

VI Semana da Engenharia Elétrica

Parte II: Controle de Quadrrrotores

Leandro Lima Gomes

Lucas Pires Leal

Tiago Roux de Oliveira

José Paulo Vilela Soares da Cunha 😊

Téo Cerqueira Revoredo

Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia
Departamento de Eletrônica e Telecomunicações

Rio de Janeiro, 24 de outubro de 2018



INTRODUÇÃO

- ✓ VANT – Veículo Aéreo Não Tripulado;
- ✓ Pilotado remotamente ou autônomo;
- ✓ Aplicações:
 - ✓ Militares;
 - ✓ Civis: filmagem, inspeção de linhas de transmissão, etc.



INTRODUÇÃO

➤ Quadricópteros:

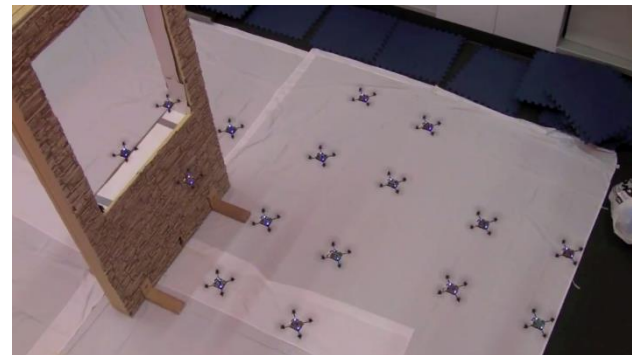
- ✓ Veículos aéreos com quatro rotores com pás rotativas;
- ✓ **Vantagens:**
 - ✓ simplicidade mecânica;
 - ✓ baixo custo de implementação;
 - ✓ boa manobrabilidade;
 - ✓ permitem adição de periféricos;
 - ✓ etc.
- ✓ **Desvantagem:** curta duração da bateria



INTRODUÇÃO

➤ Aplicações:

- ✓ Cooperação entre veículos para içar cargas;
- ✓ Vigilância de fronteiras;
- ✓ Plataformas de pesquisa em controle aplicado.



INTRODUÇÃO

➤ Servovisão Robótica:

- ✓ Uso de realimentação visual para controle de robôs;
- ✓ Utilização de câmeras para extração das informações requeridas;
- ✓ Resulta da convergência de áreas como processamento de imagens, controle, computação em tempo real, etc.



INTRODUÇÃO

➤ Aplicações:

- ✓ Controle de manipuladores robóticos;
- ✓ Controle de veículos remotamente operados.



INTRODUÇÃO

➤ **Objetivo:**

- ✓ Integração do quadricóptero AR.Drone com o sistema de visual de medição de posição Vicon tendo em vista o controle de trajetória.



PLATAFORMA AR.DRONE

- Quadricóptero desenvolvido pela Parrot, inicialmente projetado para ser um brinquedo.
- Jogos de realidade aumentada.
- Desenvolvimento de aplicações para utilização conjunta com plataformas móveis (iOS e Android).



PLATAFORMA AR.DRONE

➤ Especificações:

- Microprocessador ARM9 468 MHz
- 128 MB de memória RAM
- Sistema operacional Linux embarcado – BusyBox
- 4 rotores com unidades de controle e acelerômetro
- 2 câmeras integradas VGA
- Altímetro ultrassônico
- Controle por sistemas Android e iOS
- Comunicação *Wi-Fi* b/g



PLATAFORMA AR.DRONE

- O AR.Drone é controlado por qualquer dispositivo cliente que suporta o modo *ad-hoc*;
- O AR.Drone cria uma rede *Wi-Fi*. O usuário conecta o dispositivo cliente na mesma rede;
- O servidor DHCP AR.Drone concede ao cliente um endereço IP.



AR.DRONE

➤ *Open API Plataforma:*

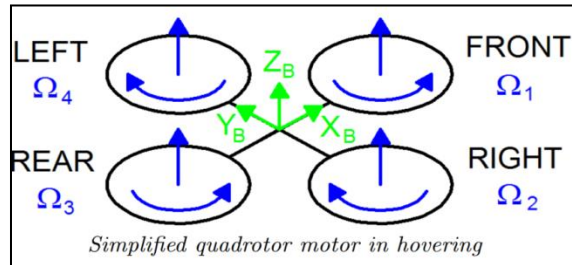
- ✓ Desenvolvimento de Jogos em realidade aumentada;
- ✓ Bibliotecas desenvolvidas na linguagem C;
- ✓ Compatibilidade com Windows e Linux;
- ✓ Diversas possibilidades de integração com outros sistemas.



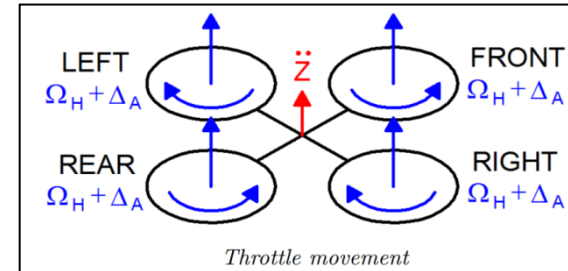
AR.DRONE

➤ Dinâmica de movimento:

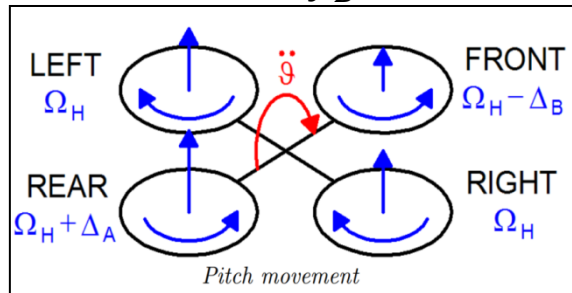
Modo Planar



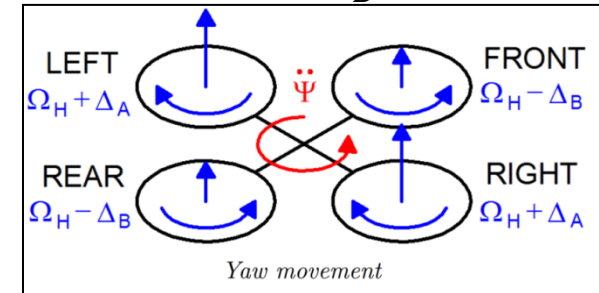
Movimento Ascensão Vertical



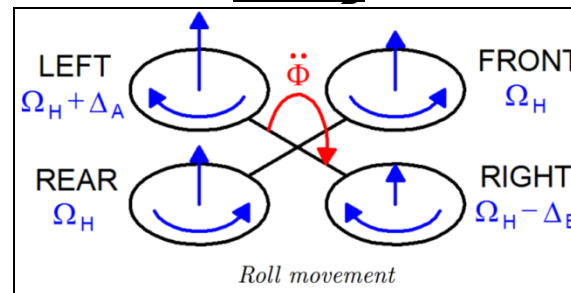
Movimento de rotação sobre eixo y_B



Movimento de rotação sobre o eixo z_B



Movimento de rotação sobre o eixo x_B



SISTEMA VICON

➤ *Software de medição de posição Vicon Tracker:*

The screenshot displays the Vicon Tracker 1.3 software interface. The main window shows a 3D orthogonal view of a 'Posicao_barco' object with several green wireframe markers. The left sidebar contains a 'CONNECTED' status, system information, and a list of four Vicon cameras (#1 to #4). The bottom panel shows a 'COMMUNICATIONS' log with entries for VSTImporter, Import XCP, and Export XCP operations.

