

**UNIVERSIDADE FEDERAL DE VIÇOSA
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

CARLOS ROBERTO CASTELANO JÚNIOR

**SISTEMA DE COMUNICAÇÃO POR UDP PARA
SIMULAÇÃO HARDWARE-IN-LOOP DE UM
VEÍCULO AÉREO NÃO TRIPULADO**

**VIÇOSA
2012**

CARLOS ROBERTO CASTELANO JÚNIOR

**SISTEMA DE COMUNICAÇÃO POR UDP PARA
SIMULAÇÃO HARDWARE-IN-LOOP DE UM
VEÍCULO AÉREO NÃO TRIPULADO**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 - Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. M.Sc. Alexandre S. Brandão.

VIÇOSA
2012

CARLOS ROBERTO CASTELANO JÚNIOR

**SISTEMA DE COMUNICAÇÃO POR UDP PARA SIMULAÇÃO
HARDWARE-IN-LOOP DE UM VEÍCULO AÉREO NÃO
TRIPULADO**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 - Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 05 de Novembro de 2012.

COMISSÃO EXAMINADORA

Prof. M.Sc. Alexandre Santos Brandão - Orientador
Universidade Federal de Viçosa, Brasil

Eng. B.Sc. Claudio Dario Rosales - Membro
Universidade Nacional de San Juan, Argentina

M.Sc. Mauro de Oliveira Prates - Membro
Universidade Federal de Viçosa, Brasil

“A meus pais e minha irmã”

Agradecimentos

Aos meus pais Carlos e Rachel, que me ensinaram que o maior herança que poderiam deixar para um filho é o conhecimento.

À minha irmã Karine, por sempre me apoiar na busca pelos meus objetivos.

Ao meu professor Alexandre Santos Brandão, por ter também desempenhado o papel de amigo.

Aos meus tios e tias, de sangue e de coração, pela confiança e apoio familiar.

E por fim, a todos os meus amigos, que tornaram a rotina pesada de estudante da Engenharia Elétrica menos cansativa.

Carlos Roberto Castelano Júnior

“Genialidade é 1% inspiração e 99% transpiração.”

Thomas Edison

Resumo

A simulação *hardware-in-loop simulation* (HILS) é uma técnica utilizada para controles de processos que permite desenvolver, estudar e monitorar sistemas, em tempo real. Este tipo de simulação se diferencia por agregar dados reais do sistema ao laço de controle do processo. Esses dados são normalmente fornecidos por sensores e, a partir deles, são gerados os parâmetros que informam a situação real da planta, para que seja feita a realimentação de um controlador. Uma comunicação de rede WiFi bidirecional permite a transferência dos dados colhidos para um ambiente computacional, onde os movimentos da aeronave podem ser reproduzidos através de uma interface gráfica de usuário (GUI, *Graphical User Interface*). No entanto, diante do esforço computacional exigido pelas tarefas de plotagem e realimentação periódica do controlador, as informações recebidas do VANT (Veículo Aéreo Não Tripulado) devem ser repassadas para outra estação terrestre, através de uma comunicação UDP (*User Datagram Protocol*). Dessa forma, os benefícios trazidos pela simulação HIL são capazes de tornar mais eficiente a implementação de novos sistemas de controle da aeronave, reduzindo custos e tempo de projeto, além de fornecer mais segurança aos equipamentos e aos seus operadores. Nesse contexto, o presente trabalho tem como objetivo desenvolver um sistema que permita reduzir o tempo gasto para execução das tarefas de controle e exibição gráfica de um VANT, de forma a garantir sua estabilidade.

Sumário

Lista de Figuras

1	Introdução	8
1.1	<i>Hardware-in-loop Simulation</i> (HILS)	8
1.2	AR-Drone	10
1.3	Objetivos	12
2	Plataforma de Simulação em Tempo Real: AuRoRA	13
2.1	O funcionamento da plataforma	15
3	Sistema de Comunicação em Rede	17
3.1	Estrutura do sistema de comunicação UDP	18
3.1.1	Protocolo de codificação de pacote	19
3.1.2	Protocolo de decodificação de pacote	19
4	Análise dos Resultados	21
5	Considerações Finais	23
	Referências	24

Lista de Figuras

1	Estrutura do <i>Hardware-in-Loop Simulation</i>	9
2	Estrutura de processamento de dados.	9
3	Quadrotores.	10
4	Sistema de coordenadas de um quadrotor.	10
5	Veículo Aéreo Utilizado no trabalho.	11
6	Comunicação entre o VANT e a estação terrestre.	11
7	Diagrama de blocos do funcionamento interno da plataforma de simulação utilizada.	14
8	Demonstração do envio de instruções.	14
9	Perda de pacote durante a tarefa de plotagem.	16
10	Tipos de transmissão de rede.	17
11	Estrutura geral do sistema.	18
12	Protocolo de codificação de dados.	19
13	Estrutura geral do sistema, sem distribuição de tarefas entre dois computadores.	21
14	Duração das iterações da plataforma AuRoRA, com estrutura centralizada.	22
15	Duração das iterações da plataforma AuRoRA, com estrutura descentralizada.	22

1 *Introdução*

O avanço tecnológico de sistemas computacionais embarcados permitiu o desenvolvimento de uma nova geração de veículos aéreos inteligentes, capazes de interagir e de se conduzir nos mais variados tipos de ambientes de forma autônoma.

