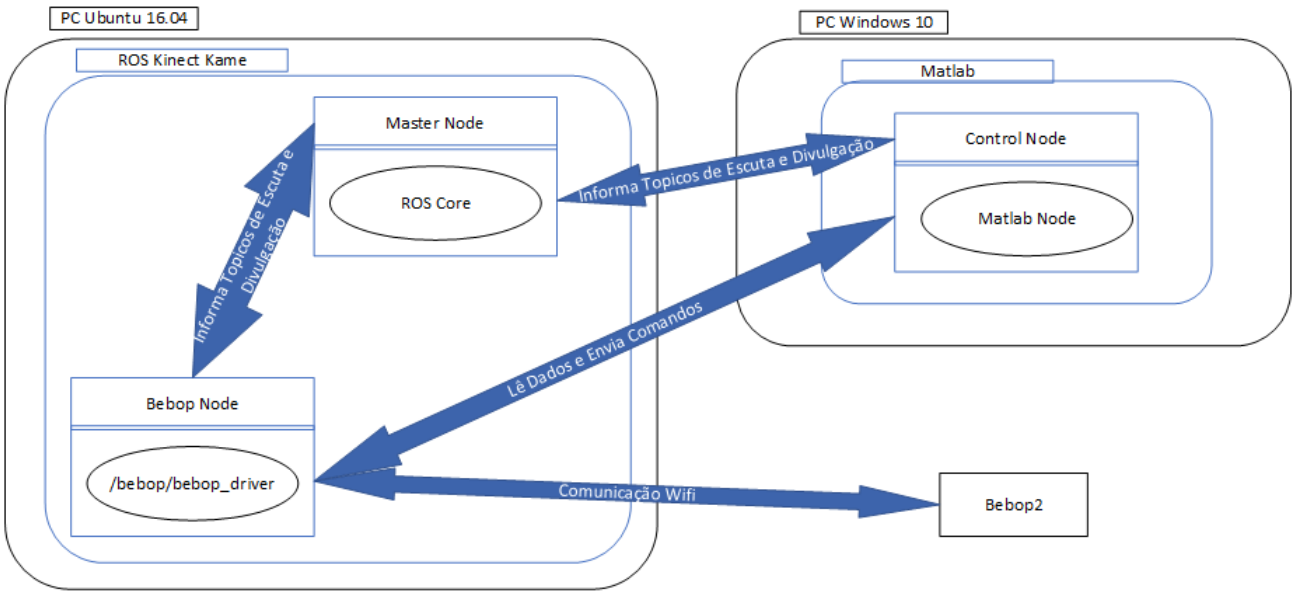


Figura 18 – Arquitetura de comunicação do *driver bebop_autonomy*

2.4.4 A Plataforma *Aurora*

A Plataforma *AuRoRa* (acrônimo para *Autonomous Robots for Research and Application*), consiste em um *framework* desenvolvido em Matlab de forma cooperativa entre os grupos de pesquisa da Universidade Federal do Espírito Santo e Universidade Federal de Viçosa. Sua primeira versão surgiu em 2016 (PIZETTA; BRANDÃO; SARCINELLI-FILHO, 2016), e desde então ela vem sendo continuamente ampliada, para incluir novos veículos, com os respectivos modelos, e novos controladores. Ela é composta por um conjunto de bibliotecas cujas características proporcionam o reaproveitamento de código e a consequente otimização do fluxo de informação. Um detalhe importante é que tal plataforma pode ser usada tanto para simulação de tarefas considerando o(s) modelo(s) do(s) veículo(s) utilizado(s) (*software-in-the-loop*), quanto para a execução de experimentos (configuração *hardware-in-the-loop*).

Dentre suas principais características se destacam a camada de baixo nível, responsável pela comunicação com os robôs, a camada de controle, a qual é estruturada para receber comportamentos individuais e colaborativos (formação de robôs) e a camada de relatórios, responsável por armazenar e realizar o pós processamento básico dos dados coletados, para produção dos gráficos que permitem a análise dos resultados obtidos.

A plataforma oferece um conjunto de ferramentas que agiliza o desenvolvimento, assim como aumenta a robustez do código e a confiabilidade dos dados coletados. Tais características embasam sua adoção como alicerce para pesquisa.

2.4.5 *Driver* de Comunicação Aurora - ROS

Apesar de ser possível o desenvolvimento da pesquisa inteiramente no ambiente Linux, onde é executado o *nó Mestre* e utilizando a linguagem de programação Python, optou-se por criar um *driver* de comunicação entre a biblioteca *bebop_autonomy* e a a Plataforma Aurora. A escolha desta arquitetura se deve à utilização dos conhecimentos prévios já implementados e estruturados na Plataforma Aurora. Isto agiliza a obtenção de resultados e aumenta a confiabilidade do controle.

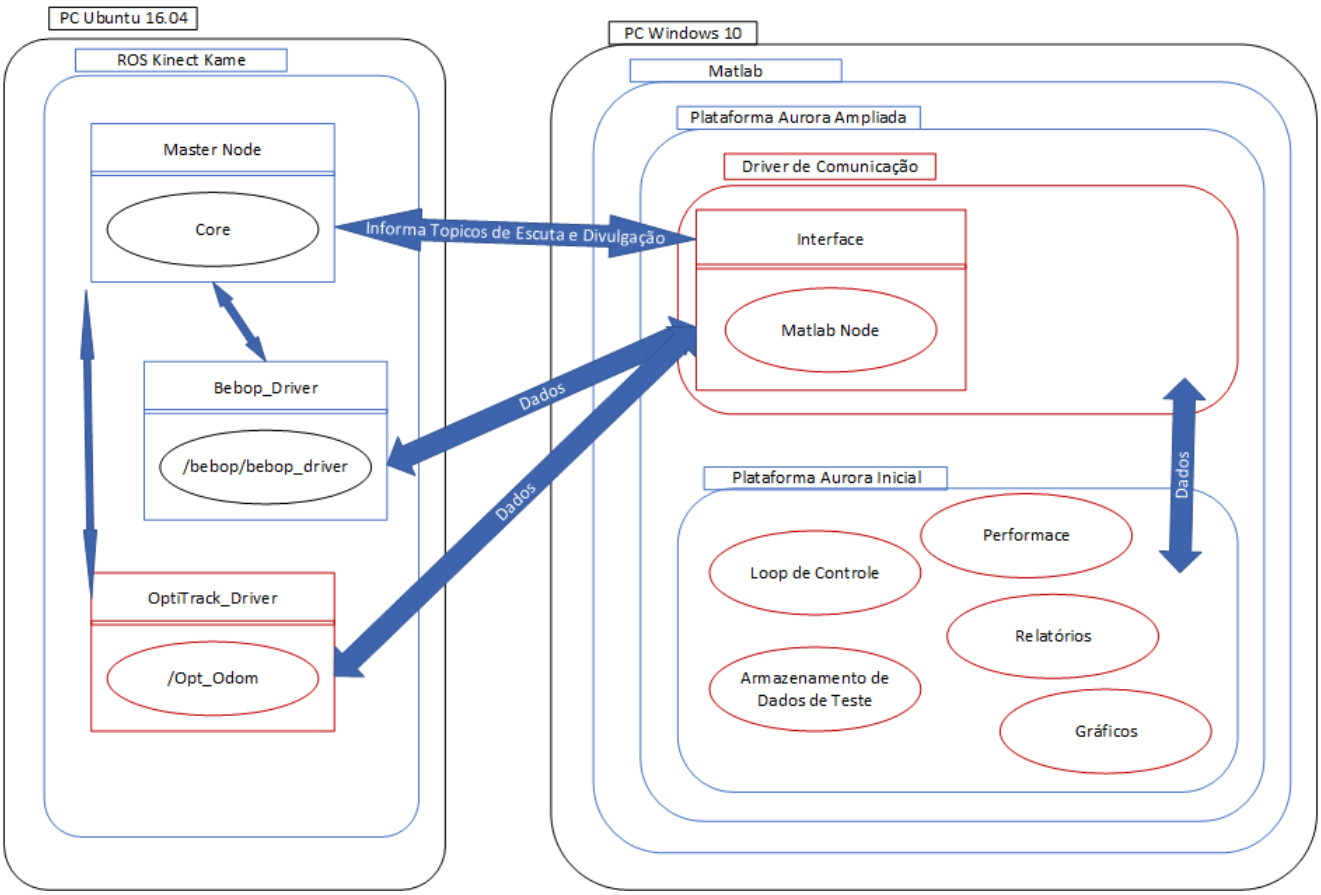


Figura 19 – *Driver* de comunicação Aurora - ROS

.

A integração da Plataforma Aurora com o ROS é o primeiro passo para que os grupos de pesquisa consolidem o ROS como sistema padrão de comunicação entre seus robôs. Tendo em vista que o ROS é capaz de integrar vários sistemas distintos e aliado à padronização proporcionada por ele, considera-se primordial sua utilização nesta e em pesquisas posteriores que se relacionem com os conhecimentos aqui compartilhados. A Figura 19 mostra a arquitetura do *driver* de comunicação Aurora - ROS. Em tal figura destaca-se que:

- Estruturas físicas como computadores, são destacadas na cor preta.
- Módulos desenvolvidos nesta pesquisa são destacados na cor vermelha.
- Módulos já desenvolvidos anteriormente são destacados na cor azul.
- As setas indicam o fluxo de informação entre as estruturas especializadas utilizadas e implementadas neste trabalho.

3 Formações de Múltiplos Robôs

Este capítulo aborda os conceitos fundamentais sobre formações de vários robôs, também conhecidas como sistemas multirrobôs, motivações para sua utilização e aspectos técnicos sobre formações no plano (2D) e formações no espaço (3D). São destacados alguns conceitos primordiais, como as relações entre as variáveis dos robôs e as variáveis que descrevem a formação.

3.1 Introdução

3.1.1 Motivações

A utilização de robôs e veículos autônomos tem aumentado consideravelmente em várias áreas do conhecimento. Eles ampliam a capacidade de interagir com o ambiente, executando tarefas para suportar, substituir ou estender habilidades humanas.

Para aumentar ainda mais sua aplicabilidade, mais robôs podem ser utilizados de forma cooperativa para realizar uma tarefa em comum. Sua utilização conjunta em uma missão viabiliza a execução de tarefas complexas, pois possibilitam sua divisão em sub-tarefas de menor complexidade, as quais podem ser executadas por um ou mais agentes. Usualmente a abordagem multirrobôs também reduz os custos com equipamentos, uma vez que tarefas de menor complexidade normalmente podem ser executadas por equipamentos mais simples, com menor capacidade de carga e de sensoriamento. Outro fator que corrobora a divisão de tarefas e utilização de mais equipamentos está relacionado com a otimização do tempo de execução da missão: vários robôs executando uma missão podem trazer uma redução considerável no tempo necessário para cumpri-la (PARKER, 2016).

Aplicações como mapeamento de áreas (LI et al., 2014; BARANZADEH; SAVKIN, 2017), buscas de sobreviventes em escombros (SATO et al., 2004; BECK et al., 2016) e locomoção de grandes objetos (BAI; WEN, 2010; MASONE; BÜLTHOFF; STEGAGNO, 2016; VILLA; BRANDÃO; SARCINELLI-FILHO, 2019), para citar umas poucas, são exemplos onde há ganhos consideráveis ao se utilizar a abordagem de múltiplos robôs operando em cooperação.

Em se tratando de uma formação, aqui entendido como um agrupamento de robôs, similares (caso de uma formação homogênea) ou não (caso de uma formação heterogênea), dois conjuntos de variáveis são fundamentais. O primeiro deles corresponde às variáveis que descrevem as posições dos robôs, chamadas variáveis dos robôs, vinculadas ao que se chamará daqui em diante espaço dos robôs. O segundo conjunto corresponde às variáveis que caracterizam a formação como um todo, envolvendo variáveis que descrevem a posição e também a forma da formação, as quais são vinculadas ao que se chamará daqui em diante espaço da formação.

O espaço dos robôs corresponde à caracterização dos robôs que fazem parte da formação. O interesse aqui é as posições dos referidos robôs, uma vez que elas caracterizam o estado atual da formação, mas as velocidades também são de interesse, uma vez que os comandos enviados para os robôs, para fazê-los mudar sua posição atual, são sinais de velocidade, nesse caso relativas ao sistema de coordenadas de cada robô. No caso de posição, a referência é o sistema de coordenadas global. Nesta dissertação as variáveis dos robôs são representadas pelo vetor $\mathbf{x}$.

Já no que se refere ao espaço da formação, a referência de posicionamento é o sistema de coordenadas

global, o qual é posicionado no instante inicial de teste. As variáveis que descrevem a formação, envolvendo uma caracterização de sua posição e de sua forma, as variáveis da formação, são representadas pelo vetor $\mathbf{q}$.

3.2 Formação Planar (2D)

Para melhor compreensão dos conceitos relacionados à formação de robôs, inicia-se o estudo por um caso particular correspondente à formação de robôs em duas dimensões, ou seja, em um plano, em geral o plano horizontal $x - y$. Uma formação desse tipo pode ser exemplificada pelo caso mostrado na Figura 20, que mostra uma formação envolvendo dois robôs a rodas do tipo unicilo (modelo *Pioneer 3-DX*). Vale destacar que o exemplo envolve apenas dois robôs, embora seja possível utilizar qualquer número de robôs. O objeto em estudo restringe-se em uma formações heterogênea composta de VANT-VTNT (ver Seção 3.3 e capítulos subsequentes), ou seja de dois robôs. Note-se que a formação ilustrada na Figura 20 é uma formação homogênea, enquanto a formação VANT-VTNT é uma formação heterogênea.