TIME	USER	TEXT
10:26:00	vstimporter	C:\ProgramData\Vicon\Tracker\Objects\wanda.vsk
10:26:00	VSTImporter	Succeeded
10:26:00	Import XCP	C:\ProgramData\Vicon\Tracker\Calibrations\LatestCalibration.xcp
10:26:00	Import XCP	Import Succeeded
10:26:00	Export XCP	C:\ProgramData\Vicon\Tracker\Calibrations\LatestCalibration.xcp
10:26:00	Export XCP	Export Succeeded



INTEGRAÇÃO DOS SISTEMAS

➤ SDK AR.Drone:

- ✓ Seu *kit* de desenvolvimento é composto por dois principais pacotes de funções:
 - ✓ ***ArDroneLIB***
 - ✓ ***ArDroneTool***

INTEGRAÇÃO DOS SISTEMAS

➤ *ARDroneLIB:*

- ✓ Controla e configura a comunicação sem fio;
- ✓ API de processamento de vídeo;
- ✓ Biblioteca de configuração de controle, transmissão de vídeo e dados.

INTEGRAÇÃO DOS SISTEMAS

➤ *ARDroneTool*:

- ✓ Fornece as funções de controle de movimento;
- ✓ Controla a transmissão de dados de voo (velocidade, altitude, etc...) – navdata;
- ✓ Gerenciamento da *thread* de vídeo.

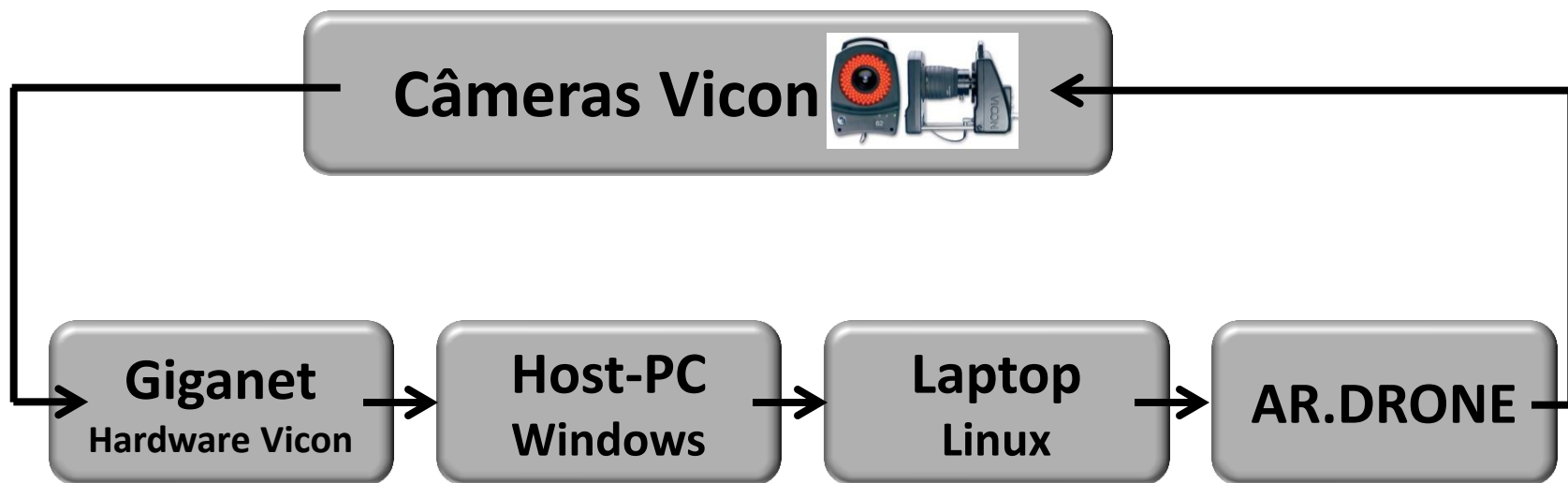
INTEGRAÇÃO DOS SISTEMAS

➤ SDK Vicon:

- ✓ Biblioteca desenvolvida em linguagem C++;
- ✓ Fornece um conjunto de funções para a aquisição e tratamento de dados;
- ✓ Utilização somente e conjunto com o *software* proprietário *Vicon Tracker*.

INTEGRAÇÃO DOS SISTEMAS

➤ Diagrama de Blocos:



MODELAGEM DINÂMICA

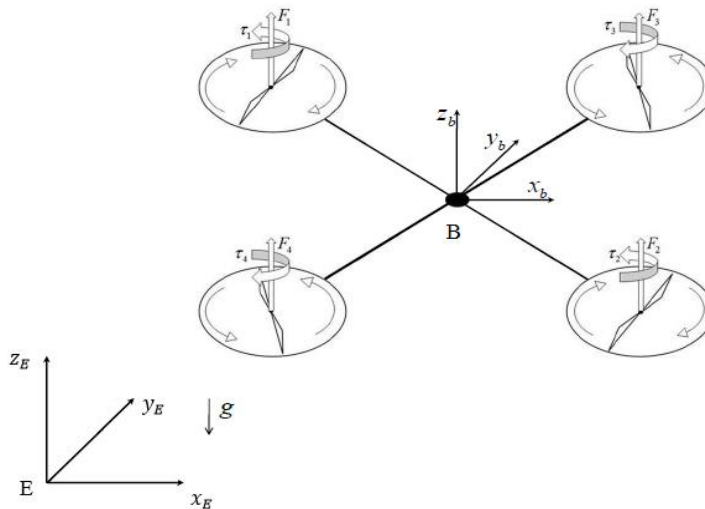
► Sistemas de Coordenadas:

- Referencial fixo usado pelo sistema de captura *Vicon System*:

$$E = \{x_E, y_E, z_E\}$$

- Referencial móvel, com eixos fixados próximo ao centro de gravidade do quadricóptero:

$$B = \{x_B, y_B, z_B\}$$



MODELAGEM DINÂMICA

➤ Relação entre os referencias:

- ✓ Seja $\mathbf{p}_E = \{x_E, y_E, z_E\}$ um vetor posição no referencial inercial e um vetor posição no referencial do veículo $\mathbf{p}_B = \{x_B, y_B, z_B\}$, podemos obter a seguinte relação entre eles: $\mathbf{p}_B = R\mathbf{p}_E$, onde R é uma matriz de rotação nos eixos x,y,z.