A navegação autônoma torna-se possível graças à implementações de algoritmos de controle não-lineares que modelam o voo de uma aeronave, permitindo que sejam gerados os sinais de controle necessários para o seu manejo ao longo de uma trajetória.

Nesse contexto, os VANTs (**V**eículos **A**éreos **N**ão **T**ripulados) se tornaram objetos ideais para a realização de missões de voo perigosas e repetitivas, inadequadas para um piloto humano, como por exemplo, em aplicações militares, incluindo mapeamento, patrulhamento, busca e resgate [1, 2, 3]

Para a validação de modelos cinemáticos e dinâmicos eventualmente propostos, a simulação em ambientes computacionais é amplamente utilizada, reduzindo custos, riscos e tempo para implementação desses sistemas [4, 5].

1.1 *Hardware-in-loop Simulation (HILS)*

A simulação computacional é um recurso muito utilizado para validação de sistemas, uma vez que é possível estimar o comportamento de suas variáveis antes mesmo de ser realizado qualquer teste prático.

Existem ainda os casos onde os dados coletados de experimentos práticos são agregados ao laço principal de controle de uma simulação, permitindo a criação de parâmetros reais para que seja feita a realimentação de um controlador, conforme ilustra a Figura 1.

Nesse contexto, o HILS (**H**ardware-**I**n-**L**oop **S**imulation) é um tipo de simulação em tempo real onde os sinais de entrada e saída de um simulador dependem dos valores reais do processo corrente, em um específico instante de tempo [6]. Tal simulação permite



Figura 1: Estrutura do *Hardware-in-Loop Simulation*.

testar um controlador real e um sistema de controle embarcado com uma maior confiabilidade, em diferentes condições de trabalho, uma vez que esse método de simulação não é exclusivamente computacional.

Do ponto de vista organizacional, a estrutura de uma HILS pode ser centralizada ou descentralizada. Quando centralizada, todos os dados referentes à leitura dos sensores embarcados ao VANT são transmitidos e processados por uma única unidade de processamento na estação em terra, responsável por gerar os sinais de controle e enviá-los de volta ao veículo (Figura 2(a)). Essa tarefa pode ser realizada através de uma plataforma de simulação computacional, que será apresentada no Capítulo 2. No entanto, em alguns casos, as tarefas realizadas pela plataforma de simulação podem exigir um tempo de processamento maior que o período de amostragem da aeronave, comprometendo sua estabilidade. Por esse motivo, muitas vezes é necessário que as atividades sejam distribuídas entre dois ou mais computadores, caracterizando uma estrutura de processamento descentralizada (Figura 2(b)).

Uma vez que dois computadores estejam conectados a uma mesma rede, é possível

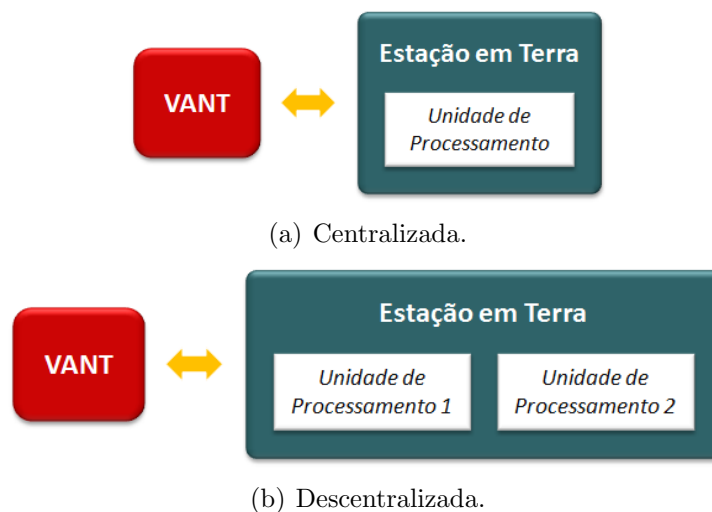


Figura 2: Estrutura de processamento de dados.



(a)

(b)

Figura 3: Quadrotores.

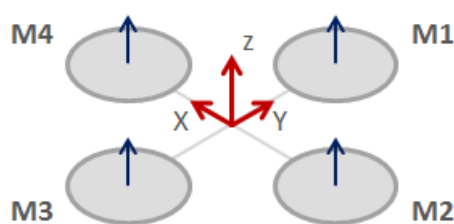


Figura 4: Sistema de coordenadas de um quadrotor.

compartilhar informações do processo monitorado, dividindo as tarefas a serem processadas e reduzindo o custo computacional exigido para cada uma das unidades de processamento em terra. Um método que permite tal comunicação em rede será proposto no Capítulo 3.

1.2 AR-Drone

Um quadrotor (Figura 3) é um veículo aéreo impulsionado por quatro hélices independentes, cada uma acoplada rigidamente a seu próprio motor. Isso equivale dizer que todas as forças que agem sob essa aeronave estão apontadas na mesma direção do eixo Z de seu sistema de coordenadas, conforme ilustrado na Figura 4.