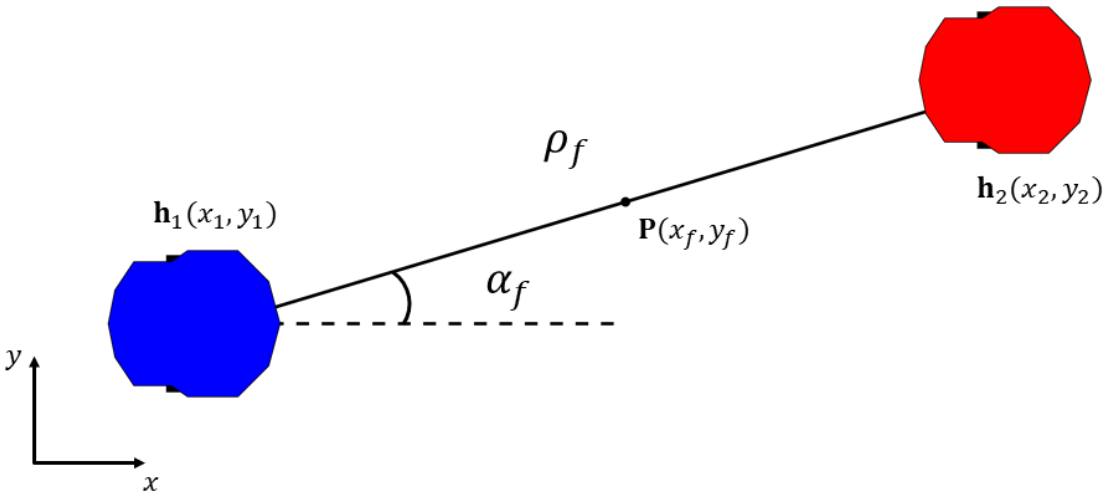


Figura 20 – Formação planar (2D).

Conforme mostrado na figura, pode-se entender a formação de dois robôs como sendo um corpo cujas dimensões e posição no plano devem ser controladas. Neste contexto apresenta-se a ideia de estrutura virtual, a qual representa uma forma geométrica cujas dimensões e posicionamento do plano devem ser mantidos sob controle. Primeiramente, escolhe-se um ponto o qual será nomeado como *ponto de controle*, e a partir deste ponto mapeia-se a posição de cada robô em relação a ele, através de relações geométricas.

Os pontos $\mathbf{h}_1 = \begin{bmatrix} x_1 & y_1 \end{bmatrix}^T$ e $\mathbf{h}_2 = \begin{bmatrix} x_2 & y_2 \end{bmatrix}^T$ representam as posições dos dois robôs em relação ao sistema de coordenadas global. Utilizando geometria, pode-se estabelecer relações entre o ponto de controle e as posições de cada robô, as quais formam a base para a atuação conjunta dos membros da formação.

A formação em 2D pode ser entendida como sendo um conjunto de variáveis que descrevem as relações entre os robôs no plano, a partir do ponto de controle $\begin{bmatrix} x_f & y_f \end{bmatrix}^T$. Este ponto é representado por um ponto tomado sobre o eixo x^r do robô (eixo na referência do robô), a uma distância a do eixo virtual que une as suas rodas de tração, e a partir deles se obterão as relações geométricas que caracterizam a forma da formação, a saber, a distância entre os dois robôs, denominada ρ_f, e o ângulo que a linha que une os robôs faz com o eixo das abscissas do sistema global de coordenadas, denominado α_f.

Portanto, a formação é composta por duas variáveis de *posição* (x_f, y_f) e duas variáveis de *forma* (ρ_f,α_f), que correspondem a uma distância e a um ângulo, respectivamente. Expressando de forma matemática, a formação em duas dimensões é representada por um vetor nomeado $\mathbf{q}_f$ composto por quatro elementos de controle, conforme a equação

$$q_f = \begin{bmatrix} x_f & y_f & \rho_f & \alpha_f \end{bmatrix}^T,\ x_f, y_f \in \mathcal{R},\ \rho_f \in \mathcal{R}^+ \text{ e } \alpha \in [0, 2\pi] \tag{3.1}$$

Note-se que a distância ρ_f está contida no conjunto dos números reais positivos incluindo o zero pois, para que haja uma representação física, a distância de dois corpos não pode assumir valores negativos.

A posição dos robôs em relação ao sistema de coordenadas global, as variáveis dos robôs, são representadas pelo vetor $\mathbf{x}$, sendo

$$\mathbf{x} = \begin{bmatrix} \mathbf{h}_1^T & \mathbf{h}_2^T \end{bmatrix}^T = \begin{bmatrix} x_1 & y_1 & x_2 & y_2 \end{bmatrix}^T, x_1, y_1, x_2, y_2 \in \mathcal{R}. \tag{3.2}$$

As variáveis de formação podem ser expressas em função das variáveis dos robôs que a compõem. Neste caso tem-se uma transformação de variáveis na qual a entrada dessa função é composta pelas variáveis dos robôs $\mathbf{x}$, representada pelo vetor da Equação 3.2, e sua saída são as variáveis de formação $\mathbf{q}_f$, representada pelo vetor da Equação 3.3. Então, ao aplicar uma transformação direta se obtém as variáveis de formação a partir das variáveis dos robôs. Tal transformação é descrita de forma matemática através da equação

$$\mathbf{q}_f = f(\mathbf{x}), \tag{3.3}$$

onde $f(x)$ é expressa de forma detalhada como

$$x_f = \frac{x_1 + x_2}{2} \tag{3.4}$$

$$y_f = \frac{y_1 + y_2}{2} \tag{3.5}$$

$$\rho_f = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.6}$$

$$\alpha_f = tg^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right). \tag{3.7}$$

Seguindo o raciocínio de relacionamento entre as variáveis de formação e variáveis dos robôs, uma transformação inversa obtém as variáveis dos robôs a partir das variáveis da formação, transformação esta representada de forma matemática como

$$\mathbf{x} = f^{-1}(\mathbf{q}_f), \tag{3.8}$$

onde $f^{-1}(\mathbf{q}_f)$ é expressa de forma detalha como

$$x_1 = x_f - \frac{\rho_f}{2}\cos(\alpha_f) \tag{3.9}$$

$$y_1 = y_f - \frac{\rho_f}{2}\,\text{sen}(\alpha_f) \tag{3.10}$$

$$x_2 = x_f + \frac{\rho_f}{2}\cos(\alpha_f) \tag{3.11}$$

$$y_2 = y_f + \frac{\rho_f}{2}\,\mathrm{sen}(\alpha_f). \tag{3.12}$$

Através das transformações direta e inversa pode-se calcular em cada instante de tempo as variáveis de formação e posições desejadas dos robôs, o que possibilita projetar controladores que sejam capazes de identificar o erro de formação a cada instante, e atuar nos robôs de forma a reduzir tais erros.

Os conhecimentos apresentados nesta seção representam os alicerces para estudo e compreensão geral de formações de robôs. Mais detalhes serão apresentados nas próximas seções.

3.3 Formação Espacial (3D)

A formação de robôs em 3D representa uma generalização da formação em 2D explicada na seção anterior. Neste novo contexto os elementos da formação possuem mais graus de liberdade. Para ser mais preciso, são acrescidos três graus de liberdade ao sistema. Há casos em que se pode ter graus de liberdade fixos, como visto em uma formação homogênea composta de dois *Pioneers* se movendo em um plano: apesar de se ter a representação de todos os graus de liberdade, três são fixos.

O escopo desta dissertação compreende uma formação heterogênea composta por um robô terrestre (*Pioneer 3-Dx*) e um quadrimotor (*Bebop 2*) que se deslocam no espaço, daí a designação de formação 3D.

Para explicar a formação em 3D, se extrapolarão os conceitos discutidos para a formação 2D. Assim é que no novo contexto a estrutura virtual gerada pela formação de dois ou mais robôs pode se mover em até seis graus de liberdade.

Os mesmos conceitos utilizados para se obter as variáveis de formação a partir das variáveis dos robôs são aplicados no contexto de formações em 3D. A referida formação, que será utilizada deste ponto em diante ao longo da dissertação, está representada na Figura 21. Nela a posição dos robôs em relação ao sistema de coordenadas global é dada pelos vetores $\mathbf{h}_1 = \begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$ e $\mathbf{h}_2 = \begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}^T$, devendo ser mencionado que embora se admita um valor z_1 qualquer, dado que o primeiro robô não se desloca do solo, se terá sempre $z_1 = 0$. Porém, se manteve o valor z_1 na formulação do problema para permitir que essa mesma formulação seja adotada para uma formação homogênea de dois VANTs.

Assim como a formação 2D, a formação em 3D pode ser entendida como sendo um conjunto de variáveis que descrevem as relações entre os robôs no espaço em relação ao ponto de controle $\begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^T$. Para o caso em estudo, ele está posicionado sobre a estrutura do robô terrestre (o *Pioneer 3 DX*), sobre seu eixo x^r, a uma distância a do ponto central do eixo virtual que une as rodas de tração, e a partir dele se obtém as relações geométricas que descrevem a forma da formação. A distância a é determinada por um valor não nulo de forma a assegura a matriz de de cinemática inversão seja não singular.

A Figura 21 ilustra o caso em estudo nesta dissertação. Nela tem-se que ρ_f representa a distância entre os dois robôs, α_f é o ângulo entre a projeção no plano $x-y$ da linha imaginária que une o *Pioneer* ao quadrimotor e o eixo global das abscissas. Em complemento ao modelo 2D tem-se agora o ângulo β_f, que corresponde ao ângulo formado pela reta imaginária que une o *Pioneer* ao quadrimotor e sua projeção no plano $x-y$ do sistema global de coordenadas. Portanto, a formação em 3D é descrita por três variáveis que descrevem sua posição, dadas por $\begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^T$, e três variáveis de forma representadas por $\begin{bmatrix} \rho_f & \alpha_f & \beta_f \end{bmatrix}^T$.

Expressando de forma matemática, a formação em três dimensões é representada por um vetor $\mathbf{q}_f$ composto por seis elementos de controle conforme a equação

$$\mathbf{q}_f = \begin{bmatrix} x_f & y_f & z_f & \rho_f & \alpha_f & \beta_f \end{bmatrix}^T, \tag{3.13}$$

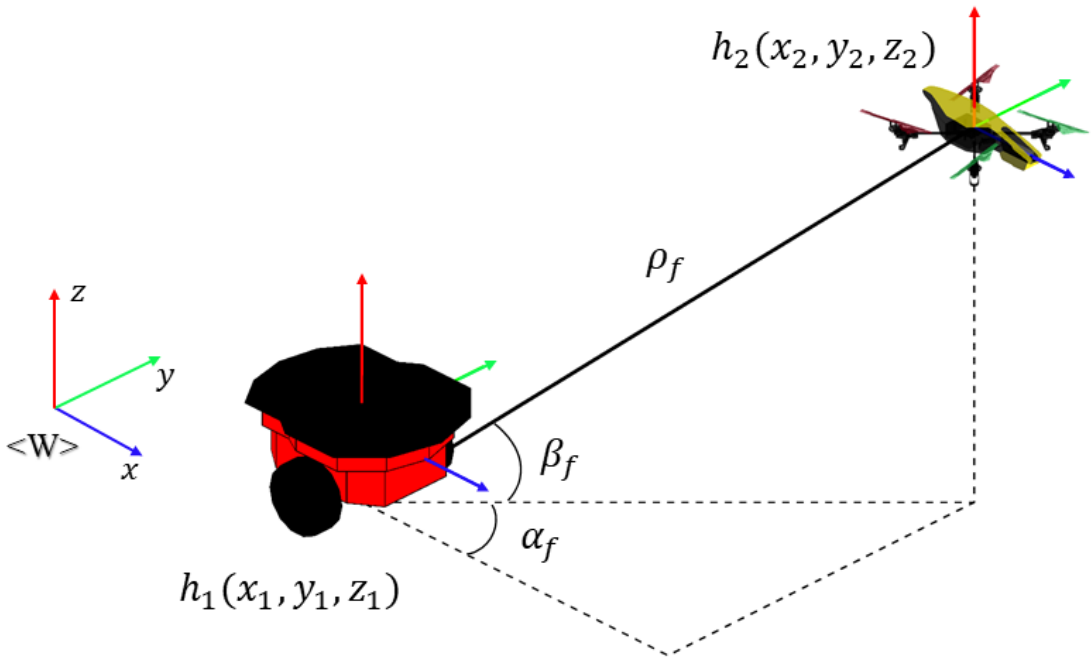


Figura 21 – Formação espacial (3D)

onde $(x_f, y_f) \in \mathcal{R}, (z_f, \rho_f) \in \mathcal{R}^+, \alpha_f \in [0, 2\pi]$ e $\beta_f \in [0, \pi]$. Note-se que a distância ρ_f e a altitude z_f estão contidas no conjunto dos números reais positivos incluindo o zero, uma vez que representam variáveis físicas, α_f pode assumir qualquer valor no intervalo de 0 a 2π e β_f, no caso estudado, está limitado entre $[0 \quad \pi]$. Contudo, para uma formação composta por dois quadrimotores esta restrição não mais seria válida uma vez que ambos os robôs podem mover-se no espaço em três dimensões.

A posição dos robôs em relação ao sistema de coordenadas global é dada pelo vetor $\mathbf{x}$ definido como

$$\mathbf{x} = \begin{bmatrix} \mathbf{h}_1^T & \mathbf{h}_2^T \end{bmatrix}^T = \begin{bmatrix} x_1 & y_1 & z_1 & x_2 & y_2 & z_2 \end{bmatrix}^T. \tag{3.14}$$

Assim como no modelo bidimensional, as relações de transformação direta, ou seja, obtenção das variáveis de formação a partir das variáveis dos robôs, e transformação inversa, ou seja, obtenção das variáveis dos robôs em função das variáveis da formação, também são aplicadas ao caso presente, com a diferença que agora mais três graus de liberdade são adicionados.

De forma análoga, ou seja, usando relações geométricas, sendo que agora a descrição deixa de ser por coordenadas polares e passa a ser por coordenadas esféricas, a transformação direta é dada por

$$\mathbf{q}_f = f(\mathbf{x}), \tag{3.15}$$

onde

$$x_f = x_1, \tag{3.16}$$

$$y_f = y_1, \tag{3.17}$$

$$z_f = z_1, \tag{3.18}$$

$$\rho_f = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \tag{3.19}$$

$$\alpha_f = \text{tg}^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right), \tag{3.20}$$

$$\beta_f = \text{tg}^{-1}\left(\frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\right). \tag{3.21}$$

Por sua vez, a transformação inversa é dada por

$$\mathbf{x} = f^{-1}(\mathbf{q}_f), \tag{3.22}$$

onde

$$x_1 = x_f, \tag{3.23}$$

$$y_1 = y_f, \tag{3.24}$$

$$z_1 = z_f, \tag{3.25}$$

$$x_2 = x_f + \rho_f \cos(\alpha_f)\cos(\beta_f), \tag{3.26}$$

$$y_2 = y_f + \rho_f \,\text{sen}(\alpha_f)\cos(\beta_f), \tag{3.27}$$

$$z_2 = z_f + \rho_f \,\text{sen}(\beta_f). \tag{3.28}$$

Este conjunto de relações geométricas forma as bases para o controle de robôs em formação, problema em que o projeto de controladores tem como objetivo manter as variáveis de formação dentro de limites estabelecidos, ou seja, o sinal de controle enviado individualmente a cada robô visa reduzir o erro de formação da estrutura virtual por eles formada, no caso a linha reta que une os dois veículos.