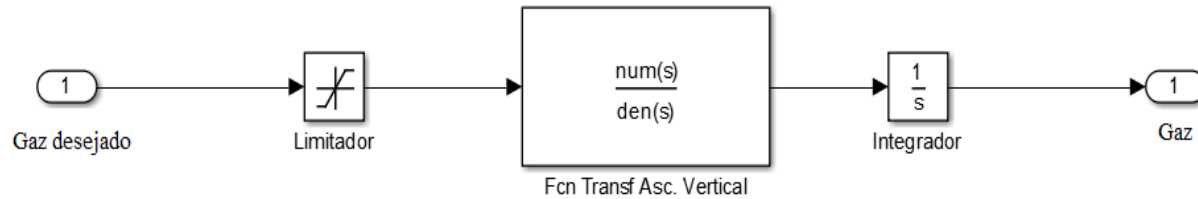
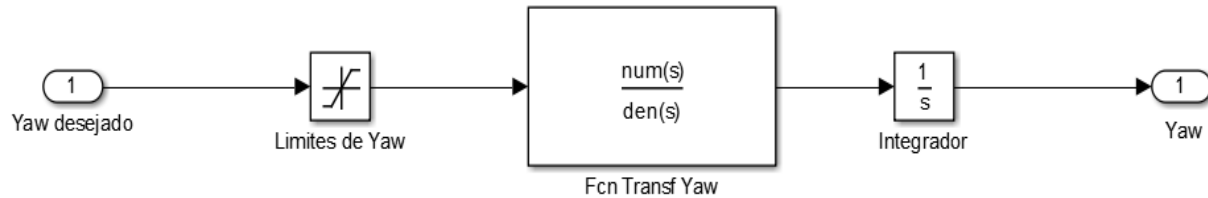
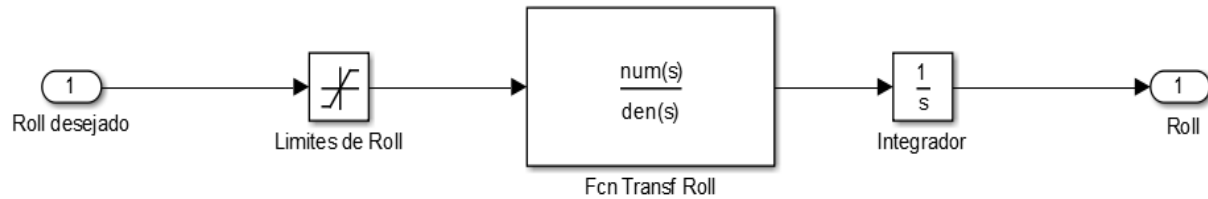
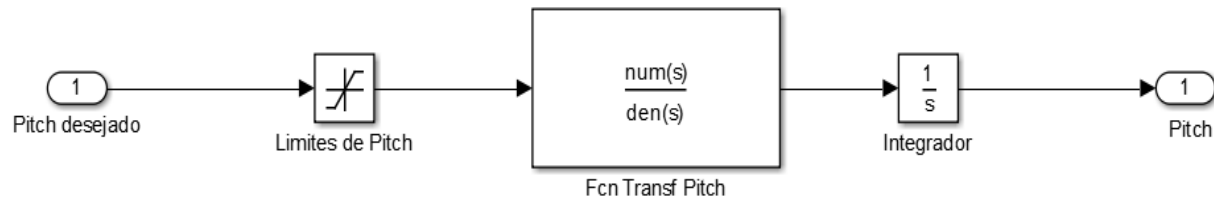
$$R(\gamma, \alpha, \beta) = \begin{bmatrix} \cos \beta \cos \alpha & -\cos \gamma \sin \alpha & -\sin \beta \\ \cos \gamma \sin \alpha + \sin \gamma \sin \beta \cos \alpha & \cos \gamma \sin \alpha - \sin \gamma \sin \beta \sin \alpha & -\cos \beta \cos \alpha \\ \sin \gamma \sin \alpha - \cos \gamma \sin \beta \cos \alpha & \sin \gamma \cos \alpha + \cos \gamma \sin \beta \sin \alpha & \cos \gamma \cos \beta \end{bmatrix}$$

MODELAGEM DINÂMICA

➤ Modelagem utilizada:

- ✓ Em vez de usarmos a modelagem clássica de um quadricóptero, foi utilizada uma modelagem levando em consideração seu controle interno, sendo este já modelado seguindo o modelo clássico, isto é, considerando as velocidades dos rotores como entradas e os ângulos de Euler como saída.

MODELAGEM DINÂMICA



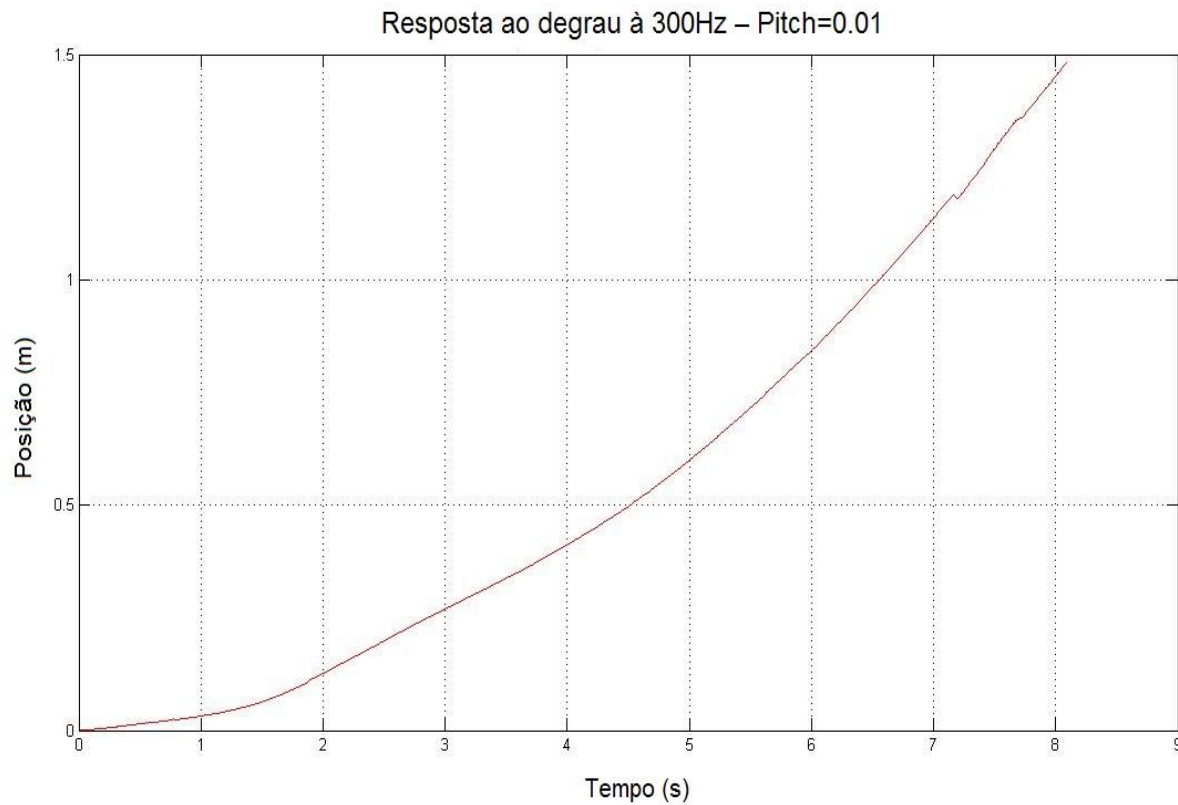
MODELAGEM DINÂMICA

➤ Modelagem utilizada:

- ✓ Para a obtenção desse modelo, foi aplicado um degrau em cada função de controle do quadricóptero;
- ✓ Com auxílio do sistema Vicon, foi medida a posição do veículo com amostragem periódica;
- ✓ Os dados foram então guardados em um arquivo.