Nesse contexto, o quadrotor AR-Drone (Figura 5(a)) tem desempenhado importante papel como plataforma para o desenvolvimento de pesquisas na área da robótica.

A movimentação dessa aeronave é feita por um computador embarcado, responsável por realizar o controle de voo em baixo nível, permitindo estabilizar, controlar a posição e navegar o quadrotor através de laços de controle internos, apenas variando-se a velocidade relativa de rotação de seus motores (M1, M2, M3 e M4). Devido a seu baixo peso (380 gramas) e pequenas dimensões (73,66 x 27,94 cm), a autonomia máxima da bateria do



(a) Quadrotor AR-Drone Parrot. (b) Sistema referencial fixo do AR-Drone.

Figura 5: Veículo Aéreo Utilizado no trabalho.



Figura 6: Comunicação entre o VANT e a estação terrestre.

AR-Drone permite cerca de 13 minutos contínuos de voo, podendo alcançar velocidades de até 5 m/s .

Além de duas câmeras embarcadas e de um altímetro, o AR-Drone também possui um sistema de sensoriamento baseado em uma IMU (*Inertial Measurement Unit*), com 6 graus de liberdade, permitindo o rastreamento de sua orientação em relação a um sistema referencial fixo (Figura 5(b)), formado pelos ângulos *Yaw*, *Pitch* e *Roll*, também conhecidos como ângulos de Euler.

Após ser ligado o computador embarcado ao sistema, é criada uma conexão de rede WiFi, permitindo que um computador externo se conecte remotamente à aeronave. Essa conexão estabelece um canal de comunicação bidirecional com o veículo, por onde são enviados e recebidos as leituras de seus sensores e os comandos para navegação (Figura 6).

1.3 Objetivos

O objetivo desse trabalho é desenvolver um sistema descentralizado de processamento de dados, que permita a redução do tempo gasto para geração de sinais de controle e exibição gráfica para um veículo aéreo não tripulado, de forma a garantir seu controle e estabilidade. Para isso, foram traçados os seguintes objetivos específicos:

- Adequação da plataforma de simulação já existente (AuRoRA), através da inserção de um bloco para comunicação UDP);
- Divisão das tarefas de geração de sinais de controle e exibição gráfica entre dois computadores, que se comunicam através de um *link* UDP;
- Desenvolvimento de um protocolo de codificação/ decodificação para validação dos pacotes de dados recebidos;
- Análise do desempenho do sistema descentralizado.

2 *Plataforma de Simulação em Tempo Real: AuRoRA*

Para realização desse trabalho, foi utilizada a plataforma de simulação em tempo real AuRoRA [7] (*Autonomous Robot Research and Application*), desenvolvida pelo grupo de pesquisa em robótica aérea da Universidade Federal de Viçosa (UFV) em parceria com a Universidade Federal do Espírito Santo (UFES). Sua estrutura pode ser observada na Figura 7. Tal sistema permite a simulação em tempo real de múltiplos VANTs, baseada em um algoritmo onde cada ação necessita de sua respectiva autorização para ser executada. A permissão para execução de uma tarefa somente é fornecida quando o período de amostragem do veículo aéreo simulado é atingido. O Algoritmo 1 apresenta a lógica de acesso a uma permissão, onde t é o tempo corrente e t_{max} é o tempo máximo da simulação. Tais permissões são monitoradas de forma independente para cada veículo aéreo simulado.

Algoritmo 1: Algoritmo de controle de permissão.

```

Inicialização;
while  $t < t_{max}$  do
  if Permissão de execução then
    Ler sensores;
    Calcular posição desejada;
    if Existe joystick then
      | Ler comandos;
    else
      | Aplicar controlador;
    end
    Enviar sinais de controle;
    Armazenar variáveis;
  end
  if Permissão para gráfico then
    | Exibir parte gráfica;
  end
end