4 Controle de Formação

Este capítulo descreve a metodologia utilizada para controlar dois ou mais robôs atuando de forma cooperativa. Os dois veículos conjuntamente são caracterizados por uma forma geométrica, que é a estrutura virtual adotada (o controle é baseado no paradigma de estrutura virtual). No caso presente, tal estrutura virtual é a reta imaginária que conecta os pontos de controle dos dois veículos. Serão abordadas as relações matemáticas utilizadas nos controladores e os respectivos comportamentos obtidos segundo cada estratégia adotada no projeto do controlador. São estudadas três estratégias, a saber o Controle Clássico, em que as tarefas de controlar a posição e a forma da formação são realizadas simultaneamente, e duas estratégias baseadas no uso da técnica de espaço nulo para priorizar uma dessas duas tarefas, a saber, Controle Baseado em Espaço Nulo com Prioridade de Posição e Controle Baseado em Espaço Nulo com Prioridade de Forma. A partir dos resultados, será discutido qual dos modelos melhor atende às necessidades de movimentação de carga, foco de estudo desta dissertação.

Na próxima seção serão listadas as principais estratégias de controle comumente utilizadas na literatura, enfatizando-se principalmente o modelo utilizado na pesquisa.

4.1 Filosofia de Controle

O controle de uma formação consiste em movimentar um conjunto de robôs de maneira coordenada, mantendo uma forma geométrica definida pelo número de agentes. A movimentação dessa estrutura possui dois aspectos básicos, que são mover a estrutura e mover os robôs que compõem a estrutura. O objetivo é relacionar como os robôs devem se mover, individualmente falando, para manter a formação ao longo da navegação em busca de uma posição e forma desejadas para ela.

Para a tarefa de movimentação, usualmente se utilizam três abordagens de controle, que são a formação líder-seguidor (BRANDÃO et al., 2009; BRANDÃO; SARCINELLI-FILHO; CARELLI, 2013b; BOEL; MARINICA; SARLETTE, 2015; LIANG et al., 2016; DEHGHANI; MENHAJ; AZIMI, 2016), a abordagem via estruturas virtuais (LEWIS; TAN, 1997) (KITTS; MAS, 2009) (LOW, 2014) e os métodos baseados em comportamento (BROOKS, 1986; ARKIN, 1990; BALCH; ARKIN, 1998; ANTONELLI; ARRICHIELLO; CHIAVERINI, 2010). Em adição a esta classificação, ainda pode-se ter um *Controle Centralizado* (BRANDÃO; SARCINELLI-FILHO, 2016; BRANDÃO et al., 2015) ou *Controle Descentralizado* (BRANDÃO et al., 2009).

Neste estudo, utiliza-se o paradigma de estruturas virtuais, operando em uma arquitetura de processamento centralizado. Para efeitos de comparação, serão adotados três tipos de controladores que modificam a forma como os robôs se relacionam entre si. Os resultados obtidos serão comparados e o desempenho de cada controlador avaliado segundo a necessidade de controle descrita no Capítulo1, Seção 1.1 (Definição do Problema).

Conforme dito na introdução deste capítulo, os controles utilizados são classificados como *Controle Clássico*, *Controle Baseado em Espaço Nulo com Prioridade de Forma* e *Controle Baseado em Espaço Nulo com Prioridade de Posição*, e lidam com a tarefa de manter a forma e buscar uma posição desejada para a formação considerando-a como duas sub-tarefas, sendo que no primeiro caso as duas sub-tarefas são tratadas simultaneamente, sem priorização de nenhuma delas, enquanto que no segundo caso a sub-tarefa de manter a forma da formação tem prioridade mais alta e no último caso mover a formação para uma posição desejada tem a maior prioridade. Controladores baseados em espaço nulo, como nos dois últimos casos, possuem a

capacidade de hierarquizar comportamentos o que significa que uma tarefa será executada com maior ênfase, em detrimento de outra que esteja competindo com ela (ANTONELLI; ARRICHIELLO; CHIAVERINI, 2010; ROSALES et al., 2016; MANSFELD et al., 2017).

Controles baseados em comportamentos podem ser entendidos como um controle de fluxo matemático, uma espécie de se/senão (*if/else*) que realiza o chaveamento entre as tarefas priorizadas, de acordo com o erro de *forma* ou *posição* da formação (BALCH; ARKIN, 1998; ANTONELLI; ARRICHIELLO; CHIAVERINI, 2010; ROSALES et al., 2016). Quando se deseja priorizar a manutenção da forma da formação o controlador enfatiza a redução do erro de forma e, portanto, o comportamento esperado dos robôs é sua movimentação de maneira a compor a estrutura geométrica virtual o mais rápido possível. Assim que o erro de formação está reduzido, a segunda tarefa é executada com maior ênfase o que acelera a movimentação da formação em direção à posição desejada.

4.2 Sistema de Controle Multicamadas

O sistema de controle multicamadas tem início com base nos trabalhos de Christopher Kitts e Ignacio Mas (2009). Em seus estudos foi utilizado um tipo de controle baseado em estruturas virtuais, criando-se o assim denominado *Cluster Space* (espaço do agrupamento, ou espaço da formação) onde o ponto de controle, para uma formação de três robôs é o centroide da figura geométrica triangular formada pelos robôs. Mais trabalhos se embasaram nessa filosofia, tais como (RESENDE; CARELLI; SARCINELLI-FILHO, 2014), (BRANDÃO et al., 2015), (ADAMEK; KITTS; MAS, 2015), (BRANDÃO; SARCINELLI-FILHO, 2016) e (ROSALES et al., 2016), os quais correspondem à base desse estudo. Porém, a estrutura triangular básica ali utilizada será adaptada para uma estrutura correspondente a uma linha reta (dois robôs, e não mais três), como também foi feito em (**??**), sendo o sistema de controle multicamadas alterado para aquele ilustrado no diagrama da Figura 22, o qual foi estruturado a partir da ideia do sistema de controle multicamadas, este ilustrado na Figura 23.

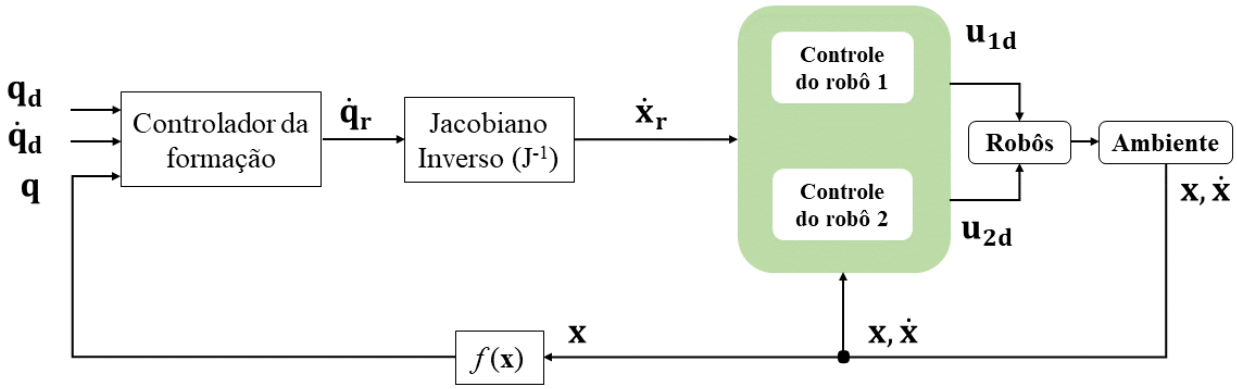


Figura 22 – Diagrama de controle detalhado.

Como pode ser visto na Figura 23 a tarefa de controle é disposta como uma sucessão de etapas, organizadas segundo camadas, cada uma delas responsável por executar uma função específica, como planejar *o que*, *como* e *quando* deve ser realizada uma determinada ação. Os tópicos abaixo apresentam, com mais detalhes, a ação correspondente a cada camada.

- Camada de Planejamento *Off-line*

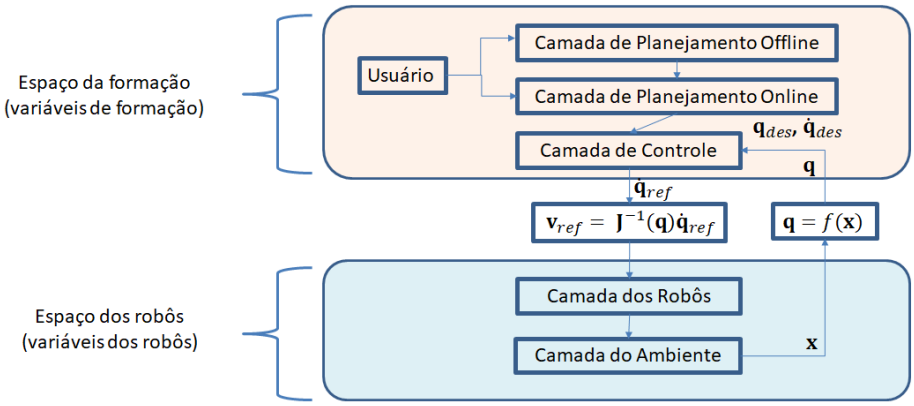


Figura 23 – Fluxograma correspondente ao sistema de controle multicamadas

O Planejamento *Off-line* é realizado previamente ao *Loop de Controle*. Nele são definidos os objetivos da formação e como ela deve se mover ao passar do tempo. De forma explícita, define-se como $\mathbf{q}_d$ e $\dot{\mathbf{q}}_d$ devem variar ao longo do tempo. Pode-se lidar com duas situações, que são o controle de posicionamento, em que a forma geométrica virtual desejada, no caso a reta que une os dois veículos, deve buscar uma posição fixa no espaço, e o controle de seguimento de trajetória, em que a formação deve seguir uma trajetória definida no espaço. Para o caso de seguimento de trajetória, $\mathbf{q}_d$ varia no tempo conforme uma função predefinida, e $\dot{\mathbf{q}}_d$ é a primeira derivada de tal função, ou seja, a velocidade da trajetória. Para o caso de posicionamento, que é o caso abordado nesta dissertação, $\mathbf{q}_d$ é um valor fixo que define a forma e a posição desejadas para a formação, com $\dot{\mathbf{q}}_d = \mathbf{0}$, configuração esta que garante que a formação busque a posição desejada o mais rápido possível.

As posições iniciais dos robôs $\mathbf{x}_0$ também são fornecidas nesta etapa. No experimento executado esta informação é proveniente do sistema *OptiTrack* descrito na Seção 2.4.1. Contudo, ela também pode ser proveniente de outras fontes como GPS, no caso de navegação em ambientes exteriores, ou mesmo ajustada manualmente.

- Camada de Planejamento *On-line*

  Esta camada é responsável pelas ações estratégicas que são executadas no interior do *loop de controle*. Um exemplo seria uma possível alteração da formação ao longo da navegação. Outro exemplo seria qual ação a formação iria executar ao encontrar um obstáculo. Nesse caso, pode-se considerar uma formação rígida, quando todos os robôs se movem de forma coordenada para desviar do obstáculo mantendo a forma geométrica, ou uma formação flexível, permitindo a deformação da forma geométrica formada pelos robôs de forma que a nova forma seria capaz de transpor o obstáculo sem contorná-lo.

- Camada de Controle

  A Camada de Controle está inserida no *loop de controle*, e é responsável por coordenar o deslocamento da formação e dos robôs segundo especificado na camada anterior. Aqui são geradas as velocidades $\dot{\mathbf{q}}_r$ de referência da formação, ou seja, um controlador de formação define como a formação como um todo deve se mover para buscar a posição desejada e estabelecer a forma desejada.

Os erros de posição e forma da formação são obtidos a partir da comparação dos valores desejados $\mathbf{q}_{des}$ e aqueles atuais $\mathbf{q}$, estes últimos obtidos a partir da informação realimentada pelo sistema *OptiTrack*. O sinal de controle $\dot{\mathbf{q}}_r$ é gerado, o qual vai ser utilizado como referência para o controle de cada um dos robôs da formação. Para tanto, as velocidades de referência para os robôs, em coordenadas globais, são obtidas através da matriz inversa do Jacobiano correspondente à transformação das variáveis dos robôs para as variáveis da formação, descrita pela Equação (3.15).

- Camada dos Robôs

  Nesta camada, os sinais de referência de velocidade, ainda em coordenadas globais, são utilizados para efetuar o controle individual dos robôs, que são, assim, comandados como num seguimento de trajetória, já que recebem como referência velocidades (RESENDE; CARELLI; SARCINELLI-FILHO, 2014). Para tanto, o primeiro passo é aplicar a cinemática inversa de cada robô às referências de velocidade que lhe correspondem, conforme as equações (2.5) e (2.11), obtendo-se assim as referências de velocidade em coordenadas dos veículos, para então serem aplicadas a eles.

  Nesse momento, as referências de velocidade já podem ser aplicadas aos robôs. Entretanto, também se pode optar por compensar a dinâmica de cada um dos robôs, para se ter um erro de seguimento de velocidade (que é a diferença entre a velocidade comandada e aquela efetivamente desenvolvida pelo robô). Nesse caso os sinais de velocidade são ajustados por um compensador dinâmico antes de efetivamente serem enviados aos robôs, obtendo-se assim os sinais $\mathbf{u}_{1d}$ e $\mathbf{u}_{2d}$ que são então enviados aos robôs, conforme mostra a Figura 22 (ver (MARTINS, 2009) e (SANTOS et al., 2019), para detalhes sobre tais compensadores dinâmicos).