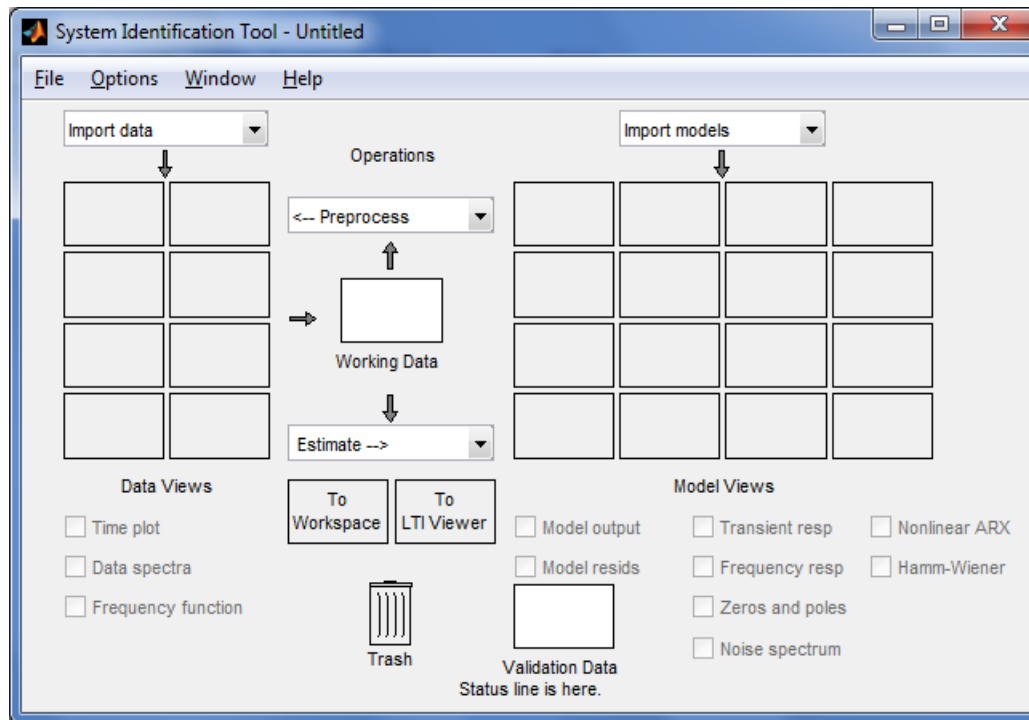
MODELAGEM DINÂMICA

- ✓ Resposta a um degrau medida:



MODELAGEM DINÂMICA

- ✓ Para se obter um modelo que aproxime a dinâmica do movimento do quadricóptero, foi usada a ferramenta *System Identification Toolbox* do MATLAB.
- ✓ Assim, foi identificada a função de transferência de cada uma das funções de controle do AR.Drone.



MODELAGEM DINÂMICA

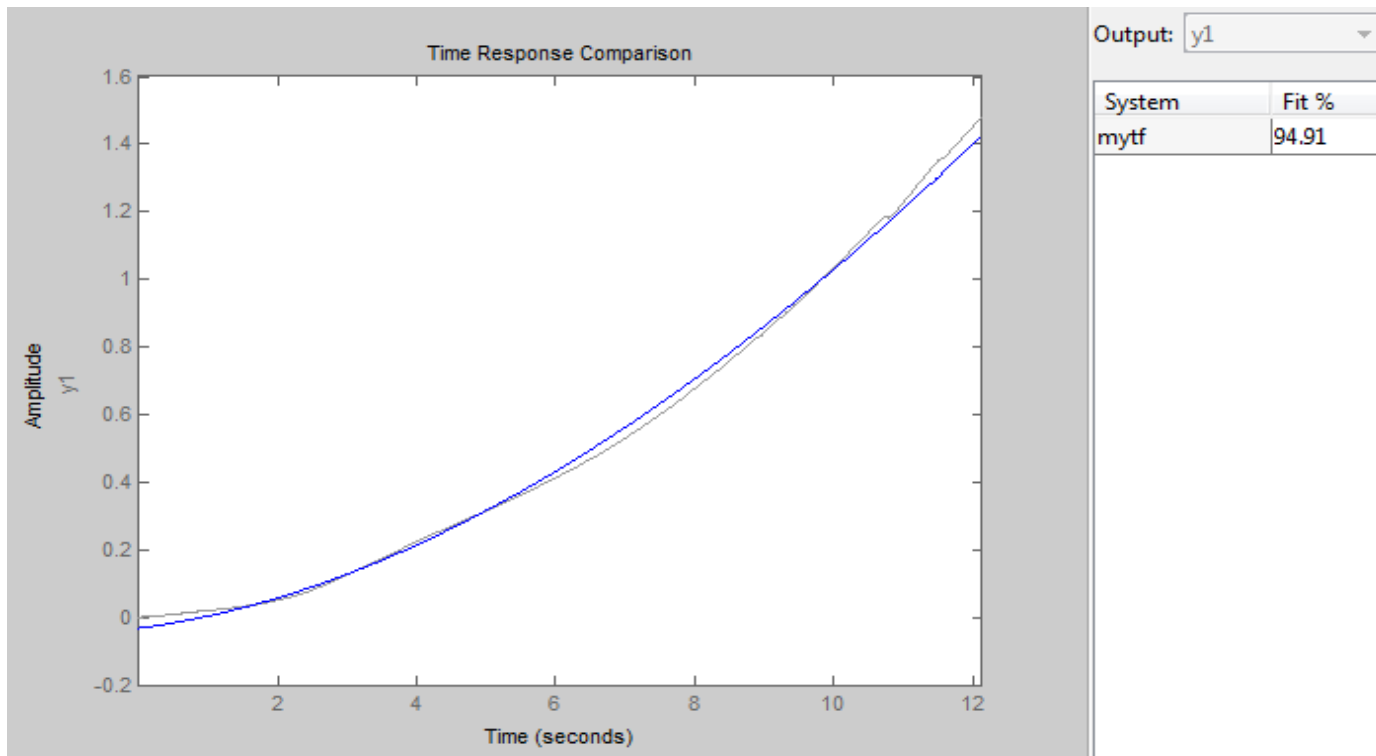
➤ Modelagem utilizada:

- ✓ Obteve-se a função de transferência :

$$H(s) = 2 \frac{s+1}{s(s+0,05)}$$

MODELAGEM DINÂMICA

- ✓ Para a validação do modelo obtido, foi usada a função *compare* do MATLAB, que compara a resposta de um modelo dinâmico aos dados obtidos na resposta ao degrau.



CONTROLE

➤ Problema Proposto:

- ✓ Uma vez feita a modelagem dinâmica do AR.Drone, pode-se projetar um controlador adequado ao veículo e à aplicação proposta:
 - ✓ Rastreamento de um alvo móvel.

CONTROLE

➤ Definição do tipo de controle:

- ✓ Avaliando o modelo dinâmico do quadricóptero, foi proposto um controle com ações proporcional e derivativa (PD).
- ✓ Como a função de transferência do AR.Drone já possui um integrador, um controlador do tipo PD, permitiria eliminar o erro quando o alvo permanecer parado.