```

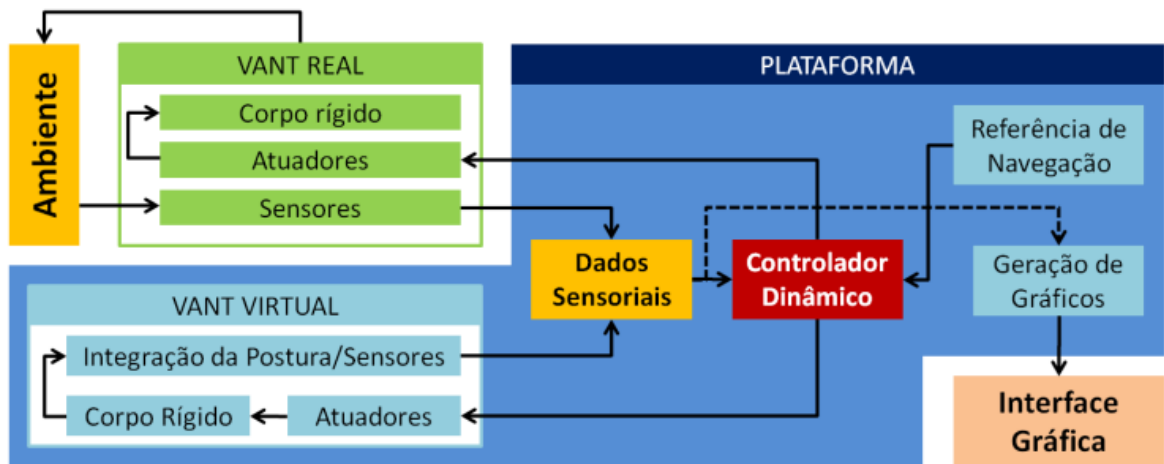
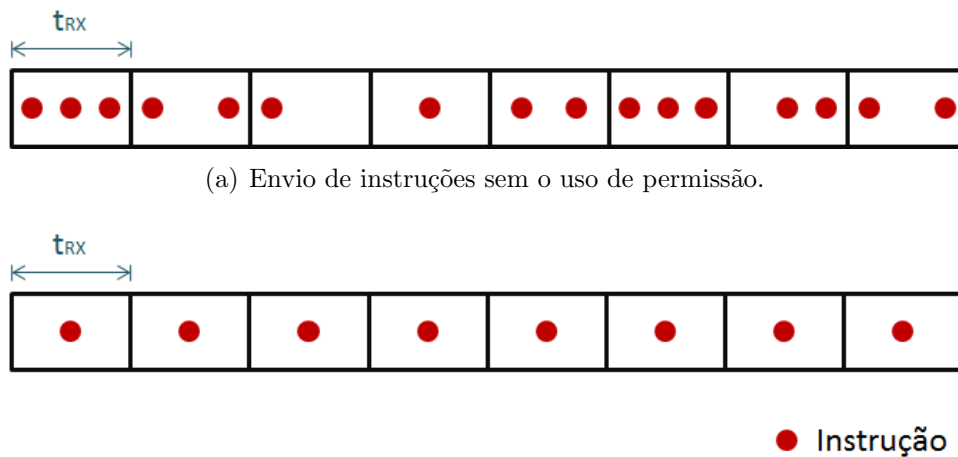


Figura 7: Diagrama de blocos do funcionamento interno da plataforma de simulação utilizada.



(a) Envio de instruções sem o uso de permissão.

(b) Envio de instruções com o uso de permissão.

Figura 8: Demonstração do envio de instruções.

Para o VANT em questão (Ar-Drone Parrot), os períodos de amostragem para recebimento e envio de dados são respectivamente de $t_{rx} = 1/30s$ e $t_{tx} = 1/50s$ (dados fornecidos pelo fabricante).

Tal técnica visa minimizar o esforço computacional, evitando que uma sequência de instruções seja desnecessariamente processada e enviada ao veículo em um mesmo período de amostragem, como ilustrado na Figura 8.

2.1 O funcionamento da plataforma

Conforme já mencionado, a comunicação entre o AR-Drone Parrot e a plataforma descrita é estabelecida através de um *link wireless* bidirecional, permitindo que os dados originados da leitura dos sensores embarcados no VANT sejam recebidos pela plataforma terrestre, sempre respeitando o tempo de permissão para recebimento de dados t_{rx} .

Uma vez capturados, os dados fornecidos por esses sensores são comparados com uma trajetória de referência pré-estabelecida pelo usuário. Os sinais de controle, calculados através do erro de navegação, são por fim fornecidos e transmitidos de volta ao veículo aéreo, também respeitando seu tempo de permissão t_{tx} .

Após reagir às ações do controlador, uma nova postura é determinada pelo veículo aéreo. Finalmente, seus sensores embarcados registram as novas condições de voo, que são novamente transmitidas à plataforma, reiniciando o ciclo de controle.

A estrutura dessa plataforma permite também habilitar ou desabilitar vários módulos, como por exemplo, a geração de gráficos, sem comprometer a execução de qualquer outra tarefa. Como forma de segurança, também é possível habilitar um joystick para controle emergencial da aeronave, uma vez que sua atuação se sobrepõe a ação do controlador.

Ao final de cada iteração, os dados de navegação são armazenados, criando-se assim um registro, para possíveis utilizações futuras.

A geração de gráficos permite o monitoramento em tempo real de uma situação de voo, que muitas vezes pode ser realizada em ambientes fora do alcance da visão do usuário. Além disso, a reprodução gráfica permite a reprodução de situações de voo já registradas e até mesmo a visualização do comportamento do VANT diante da realização de alguma outra tarefa ainda não realizada.

No entanto, a representação gráfica em um ambiente virtual demanda um esforço computacional que pode, na maior parte dos casos, comprometer o tempo de permissão estabelecido para o cumprimento de alguma tarefa.

Supondo-se que a exibição gráfica do VANT seja realizada a cada t_g segundos, e que o tempo médio para execução da tarefa de plotagem seja de t_m segundos. Conforme mostrado na Figura 9, sempre que uma tarefa de geração gráfica é solicitada, o tempo de permissão para envio/recebimento de dados é estourado, gerando lacunas nas quais o sistema de controle permanece sem receber dados dos sensores e sem enviar novos sinais de controle ao VANT. É importante ressaltar que, durante esse período de lacunas, a última

leitura dos sensores enviada à plataforma é mantida, comprometendo a realimentação do controlador. Da mesma forma, o último sinal de controle enviado à estação aérea permanece exercendo influência sobre seus atuadores, levando o veículo aéreo à posições indesejadas.