- Camada do Ambiente

  Aqui são obtidas as informações de posições momentâneas dos robôs, através de sensores embarcados ou localizados no espaço físico de teste. Os dados coletados nesta camada são utilizados como referência para se obter o erro de formação na camada de controle.

Para efeito didático e melhor compreensão da estrutura, segue-se uma iteração do controle, pontuando as principais ações executadas em um ciclo do *loop* de controle.

O controlador de formação, que está na Camada de Controle, e que é um controlador apenas cinemático, já que a estrutura virtual é considerada como um agente sem massa, e portanto sem inércia, toma as variações temporais $\dot{\mathbf{q}}_d$ e configuração $\mathbf{q}_d$ desejadas para a formação, provenientes da camada de Planejamento *Offline* ou da camada de Planejamento *Online*, e a posição e forma atuais da formação $\mathbf{q}$, provenientes da camada do Ambiente. De posse dessas informações o controlador calcula os erros de formação $\tilde{\mathbf{q}}$, gerando em seguida as referências de velocidades $\dot{\mathbf{q}}_r$ que vão definir como os dois robôs devem mover-se. Este vetor de referência é multiplicado pela matriz inversa do Jacobiano, resultando em velocidades de referência $\dot{\mathbf{x}}_r$, ainda em coordenadas globais.

O controle dos robôs multiplicaria as referências de velocidade em coordenadas globais correspondentes a cada robô pela sua matriz de cinemática inversa, gerando as referências de velocidade em coordenadas locais, a serem enviadas aos robôs correspondentes, no caso de se adotar apenas o controle cinemático. Entretanto, antes de enviar tais comandos de velocidade aos robôs, seria possível fazê-los passar por um compensador dinâmico, conforme descrito em (MARTINS, 2009) para o caso do *Pioneer 3-DX* e em (SANTOS et al., 2019) para o caso do drone, gerando-se assim os comandos de velocidade $\mathbf{u}_d = \begin{bmatrix} \mathbf{u}_{1d}^T & \mathbf{u}_{2d}^T \end{bmatrix}^T$ efetivamente enviados aos robôs.

Porém, no caso dessa dissertação foi adotada uma estratégia diferente, tanto no caso do VTNT quanto do VANT. Para o *Bebop 2* utilizou-se o controlador dinâmico caracterizado na Equação (2.14), proposto originalmente em (SANTANA et al., 2014), obtendo-se o sinal $\mathbf{u}_{2d}$. Para isso, utilizou-se como informação a referência de velocidade $\dot{\mathbf{X}}_r$, o valor $\mathbf{X}_r$ obtido integrando-se numericamente $\dot{\mathbf{X}}_r$, para caracterizar as posições desejadas dos robôs, as posições reais dos robôs no sistema de coordenadas global, dadas por $\mathbf{X}$, medidas através do sistema *OptiTrack*, e sua derivada $\dot{\mathbf{X}}$, obtida numericamente a partir de $\mathbf{X}$.

No que se refere ao *Pioneer 3-DX*, foi utilizado simplesmente um controlador cinemático, que gera o sinal $\mathbf{u}_{1d}$ considerando como entradas a velocidade desejada para o robô $\dot{\mathbf{X}}_r$, novamente a posição desejada $\mathbf{X}_r$, obtida integrando-se numericamente $\dot{\mathbf{X}}_r$, e os valores $\mathbf{X}$, obtidos através do sistema *OptiTrack*. A partir de tais valores se obtém o comando $\mathbf{u}_{1d}$ como

$$\mathbf{u}_{1d} = \mathbf{A}_P^{-1}\left(\dot{\mathbf{X}}_r + \mathbf{K}_c \tanh\left(\tilde{\mathbf{X}}\right)\right), \tag{4.1}$$

onde $\mathbf{A}_P^{-1}$ é a matriz de cinemática inversa do *Pioneer 3-DX*, caracterizada em (2.6), $\tilde{\mathbf{X}} = \mathbf{X}_r - \mathbf{X}$ é o erro de posição correspondente ao VTNT e $\mathbf{K}_c$ é uma matriz de ganhos diagonal e positiva definida.

Assim, os sinais $\mathbf{u}_{1d}^T$ e $\mathbf{u}_{2d}^T$ são enviados aos respectivos robôs, o ciclo atual se encerra e uma nova iteração se inicia, até que uma condição de parada seja atingida. Tal condição, no caso presente, é quando os valores absolutos dos erros de formação fiquem abaixo de um limiar predefinido.

4.3 O Controle da Formação

Nesta seção se detalharão as estratégias de controle tratadas, pontuando as diferenças entre a utilização do controle clássico e o controle baseado em comportamento utilizando espaço nulo.

4.3.1 Controlador Clássico

Para o caso em estudo, o objetivo é controlar dois robôs de suas posições iniciais até suas respectivas posições finais, mantendo uma formação específica ao longo da navegação. Do ponto de vista matemático, deve-se manter a linha imaginária e ângulos correspondente à estrutura virtual com o mínimo erro possível. Para que esta tarefa seja executada, monitoram-se as variáveis $\mathbf{x}$ correspondentes à pose dos robôs, provenientes da camada Ambiente, as variáveis de formação $\mathbf{q}$ e a posição e forma desejadas da formação $\mathbf{q}_d$ provenientes da camada de planejamento.

Na abordagem clássica, o controle da formação é executado contemplando uma única tarefa a ser controlada, ou seja, controla-se todo o vetor $\mathbf{q} = \begin{bmatrix} x_f & y_f & z_f & \rho_f & \alpha_f & \beta_f \end{bmatrix}^T$ de uma só vez, conforme é feito em (RABELO; BRANDÃO; SARCINELLI-FILHO, 2018).

O projeto do controlador proposto considera a matriz Jacobiana correspondente à derivada da equação (3.15), dada por $\mathbf{J}(\mathbf{x})$ em

$$\dot{\mathbf{q}} = \mathbf{J}(\mathbf{x})\dot{\mathbf{x}}. \tag{4.2}$$

Tal matriz possui dimensão 6×6, e é definida como

$$\mathbf{J}(\mathbf{x}) = \frac{\partial \mathbf{q}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial x_f}{\partial x_1} & \frac{\partial x_f}{\partial y_1} & \cdots & \frac{\partial x_f}{\partial z_2} \\ \frac{\partial y_f}{\partial x_1} & \frac{\partial y_f}{\partial y_1} & \cdots & \frac{\partial y_1}{\partial z_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \alpha_f}{\partial x_1} & \frac{\partial \alpha_f}{\partial y_1} & \cdots & \frac{\partial \alpha_f}{\partial z_2} \end{bmatrix},$$

o que, no caso presente, corresponde a

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{d_x}{r} & \frac{d_y}{r} & \frac{d_z}{r} & \frac{-d_x}{r} & \frac{-d_y}{r} & \frac{-d_z}{r} \\ \frac{-d_y}{d^2} & \frac{d_x}{d^2} & 0 & \frac{d_y}{d^2} & \frac{-d_x}{d^2} & 0 \\ \frac{d_x d_z}{r^2 d} & \frac{d_y d_z}{r^2 d} & \frac{-d}{r^2} & \frac{-d_x d_z}{r^2 d} & \frac{-d_y d_z}{r^2 d} & \frac{d}{r^2} \end{bmatrix}, \tag{4.3}$$

sendo $d_x = x_2 - x_1, d_y = y_2 - y_1$ e $d_z = z_2 - z_1$, $r = \rho_f$ e $d = \sqrt{d_x^2 + d_y^2}$.

As derivadas no tempo $\dot{\mathbf{q}}_d$ das variáveis de formação desejadas e o erro de formação $\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}$ são usados na lei de controle

$$\dot{\mathbf{x}}_\mathbf{r} = \mathbf{J}^{-1}(\mathbf{x}) \left[\dot{\mathbf{q}}_d + \mathbf{k}_1 \tanh\left(\mathbf{k}_2 \tilde{\mathbf{q}}\right)\right] \tag{4.4}$$

para gerar as velocidades desejadas para os robôs individuais, onde $\mathbf{k}_1$ e $\mathbf{k}_2$ são matrizes de saturação e ganho, respectivamente, ambas diagonais e positivas definidas. Quanto à função tanh, ela é utilizada como um saturador suave, para impedir que erros de formação muito grandes gerem sinais de referência de velocidade que os robôs da formação não possam atingir. Salienta-se, por fim, que o sistema de controle de malha fechada constituído quando se adota tal lei de controle é estável, conforme provado em (RABELO; BRANDÃO; SARCINELLI-FILHO, 2018).

4.3.2 Controle Baseado em Espaço Nulo

A abordagem utilizando a metodologia de espaço nulo pode ser considerada como uma maneira analítica de implementar uma comutação lógica. Tal abordagem permite executar tarefas diferentes mesmo que elas sejam conflitantes. A estratégia utilizada consiste em dividir a tarefa a ser executada pela formação em subtarefas complementares, que devem ser executadas obedecendo uma hierarquia de execução (ANTONELLI; ARRICHIELLO; CHIAVERINI, 2010).

Considerando o mesmo objetivo citado no controle clássico, pode-se dividir a ação de controle em duas tarefas distintas, a saber, manter a forma da formação ($\mathbf{q}_f = \begin{bmatrix} \rho_f & \alpha_f & \beta_f \end{bmatrix}^T$), e controlar a posição da formação ($\mathbf{q}_p = \begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^T$).

Conforme ilustrado nas figuras 24 e 25, o controlador é projetado considerando a execução independente das tarefas, e os sinais de controle gerados pelas duas tarefas são então combinados para obter as referências

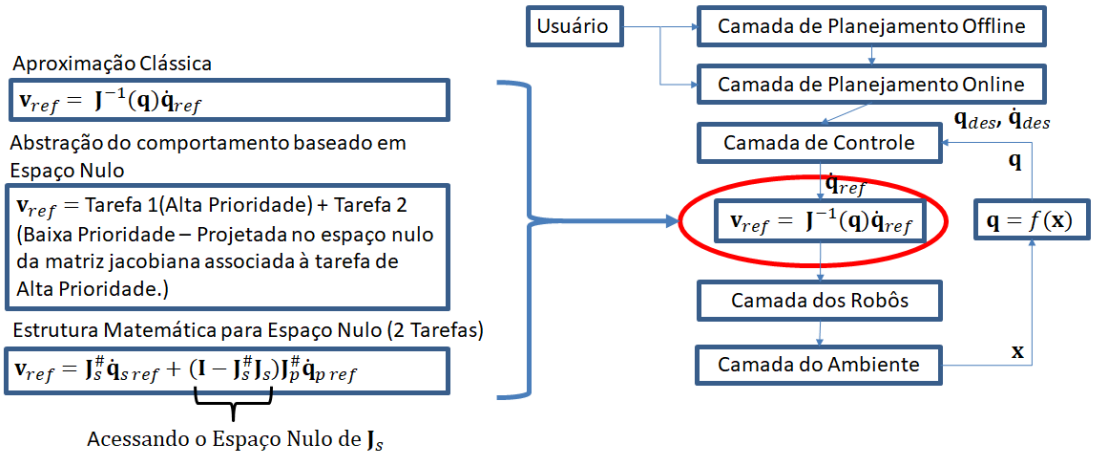


Figura 24 – Controle baseado em espaço nulo

de movimento dos robôs. Portanto, definem-se duas leis de controle, uma para controlar cada tarefa associada às variáveis de formação $\mathbf{q}_p$ e $\mathbf{q}_f$. De forma matemática elas podem ser representadas como

$$\dot{\mathbf{x}}_\mathbf{r} = \mathbf{J}_f^{\#}(\mathbf{x})\left[\dot{\mathbf{q}}_{fd} + \mathbf{k}_{1f}\tanh\left(\mathbf{k}_{2f}\tilde{\mathbf{q}}_f\right)\right] \tag{4.5}$$

e

$$\dot{\mathbf{x}}_\mathbf{r} = \mathbf{J}_p^{\#}(\mathbf{x})\left[\dot{\mathbf{q}}_{pd} + \mathbf{k}_{1p}\tanh\left(\mathbf{k}_{2p}\tilde{\mathbf{q}}_p\right)\right], \tag{4.6}$$

onde $\mathbf{J}_p$ e $\mathbf{J}_f$ são matrizes não quadradas obtidas particionando-se $\mathbf{J}(\mathbf{x})$, $\mathbf{J}_p$ correspondendo às três primeiras linhas de $\mathbf{J}(\mathbf{x})$ e $\mathbf{J}_f$ correspondendo às últimas três linhas de $\mathbf{J}(\mathbf{x})$, com $\mathbf{k}_{1f}$, $\mathbf{k}_{1p}$, $\mathbf{k}_{2f}$, $\mathbf{k}_{2p}$, $\mathbf{k}_1$ e $\mathbf{k}_2$ sendo matrizes de ganho/saturação diagonais e positivas definidas. Quanto a $\mathbf{J}_p^{\#}$ e $\mathbf{J}_f^{\#}$, elas são as pseudo-inversas Moore-Penrose de $\mathbf{J}_p$ e $\mathbf{J}_f$. A função tanh, por sua vez, é usada, mais uma vez, como uma função de saturação suave, para evitar que grandes valores de $\tilde{\mathbf{q}}$ gerem sinais de controle inviáveis para os robôs.

Depois de obter as leis de controle para as duas subtarefas, existem duas maneiras de compor o sinal de controle resultante: a primeira é conceder maior prioridade à tarefa de manter a forma da formação e a segunda é conceder maior prioridade para mover a formação. Neste trabalho o objetivo é priorizar a sub-tarefa de controlar a forma da formação, uma vez que se deseja assegurar as melhores condições para que o VANT pouse sobre o VTNT.

Para efeito de análise de desempenho, é realizada uma comparação entre o controlador baseado no espaço nulo e o controlador convencional (sem prioridade).

4.4 O Controlador Proposto

Para entender como é realizada a hierarquização de tarefas, é preciso entender o conceito matemático que realiza esta operação: a ação de atribuir menor prioridade a uma certa subtarefa significa projetá-la no espaço nulo da tarefa de prioridade imediatamente acima. Isto significa que a tarefa de menor prioridade será executada quando não conflitar com a de maior prioridade. Explicando de forma prática, o erro associado à tarefa de **maior** prioridade deve ser pequeno para que a tarefa de **menor** prioridade na hierarquia seja executada.