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

CONTROLE

➤ Implementação do PD:

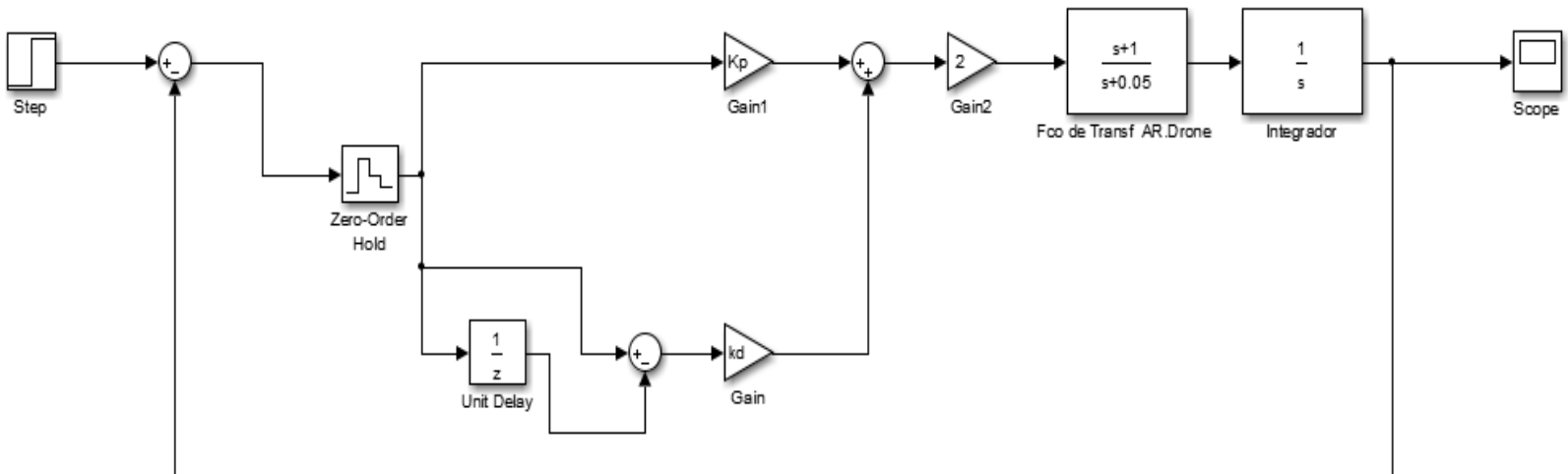
- ✓ Como o sistema de captura amostra a posição medida, foi necessário realizar o controle em tempo discreto.
- ✓ Escolheu-se a aproximação *Backward Euler* :

$$u(k) = K_p e(kh) + K_d \frac{e(kh) - e(kh - h)}{h}$$

CONTROLE

➤ Simulação do controle:

- ✓ Foram feitas simulações no SIMULINK/MATLAB, para obter-se os parâmetros do controlador.



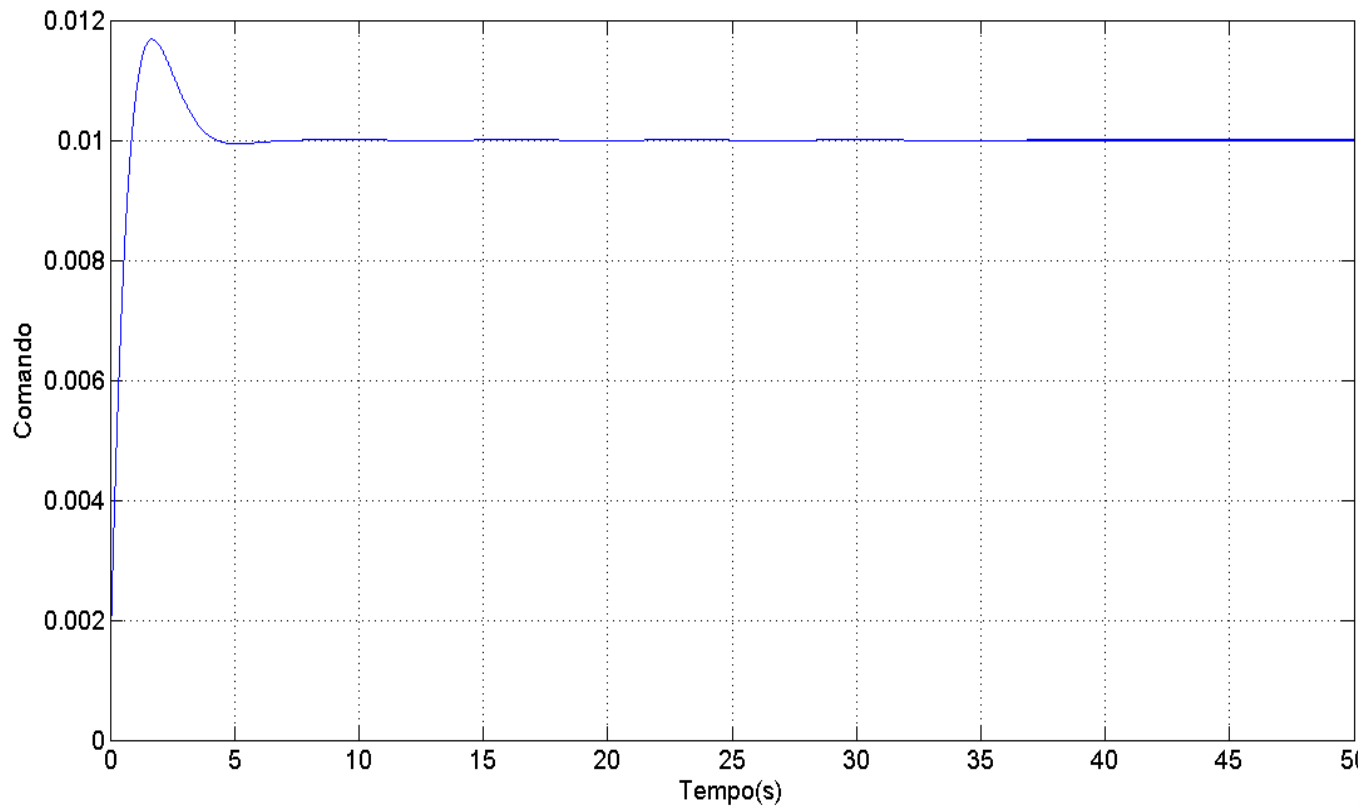
CONTROLE

➤ Resultados da simulação:

- ✓ A partir de iterações, foram obtidos os parâmetros K_p e K_d de maneira que o controle apresentasse uma resposta satisfatória.
- ✓ Foi proposto como restrição um *overshoot* inferior a 25%.
- ✓ Com esta especificação, foi obtida a resposta do sistema controlado apresentado com $K_p=1$ e $K_d=0,07$.

CONTROLE

➤ Resposta ao degrau do controle simulado:



CONTROLE

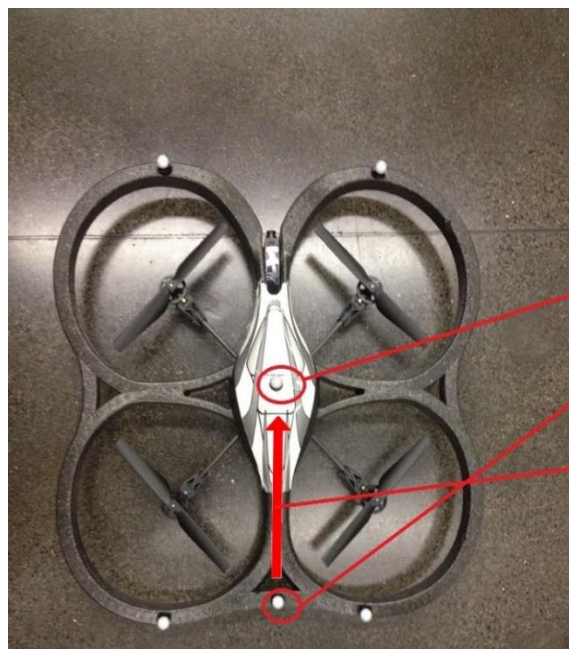
➤ Algoritmo de controle:

- ✓ Foi elaborado um algoritmo que implementa o controle proposto, em linguagem C++.
- ✓ Embora o AR.Drone possa operar no espaço tridimensional, o algoritmo implementa o controle somente no plano horizontal (XY), pois a área de captura das câmeras limitou a altura em até 50 cm.