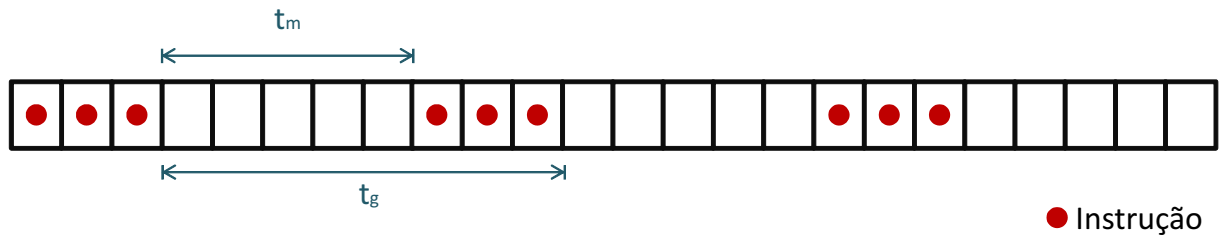


Figura 9: Perda de pacote durante a tarefa de plotagem.

Uma possível solução para esse problema é distribuir as atividades de controle e exibição gráfica entre dois computadores interconectados. Nesse contexto, o presente trabalho propõe a utilização de uma segunda estação terrestre, conectada à primeira através de um *link* UDP (*User Datagram Protocol*) [8]. Sua estrutura será apresentada no capítulo seguinte.

3 *Sistema de Comunicação em Rede*

O desenvolvimento de diferentes tipos de protocolos de comunicação em rede possibilitou o surgimento de ferramentas que tornaram possível a interconexão remota entre dois ou mais pontos de acesso distintos. Nesse contexto, o TCP-IP e o UDP [9] são dois protocolos de comunicação amplamente utilizados para troca informações entre computadores, em uma mesma rede.

Do ponto de vista estrutural, a transmissão de dados em uma rede pode ser realizada seguindo três métodos distintos: *Unicasting*, *Broadcasting* ou *Multicasting* [10, 11]. A transmissão por Unicasting (Figura 10(a)) é aquela em que uma mensagem é transmitida de um servidor para um único cliente de destino, de endereço IP preestabelecido pelo usuário. Quando em modo Broadcasting (Figura 10(b)), essa transmissão é feita para todos os pontos de acesso presentes na rede. No entanto, essa transmissão também pode ser feita apenas para clientes específicos de uma rede, através da inserção de seus respectivos endereços IP, segundo os interesses do usuário. Nesse caso, diz-se que a transmissão é do tipo Multicasting (Figura 10(c)). No presente trabalho, a comunicação é feita através de uma comunicação bidirecional direta entre apenas dois computadores, caracterizando uma estrutura de transmissão de dados do tipo *Unicasting*. No entanto, a mesma plataforma também oferece suporte ao *Multicasting*, permitindo o envio dos dados desejados para vários outros computadores conectados a uma mesma rede.

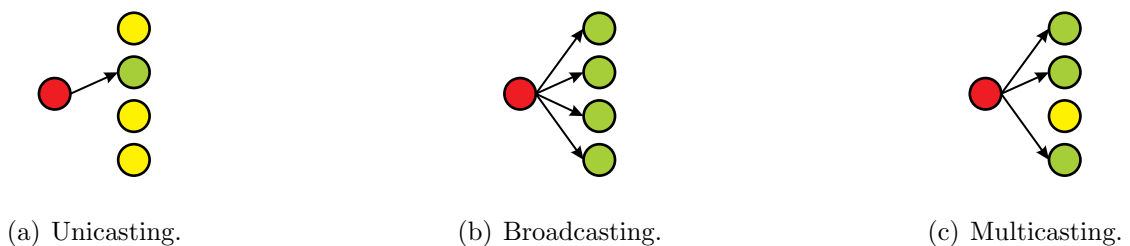


Figura 10: Tipos de transmissão de rede.

Uma vez que o protocolo de comunicação por TCP-IP não oferece suporte ao Multicasting, o UDP torna-se um protocolo largamente utilizado para implementação de sistemas desse tipo [11]. Além disso, a comunicação por UDP possui uma latência de transmissão muito mais baixa, já que o protocolo TCP-IP aguarda o recebimento dos pacotes perdidos até a ocorrência de um *timeout* (estouro no tempo de transmissão). Por esse motivo, o UDP é considerado uma ferramenta muito propícia para a transmissão de dados em tempo real [10], como mostrado em [8, 12], onde a transmissão dos sinais de controle para diferentes tipo de robôs é feita utilizando-se esse mesmo protocolo.