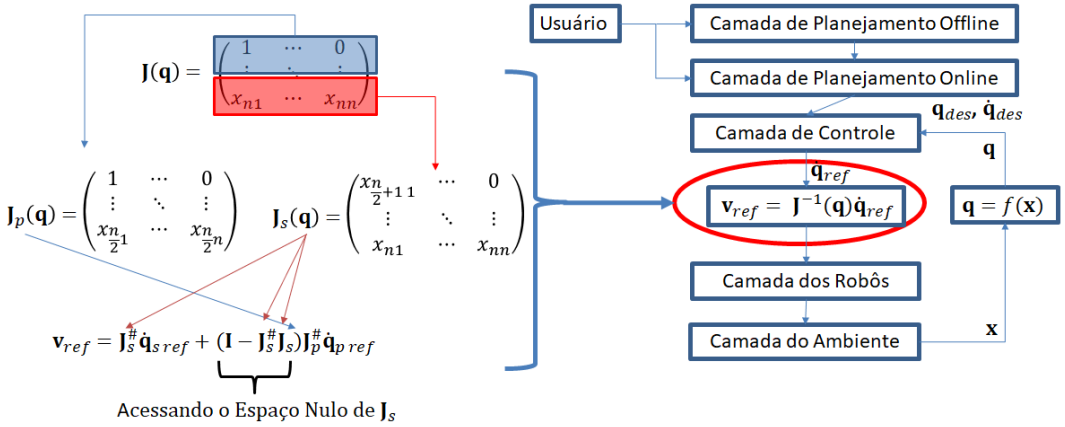


Figura 25 – Controle baseado em espaço nulo com prioridade de forma

As expressões matemáticas que realizam o comportamento descrito no parágrafo anterior são:

$$\dot{\mathbf{x}}_\mathbf{r} = \mathbf{v}_{fref} + (\mathbf{I} - \mathbf{J}_f^{\#}\mathbf{J}_f)\mathbf{v}_{pref}, \tag{4.7}$$

para atribuir a prioridade mais alta ao controle da forma da formação, ou

$$\dot{\mathbf{x}}_\mathbf{r} = \mathbf{v}_{pref} + (\mathbf{I} - \mathbf{J}_p^{\#}\mathbf{J}_p)\mathbf{v}_{fref}, \tag{4.8}$$

para atribuir a prioridade mais alta ao controle da posição da formação.

No primeiro e segundo casos, obtém-se as formações rígida e flexível respectivamente.

Para o primeiro caso, o sistema de controle referente à formação de um VTNT e um VANT, é aquele mostrado na Figura 26, que é o sistema de controle que se propõe para guiar o VTNT e o VANT em uma tarefa de posicionamento, e permitir o pouso do segundo no primeiro. Para implementar o segundo caso, basta alterar os índices $_f$ para $_p$ e $_p$ para $_f$ no diagrama. Quanto à análise de estabilidade desses dois controladores, o leitor deve consultar (ROSALES et al., 2016).

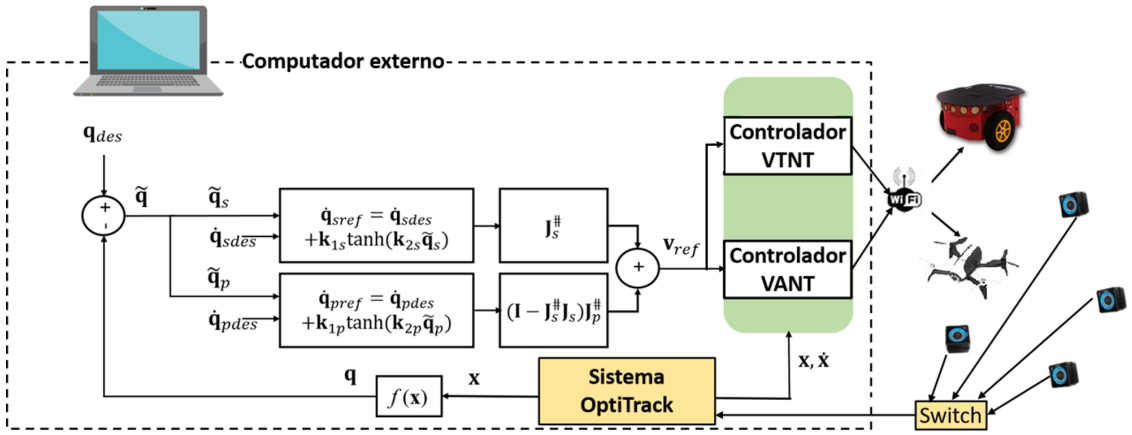


Figura 26 – Diagrama de blocos para o controlador proposto, que atribui maior prioridade ao controle da forma da formação.

5 Resultados

Este capítulo apresenta resultados experimentais obtidos com a formação já descrita, composta por um robô terrestre a rodas do tipo tração diferencial modelo *Pioneer 3-DX* e o quadrimotor modelo *Bebop 2*. Os resultados apresentados se referem a uma tarefa de posicionamento da formação.

Num primeiro momento se comparam os resultados obtidos com três controladores, a saber aquele proposto em (**??**) e os dois outros discutidos na Seção 4.4. O objetivo é verificar qual opção resulta num melhor desempenho quanto à forma da formação, visando assegurar um erro de forma o menor possível, para que o VANT esteja exatamente acima do VTNT no momento de pousar nele.

Num segundo momento é mostrado o resultado utilizando o controlador da Figura 26, aqui proposto, conduzindo a formação a uma posição desejada, na qual o VANT deve estar exatamente sobre o VTNT e pousar sobre o mesmo.

5.1 Comparação de Desempenho de Controladores

Nesta seção o controlador da Figura 26, doravante denominado Controle *NSB* (do Inglês Null Space Behavior) com Prioridade de Forma, é adotado para guiar a formação VTNT-VANT em uma tarefa de posicionamento.

Além do controlador da Figura 26 outros dois controladores são também implementados e testados utilizando tal formação, para fins de comparação de desempenho, a saber o controlador proposto em (**??**), daqui em diante denominado Controle Convencional, que considera a lei de controle em (4.4), e a configuração semelhante à da Figura 26 que implementa a lei de controle em (4.8), daqui em diante denominada Controle NSB com Prioridade de Posição. O objetivo é comparar o desempenho de tais controladores, para verificar, por exemplo, se dar maior prioridade para manter a forma da formação assegura desempenho adequado no que se refere ao erro de forma, quando se deseja fazer o VANT pousar sobre o VTNT.

As posições iniciais dos dois robôs da formação são $\mathbf{h}_{1_0} = \begin{bmatrix} -1\ m & -1\ m & 0\ m \end{bmatrix}$, para o *Pioneer 3-DX*, e $\mathbf{h}_{2_0} = \begin{bmatrix} -1\ m & 1\ m & 1\ m \end{bmatrix}$, para o *Bebop 2*. Quanto à posição e à forma desejada para a formação, elas são $\mathbf{q}_{des} = \begin{bmatrix} 0\ m & 1\ m & 0\ m & 1\ m & pi/3\ rad & 0\ rad \end{bmatrix}$. Como $\mathbf{q}_{des}$ é um vetor constante, dado que a tarefa a ser cumprida é um posicionamento, $\dot{\mathbf{q}}_{des} = \mathbf{0}$. Finalmente, as matrizes de ganho associadas aos controladores são $\mathbf{k}_{1p} = diag\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$, $\mathbf{k}_{1f} = diag\begin{bmatrix} 1 & 0,25 & 0,35 \end{bmatrix}$, $\mathbf{k}_{2p} = diag\begin{bmatrix} 0,25 & 0,1 & 0,1 \end{bmatrix}$, e $\mathbf{k}_{2f} = diag\begin{bmatrix} 0,1 & 0.25 & 0,2 \end{bmatrix}$. Note-se que $\mathbf{k}_{1p}(3,3) = 0$ porque $z_f = z_1 = 0$ todo o tempo, já que tal coordenada se refere ao VTNT. Por sua vez, $\mathbf{k}_1 = diag\begin{bmatrix} \mathbf{k}_{1p} & \mathbf{k}_{1f} \end{bmatrix}$ e $\mathbf{k}_2 = diag\begin{bmatrix} \mathbf{k}_{2p} & \mathbf{k}_{2f} \end{bmatrix}$.

Os trajetos dos robôs no experimento podem ser vistos nas Figuras 27, 28 e 29, que se referem ao Controle Convencional, ao Controle NSB com Prioridade de Forma, e ao Controle NSB com Prioridade de Posição, respectivamente.

Os resultados do experimento evidenciam uma diferença significativa de desempenho entre os controladores, como se pode ver comparando os gráficos das Figuras 30, 31 e 32, que mostram os erros nas variáveis de formação para os três casos analisados. Os ângulos que caracterizam a formação são menos estáveis para o Controle Convencional, em comparação com os do Controle NSB com Prioridade de Posição e do Controle NSB com Prioridade de Forma, especialmente este último.

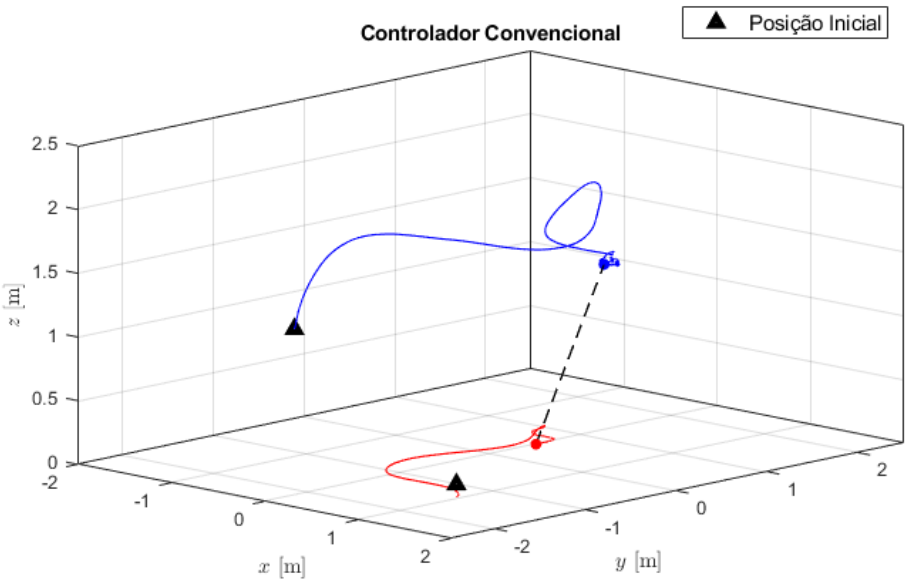


Figura 27 – Percursos dos robôs no Controle Convencional.

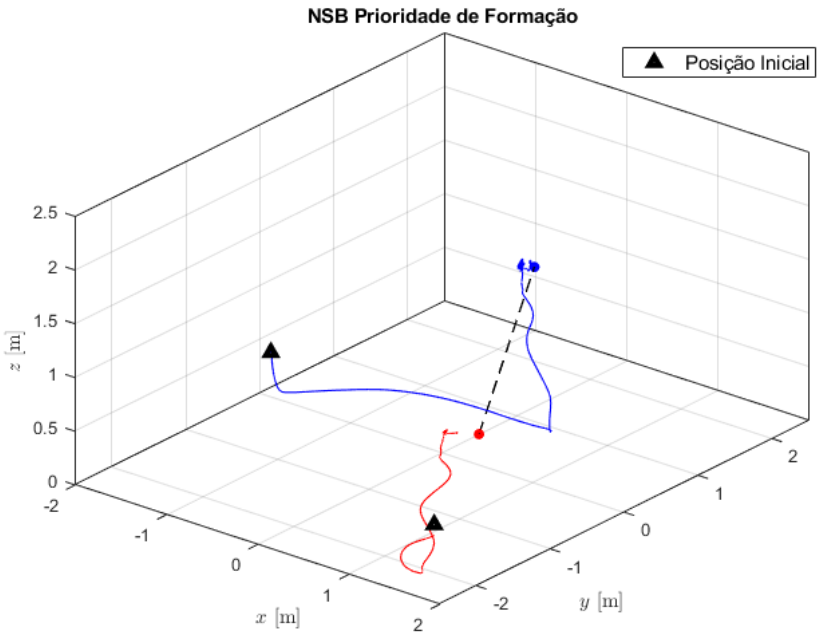


Figura 28 – Percursos dos robôs no Controle NSB com Prioridade de Forma.

Observando a Figura 30, o caso do Controle Convencional apresenta alta variação de ângulos para os parâmetros adotados, embora todos os erros tendam a zero ao longo do tempo. Quando manter a forma

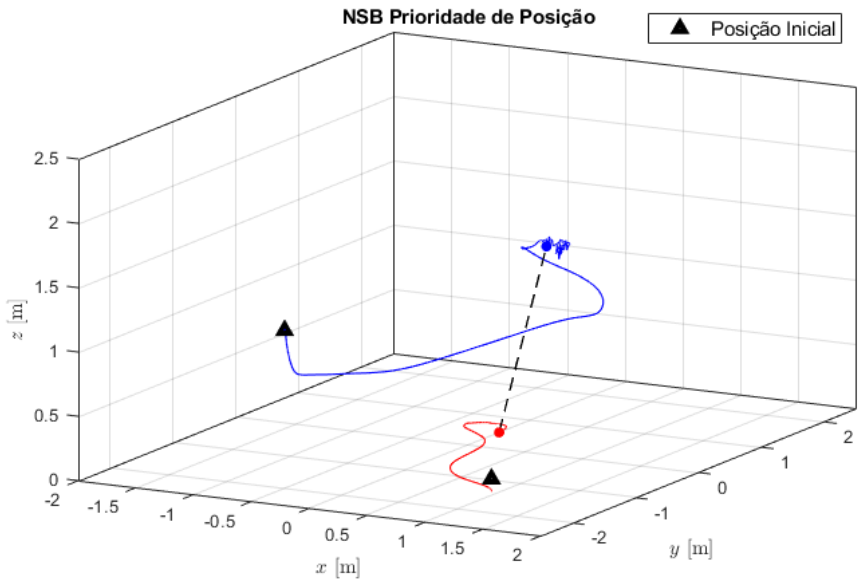


Figura 29 – Percursos dos robôs no Controle NSB com Prioridade de Posição.