CONTROLE

➤ Algoritmo de controle:

- ✓ Foi criado no algoritmo um vetor orientação que fornece o ângulo de rumo, cujo intuito é poder mapear tanto a posição do alvo, quanto do AR.Drone, em relação ao sistema de coordenadas fixo do veículo;



Marcadores centrais

Vetor orientação criado no algoritmo

CONTROLE

➤ Algoritmo de controle:

- ✓ O ângulo de rumo (A_{rumo}) é ângulo entre o vetor orientação, definido como $\mathbf{R}_o = \{x_b, y_b\}$ e o eixo y_E .
- ✓ O cálculo deste ângulo é realizado pela equação :

$$A_{rumo} = \cos^{-1} \frac{\mathbf{R}_o \cdot \mathbf{y}_e}{\|\mathbf{R}_o\| \cdot \|\mathbf{y}_e\|}$$

TESTES E RESULTADOS EXPERIMENTAIS

➤ Ambiente de teste:

- ✓ Os experimentos foram realizados no Laboratório de Controle e Automação da UERJ, onde o sistema Vicon está instalado;



TESTES E RESULTADOS EXPERIMENTAIS

➤ Configuração do experimento:

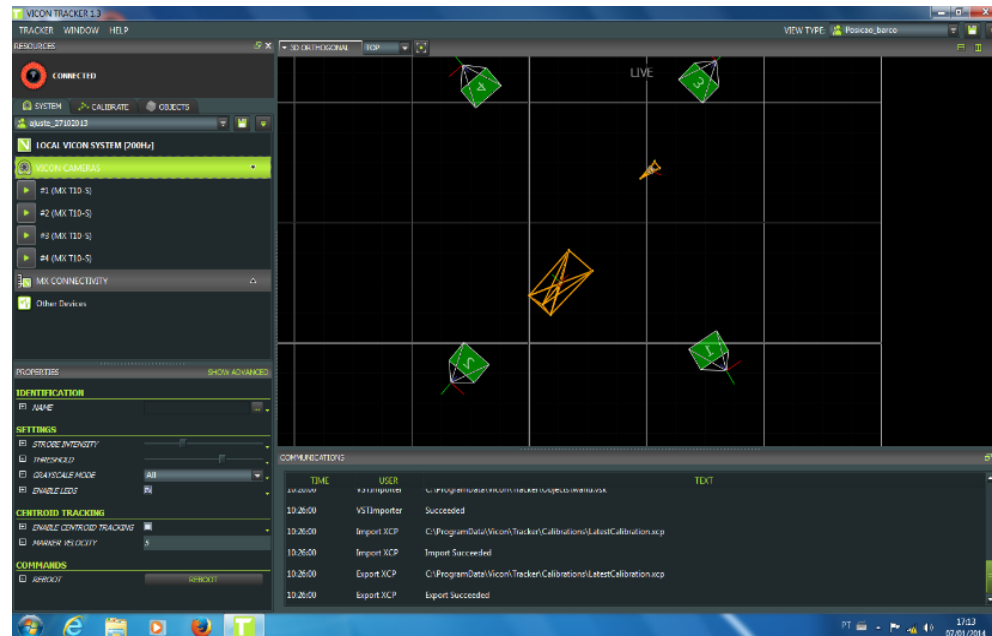
- ✓ Foram demarcadas as posição de partida do AR.Drone e do alvo, de modo que a distância entre eles seja máxima.
- ✓ Foi considerando os marcadores centrais de ambos como referencial da posição marcada na área de captura.
- ✓ Após isto, foi configurado como alvo a ser seguido, um carro de controle remoto, e posicionados o AR.Drone e o alvo.



TESTES E RESULTADOS EXPERIMENTAIS

➤ Configuração do experimento:

- ✓ No computador do laboratório, onde está instalado o *software Vicon Tracker 1.3*, foram criados os objetos virtuais.
- ✓ Foram capturadas as imagens dos marcadores e gerados objetos com formato referente à disposição dos marcadores de cada veículo.