3.1 Estrutura do sistema de comunicação UDP

Conforme já mencionado, esse trabalho propõem a distribuição das tarefas de controle e de exibição gráfica entre dois computadores pessoais interconectados através de um *link* UDP bidirecional, como realizado em [13]. A partir da Figura 11, tem-se que a estação terrestre “A” estabelece uma conexão WiFi direta com o ArDrone, por onde são continuamente capturadas as leituras dos sensores embarcados à aeronave. Após processadas na estação terrestre “A”, os sinais de controle resultantes do controlador de alto nível são enviados aos atuadores do VANT, conduzindo-o através de uma trajetória preestabelecida pela plataforma computacional. Simultaneamente, as informações sobre postura são enviadas através do *link* UDP em direção à estação terrestre “B”. Diz-se que as estações terrestres “A” e “B” desempenham uma relação de servidor-cliente, respectivamente, como descrito em [14].

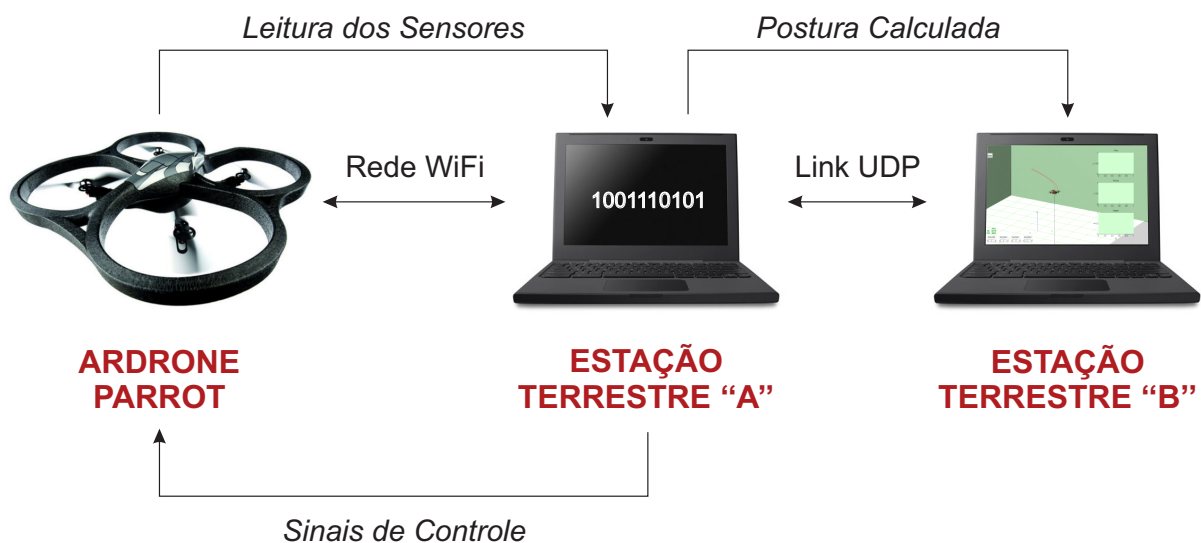


Figura 11: Estrutura geral do sistema.

Com o objetivo de estabelecer uma conexão UDP entre servidor e cliente, fez-se

necessária a inserção de um novo bloco de programa na estrutura geral da plataforma AuRoRA, anteriormente apresentada. Com isso, o bloco que desempenhava a função de plotagem da interface gráfica foi desabilitado, e substituído por um bloco de servidor UDP. Já na estação terrestre “B”, essa mesma plataforma foi preparada para receber os dados referentes à postura do ArDrone. Através de um bloco de cliente UDP, esses dados podem ser coletados, validados, passam pela plataforma, e por fim, é realizada a tarefa de plotagem.

Os protocolos para encapsulamento e desencapsulamento dos pacotes será apresentado nas subseções a seguir.

3.1.1 Protocolo de codificação de pacote

Apesar de suas aparentes vantagens, o protocolo de comunicação UDP não garante a recepção correta dos dados enviados, havendo a necessidade da criação de um algoritmo para verificação de perdas de pacote durante a transmissão [14, 15]. No entanto, não é necessário realizar a retransmissão do pacotes perdidos, uma vez que eles contém dados antigos sobre o sistema, que possui característica de plotagem em tempo real. Dessa forma, os pacotes inválidos são simplesmente descartados.

Uma vez processados pela estação terrestre “A”, a codificação dos dados para transmissão UDP é feita segundo o protocolo padrão (Figura 12). Primeiramente, os caracteres ‘N’ e ‘E’ são enviados, sequencialmente, identificando o início da transmissão do pacote de dados. Em seguida, são transmitidos o número de robôs (Nr) presentes na simulação, os respectivos modelos dos veículos aéreos utilizados (Mr), e um último pacote (dados) contendo informações como posição, orientação e posição desejada para cada robô. Finalmente, os caracteres ‘R’ e ‘O’ são transmitidos, indicando o fim do pacote.



Figura 12: Protocolo de codificação de dados.

3.1.2 Protocolo de decodificação de pacote

Toda vez que é detectada a chegada de um pacote de dados UDP na estação cliente, a leitura do canal de entrada UDP é habilitada, iniciando os processos de recepção, decodificação e validação do pacote. A decodificação dos dados deve seguir a mesma estrutura

apresentada na Figura 12. Uma vez validado, os dados contidos no pacote são utilizados para plotagem do VANT. Caso a validação não tenha sucesso, a tarefa de plotagem referente à aquele pacote não é realizada, e o sistema aguarda o recebimento da próxima transmissão válida.