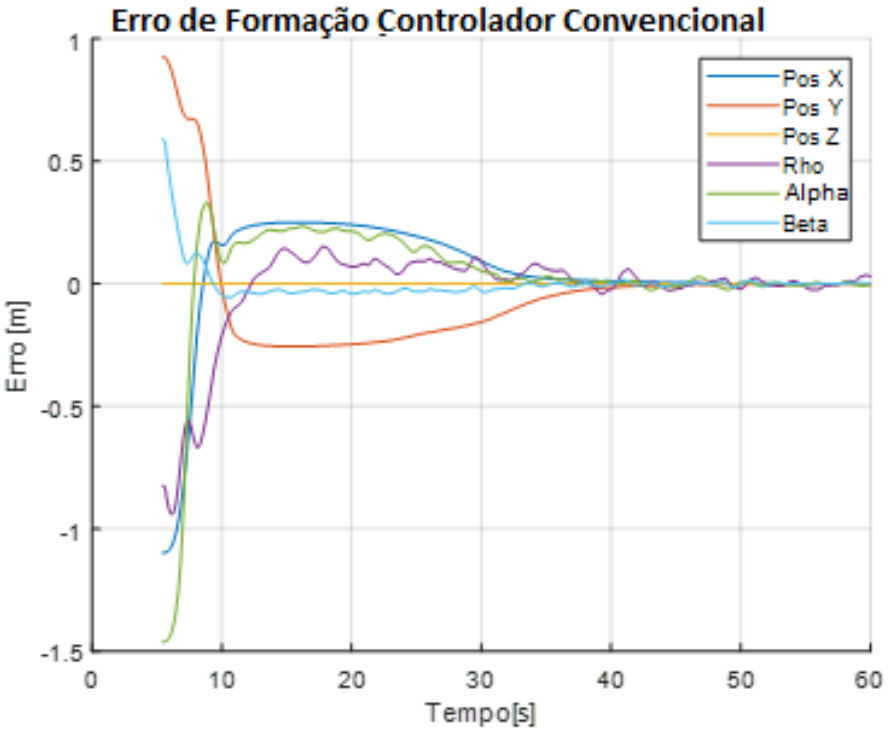


Figura 30 – Erros de formação para o Controle Convencional.

da formação é a prioridade mais alta, ou seja, no Controle NSB com Prioridade de Forma, ambos os robôs primeiro procuram um ao outro, tentando reduzir os erros de forma. Ambos os objetivos, reduzir os erros de

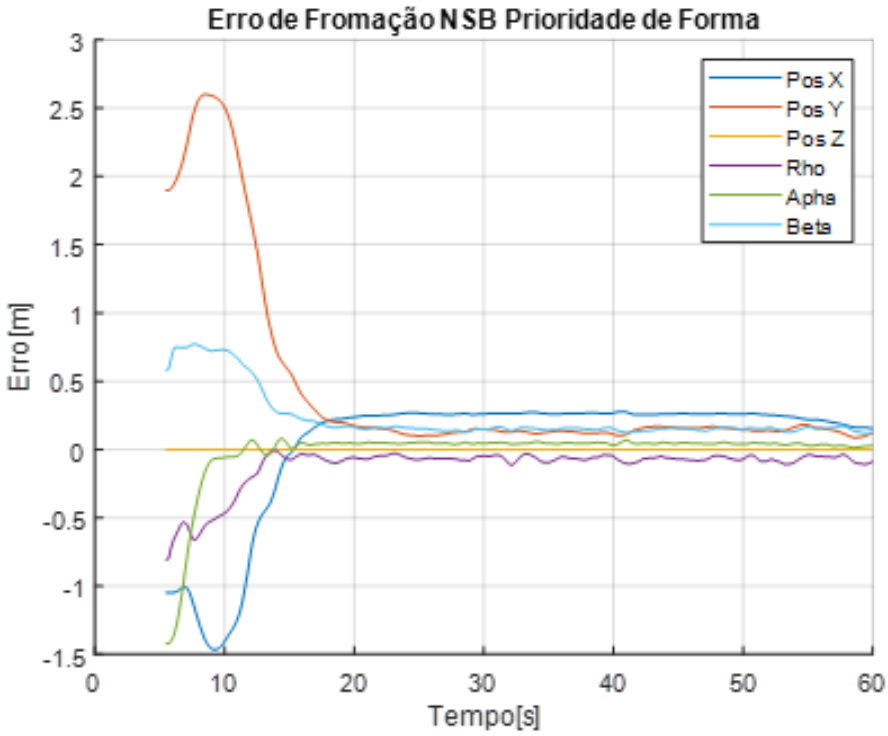


Figura 31 – Erros de formação para o Controle NSB com Prioridade de Forma.

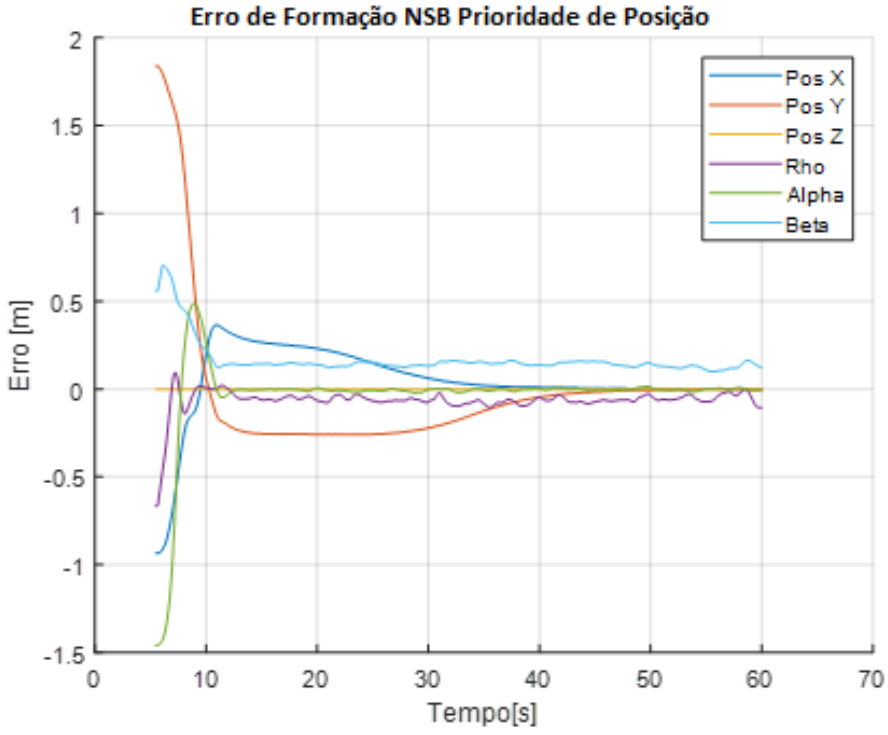


Figura 32 – Erros de formação para o Controle NSB com Prioridade de Posição.

forma e de posição, são buscados, mas o controlador prioriza a manutenção do formato da formação.

No caso do Controle NSB com Prioridade de Posição (ver Figura 32) o controlador permite mais aceleração, o que causa maiores erros de forma. Ou seja, o controlador está tentando reduzir o erro de posição o mais rápido possível, mesmo à custa de erros de forma maiores.

Como pode ser observado na Figura 32, o controlador NSB com Prioridade de Forma apresenta grande aceleração em direção ao Pioneer, conforme esperado, pois ele busca corrigir a forma da formação com uma ação de controle mais acentuada na variável de formação que apresenta o maior erro, representada neste caso pelo ângulo α.

Uma característica importante a ser salientada é que o controlador adotado realiza a correção de posição utilizando coordenadas polares. Logo, em alguns casos verifica-se um comportamento não esperado para controladores baseados em coordenadas retangulares. Este fato é evidenciado na Figura 31, na qual se pode verificar um pequeno aumento de erro em algumas variáveis no início do teste. Este aumento de erro também é verificado na Figura 32 no conjunto correspondente às variáveis de forma, porém com menor valor absoluto.

Como a premissa desta seção é comparar o comportamento dos controladores, o experimento se realizou em igualdade de condições, ou seja, todos os três controladores tiveram os mesmos ganhos, no que se refere aos controladores de cada robô. Entretanto, pode-se perceber que os controladores analisados poderiam ter tais ganhos ajustados para cada caso, buscando uma condição ótima individual. O estudo das condições ótimas de cada controlador, porém, foge ao escopo deste trabalho, uma vez que aqui o que se estuda é a viabilidade de atribuir maior prioridade à manutenção da forma da formação, num contexto de pouso do VANT sobre o VTNT.

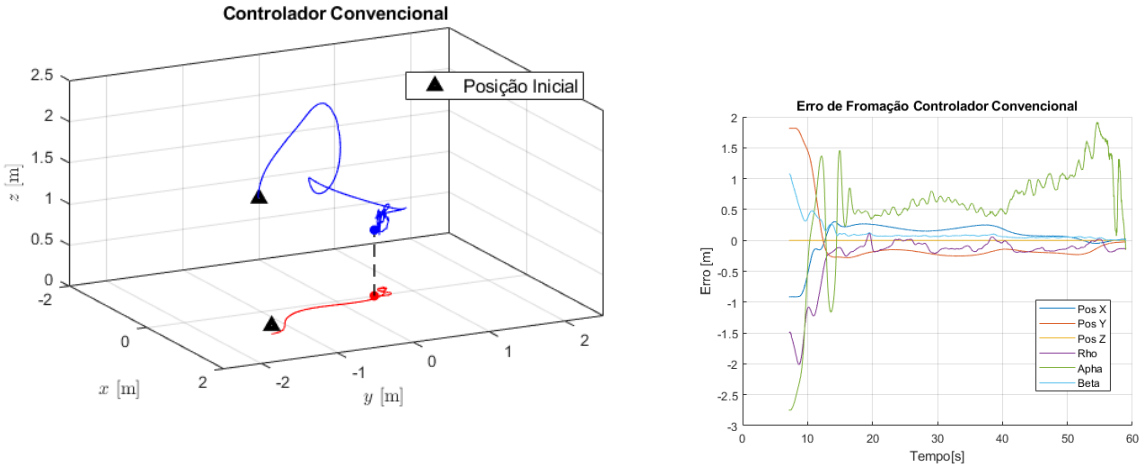
Em resumo, todos os três controladores fazem os erros de forma e posição tenderem a zero ao longo do tempo. Mas se o foco principal é manter a forma da formação, o Controle NSB com Prioridade de Forma tem melhor desempenho, como mostram os resultados do experimento. Como exemplo, suponha-se que o VTNT mude seu caminho para evitar um obstáculo quando o VANT estiver tentando pousar nele. Nesse caso, a melhor opção seria certamente usar o Controle NSB com Prioridade de Forma aqui discutido, para que se tenha maior confiança que o VANT estará em condição de pousar no VTNT. A Figura 31 ilustra bem este comportamento: o VANT, por ter uma dinâmica mais rápida, se desloca mais em direção ao *Pioneer*; porém, também é possível observar o VTNT se deslocar em direção ao VANT. Uma vez que a forma é estabelecida, o conjunto se desloca em direção à posição desejada.

5.2 Pouso do VANT sobre o VTNT

O pouso do VANT sobre o VTNT foi testado em duas condições distintas, no primeiro caso o ângulo β assume o valor de $\pi/3$, e no segundo caso, o valor é modificado para $\pi/2$. Os comportamentos são observados nas Figuras 30, 31 e 32 e Figuras 33 e 34 para o primeiro e segundo caso respectivamente.

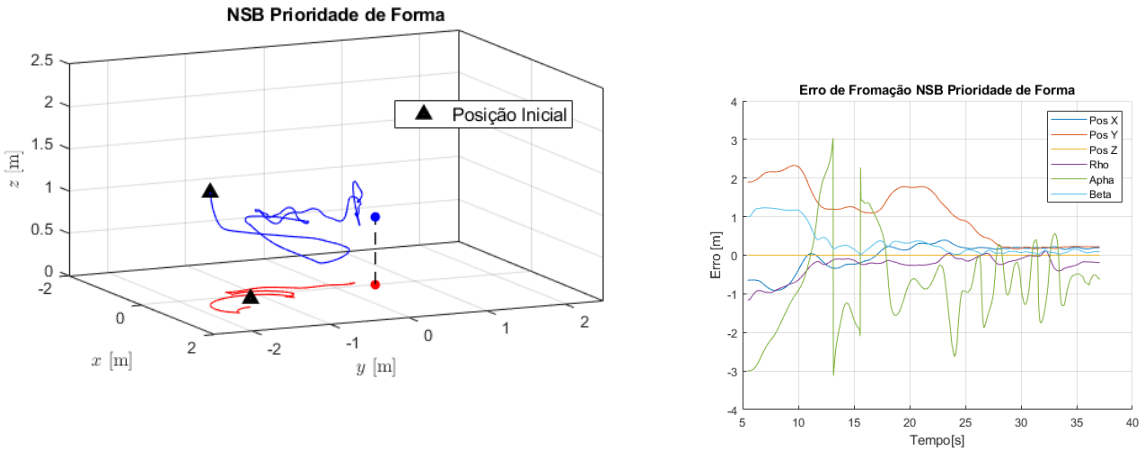
O comportamento observado utilizando a técnica de espaço nulo, assegura que o controle é capaz de manter o drone sobre o ponto especificado acima do ponto de interesse, este comportamento foi detalhado na seção 5.1.

No segundo caso observa-se um comportamento distinto em relação ao primeiro caso no que tange à estabilidade do controlador. Mesmo utilizando os mesmos parâmetros e tipos de controle, há maior oscilação no segundo caso a qual é justificada pela escolha do sistema de coordenadas. A convenção adotada apresenta uma singularidade quando $\beta = \pi/2$, nessa condição o ângulo α pode assumir qualquer valor uma vez que sua projeção é representada por um ponto no plano XY e qualquer variação do angulo β possibilita que α varie com grande amplitude. Tal comportamento pode ser observado nas Figuras 33 e 34.