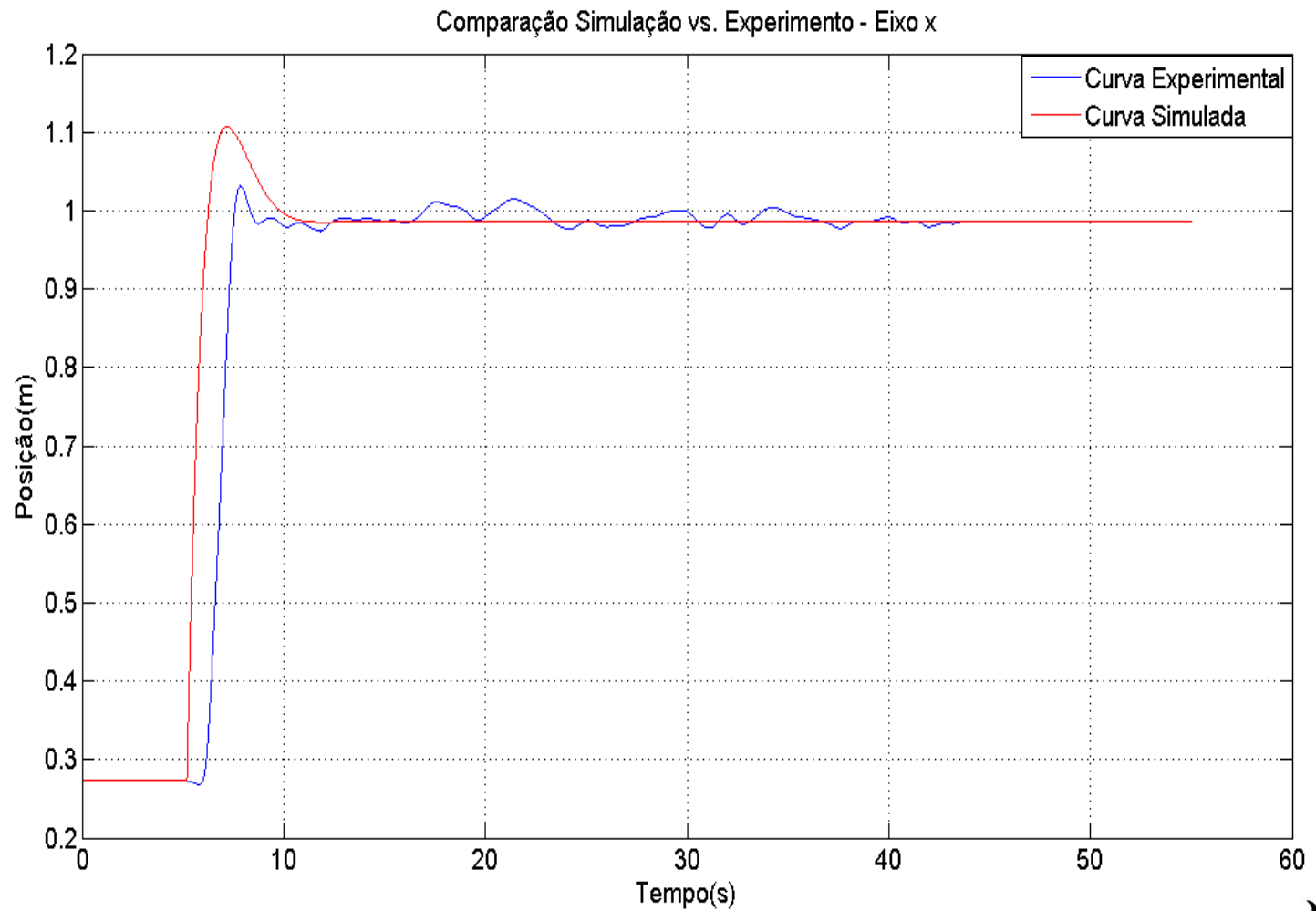


TESTES E RESULTADOS EXPERIMENTAIS

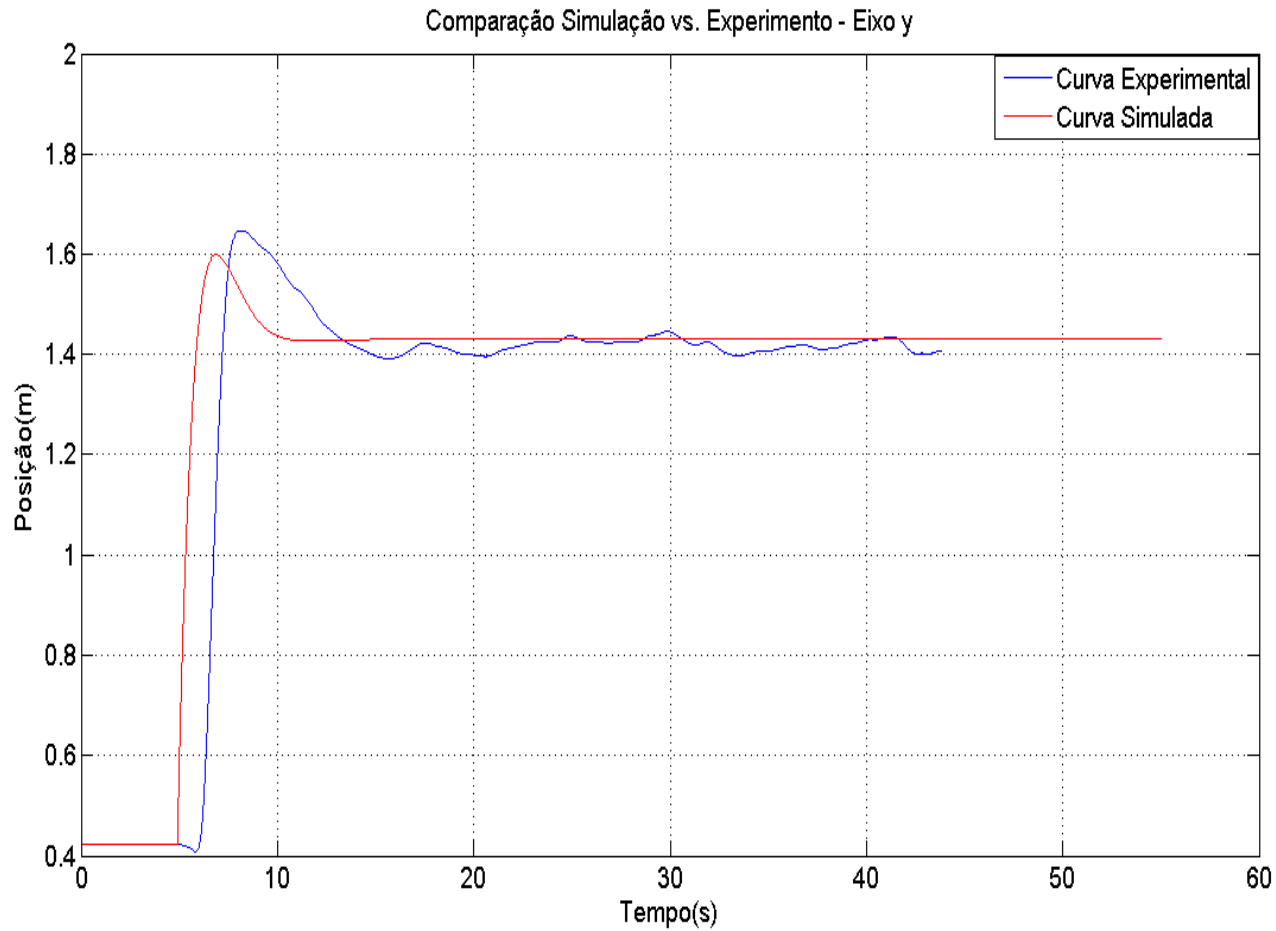
➤ Resultados Experimentais:

- ✓ Nos experimentos usando os parâmetros do controle simulado, foi constatado que os ganhos K_p e K_d necessitavam de alguns ajustes, pois o comportamento do veículo não de acordo com a simulação.
- ✓ Após alguns ajustes, K_p e K_d receberam os valores de 0,6 e 0,2, respectivamente.

TESTES E RESULTADOS EXPERIMENTAIS



TESTES E RESULTADOS EXPERIMENTAIS



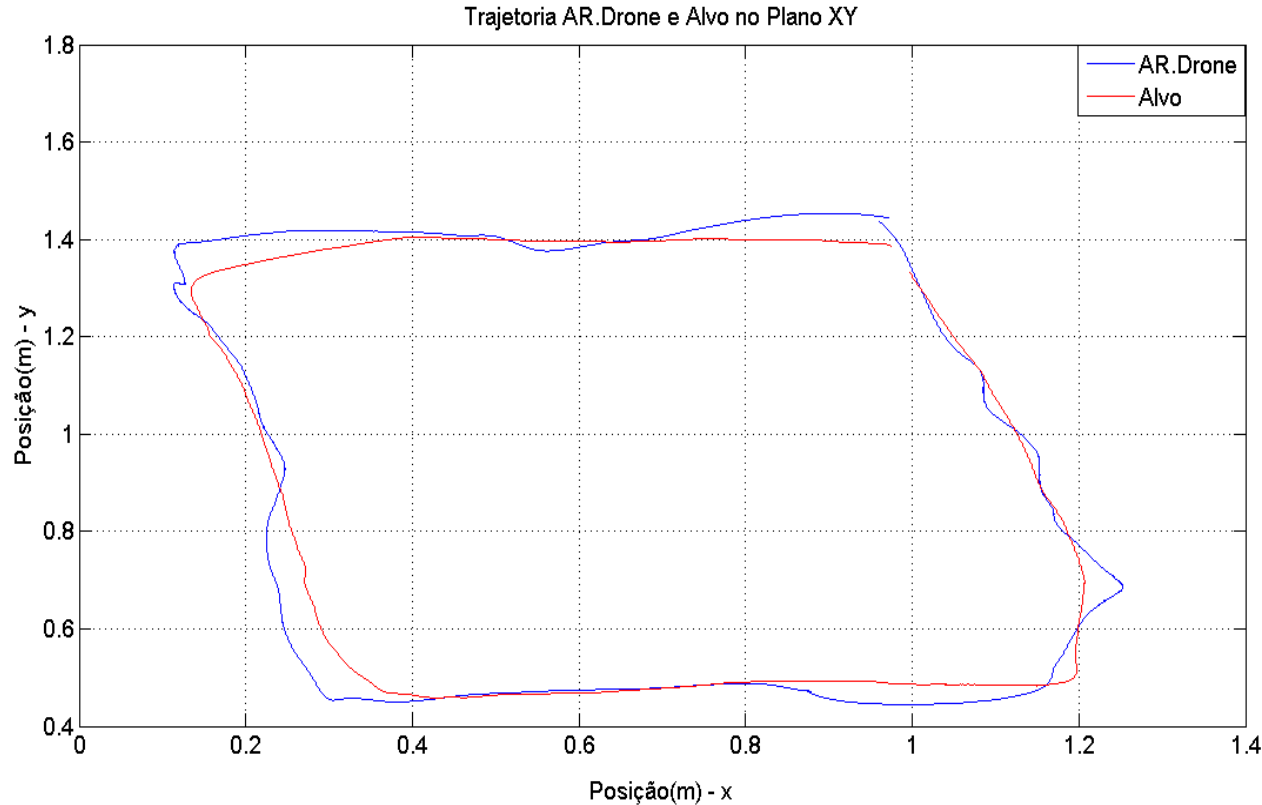
TESTES E RESULTADOS EXPERIMENTAIS

- ✓ Verificou-se um transitório em que a resposta do sistema se comporta como um sistema de fase não-mínima. Isto pode ser relacionado:
 - ✓ ao sistema de controle para a estabilização embarcado e
 - ✓ a dinâmica de decolagem do veículo.
- ✓ Também verifica-se um atraso médio de cinco segundos para a partida do veículo.
 - ✓ Isso se deve aos atrasos gerados pelos encadeamentos de rede.