4 *Análise dos Resultados*

Como forma de avaliar a proposta desse trabalho, analisou-se o esforço computacional requerido pela plataforma AuRoRA, em duas situações distintas. Primeiramente, considerou-se o caso onde apenas um computador é responsável pela comunicação com o VANT, geração dos sinais de controle e exibição da interface gráfica (Figura 13).

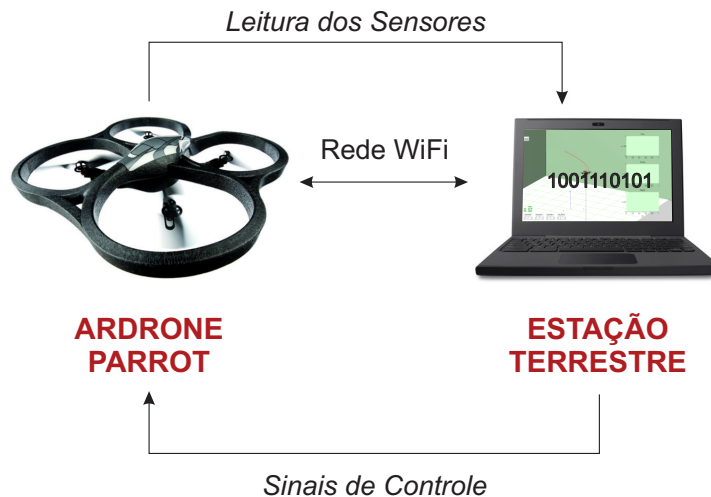


Figura 13: Estrutura geral do sistema, sem distribuição de tarefas entre dois computadores.

Nessas condições, a Figura 14 apresenta o tempo gasto para execução de cada ciclo do laço principal da plataforma AuRoRA, para uma simulação com duração total de 6 segundos.

Conforme mencionado no Capítulo 2, o período de amostragem do veículo aéreo utilizado na simulação é de $t_{rx} = 1/30s$, e portanto, para garantir a estabilidade do VANT, é preciso que uma instrução seja recebida pelo seu controlador de bordo a cada t_{rx} segundos.

No entanto, note que a cada intervalo de aproximadamente Δi iterações, o tempo de duração do ciclo é visivelmente maior que t_{rx} , alcançado picos de até 0,11 segundos. Tal situação ocorre sempre que é exigida uma tarefa de plotagem do VANT, definindo uma condição que compromete o controle da aeronave.

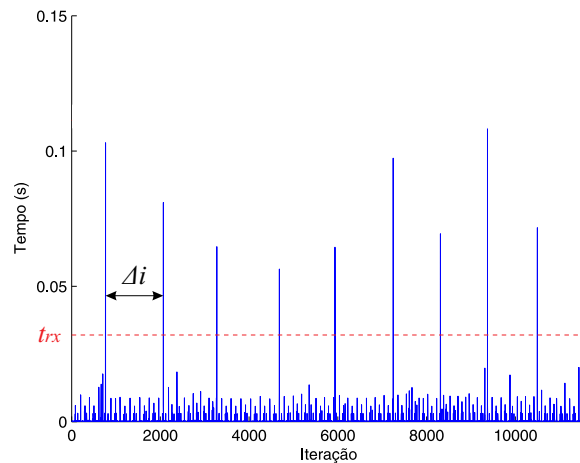


Figura 14: Duração das iterações da plataforma AuRoRA, com estrutura centralizada.

Agora considerando a inserção de um novo computador na estrutura geral do sistema, que se comunica por meio de um *link* UDP, e é responsável unicamente pela tarefa de exibição gráfica, conforme já apresentado anteriormente na Figura 11, é possível constatar que a divisão das tarefas entre estes computadores é suficiente para manter o tempo de duração das iterações do controlador abaixo do período de amostragem t_{rx} do ArDrone (ver Figura 15).

Note ainda que a comunicação UDP entre os dois computadores mostrou-se suficientemente rápida a ponto de não demandar um esforço computacional maior por parte da estação terrestre “A”, uma vez que o tempo médio gasto para transmissão dos dados por UDP está na ordem de 1ms. Neste caso, os picos observados no gráfico representam os momentos em que foi estabelecida a comunicação UDP entre as duas estações em terra.

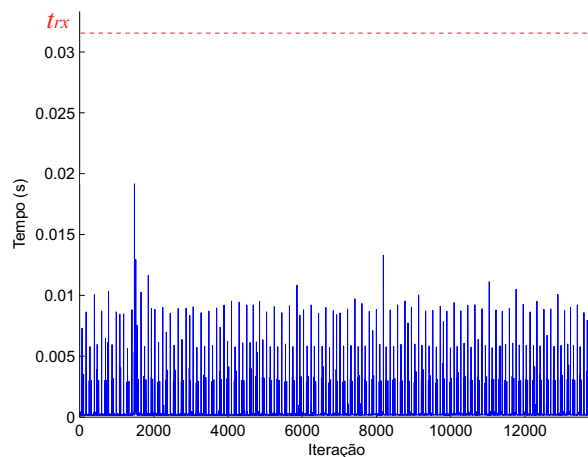


Figura 15: Duração das iterações da plataforma AuRoRA, com estrutura descentralizada.