(a) Trajetória em 3D (b) Erro de Formação

Figura 33 – Pouso do VANT sobre o VTNT usando o Controle Convencional



(a) Trajetória em 3D (b) Erro de Formação

Figura 34 – Pouso do VANT sobre o VTNT usando o Controle NSB com Prioridade de Forma

Algumas soluções podem ser adotadas para correção deste cenário, tais como:

- Mudança do sistema de referência do plano XY para XZ ou YZ;
- Deslocamento do ponto de controle para uma posição à frente do Pioneer (aumento do valor do parâmetro a que resenta a distância entre o ponto de controle e a linha de centro que une as rodas do Pioneer; este parâmetro é observado na matriz de cinemática inversa em (2.6));
- Modificação do ponto de pouso para um reboque (*trailer*) acoplado ao ao Pioneer.

Tais propostas de melhoria, porém, não fazem parte do escopo deste trabalho.

6 Conclusões

Neste trabalho um sistema de controle foi proposto para conduzir a formação de um VTNT e um VANT, visando o pouso do VANT no VTNT em movimento. O VANT deve manter uma certa posição em relação ao VTNT, para então aterrissar nele. A proposta consiste em controlar a formação VTNT-VANT associando-a a uma estrutura virtual, que é a linha reta ligando os dois veículos. Para permitir que o VANT aterrisse no VTNT, a estrutura virtual deve manter a forma desejada (comprimento ρ_f e ângulos β_f em relação ao eixo z e α_f da projeção de linha sobre o plano xy e o eixo x) o mais próximo possível dos valores especificados. A formação é descrita a partir da posição do VTNT (o vetor $\begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$), cuja posição define a posição da formação (o vetor $\begin{bmatrix} x_f & y_f & z_f \end{bmatrix}^T$), cuja forma (variáveis $\rho_f, \beta_f, \alpha_f$) está relacionada às posições do VTNT e do VANT (vetor $\begin{bmatrix} x_2 & y_2 & z_2 \end{bmatrix}^T$).

Para assegurar que os erros nas variáveis de forma associadas à formação ao longo do seu movimento sejam mantidos em valores baixos, a técnica de controle baseada no espaço nulo é adotada, atribuindo-se maior prioridade ao controle da forma da formação. A ideia é que o VANT possa pousar no VTNT mesmo que a posição deste inclua grandes erros.

Finalmente, o esquema de controle proposto é validado através de resultados experimentais, após comparações com outros dois controladores similares, incluindo-se o caso do pouso do VANT sobre o VTNT.

Assim é que se pode afirmar que a hipótese que conduziu esta pesquisa, apresentada na Seção 1.5, foi plenamente comprovada.

Como sugestões de continuidade desta pesquisa, pode-se aplicar o esquema de controle baseado em espaço nulo com prioridade de forma em experimentos nos quais o VANT deve pousar sobre o VTNT na presença de um obstáculo em seu caminho.

Também como sugestão de continuidade desta pesquisa, pode-se expandir o sistema de controle proposto para situações envolvendo mais agentes. A ideia é aplicar o sistema proposto para cada par de agentes, decompondo-se a formação de mais de dois agentes em diversas formações menores, homogêneas ou heterogêneas.

Referências

ADAMEK, T.; KITTS, C. A.; MAS, I. Gradient-based cluster space navigation for autonomous surface vessels. *IEEE/ASME Transactions on Mechatronics*, v. 20, n. 2, p. 506–518, April 2015. ISSN 1083-4435. Citado 2 vezes nas páginas 30 e 58.

ANTONELLI, G.; ARRICHIELLO, F.; CHIAVERINI, S. The nsb control: a behavior-based approach for multi-robot systems. *Paladyn*, v. 1, n. 1, p. 48–56, March 2010. ISSN 2080-9778. Citado 4 vezes nas páginas 30, 57, 58 e 62.

ANTONELLI, G.; CHIAVERINI, S. Kinematic control of platoons of autonomous vehicles. *IEEE Transactions on Robotics*, v. 22, n. 6, p. 1285–1292, December 2006. ISSN 1552-3098. Citado na página 30.

ARKIN, R. C. Integrating behavioral, perceptual, and world knowledge in reactive navigation. *Robotics and Autonomous Systems*, v. 6, n. 1, p. 105 – 122, 1990. ISSN 0921-8890. Citado na página 57.

BAI, H.; WEN, J. T. Cooperative load transport: A formation-control perspective. *IEEE Transactions on Robotics*, v. 26, n. 4, p. 742–750, August 2010. ISSN 1552-3098. Citado 2 vezes nas páginas 29 e 51.

BALCH, T.; ARKIN, R. C. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, v. 14, n. 6, p. 926–939, December 1998. ISSN 1042-296X. Citado 2 vezes nas páginas 57 e 58.

BARANZADEH, A.; SAVKIN, A. V. A distributed control algorithm for area search by a multi-robot team. *Robotica*, Cambridge University Press, v. 35, n. 6, p. 1452–1472, 2017. Citado na página 51.

BECK, Z. et al. Online planning for collaborative search and rescue by heterogeneous robot teams. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. Singapore: [s.n.], 2016. p. 1024–1033. Citado na página 51.

BOEL, R. K.; MARINICA, N. E.; SARLETTE, A. Leader-follower cooperative control paradigm, with applications to urban traffic coordination control. In: *2015 European Control Conference (ECC)*. [S.l.: s.n.], 2015. p. 2208–2215. Citado na página 57.

BRANDÃO, A. S. et al. The multilayer control scheme: A strategy to guide n-robots formations with obstacle avoidance. *Journal of Control, Automation and Electrical Systems*, v. 26, n. 3, p. 201–214, June 2015. ISSN 2195-3899. Citado 3 vezes nas páginas 30, 57 e 58.

BRANDÃO, A. S.; SARCINELLI-FILHO, M. On the guidance of multiple uav using a centralized formation control scheme and delaunay triangulation. *Journal of Intelligent & Robotic Systems*, Springer, v. 84, n. 1-4, p. 397–413, 2016. Citado 2 vezes nas páginas 57 e 58.

BRANDÃO, A. S.; SARCINELLI-FILHO, M.; CARELLI, R. High-level underactuated nonlinear control for rotorcraft machines. In: IEEE. *Proceedings of the 2013 IEEE International Conference on Mechatronics (ICM2013)*. Vicenza, Italy, 2013. p. 279–285. Citado na página 26.

BRANDÃO, A. S.; SARCINELLI-FILHO, M.; CARELLI, R. Leader-following control of a uav-ugv formation. In: IEEE. *2013 16th International Conference on Advanced Robotics (ICAR 2013)*. Montevideo, Uruguay, 2013. p. 1–6. Citado na página 57.

BRANDÃO, A. S. et al. Decentralized control of leader-follower formations of mobile robots with obstacle avoidance. In: *2009 IEEE International Conference on Mechatronics*. Malaga, Spain: [s.n.], 2009. p. 1–6. Citado na página 57.

BROOKS, R. A. A robust layered control system for a mobile robot. *IEEE Journal on Robotics and Automation*, v. 2, n. 1, p. 14–23, March 1986. ISSN 2374-8710. Citado na página 57.

CHIAVERINI, S.; MEDDAHI, A. A null-space based behavioural control approach to coordinated motion of a humanoid robot. In: *2015 IEEE International Conference on Information and Automation*. Lijiang, China: [s.n.], 2015. p. 20–25. Citado na página 30.

DEHGHANI, M. A.; MENHAJ, M. B.; AZIMI, M. Leader-follower formation control using an onboard leader tracker. In: *4th International Conference on Control, Instrumentation, and Automation (ICCIA)*. Qazvin, Iran: [s.n.], 2016. p. 99–104. Citado na página 57.

KITTS, C. A.; MAS, I. Cluster space specification and control of mobile multirobot systems. *IEEE/ASME Transactions on Mechatronics*, v. 14, n. 2, p. 207–218, April 2009. ISSN 1941-014X. Citado 3 vezes nas páginas 30, 57 e 58.

LEWIS, M. A.; TAN, K.-H. High precision formation control of mobile robots using virtual structures. *Autonomous Robots*, v. 4, n. 4, p. 387–403, October 1997. ISSN 1573-7527. Citado 2 vezes nas páginas 30 e 57.

LI, H. et al. Multivehicle cooperative local mapping: A methodology based on occupancy grid map merging. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 15, n. 5, p. 2089–2100, 2014. Citado na página 51.

LIANG, X. et al. Leader-following formation tracking control of mobile robots without direct position measurements. *IEEE Transactions on Automatic Control*, IEEE, v. 61, n. 12, p. 4131–4137, 2016. Citado na página 57.

LOW, C. B. A flexible virtual structure formation keeping control design for nonholonomic mobile robots with low-level control systems, with experiments. In: IEEE. *Intelligent Control (ISIC), 2014 IEEE International Symposium on*. [S.l.], 2014. p. 1576–1582. Citado na página 57.

MANSFELD, N. et al. Interactive null space control for intuitively interpretable reconfiguration of redundant manipulators. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, BC, Canada: [s.n.], 2017. p. 5368–5375. ISSN 2153-0866. Citado 2 vezes nas páginas 30 e 58.

MARTINS, F. N. *Modelagem e compensação da dinâmica de robôs móveis e sua aplicação em controle de formação*. Tese (Doutorado) — PhD thesis, Universidade Federal do Espırito Santo, 2009. Citado 3 vezes nas páginas 29, 35 e 60.

MARTINS, F. N.; SARCINELLI-FILHO, M.; CARELLI, R. A velocity-based dynamic model and its properties for differential drive mobile robots. *Journal of Intelligent & Robotic Systems*, Springer, v. 85, n. 2, p. 277–292, 2017. Citado 3 vezes nas páginas 29, 33 e 35.

MASONE, C.; BÜLTHOFF, H. H.; STEGAGNO, P. Cooperative transportation of a payload using quadrotors: A reconfigurable cable-driven parallel robot. In: IEEE. *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. [S.l.], 2016. p. 1623–1630. Citado 2 vezes nas páginas 29 e 51.

PARKER, L. E. Multiple Robot Systems. In: BRUNO SICILIANO AND OUSSAMA KHATIB. *Springer Handbook of Robotics*. 2nd. ed. [S.l.]: Springer International Publishing, 2016. (Springer Handbooks), p. 1335–1384. ISBN 978-3-319-32552-1. Citado 2 vezes nas páginas 29 e 51.

PIZETTA, I. H. B.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. A hardware-in-the-loop platform for rotary-wing unmanned aerial vehicles. *Journal of Intelligent & Robotic Systems*, v. 84, n. 1, p. 725–743, Dec 2016. Citado na página 49.

PIZETTA, I. H. B.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Avoiding obstacles in cooperative load transportation. *ISA Transactions*, v. 91, p. 253 – 261, 2019. ISSN 0019-0578. Citado na página 29.

PIZETTA, I. H. B.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Cooperative load transportation using three quadrotors. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS'19)*. Atlanta, GA, USA: [s.n.], 2019. p. 644–650. ISSN 2373-6720. Citado na página 29.

RABELO, M. F. S.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Centralized control for an heterogeneous line formation using virtual structure approach. In: IEEE. *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*. [S.l.], 2018. p. 135–140. Citado 3 vezes nas páginas 35, 61 e 62.

RESENDE, C. Z.; CARELLI, R.; SARCINELLI-FILHO, M. Coordinated path-following for multi-robot systems using the cluster space framework approach. In: *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*. Porto Alegre, RS, Brazil: [s.n.], 2014. p. 332–337. ISSN 2378-363X. Citado 2 vezes nas páginas 58 e 60.

ROSALES, C. et al. 3d formation control of autonomous vehicles based on null-space. *Journal of Intelligent & Robotic Systems*, Springer, v. 84, n. 1-4, p. 453–467, 2016. Citado 3 vezes nas páginas 30, 58 e 64.

SANTANA, L. V. *Sistemas de Navegação e Controle para Veículos Aéreos Não Tripulados e suas Aplicações*. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2016. Citado 4 vezes nas páginas 29, 36, 38 e 42.

SANTANA, L. V.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Heterogeneous leader-follower formation based on kinematic models. In: IEEE. *Unmanned Aircraft Systems (ICUAS), 2016 International Conference on*. Arlington, VA, USA, 2016. p. 342–346. Citado 2 vezes nas páginas 33 e 38.

SANTANA, L. V. et al. A trajectory tracking and 3d positioning controller for the ar.drone quadrotor. In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. Orlando, FL, USA: [s.n.], 2014. p. 756–767. Citado 3 vezes nas páginas 29, 38 e 61.

SANTOS, M. C. et al. Indoor low-cost localization system for controlling aerial robots. *Control Engineering Practice*, v. 61, p. 93 – 111, 2017. ISSN 0967-0661. Citado 2 vezes nas páginas 29 e 38.

SANTOS, M. C. P. *Controle em Ambientes Interiores de Veículos Aéreos Não Tripulados*. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2017. Citado 3 vezes nas páginas 29, 38 e 39.

SANTOS, M. C. P. et al. An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks. *Journal of Intelligent & Robotic Systems*, v. 93, n. 1, p. 5–16, February 2019. ISSN 1573-0409. Citado 2 vezes nas páginas 38 e 60.

SANTOS, M. C. P. et al. A novel null-space-based uav trajectory tracking controller with collision avoidance. *IEEE/ASME Transactions on Mechatronics*, v. 22, n. 6, p. 2543–2553, Dec 2017. ISSN 1941-014X. Citado na página 38.

SATO, N. et al. Cooperative task execution by a multiple robot team and its operators in search and rescue operations. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sendai, Japan: [s.n.], 2004. v. 2, p. 1083–1088. Citado na página 51.

SURMANN, H.; BERNINGER, N.; WORST, R. 3d mapping for multi hybrid robot cooperation. In: IEEE. *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. Vancouver, BC, Canada, 2017. p. 626–633. Citado na página 29.

VIDYASAGAR, M. *Nonlinear System Analysis*. 2nd. ed. New Jersey: Prentice Hall, 1993. Citado na página 39.