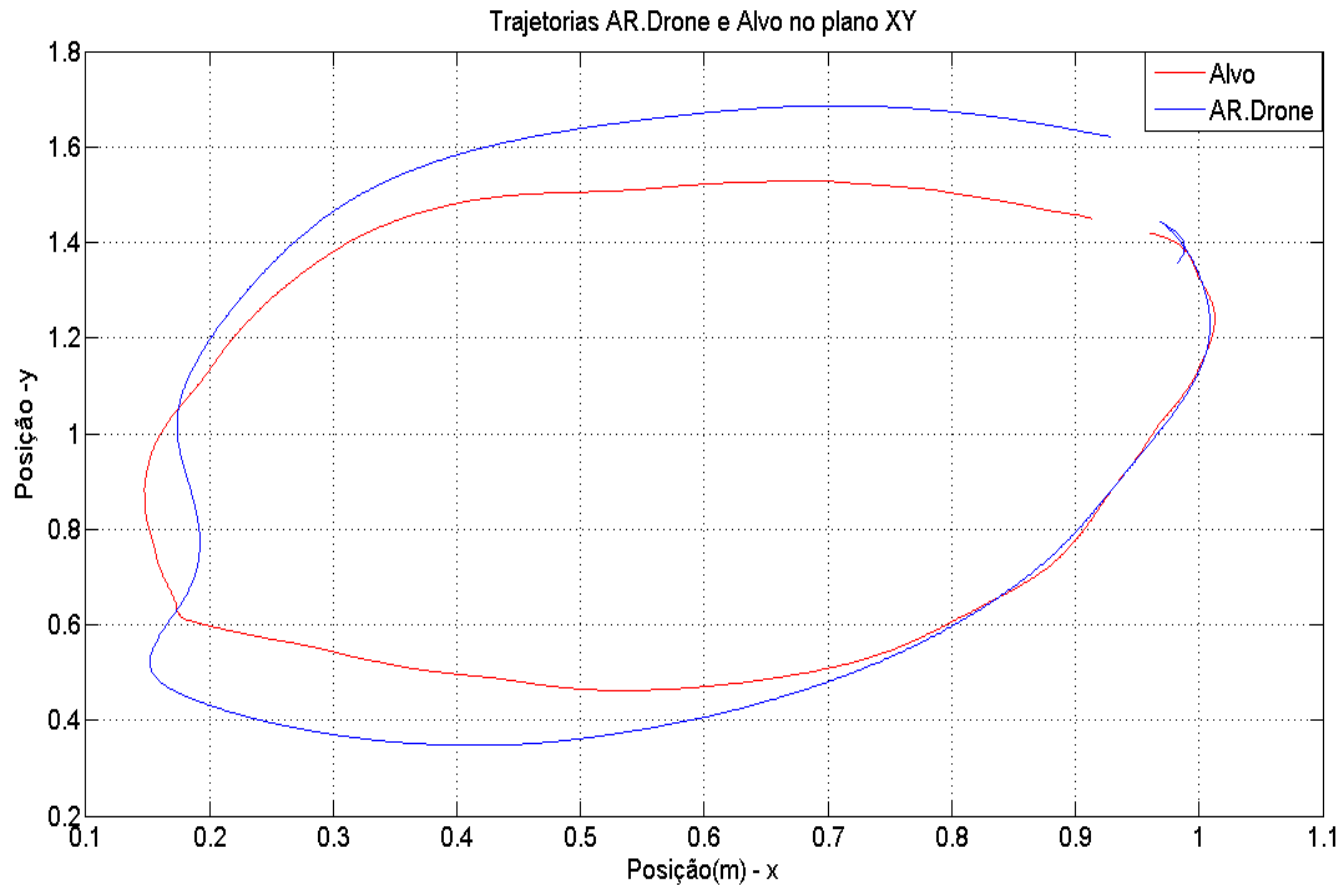
TESTES E RESULTADOS EXPERIMENTAIS

- ✓ No gráfico das trajetórias no plano XY, pode-se concluir que quando o alvo se move com uma velocidade baixa, o AR.Drone tem um desempenho melhor ao segui-lo.



TESTES E RESULTADOS EXPERIMENTAIS

- ✓ Ao aumentarmos a velocidade do alvo, o desempenho piora, porém ainda é satisfatório.
- ✓ Verifica-se um erro de no máximo 15 cm entre as posições do AR.Drone e do alvo.



Conclusão – Parte II

- ✓ Controle PD foi eficaz para o problema proposto.
- ✓ Dificuldade de obtenção de um modelo dinâmico do veículo, levando-se em consideração o seu sistema de controle embarcado.
- ✓ Foi usado um artifício (*extern "c"*) para permitir a compilação do *software* de integração entre os SDK's.



Trabalhos Futuros

- ✓ Reformar o *layout* do Laboratório de Controle e Automação (LCA) tendo-se em vista:
 - ✓ Aumento da área de captura do sistema Vicon;
 - ✓ Compartilhamento do LCA entre diversos experimentos e aulas e outras atividades.
- ✓ Aplicações em cooperação entre USVs, quadricópteros e outros veículos.
- ✓ Desenvolver e testar outras estratégias de controle.



Agradecimentos

- Aos docentes, discentes e técnicos colaboradores deste projeto.
- Convite para esta apresentação feito pelo Ramo Estudantil IEEE UERJ.
- Financiamentos:
 - Parceria CNPq/FAPERJ no Edital nº 25/2010: Apoio a Núcleos Emergentes de Pesquisa no Estado do Rio de Janeiro – 2010 – PRONEM;
 - Edital FAPERJ nº 04/2011: Apoio às Engenharias – 2011.



Contato

- *Homepage:* <http://www.lee.uerj.br/~jpaulo>
- *E-mail:* jpaulo@ieee.org
- *Telefone:* 0xx-21-2334-0027



Referências

- Gomes, L. L. e Leal, L. P. (2014). Controle de um quadricóptero por servovisão. Projeto de Graduação em Engenharia Eletrônica — UERJ, Rio de Janeiro.
- Gomes, L. L., Leal, L., Oliveira, T. R. e Cunha, J. P. V. S. (2014). “Controle de um veículo quadrirrotor usando um sistema de captura de movimentos,” in *Anais do XX Congresso Brasileiro de Automática*, Belo Horizonte, pp. 1474-1481, setembro.
- Gomes, L. L., Leal, L., Oliveira, T. R., Cunha, J. P. V. S. e Revoredo, T. C. (2016). “Unmanned quadcopter control using a motion capture system,” *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3606-3613, August.