5 *Considerações Finais*

Conforme mencionado no Capítulo 2, a geração de gráficos permite o monitoramento em tempo real de situações de voo, auxiliando a visualização do comportamento do VANT por parte do usuário. No entanto, o custo computacional exigido pelo processamento gráfico pode, na maioria das vezes, comprometer o tempo de permissão estabelecido para o cumprimento de determinadas tarefas, culminando em uma maior dificuldade para controlar a aeronave.

Como forma de contornar essa limitação, o presente trabalho propôs a descentralização do processamento das atividades do sistema entre dois computadores, estabelecendo uma relação cliente-servidor entre os mesmos.

Para estabelecer a troca de informações entre as duas unidades computacionais, foi escolhido o protocolo de comunicação UDP, que se mostrou rápido, de fácil uso e ideal para aplicações em tempo real.

A partir da análise dos resultados apresentada no Capítulo 4, foi mostrado que o tempo de duração do ciclo da plataforma, que antes alcançava picos de até 0,11 segundos, pode ser reduzido para valores da ordem de 0,02 segundos. Tal resultado se mostrou suficiente para suprir o envio de sinais de controle pelo WiFi e de dados através do canal UDP, em um tempo inferior ao período de amostragem do veículo aéreo utilizado (inferior a 1/30 segundos, para o caso do AR-Drone Parrot). Dessa forma, é possível executar a tarefa de exibição gráfica do VANT, sem comprometer a estabilidade em seu controle.

Referências

- [1] MUTTER, F. et al. Model-driven in-the-loop validation: Simulation-based testing of uav software using virtual environments. In: *Engineering of Computer Based Systems (ECBS), 2011 18th IEEE International Conference and Workshops on*. [S.l.: s.n.], 2011. p. 269 –275.
- [2] KRAJNÍK, T. et al. Ar-drone as a plataform for robotic research and educatino, 2011. In: . [S.l.: s.n.], 2011.
- [3] PAW, Y. C.; BALAS, G. J. Developemnt and application of an integrated framework for small uav flight control development. In: . [S.l.: s.n.], 2010.
- [4] CAI, G. et al. Design and implementation of a hardware-in-the-loop simulation system for small-scale uav helicopters. In: *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*. [S.l.: s.n.], 2008. p. 29 –34.
- [5] LUI, J.; CHAN, M. An efficient partitioning algorithm for distributed virtual environment systems. *Parallel and Distributed Systems, IEEE Transactions on*, v. 13, n. 3, p. 193 –211, mar 2002. ISSN 1045-9219.
- [6] TJERNBERG, I.; LINDBERG, J.; HANSSON, K. Cooperative networked controle of unmanned air vehicles. In: *A Study of Quad-Rotors in a Smart Building Context*. [S.l.: s.n.], 2011.
- [7] PIZETTA, I. H. B. et al. Uma plataforma *Hardware-In-Loop* para vants de asas rotativas. In: . [S.l.: s.n.].
- [8] CHO, D. K. et al. A udp-based protocol for mobile robot control over wireless internet. In: *Robot Communication and Coordination, 2009. ROBOCOMM '09. Second International Conference on*. [S.l.: s.n.], 2009. p. 1 –4.
- [9] VUCKOVIC, V. The realization of the parallel chess system using udp communication protocol. In: *Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2007. TELSIS 2007. 8th International Conference on*. [S.l.: s.n.], 2007. p. 450 –453.
- [10] GEHLEN, G.; AIJAZ, F.; WALKE, B. Mobile web service communication over udp. In: *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*. [S.l.: s.n.], 2006. p. 1 –5.
- [11] MALHOTRA, A. et al. Udp based chat application. In: *Computer Engineering and Technology (ICCT), 2010 2nd International Conference on*. [S.l.: s.n.], 2010. v. 6, p. V6–374 –V6–377.

-
- [12] KATO, T.; HIGASHI, T.; SHIMIZU, K. Teleoperation of a robot arm system using pneumatic artificial rubber muscles: Teleoperation over the internet using udp and a web camera. In: *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 714 –718.
- [13] SANTOS, S. Barros dos; NASCIMENTO, C.; GIVIGI, S. Design of attitude and path tracking controllers for quad-rotor robots using reinforcement learning. In: *Aerospace Conference, 2012 IEEE*. [S.l.: s.n.], 2012. p. 1 –16. ISSN 1095-323X.
- [14] XIAOHUI, X.; ZHIJIANG, D.; LINING, S. The design and implementation of real-time internet-based telerobotics. In: *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*. [S.l.: s.n.], 2003. v. 2, p. 815 – 819 vol.2.
- [15] ZHU, M. et al. Experimental analysis of communication quality of teleoperation with force feedback via ip network. In: *Integration Technology, 2007. ICIT '07. IEEE International Conference on*. [S.l.: s.n.], 2007. p. 155 –160.