VILLA, D. K. D.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. A survey on load transportation using multirotor uavs. *Journal of Intelligent & Robotic Systems,* on line access, DOI: 10.1007/s10846-019-01088-w, October 2019. ISSN 1573-0409. Citado 2 vezes nas páginas 29 e 51.

APÊNDICE A – *Conceitos Básicos de Sistemas Lineares*

Este apêndice fornece o alicerce matemático básico para compreensão e utilização do conceito de espaço nulo que será aplicado ao controle de formação de robôs. Salienta-se que todos os conceitos apresentados neste apêndice se encontram no âmbito de sistemas lineares. Portanto, é necessário que o leitor tenha conhecimento prévio sobre tal assunto.

A.1 Escalonamento de Matrizes

A.1.1 Conceito Geral

O processo de escalonamento de um sistema linear ocorre por meio de operações elementares, que são iguais às utilizadas no teorema de Jacobi. Vale ressaltar que dois sistemas são ditos equivalentes quando estes possuem o mesmo conjunto solução.

Teorema 1 (Jacob) *Seja* $\mathbf{A}$ *uma matriz quadrada n por n. Se multiplicarmos todos os elementos de uma fila (linha ou coluna) por um mesmo número, e somarmos os resultados dos elementos aos seus correspondentes de outra fila (linha ou coluna), obteremos outra matriz* $\mathbf{B}$*, de mesma dimensão. Entretanto, podemos afirmar que* $det(\mathbf{A}) = det(\mathbf{B})$.

A.1.2 Matriz Escalonada na Forma Reduzida

Seja $\mathbf{A}_{mxn}$ uma matriz composta de elementos constantes, escalonada por linhas na forma reduzida ou seja, foram aplicadas sucessivas operações elementares, transformações lineares, à matriz $\mathbf{A}$ de forma a obter um novo sistema linear de mesma dimensão que apresenta exatamente o mesmo conjunto solução que o sistema original. Caso o resultado final do escalonamento atenda ao conjunto de restrições

- todas as linhas não-nulas estão acima de qualquer linha composta só de zeros,
- o pivô de cada linha está numa coluna à direita do pivô da linha acima,
- todos os elementos de uma coluna abaixo de um pivô são zero,
- o pivô de cada linha não-nula é 1,
- cada pivô 1 é o único elemento não-nulo de sua coluna,

a nova matriz encontrada após as manipulações algébricas recebe a designação especial de matriz Escalonada na Forma Reduzida.

A.2 Independência e Dependência Linear

Dado o conjunto de vetores $\mathbf{v}_1, \cdots, \mathbf{v}_n$, o conjunto é dito ser linearmente independente (abreviadamente l.i.), se $\alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n = \mathbf{0}$ implicar em $\alpha_1 = \cdots = \alpha_n = \mathbf{0}$. Isto é, se a combinação linear for nula, seus coeficientes devem ser nulos. Como o vetor nulo pode ser obtido como a combinação linear com coeficientes nulos, ser l.i. significa que só tem uma forma de escrever o vetor nulo como combinação linear dos vetores dados.

Suponha-se que o conjunto de vetores $\mathbf{v}_1, \cdots, \mathbf{v}_n$ é l.i. Então $\alpha_1\mathbf{v}_1 + \cdots + \alpha_{\mathbf{v}_n=\beta_1\mathbf{v}_1+\cdots+\beta_n\mathbf{v}_n}$ implica que $(\alpha_1\beta_1)\mathbf{v}_1 + \cdots + (\alpha_n\beta_n)\mathbf{v}_n = \mathbf{0}$, e, consequentemente, $\alpha_i\beta_i = 0$ para todo i. Logo, $\alpha_i = \beta_i$.

Teorema 2 *Um conjunto de vetores $\mathbf{v}_1, \cdots, \mathbf{v}_n$ é l.i., se, e somente se, a combinação linear é única, isto é, se, e somente se, $\alpha_1\mathbf{v}_1 + \cdots + \alpha_{\mathbf{v}_n=\beta_1\mathbf{v}_1+\cdots+\beta_n\mathbf{v}_n}$ implica que $\alpha_i = \beta_i$.*

Suponha $\alpha_1\mathbf{v}_1 + \cdots + \alpha_n\mathbf{v}_n = \mathbf{0}$ de modo que $\alpha_i \neq 0$ para algum i. Então tem-se que $\alpha_i\mathbf{v}_i = \alpha_1\mathbf{v}_1\alpha_{i-1}\mathbf{v}_{i-1} - \alpha_{i+1}\mathbf{v}_{i+1} - \cdots\alpha_n\mathbf{v}_n$. Como $\alpha_i \neq 0$, pode-se dividir toda a expressão por tal valor e obter $\mathbf{v}_i = \frac{\alpha_1}{\alpha_i}\mathbf{v}_1\frac{\alpha_{i-1}}{\alpha_i}\mathbf{v}_{i-1}\frac{\alpha_{i+1}}{\alpha_i}\mathbf{v}_{i+1}\cdots\frac{\alpha_n}{\alpha_i}\mathbf{v}_n$, ou seja, $\mathbf{v}_i$ é uma combinação linear dos vetores restantes.

O conjunto de vetores que não é linearmente independente é dito ser linearmente dependente (abreviadamente, l.d.).

Teorema 3 *O conjunto de vetores $\mathbf{v}_1, \cdots, \mathbf{v}_n$ é l.d. se, e somente se, algum desses vetores é combinação linear dos restantes.*

A.3 Posto de uma Matriz

O posto da matriz $\mathbf{A}$ escalonada e em sua forma reduzida é definido como sendo o número de linhas não nulas linearmente independentes após o escalonamento. Comumente se denota o posto de uma matriz $\mathbf{A}$ como $\rho(A)$, sendo comum também se usar o termo *rank*, originário do inglês. Por definição,

$$\rho(\mathbf{A}_{mxn}) = \text{ número de linhas l. i. de } \mathbf{A}_{mxn}) \tag{A.1}$$

O poeto de uma matriz permite classificar um sistema de equações lineares quanto às suas possíveis soluções, assim como saber quantas variáveis livres existem na solução do sistema.

Como exemplo, seja uma matriz $\mathbf{M}_{3x3}$ escalonada e em sua forma reduzida, onde sua diagonal principal seja composta por elementos unitários ($(\mathbf{M}) = \begin{bmatrix}1 & 1 & 1\end{bmatrix}$). Neste caso tem-se que $\rho(M) = 3$ ou seja, a matriz $\mathbf{M}$ possui posto completo. Logo, o vetor $\mathbf{x}$ solução do sistema de equações lineares $\mathbf{Mx} = \mathbf{y}$, se existir, é único.

Da mesma forma, suponha-se uma segunda matriz $\mathbf{N}_{3x3}$ também escalonada e em sua forma reduzida, mas, neste caso, com sua diagonal principal composta por apenas dois elementos unitários e o terceiro sendo 0 ($(\mathbf{N}) = \begin{bmatrix}1 & 1 & 0\end{bmatrix}$). Pelo exposto acima vê-se que o posto da matriz $\mathbf{N}$ não é completo. Logo, há uma linha que pode ser expressa como combinação linear das demais, e, neste novo cenário, o conjunto de soluções do sistema de equações lineares $\mathbf{Nx} = \mathbf{y}$, se existir, possui um grau de liberdade, e tem-se infinitas possibilidades que atendem à equação $\mathbf{Nx} = \mathbf{0}$.

A.4 Nulidade de uma Matriz

Em álgebra linear, o espaço nulo (*null space*, em inglês) corresponde ao conjunto de vetores x_{nx1} solução de

$$\mathbf{A}_{mxn}\mathbf{x}_{nx1} = \mathbf{0}, \tag{A.2}$$

onde $\mathbf{A}_{mxn}$ representa uma matriz de elementos constantes de dimensão m por n ou seja pertencente ao espaço $\mathcal{R}^{mxn}$, $\mathbf{x}_{nx1}$ representa um vetor de dimensão n por 1, ou seja pertencente ao espaço $\mathcal{R}^n$ e $\mathbf{0}$ é um vetor nulo de dimensão m por 1, ou seja, pertencente ao espaço $\mathcal{R}^m$.

De forma matemática, pode-se representar a nulidade de $\mathbf{A}_{mxn}$ ou seja, a dimensão do espaço solução do sistema homogêneo $\mathbf{Ax} = \mathbf{0}$ como sendo

$$Null(\mathbf{A}) = \{\mathbf{x} \in \mathcal{R}^n | \mathbf{Ax} = \mathbf{0}\} \tag{A.3}$$

Teorema 4 (Teorema do Núcleo e da Imagem) *Seja*

$$\mathbf{A}_{mxn} \in \mathcal{R}^{mxn}. \tag{A.4}$$

Então

$$n = Null(\mathbf{A}) + Posto(\mathbf{A}) \tag{A.5}$$

Em outras palavras, o posto e a nulidade de uma matriz têm como soma seu número de colunas.

Deste teorema vemos que caso a matriz $\mathbf{A}_{mxn}$ possua posto completo, o espaço solução do sistema homogêneo $\mathbf{Ax} = \mathbf{0}$ se resume a um vetor de zeros.

A.5 Matriz Identidade

A Matriz Identidade de dimensão n é definida como a matriz quadrada n por n em que todos os elementos da diagonal principal são iguais a 1 e todos os demais elementos são iguais a 0 (zero). Ela é normalmente representada pela letra $\mathbf{I}_n$ em maiúsculo, e é da forma

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}.$$

A.6 Matriz Transposta

A transposta de uma matriz $\mathbf{A}$ é uma matriz que apresenta os mesmos elementos de $\mathbf{A}$, mas transportando-se ordenadamente os elementos das linhas de $\mathbf{A}$ para suas colunas. Portanto, dada uma matriz $\mathbf{A} = \{a_{ij}, i = 1, \cdots, m, j = 1, \cdots, n\}$, a transposta de $\mathbf{A}$, denotada por $\mathbf{A}^T$, é $\mathbf{A}^T = \{a_{ji}, i = 1, \cdots, m, j = 1, \cdots, n\}$, onde a_{ij} representa um elemento da matriz situado na linha i e na coluna j, e m e n são o número d elinhas e de colunas da matriz, respectivamente.

Note-se que a matriz $\mathbf{A}$ é m por n, enquanto sua transposta $\mathbf{A}^T$ é n por m.

A.7 Matriz Inversa

A matriz inversa é um tipo de matriz quadrada, ou seja, que possui o mesmo número n de linhas e colunas, que ocorre quando o produto de duas matrizes resulta numa matriz identidade de mesma dimensão. Assim, para encontrar a inversa $\mathbf{B}_{nxn}$ de uma matriz $\mathbf{A}_{nxn}$, utiliza-se a multiplicação $\mathbf{A}_{nxn}\mathbf{B}_{nxn} = \mathbf{B}_{nxn}\mathbf{A}_{nxn} = \mathbf{I}_n$. Nesse caso, a matriz $\mathbf{B}_{nxn}$ é a inversa da matriz $\mathbf{A}_{nxn}$, ou, matematicamente, $\mathbf{B}_{nxn} = \mathbf{A}_{nxn}^{-1}$.

A matriz inversa possui propriedades específicas que a caracterizam, como

- existe somente uma inversa para cada matriz quadrada, desde que esta tenha determinante diferente de zero (seja não singular);
- nem todas as matrizes possuem uma matriz inversa. Uma matriz é inversível somente quando seu determinante é diferente de zero. Isso equivale à existência de uma segunda matriz de mesma dimensão tal que seu produto pela matriz original resulta numa matriz identidade;
- a matriz inversa de uma inversa corresponde à própria matriz, ou $\mathbf{A} = (\mathbf{A}^{-1})^{-1}$;
- a matriz inversa de uma matriz identidade é igual à própria matriz identidade, ou seja, $\mathbf{I}^{-1} = \mathbf{I}$;
- a matriz transposta de uma matriz inversa também é inversa, ou seja, $(\mathbf{A}^{-1})^T = (\mathbf{A}^T)^{-1}$;
- a matriz inversa de uma matriz transposta corresponde à transposta da inversa, ou seja $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$.

A.8 Matriz Pseudo Inversa

Em diversos problemas da álgebra linear é necessário o uso da matriz inversa. Mas não é toda matriz que possui uma inversa, como já mencionado, visto que algumas condições precisam ser verdadeiras. Entre estas condições, está o fato de a matriz ser necessariamente quadrada, o que restringe a existência da matriz inversa a uma classe menor das matrizes em $\mathcal{C}^{mn}$ gerais. Mesmo assim, não é toda matriz quadrada que possui inversa, já que somente matrizes não singulares a possuem.

A matriz pseudo-inversa generaliza o conceito de matriz inversa, abrangendo sistemas assimétricos e nos fornecendo resultados otimizados, como será abordado mais adiante. Para que a matriz seja uma matriz pseudo-inversa, ela deve atender às condições de Penrose (ou equações de Penrose), a saber, considerando duas matrizes $\mathbf{A}$ e $\mathbf{X}$ quaisquer, de dimensões compatíveis para que os produtos abaixo existam,

- $\mathbf{AXA} = \mathbf{A}$
- $\mathbf{XAX} = \mathbf{X}$
- $(\mathbf{AX})^T = \mathbf{AX}$
- $(\mathbf{XA})^T = \mathbf{XA}$

Se $\mathbf{X}$ satisfizer essas quatro propriedades, então $\mathbf{X}$ é conhecida como a inversa de Moore-Penrose de $\mathbf{A}$, ou simplesmente sua pseudo-inversa, que é denotada por $\mathbf{A}^\dagger$. Ela é obtida através da equação

$$\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T. \tag{A.6}$$

Uma das propriedades teóricas mais interessantes da matriz pseudo-inversa, que certamente é utilizada na prática, é que, dado um sistema $\mathbf{Ax} = \mathbf{b}$, então $\mathbf{A}^{\dagger}\mathbf{b}$ é solução do problema de mínimos quadrados associado e, mais ainda, é a solução de norma mínima.

Como detalhe importante, note-se, também, que o conceito de matriz pseudo-inversa se aplica somente a matrizes não quadradas. No caso de matrizes quadradas a matriz pseudo-inversa recai no caso convencional de matriz inversa, ou seja, a matriz $\mathbf{X}$ das equações de Penrose é a própria matriz $\mathbf{A}^{-1}$.